BIND DNS Server

On this page the DNS protocol and the **BIND DNS server** are explained, as is the Webmin module for creating and managing DNS domains.

See Introduction to the Domain Name System

Contents

[hide]

The BIND DNS Server module

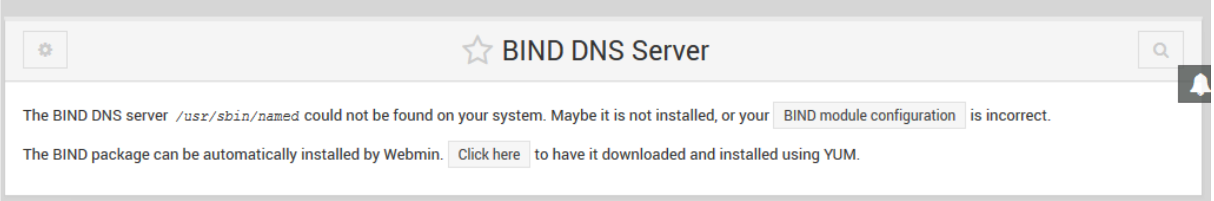
BIND (Berkeley Internet Name Domain) is the most common DNS server for Unix systems. Several versions have been released over the years, the most recent being version 9. The BIND DNS Server module (found under the Servers category) supports the configuration of versions 8 and 9. The older version 4 has a different configuration file format, and can be configured using the BIND 4 DNS Server module, documented in a later section of this chapter.

Because BIND is available for almost all Unix systems and works identically regardless of operating system, the instructions in this chapter apply not just to Linux but to other versions of Unix as well. Most versions of Unix and Linux include BIND 8 or 9 as a standard package, so it is rarely necessary to install it. If the module cannot find the DNS server, an error message will be displayed on the main page - if this happens, check your operating system CD or website for a BIND package, or download and compile the source from http://www.isc.org/.

BIND's primary configuration file is /etc/named.conf, which contains all of the zones that the server hosts, and global configuration settings that apply to all zones. The records in each zone are stored in separate files, usually found in the /var/named directory. This Webmin module always updates all of these files directly, instead of by communicating with the running BIND process. This means that if you are running some other program that dynamically updates zones by communicating with BIND (such as a DHCP server), then this module should not be used as it may interfere with these changes. However, very few systems have this kind of dynamic updating activated.

Versions 9 of BIND has some features that version 8 does not. The most important one that is supported by this Webmin module is views. A view is a set of zones that are visible to only some DNS clients. Normally all clients see the same zones, but with BIND 9 you can restrict the visibility of some domains to only particular clients, identified by their IP addresses. This can be useful for creating zones that are only visible to systems on an internal network, even if your DNS server is connected to the Internet.

If you have never set up BIND on your system, when you enter the module for the first time the main page will display a form for setting up the DNS server, as shown below. This form is only shown if Webmin detects that the configuration file named.conf does not exist, or if the zone files directory that is specifies is non-existent. If you are certain that your BIND configuration is valid and that the DNS server is already running, do not click the **Create** button, as your named.conf file will be overwritten. Instead, click on the **Module Config** link and check that all the paths are correct for your system.



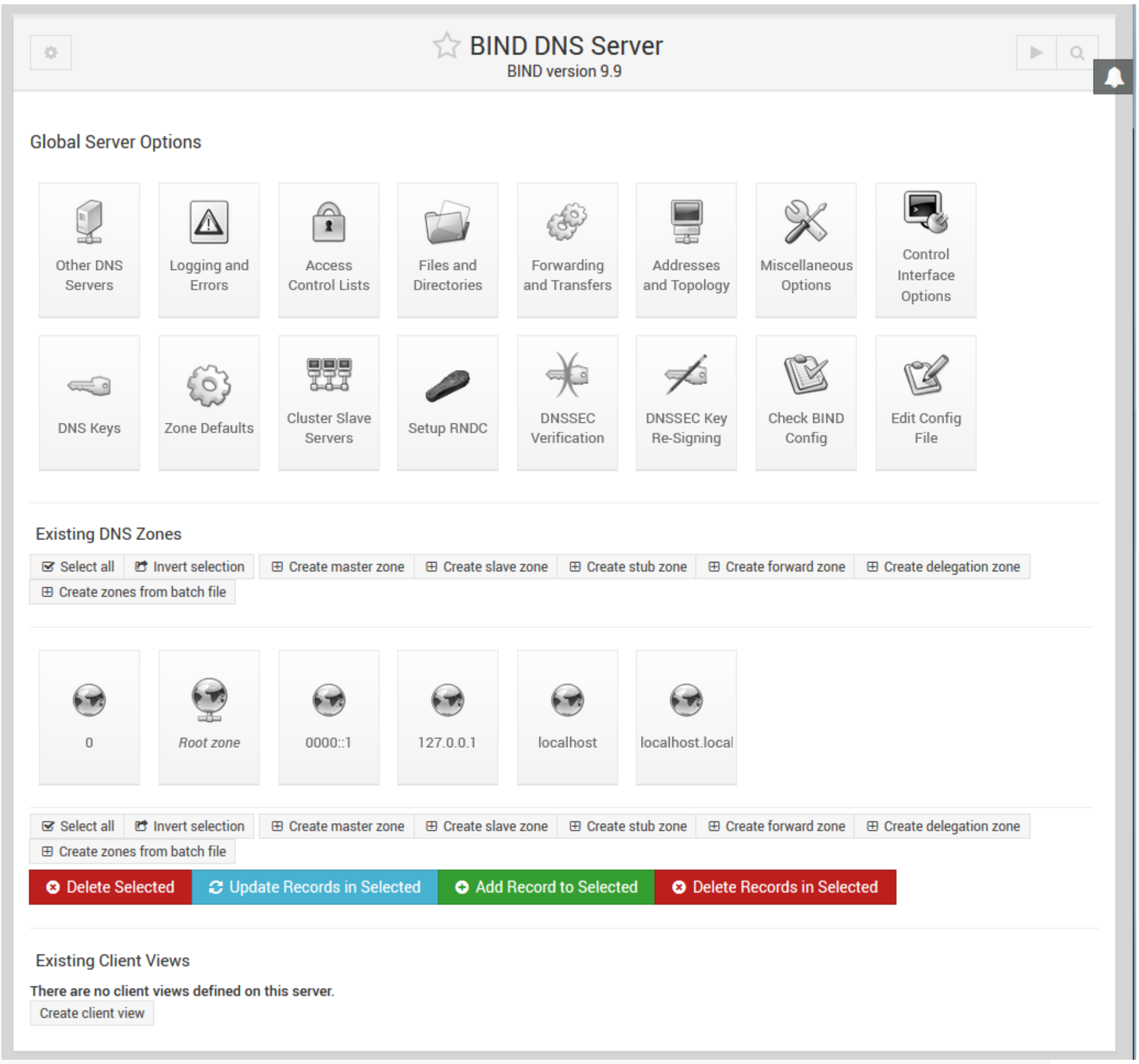
The BIND DNS server `/usr/sbin/named` could not be found on your system. Maybe it is not installed, or your BIND module configuration is incorrect.

The BIND package can be automatically installed by Webmin. Click here to have it downloaded and installed using YUM.

Bind Install

When BIND has been set up on your system, the main page will appear as shown in the screenshot below. At the top is a table of icons for setting global options that apply to your entire DNS server. Below them are icons for each of the zones your server hosts, followed by icons for views if you are running BIND version 9. At the very bottom are buttons for applying the current DNS configuration or starting the BIND server.

If you have just set up BIND for the first time, there will probably be only one zone icon - the root zone. Some Linux distributions that include a BIND package come with a basic configuration file that defines zones like localdomain and 127.0.0, which are used for resolving the localhost and 127.0.0.l loopback hostname and IP address.



There are no client views defined on this server.

Bind DNS Server Main Page

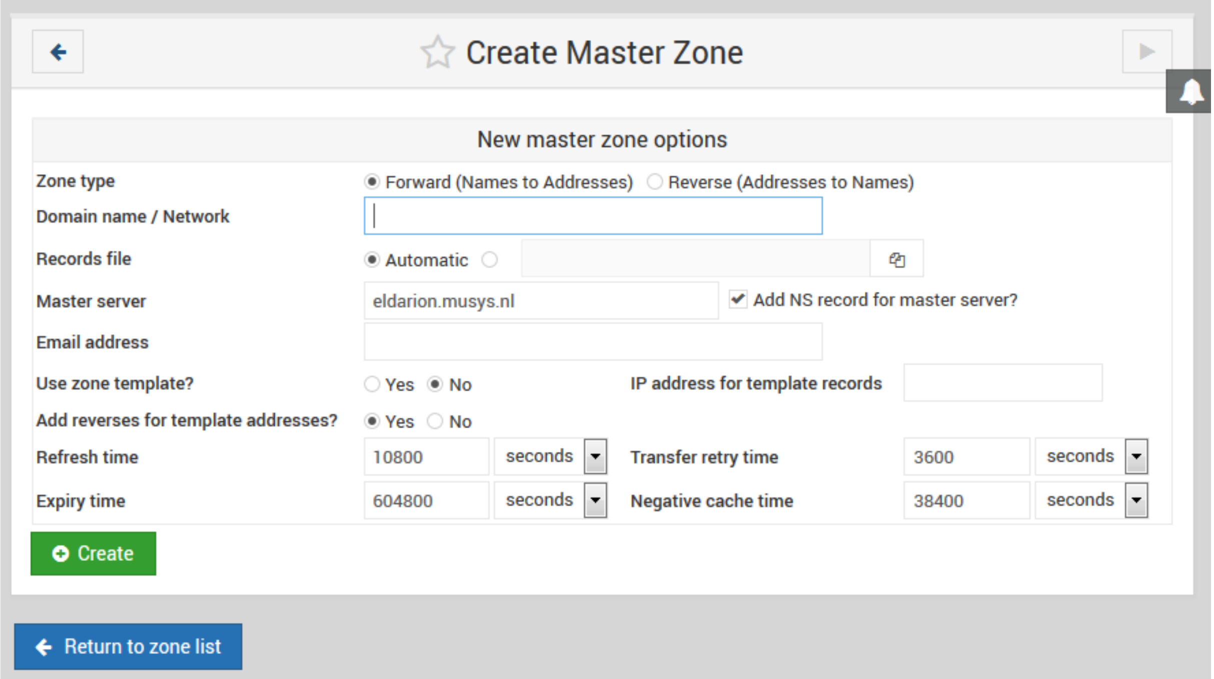
Creating a new master zone

A master zone is one for which your DNS server is the authoritative source of information. A single zone may be hosted by multiple servers, but only one is the master - all the rest are slaves. If you want to add a new master zone to your server's configuration, the steps to follow are:

1. Decide on a name for the new zone, such as example.com or internal. If this is going to be Internet domain that will be visible to other everyone in the world, the domain name must not have been registered by anyone else yet. However, you cannot normally register it yourself until your DNS server has been set up to host it.
2. On the module's main page, click on the **Create a new master zone** link below the table of existing zones. This will take you to the page shown in the image below for entering the details of the new zone.
3. If this is to be a forward zone like example.com or foo.com.au, leave the **Zone type** field set to **Forward**. However, if it is a reverse zone for looking up hostnames from IP addresses, set the field to **Reverse**.
4. In the **Domain name / Network** field, enter the name of the zone without any trailing dot. For a reverse zone, just enter the network

address like 192.168.1. Webmin will automatically convert this to the in-addr.arpa format for you when the domain is created.

5. The **Records file** field controls where the configuration file containing the zone's records is stored. If you leave it set to **Automatic**, the filename will be determined automatically based on the module's configuration and the directory setting in the named.conf file. This is usually the best option, as it will result in the records file being created in the same directory as any existing zones, such as /var/named. However, if you de-select the **Automatic** option and enter a filename instead, all records for the zone will be written to that file. If you enter the name of an existing file, it will be overwritten when the domain is created.
6. In the **Master server** field, enter the full domain name of the master DNS server for this zone. This must be the canonical name of your system, such as server.example.com, not a short name like server. This server (and the values from the next
7. fields) are used to create the new zone's SOA record.
8. In the **Email address** field, enter the address of the person responsible for this zone. You can use the @ symbol in the address, which Webmin will automatically convert to a dot for inclusion in the SOA record.
9. The **Refresh time** field determines how often secondary servers should check with this master server for updates to the zone. The default is reasonable, but you may want to increase it for zones that rarely change, or decrease it for those that are frequently updated.
10. The **Transfer retry time** field determines how long a secondary server should wait after a failed zone transfer before trying again.
11. The **Expiry time** field controls the maximum amount of time that a secondary DNS server for the zone should cache records for before re-transferring them from the master.
12. The **Default time-to-live** field determines the TTL of records in the zone that do not have one set explicitly.
13. Click the **Create** button at the bottom of the page. As long as the form has been filled in correctly and the zone does not already exist on your server, you will be taken to a page for adding new records to the zone.
14. Return to the module's main page which will now include an icon for your new zone, and click the **Apply Changes** button at the bottom to activate it.



Create Master Zone

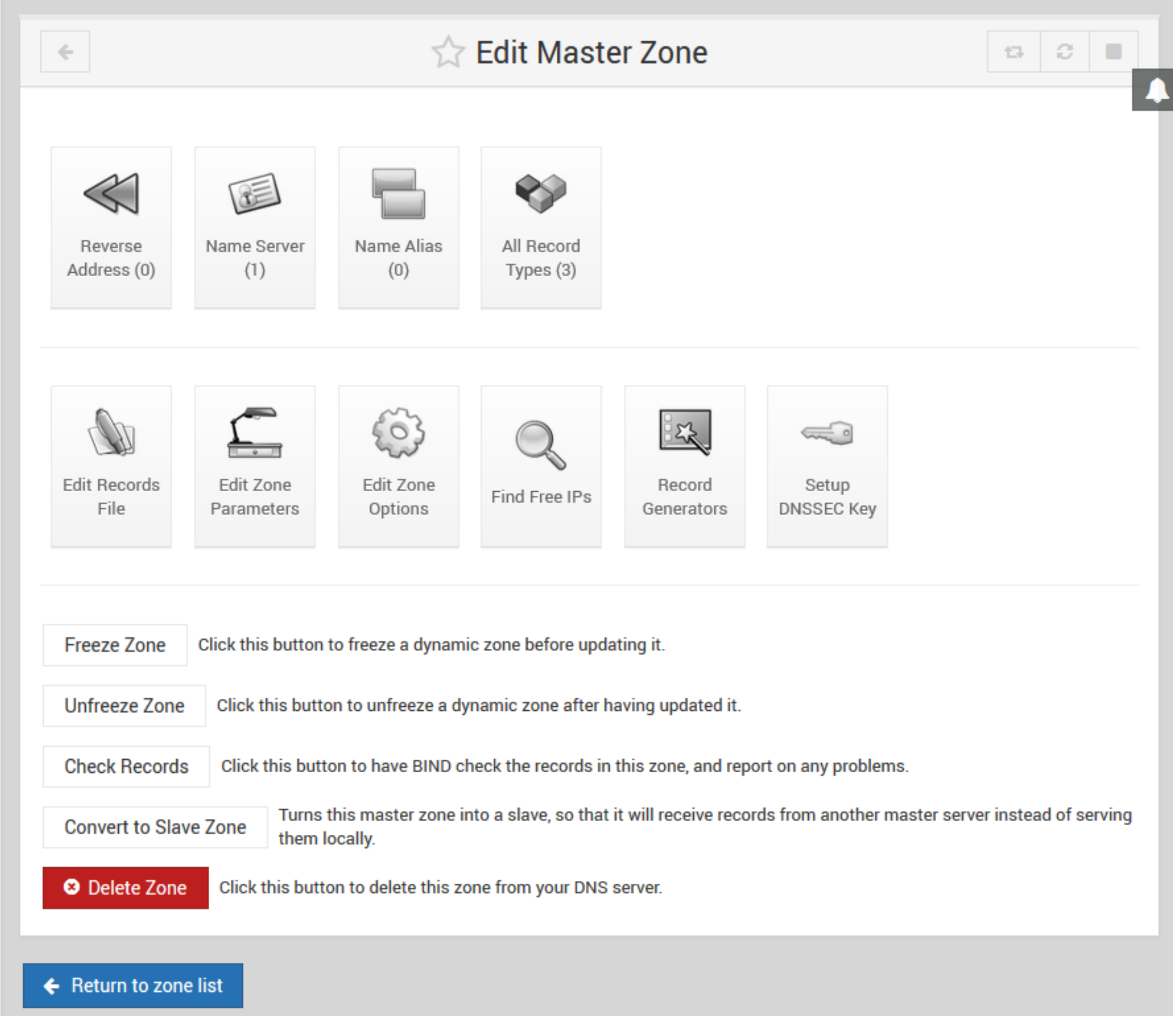
New master zone options

Zone type	● Forward (Names to Addresses) ○ Reverse (Addresses to Names)		
Domain name / Network			
Records file	● Automatic ○		
Master server	eldarion.musys.nl ☑ Add NS record for master server?		
Email address			
Use zone template?	○ Yes ● No	IP address for template records	
Add reverses for template addresses?	● Yes ○ No		
Refresh time	10800 seconds	Transfer retry time	3600 seconds
Expiry time	604800 seconds	Negative cache time	38400 seconds

Create Master Zone.png

A newly created zone will contain only one record (unless you have set up a template). To add more, follow the instructions in the next section. Once you have set up the basic records in your domain, you can register it with the authority that manages the parent domain, such as .com or .com.au. Some domain authorities will not allow you to register zones that do not have at least two servers (one master and one slave), and name server records in the zone for those servers.

Adding and editing records



Freeze Zone
Click this button to freeze a dynamic zone before updating it.

Unfreeze Zone
Click this button to unfreeze a dynamic zone after having updated it.

Check Records
Click this button to have BIND check the records in this zone, and report on any problems.

Convert to Slave Zone
Turns this master zone into a slave, so that it will receive records from another master server instead of serving them locally.

Delete Zone
Click this button to delete this zone from your DNS server.

Return to zone list

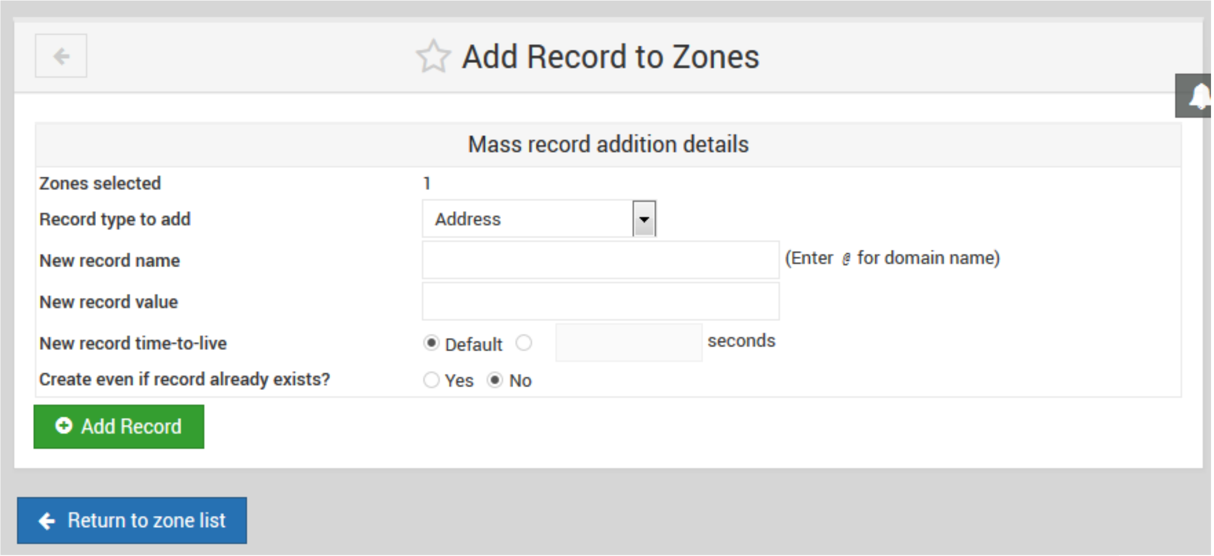
Bind - Edit Master Zone

The most useful feature of the BIND DNS Server module is the ability to add, edit and delete records in the master zones hosted by your server. For example, if you wanted to set up a webserver in your domain example.com, you would need to add an Address record for www.example.com with the IP address of the server. To add a new record like this, the steps to follow are:

1. On the module's main page, click on the icon for the zone that you want to add to. This will bring up the page shown below, at the top of which is a table of icons, one for each record type.
2. Click on the icon for the type of record that you want to add. The most common type is **Address**, which associates an IP address with a hostname. See the **#Record types** section below for a complete list of all the supported record types.
3. Clicking on the icon will take you to a page listing all existing records of that type. Above the list is a form for entering a new record.
4. In the **Name** field, enter the name of the new record relative to the zone name. For example, if you wanted to add the record www.example.com, you should just enter *www*. It is also possible to enter the full record name, as long as it has a dot at the end to indicate that it is not relative to the zone.

Do not enter just *www.example.com*, as it will be converted to www.example.com.example.com, which is probably not what you want.

5. If this record is going to change more frequently than the rest of the zone, change the **Time-To-Live** field from **Default** to the estimated time between changes. This determines how long DNS clients and other servers will cache the record for.
6. If you are adding an Address record, enter the complete IP address of the host into the **Address** field. See the table below for a description of the fields that appear when adding other types of records and what they mean.
7. The field **Update reverse?** only appears when adding an Address record. It controls the automatic creation of a corresponding record in a reverse zone which associates the hostname with the IP address. Naturally, this can only be done if the IP that you enter is in a network that your system is the primary reverse DNS server for. This keeps the forward and reverse zones synchronized, which can be very useful. If **Yes** is selected, a reverse address record will be added as long as one does not already exist in the reverse zone for the same IP address. Often many hostnames will have the same IP, such as those use for name-based virtual hosting. In cases like these, you don't want to change the reverse mapping if one already exists. The *Yes (and replace existing)* option works the same as **Yes**, but if a reverse record for the IP address already exists it will be updated with the new hostname. This can be useful if you know there is an existing record that you want to replace.
   If **No** is selected, no reverse address will be created even if it is possible.
8. When you are done filling in the form, click the **Create** button at the bottom. As long as it is filled in correctly, the record will be added to the list below the form. When writing to the zone's records file, Webmin will use the full canonical format for the record name, such as www.example.com., even if you just enter www.
9. To activate the new record so that it can be looked up by DNS clients and other servers, you will need to click the *Apply Changes* button on the module's main page. If you are planning to add or edit several records, it is usually better to wait until all the changes are complete before hitting the apply button. If it is available, you can instead use the *Apply Changes* button at the bottom of the master zone page shown below. This uses the ndc command to tell BIND to re-read only the file for this zone, which can be much faster on a system that hosts are large number of domains.



Add Record to Zones

Mass record addition details

Zones selected	1
Record type to add	Address
New record name	(Enter @ for domain name)
New record value	
New record time-to-live	● Default ○ seconds
Create even if record already exists?	○ Yes ● No

Add Record

Return to zone list

Bind - Add Record to Zones

Although the instructions above are focused on adding an Address record, the process of adding other record types to a forward zone is almost identical. The **Update reverse?** field does not exist, and the **Address** field is replaced with one or more different fields. The **Record types** section below explains in detail what fields are available for each type of record known to Webmin.

When adding a Reverse Address record to a reverse zone, the form is quite different. The **Address** field appears before the **Hostname**, and the hostname must always be entered in canonical form with a dot at the end, like *www.example.com.*. The **Update reverse?** field is replaced with **Update forward?**, which controls the automatic creation of a record in the corresponding forward zone. However, there is no option to overwrite an existing forward record - if one with the same name already exists, it will not be touched even if **Yes** is selected.

Every time a record is added to or updated in a zone using Webmin, its serial number will be automatically incremented. This also applies to reverse zones that are automatically updated when adding an Address record, and vice-versa. This means that when you apply the changes, other DNS servers will be able to detect that the zone has changed by comparing the new serial number with the old one that they have cached.

To edit an existing record in a zone, the steps to follow are:

1. On the module's main page, click on the icon for the zone that you want to edit, which will bring you to the page show above.
2. Click on the icon for the type of record that you want to change, which will display a page listing all records of that type in the zone. Alternately, you can click on the *All Record Types* icon which will bring up a list of every single record in the zone regardless of type.
3. Click on the name of the record that you want to edit. Your browser will display a form similar to the one used for adding a record, but with the fields already filled in with the details of the existing address.
4. To re-name the record, edit the contents of the **Name** field. It will be shown in canonical form with a dot at the end initially, but you can change it to a name relative to the domain if you wish.
5. Adjust the **Time-To-Live** field in you want this record to have a different TTL, or set it to **Default** to make it the same as the rest of the zone.
6. If this is an Address record, change the IP in the **Address** field. For other record types, the fields are the same as those on the record creation form, and have the same meanings.
7. For Address records, the field **Update reverse?** is displayed. Selecting **Yes** will cause the corresponding record in the reverse zone to be have its name and

address changed to match this forward record. If you change the IP so that the reverse address is no longer in the same network, it will be removed from the old reverse zone and added to the new reverse zone (if it is hosted by your server).

8. For Reverse Address records, the field **Update forward?** is shown instead. If **Yes** is selected, the corresponding Address record in its forward zone will be changed to match any changes that you make on this form.
9. Click the **Save** button to update the record in the zone file, and return to the list of record types.
10. To activate the changes, click the **Apply Changes** button back on the module's main page.

To delete a record from a zone, click on the **Delete** button on the editing form instead of **Save**. For Address records, if the **Update reverse?** field is set to **Yes**, the corresponding Reverse Address record will be deleted as well. Apart from that, the process of deleting a record is identical no matter what type it is. The same thing happens when deleting a Reverse Address record - the matching Address record is deleted as well, as long as the *Update forward?* field is set to **Yes**.

The list of records in a zone is initially sorted according to the module configuration, which usually means that records will be displayed in the order that they were added. To change this, you can click on a column heading like **Name**, **Address** or **Real Name** to sort them by that column instead. The sorting is only temporary though, and will be lost if you return to the main page and re-open the zone. To change it permanently, see the **Order to display records in** field in the section on *Configuring the BIND DNS Server module*.

Record types

Webmin does not support all of the record types that BIND knows about, only those that are most commonly used. The list below covers all of the supported types, and explains what they are used for and what fields are available when adding or editing a record of that type in Webmin. Next to each type name is the short code used by BIND itself for identifying the type in the records file.

Address (A)

An address record associates an IPv4 address with a hostname. Any system that you want to be able to connect to via HTTP, telnet or some other protocol using its hostname must have an address record so that clients can look up its IP. A single hostname can have more than one Address record, which is often done to spread the load for a website across multiple servers. It is also valid to create multiple records of this type with different names but the same IP, such as when setting up name-based Apache virtual servers. When creating or editing an Address record, the field **Address** is displayed for entering the IP associated with the hostname. A field labelled **Update reverse?** also appears, which controls the automatic creation and modification of a Reverse Address record in the appropriate reverse zone. See the **Adding and editing records** section above for more details.

IPv6 Address (AAAA)

An IPv6 address record associates an IPv6 address with a hostname similar to an A record.

Name Server (NS)

Records of this type defines a name server that is responsible for a zone. Every zone must have at least one Name Server record for itself, and may have additional records that specify the DNS servers responsible for subdomains. If you set up a secondary DNS server for a zone, be sure to add a Name Server record for the zone on the master server. In this case, the name of the record will be the canonical name of the zone, such as *example.com.*. When creating or editing a record of this type, a field labelled **Name Server** will be displayed. This must be filled in with the IP address or hostname of the DNS server that is responsible for the zone. If you enter a hostname, it must have an IP address set by an Address record in some zone on your server.

Name Alias (CNAME)

This type of record creates an additional name for an existing Address or Reverse Address record. When a DNS client requests the IP address of a record of this type, it will get the IP of the record that the Name Alias points to instead. This kind of record can be useful if you have a single host that needs to be accessible under several different names, such as a web server doing name-based virtual hosting. Even though this could also be done by creating multiple Address records, creating just a single Address and multiple Name Aliases is more flexible as it allows easier updating if the IP address of the host ever changes. The forms for editing and creating Name Alias records contains a field labelled **Real Name**. This must be filled in with either the canonical name of the record that the alias points to (such as *webserver.example.com.*), or with a short name that is relative to the zone that the Name Alias record is in.

Mail Server (MX)

Mail Server records tell mail delivery programs like Sendmail and Qmail which system to contact when delivering mail to a domain or host. Without a record of this type, mail for a domain will be delivered to the system whose IP is specified in the Address record for the zone itself. This is not always desirable, as you may want that IP to be the address of a webserver, so that web browsers can connect to http://example.com/ as well as http://www.example.com/. A Mail Server record can solve this problem by having only email for example.com sent to another hosts, and all other traffic to the webserver. Each Mail Server record has a priority, which tells mail delivery programs which mail server should be tried first. The record with the lowest priority should point to the system that actually receives and stores email for the domain, while those with higher priorities generally point to systems that will simply relay mail. Delivery programs will try each in turn starting with the lowest, so that if the primary mail server is down email will still be sent to a relay that can hold it until the primary comes back up.

When adding or editing a Mail Server record, two additional fields are displayed. The first is labelled **Mail Server**, and must be filled in with the canonical or relative hostname of a system that can accept mail for the domain or hostname entered in the **Name** field. The second is labelled **Priority**, and must be used to specify a numerical priority for this particular mail server. Normally a priority of 5 is used for the primary mail server, and 10 for backup relays. If you only have one mail server for your domain, it doesn't really matter what number is entered into this field. It is possible for two servers to have the same priority, in which case one will be chosen randomly to deliver to. A Mail Server record can use the * wildcard in its name, which indicates to mail programs that a particular mailserver is responsible for all hosts in a domain. For example, a record named like **.example.com* would match the hostname pc1.example.com and any other hosts in the zone. This can be useful if you want to force mail that would otherwise be delivered directly to workstations in your domain to go through a central mailserver instead. Webmin will not let you use wildcards unless the **Allow wildcards** module configuration option is set to **Yes** though, as explained in the *Configuring the BIND DNS Server module* section.

Host Information (HINFO)

Records of this type are used to record information about the hardware and operating system of a particular host. For example, you might create one that says that *server1.example.com* is an x86 PC running Linux. However, they are very rarely used and are in fact considered a security risk, as they give out information to potential attackers that could be used to take over a server. When creating or editing a Host Information record, the fields **Hardware** and **Operating System** are displayed for entering the architecture and operating system type of a host. The values you enter must not contain any spaces - typically, they are replaced in the hardware type and operating system strings with _ characters.

Text (TXT)

A Text record associates an arbitrary message of some kind with a name. TXT-records can be are used to provide ownership information to mail facilities as SPF and DKIM. Be aware though that any such comments will be available to anyone on the Internet that can look up records in your domain, and so should not contain sensitive information. The field **Message** is displayed when entering or editing a Text record. You can enter any text that you like, including spaces.

Well Known Service (WKS)

A record of this type associates a hostname, port and protocol with a name. It can be thought of as a generalized variant of the Mail Server record, which tells clients which host provides a particular service for some domain or hostname. However, almost no programs actually look up WKS records, so in practice they are pretty much useless. When adding or editing one of these records, the fields **Address**, **Protocol** and **Services** are available. The first is for entering the IP address of a host that provides the services for the host or domain entered into the **Name** field. The second is for selecting the network protocol that the services use, either TCP or UDP. The last is for entering a list of port numbers or names (from the /etc/services file) for services that the host provides.

Responsible Person (PR)

This type of record is used for specifying the person or group responsible for a particular host. Each of these records has two values associated with it - an email address, and the name of Text record containing the person's name. Responsible Person records are rarely seen, and are not used by any mail delivery program or Internet client. The **Email Address** field shown when editing or adding one of these records is for entering the complete address (like *jcameron@example.com*) of the person responsible for the host whose name is entered into the **Name** field. The **Text Record Name** field is for entering the relative or canonical name of a Text record that contains the person's real name.

Location (LOC)

Location records are used to specify the physical location in latitude and longitude of a host. They are hardly ever seen, and thus not used by many programs. However, they can be useful in large organizations that have hosts in many countries. When adding or editing a Location record, the field *Latitude and Longitude* is displayed for entering the location of the host in the **Name** field. It must be formatted like _42 21 43.528 N 71 05 06.284 W 12.00m 30.00m 10000.00m 10.00m_.

Service Address (SRV)

Records of this type are used to associate a domain name, service name and protocol with a particular host. They allow you to specify which server a client should contact for a particular service and hostname, instead of just connecting to the host. In a way, they are like Mail Server records but far more flexible. For example, you can specify that the POP3 server for example.com is *mail.example.com*, but the webserver is *www.example.com*. At the time of writing, SRV records are mostly used by Windows client systems. When adding or editing a Service Address record, the fields **Protocol** and **Service name** are displayed near the **Name** text box. For the protocol, you must select either TCP or UDP from the menu. For the service name, you must enter a well-known name from the /etc/services file, such as *pop3* or *telnet*. To look up an SRV record, a client combines the service name, protocol and name to get a record name like ___*telnet.*___*tcp.example.com*. Webmin does this for you automatically when editing or adding a Service Address record, but you can see the combined name on the page listing records of this type. Webmin also automatically added the _s before the service and protocol, but hides them when a SRV record is being displayed or edited. This means that there is no need to enter then manually when creating or editing a record of this type. The **Priority** field must be used to enter a numeric priority for this server, which has the same meaning as the priority in a Mail Server record. The **Weight** field must contain a weighing for this particular server, or zero if there is only one record with the same name, protocol and service name. A higher weighting tells clients to try this server more often than one with a lower weight. The **Port** field must contain a port number for clients to connect to on the server, which does not necessarily have to be the standard port for the service. In the **Server** field, you must enter the hostname or IP address of the system that actually provides the service, and that clients actually connect to.

The record types support by Webmin in reverse zones are:

Reverse Address (PTR)

A reverse address record associates a hostname with an IP address in a reverse zone. For DNS clients to be able to lookup hostnames from IP addresses in your network, you will need to create one record of this type for each host. However, most of the time this is done automatically by Webmin when adding and editing Address records. If you create your own Reverse Address records, make sure that they are synchronized with the matching Address records. When adding or editing a record of this type, the fields **Address** and **Hostname** are displayed. The first is for entering a complete IP address, like *192.168.1.10*. This will be automatically converted by Webmin to the in-addr.arpa format used internally by the DNS system for reverse addresses. The second field is for entering a hostname in canonical form, such as *pc1.example.com.*, be sure to always put a dot at the end, or else the hostname will be relative to the reverse zone, which is definitely not what you want.

Name Server (NS)

Name Server records in a reverse zone have an identical purpose to those in a forward domain - they tell other DNS servers the IP address or hostname of a server responsible for the zone or a sub-domain. This means that one must be added for each primary or secondary DNS server for the zone. The *Zone Name* field that appears when adding or editing a record of this type is for entering the name of the zone that the server is responsible for, which will typically be the zone that contains the record. However, unlike Reverse Address records this field is not automatically converted to in-addr.arpa format. Instead, you must enter it in fully qualified form like *1.168.192.in-addr.arpa.* if defining an nameserver for the *192.168.1* network. In the *Name Server* field, you must enter an IP address or canonical form hostname for the DNS server, such as *ns1.example.com.*.
</blockquote>

Name Alias (CNAME)

Records of this type behave exactly the same in reverse zones as they do in forward domains. However, you must fill in the **Name** and **Real Name** fields with reverse names in in-addr.arpa format, as Webmin will not convert them for you. Name Alias fields are most useful in reverse zones for doing partial subnet delegation, as covered in the *Partial reverse delegation* section below.

Editing a master zone

You can use Webmin to edit many of the settings that apply to an entire master zone, such as the expiry and retry times, and the clients that are allowed to query it. These settings effectively apply to all records in the zone, although some (such as the TTL) can be overridden on a per-record basis.

Webmin uses the term zone parameters to refer to all information stored in the domain's SOA record, including the primary nameserver, administrator email address and retry and expiry times. All of these are set when the zone is created, but you can edit them at any time by following these steps:

1. On the module's main page, click on the icon for the zone that you want to edit.
2. Click on the **Zone Parameters** icon, which will bring up a form for editing the parameters.
3. The **Master server** field only needs to be edited if the Internet hostname of the DNS server has changed. Enter a fully-qualified hostname, with a dot at the end.

4. To change the address of the person responsible for the zone, edit the **Email address** field. Any @ symbols that it contains will be automatically converted to dots for use in the SOA record, as BIND requires.
5. The **Refresh time**, **Transfer retry time**, **Expiry time** and **Default time-to-live** fields all have the same meanings as explained in the section on **Creating a new master zone**. If records in your zone are going to be changing frequently in future, you may want to reduce some of these times. However, any changes, may not be detected by secondary servers and DNS clients until the old refresh or expiry time has elapsed, even if the new times are much lower. This is because they will wait for the old times to elapse before checking with the master server again to discovered the new ones.
6. Click the **Save** button at the bottom of the page when you are done, and then the **Apply Changes** button back on the module's main page. The serial number in the SOA record will be automatically incremented when the form is saved, so that secondaries now that the zone has changed.

There is another set of options that you can edit for a master zone, stored in the named.conf file in the zone's section. These control which servers and clients are allowed to query records in the zone, do zone transfers and update records over the network. The most useful of these options specifies a list of slave DNS servers for the zone that should be notified when a change occurs, so that they can perform immediate zone transfers and thus remain synchronized.

To edit these master zone options, the process to follow is:

1. On the module's main page, click on the icon for the zone that you want to edit. This will take you to the form shown in Figure 30-4.
2. Click on the **Edit Zone Options** icon, which will bring up a form showing the existing settings.
3. The **Check names?** field determines the level of checking that BIND performs on records in this zone when it reads the records file. The available options are :

Warn
If an invalid record is found, an error will be written to the system log file but processing of other records continues normally.

Fail
Invalid records cause the entire zone to be rejected, but other zones will still be processed normally.

Ignore
No checking is done at all.

Default
The global default from the Zone Defaults page is used. If it is not set, then the default complied into BIND will be used instead, which is to fail when invalid records are encounterd.

4. To have secondary servers notified when records in the zone change, set the **Notify slaves of changes?** field to **Yes**. BIND works out which slaves will be notified by looking at the Name Server records for the zone, and the list of IP addresses in the **Also notify slaves** field. If your zone has an secondary servers, then you should definitely turn this option on.
5. To allow some systems to update records in the zone dynamically, fill in the **Allow updates from** field with a list of IP addresses, IP networks (like 192.168.1.0/24) and BIND ACL names. Only those hosts that match will be able to modify records using commands like nsupdate, and if the list is left empty updates will not be allowed at all. You should be careful allowing the dynamic update of zones in which Webmin is also being used to edit records, as it is very likely that updates made dynamically will be overwritten by changes made in this module, or vice-versa.

6. By default, all DNS clients and servers will be able to lookup records in the zone. This may not be what you want for a zone that is used only on an internal network, as it may give away sensitive information to potential attackers. To restrict queries, fill in the **Allow queries from** field with a list of IP addresses, IP networks and BIND ACL names. If the field is left empty, the field with the same name on the global Zone Defaults page determines which clients are allowed.
7. To restrict the clients and servers that are allowed to perform zone transfers of all the records in this domain, fill in the **Allow transfers from** field. Often you will only want to allow secondary servers to perform transfers, especially if your zone is very large or contains records that you want to hide from attackers. Enter a list of IP addresses, IP networks and ACL names into the field to limit transfers to only matching clients. If it is left empty, the **Allow transfers from** field on the Zone Defaults page applies instead.
8. To specify additional slave servers to be notified when the zone changes, fill in the **Also notify slaves** field with a list of IP addresses. BIND normally works out with addresses of all secondary servers for the zone from its Name Server records, but this may not always be complete.
9. When you are done, click the **Save** button at the bottom of the page to update the BIND configuration file with your changes. You will need to use the **Apply Changes** button on the module's main page to make them active.

If a master zone is no longer needed, you can use this Webmin module to totally delete it along with all the records that it contains. To do this, the steps to follow are:

1. On the module's main page, click on the icon for the zone that you want to edit.
2. Click on the **Delete Zone** button at the bottom of the page.
3. When deleting a forward zone, the field *Delete reverse records in other zones?* controls whether matching Reverse Address records in hosted reverse zones for all of the Address records in this one should be removed as well. This is generally safe to set to **Yes**, as only records with the exact same IP address and hostname will be deleted.
4. Similarly, when deleting a reverse zone the field *Delete forward records in other zones?* determines whether matching forward records should be deleted too.
5. Once you have made your selection and are sure you want to go ahead with the deletion, click the **Delete** button. The zone's entry in the named.conf file will be removed, and its records file deleted.

You can convert a master zone to a slave zone of the same name without needing to delete and re-create it. This can be useful if the new server is taking over as the master for some domain, or if the master and secondary servers are switching roles. The section on *Editing a slave zone* explains how to carry out the reverse action of converting a slave zone to a master, which may be useful in this situation.

To convert a zone, the steps to follow are:

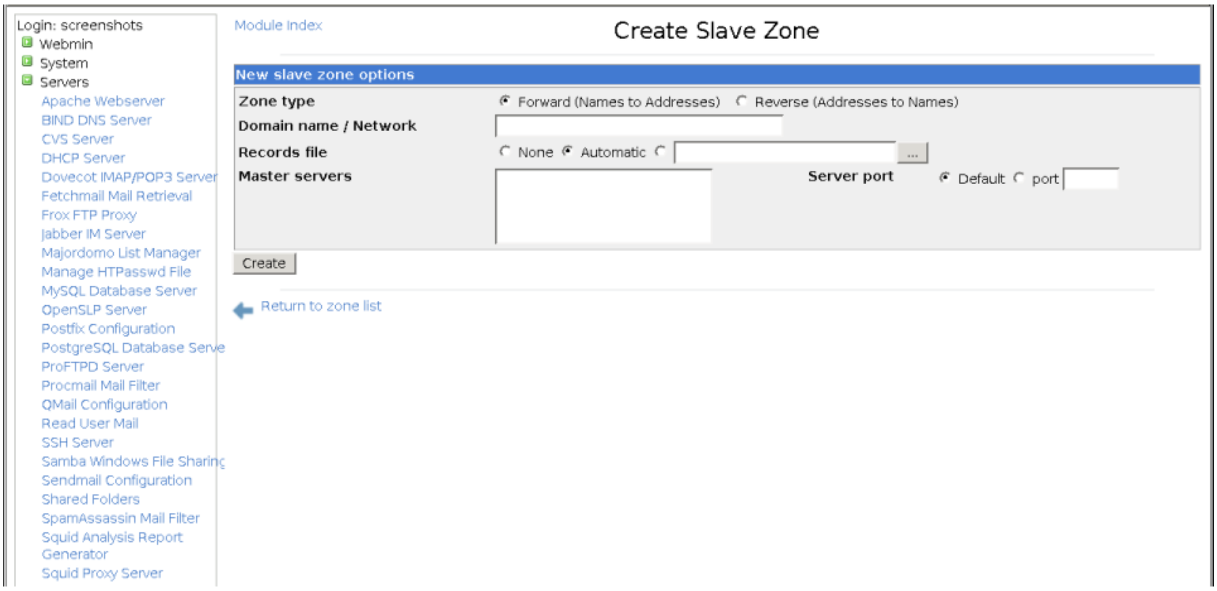
1. On the module's main page, click on the icon for the zone that you want to edit, then on the **Edit Zone Options** icon.
2. When you click on the **Convert to slave zone button**, zone's entry in named.conf will be immediately updated to convert it to a slave zone. The browser will then return to the module's main page.
3. Normally, every slave zone has a list of master server IP addresses that it can use to perform zone transfers from. In the case of converted zones, this list will be initially empty unless the **Default master server(s) for slave zones** module configuration option is set. Follow the instructions in the *Edit a slave zone* section to set the master servers addresses correctly.
4. To activate the change, click on the **Apply Changes** button the module's main page.

Creating a new slave zone

A slave or secondary zone is one for which your DNS server gets the list of records from a master server for the zone. Generally, slave servers are used to reduce the load on the primary server, or act as a backup in case it goes down. For important zones (such as a company's Internet domain), you should always have at least one slave server so that your website is still accessible and email can still be delivered even if the primary goes down.

The secondary DNS server for a domain should not usually be located on the same network as the master, so that the failure of that network cannot take them both down. Many ISPs and hosting companies will host secondary zones for their customer's domains for free, on their own DNS servers. If your ISP provides this service and you want to set up a secondary server for an Internet domain, you should take advantage of it. If so, most of this section can be skipped. However, if you want to add a slave server for an internal domain or have a large company network with many connections to the Internet, then the instructions below explain how to set it up:

1. On the main page of the BIND DNS Server module, click on the **Create a new slave zone** link above or below the list of existing zones. This will bring up the form shown below, for entering the details of the new domain.
2. For a forward zone like *example.com*, set the **Zone type** field to **Forward** and enter the zone name into the *Domain name / Network* field. For a reverse zone that maps IP addresses to hostnames for a network, choose the **Reverse** option and enter the network address (like *192.168.1*) into the *Domain name / Network* text field.
3. The **Records file** field determines if BIND keeps a cache of the records in this zone in a file, and if so where that file is located. If the option **None** is chosen, records that the DNS server transfers from the master will be kept in memory only, and lost when the server is re-started. This should only be chosen if there is a good network connect between the master and slave servers, as it will increase the number of zone transfers that your server must perform. If you choose **Automatic**, Webmin will generate a filename in the zone files directory specified in the named.conf file (usually /var/named). Whenever your server does a zone transfer, all records will be written to this file in the standard format. If the final option is selected, you can enter the full path to a file in which records should be stored into the field next to. This can be useful if you want to separate the records files for master and slave zones.
4. In the **Master servers** field, enter the IP addresses of the master DNS server and any other secondary servers for the zone. BIND will try these servers in order when doing a zone transfer, so the master should be first on the list. You must enter at least one address, so that your server knows where to get records from.
5. Click the **Create** button to have the new slave zone added to your server's configuration. Your browser will be re-directed to a page for editing options for the zone.
6. Return to the module's main page, and click the **Apply Changes** button on the main page to make the addition active.
7. On the master server, add a new Name Server (NS) record for the zone with the IP address of the secondary server. This can be easily done in Webmin by following the instructions in the **Adding and editing records** section.
8. Configure the master DNS server to notify this slave of any changes to records in the zone. The steps in the section on **Editing a master zone** explain how.
9. If this is an Internet domain, notify the registrar for the parent zone of the new secondary server. Most provide online forms for editing the list of nameservers for a domain, to which you can add the secondary's IP. This is necessary so that other hosts on the Internet know to use the slave server is the master is down.



Create Slave Zone

New slave zone options

Zone type	● Forward (Names to Addresses) ○ Reverse (Addresses to Names)
Domain name / Network	
Records file	○ None ● Automatic ○ ...
Master servers	Server port ● Default ○ port

Create

Return to zone list

The slave zone creation form

Another type of zone that is closely related to the slave zone is the stub. They are like slave zones, but only contain Name Server records that have been transferred from a master server, instead of all the records. Stub zones are rarely used, but can be useful for ensuring that the Name Server records in a zone for its sub-domains are the same as those use in the sub-domain itself. The steps for creating one are almost identical to those above, but in step 1 you must use the **Create a new stub zone** link on the main page instead.

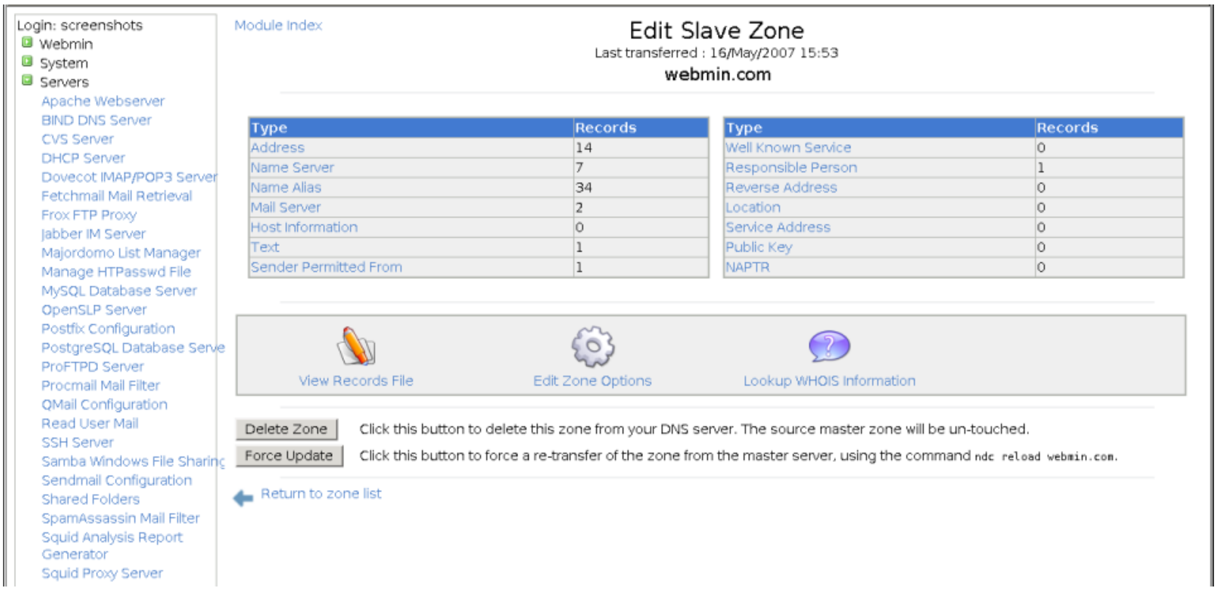
Editing a slave zone

After a slave zone has been created, it is still possible to edit several options that apply to it. Naturally there is no way to add or edit the actual records within the zone, but you can still change the list of master servers, the records file and the clients that allowed to query it. To change these setting, the steps to follow are:

1. On the module's main page, click on the icon for the slave zone that you want to edit. Your browser will display the form shown in the screenshot below.
2. Scroll down to the **Zone Options** form at the bottom of the page.
3. To edit the list of other master and slave servers for this zone, change the IP addresses in the *Master servers *field. If a new secondary server has been added, it should be added to this list on all other secondaries so that they can do zone transfers from it. If the IP address of the master has changed, the list must be updated with the new address.
4. To change the amount of time that the server will wait before giving up on a zone transfer, de-select **Default** for the **Maximum transfer time** field and enter a number of minutes into the text box next to it.
5. If the **Records file** field is set to **None**, records transferred from the master server for this zone will be kept in memory only. However if a filename is entered, records will be written to that file instead in the standard format. This is the best option, as it minimizes zone transfers and allows you to view the records on the secondary server, as explained below.
6. To have this DNS server notify others when the zone changes, change the **Notify slaves of changes?** field to **Yes**. This is only really useful if there are other secondary servers that perform zone transfers from this one, and may not be able to receive update notifications from the master server. The DNS servers to notify are determined from the Name Server records for the zone, and the contents of the *Also notify slaves* field.
7. By default, all DNS clients and servers will be able to lookup records in the zone. To change this, fill in the *Allow queries from* field with a list of IP addresses, IP

networks and BIND ACL names. If the field is left empty, the field with the same name on the global Zone Defaults page determines which clients are allowed.

8. To restrict the clients and servers that are allowed to perform zone transfers of all the records in this domain, fill in the **Allow transfers from** field with a list of IP addresses, IP networks and ACL names. If it is left empty, the *Allow transfers from* field on the Zone Defaults page applies instead.
9. The other fields on the form such as **Check names?** and *Allow updates from?* are not really used for slave zones, and so can be left unchanged.
10. When you are done making changes, click the Save button. As long as there were no syntax errors in your input, you will be returned to the module's main page. Click the **Apply Changes** button there to make the modifications active. Note that this will not always force a re-transfer of the zone, even if the master servers have changed. For slave zones that use records files, BIND will only do a transfer when it the zone expires or the server receives notification of a change.



Edit Slave Zone

Last transferred : 16/May/2007 15:53

webmin.com

Type	Records
Address	14
Name Server	7
Name Alias	34
Mail Server	2
Host Information	0
Text	1
Sender Permitted From	1

Type	Records
Well Known Service	0
Responsible Person	1
Reverse Address	0
Location	0
Service Address	0
Public Key	0
NAPTR	0

Click this button to delete this zone from your DNS server. The source master zone will be un-touched.

Click this button to force a re-transfer of the zone from the master server, using the command `ndc reload webmin.com`.

The slave zone editing form

When editing a slave zones that uses a records file, it is possible to browse the records in Webmin. At the top of the page that appears when you click on the slave zone's icon is a table of record types, just like the one that appears on the master zone form. Each can be clicked on to list the names and values of records of that type in the zone, as known to the secondary server. Editing or adding to them is impossible of course, as any changes must be made on the master server which is the authoritative source of records for the domain.

To stop your system acting as a slave server for a zone, you will need to delete it from the BIND configuration. This is generally a safe procedure, as the all the records in the zone have been copied from a master server and can be easily replaced. However, you should update the Name Server records in the zone and notify the parent domain's registrar that you system is no longer a secondary for the zone, so that other DNS servers do not waste time querying it.

To delete a slave zone, the steps to follow are:

1. On the module's main page, click on the icon for the slave zone that you want to edit. This will take you to the form shown in the screenshot above.
2. Click on the **Delete** button in the bottom right-hand corner of the page, which will display a confirmation form.
3. Hit the **Delete** button if you are sure you want to delete the zone.
4. After your browser returns to the module's main page, click on **Apply Changes** to make the deletion active.
5. On the master server, remove the Name Server (NS) record for this secondary server from the zone.

6. If this is an Internet domain, notify the parent zone registrar of the removal of this secondary server. Failure to do so could cause problems if other DNS servers attempt to query this one for records in the domain when it cannot provide answers.

The final thing that you can do to a slave zone is convert it to a master. This is only possible for zones that use a records file, so that Webmin can view and edit that file in future. If you do such a conversion, make sure that the original master server is changed to become a slave or stops hosting the zone altogether - the same domain cannot be served by two masters.

The steps to convert a zone are:

1. Click on its icon on the module's main page.
2. Scroll down to the bottom of the slave zone page, and hit the **Convert to master zone** button. This will immediately update the named.conf file to change the zone's type, but will not make any other changes.
3. To make the conversion active, click on the **Apply Changes** button on the module's main page.
4. You can now edit records in the domain just as you would with any normal master zone, by following the instructions in the section on **Adding and editing records**.

Creating and editing a forward zone

A forward zone is one for which your DNS server simply forwards queries to another server on behalf of whoever is making the request. They can be useful if the zone is actually hosted by another server that cannot be reached by clients of this server. It is possible to set up BIND to forward all requests for any non-hosted zones to another server, as explained in the *Configuring forwarding and transfers* section below. A forward zone entry does the same thing, but for just a single domain.

To set one up, the steps to follow are:

1. On the module's main page, click on the *Create a new forward zone* link above or below the list of existing domain icons. This will take you to the zone creation form.
2. Set the **Zone type** field to either **Forward** or **Reverse**, as when creating master and slave zones.
3. For a forward zone, enter its full name (without a dot at the end) into the **Domain name / Network** field. For a reverse zone, enter its network (like *192.168.1*) into the field instead - Webmin will automatically convert it to in-addr.arpa format when the zone is added.
4. In the **Master servers** field, enter a list of IP addresses for the DNS servers that can be queried to lookup records in the zone. These must all be master, slave or forward hosts for the domain. If no addresses are entered at all, BIND will always perform normal lookups of records in the zone instead of forwarding requests to another server. This can be used to override the global forwarding settings on the Forwarding and Transfers page for a single zone.
5. Click the **Create** button to have the zone added to BIND's configuration file. Your browser will be taken to a page for editing options in the new domain.
6. Return to the module's main page, and hit the **Apply Changes** button to make it active.

After a forward zone has been created, you can delete it or edit the few settings that it has by following these steps :

1. Click on the icon for the zone on the module's main page, which will bring your browser to a small form for editing its options.

2. To change the list of DNS servers that requests are forwarded to, edit the IP addresses in the **Master servers** field. If none are entered, requests for records in this domain will be looked up directly.
3. If the **Try other servers?** field is set to **Yes**, BIND will try a normal direct lookup for requests in this zone if it cannot contact any of the listed servers.
4. Click the **Save** button to store your changes, and then **Apply Changes** back on the main page to activate them. Or to delete the forward zone, click on **Delete** and then **Delete** again on the confirmation page.

Creating a root zone

As the introduction explains, a root zone is one that contains the information that your DNS server needs to contain the Internet root servers. Without one, it is impossible to resolve records in domains other than those hosted by your server. Fortunately, one will almost always exist already in your BIND configuration, created either by Webmin or included as part of the default setup.

You may need to create a root zone if one does not exist yet because you selected the **internal non-internet use only** option when setting up the module for the first time, but have now connected your system to the Internet. Adding a second root zone can also be useful when views have been configured, as explained in the **Using BIND views** section.

Webmin will only allow you to create a root zone if none exists yet, or if a view exists that does not contain one, because there is no point having two such zones. To add one, the steps to follow are:

1. On the module's main page, click on the **Create a new root zone** icon.
2. Fill in the **Store root servers in file** field with a filename to use for the root zone file. If one already exists, then this field will already contain its path - otherwise, you should enter something like /var/named/db.cache.
3. The **Get root servers from** field controls where Webmin copies the root file from. The choices are : *Download from root FTP server *This is the best option, as it tells the module to make an FTP connection to rs.internic.net and download the latest version of the file. However, this may not work if your system cannot make perform FTP downloads due to a firewall. *Use Webmin's older root server information *This option should be used if the first will not work. If selected, the module will use a copy of the root zone file that comes with Webmin, which will work but may not be up to date. *Existing root servers in file *If the file entered in step 2 already exists, then this option should be chosen. If you are adding a root zone to a view and one already exists in another view, it will be selected by default so that the file can be shared between both zones.
4. Click the **Create** button to add the zone and return to the module's main page. Then hit **Apply Changes** to make it active.

Once a root zone has been added, an icon representing it will appear on the main page. You can delete it by clicking on the icon and hitting the **Delete** button - however, this may prevent the lookup of records in non-hosted Internet domains from working as explained above.

Editing zone defaults

Defaults for new master zones

If you add lots of zones that contain similar records, then it can be a lot of work to add them manually after creating each one. For example, in a web hosting company all of your domains might contain a www Address record for the IP address of your webserver, and an Mail Server record that directs mail to a central server. Fortunately, Webmin allows you to create a list of records that get added to all new domains, called a zone template.

A template consists of one or more records, each of which has a name, type and value. For Address records, the value can be option which indicates that it can be entered by the user at zone creation time. This is useful if one of the records (such as www) in the new domains does not have a fixed address, and you want to be able to easily set it when the zone is added.

Templates can only be used when creating forward zones, as they do not make much sense for reverse zones.

It is also possible to edit the default expiry, refresh, TTL and retry times for new zones. Webmin's initial defaults are reasonable, but may not be appropriate for your network. To change these defaults and set up template records, the steps to follow are:

1. On the module's main page, click on the **Zone Defaults** icon. The form at the top of the page labeled **Defaults for new master zones** contains all the fields that need to be edited.
2. Edit the **Refresh time**, **Transfer retry time**, **Expiry time** and **Default time-to-live** fields if you want to change the defaults times for new zones. Existing master zones will not be effected by any changes you make here though.
3. If all your new domains are managed by the same person, enter his address into the **Default email address** field. This will save you from having to type it in on the master zone creation page every time.
4. In the **Template records** table, two blanks rows appear for entering new records. To add more than two, you will need to save this page and re-edit it. The records in existing rows can be edited by just changing their fields, or deleted by clearing out the record name. Under the **Record name** column you must enter the name of the record relative to the zone, such as www or ftp. To create a record for the zone itself (such as a Mail Server record for the domain), just enter a single dot. Under the **Type** column, select a type for the record from the list. See the **#Record types** section for more information on what each is used for. As its name suggests, the field under the **Value** column is for entering a value for the new record. For the Address type, you can select **From form** in which case you will be able to enter an address when creating a new domain, which will be used by all template records that have this option selected. For Mail Server records, both the priority and server name must be entered separated by a space, such as _5 mail.example.com._. Values for records of all other types should be entered in the same format as is used when adding a record to a zone.
5. If you are familiar with the records file format used by BIND, you can create your own file of records to be included in new zones. If a filename is entered into the **Additional template file** field, its contents will be added to the zone file created by Webmin for new master domains.
6. When you are done adding template records, click the **Save** button at the bottom of the page. The changes will apply to any new master zones created from now on.

Now that you have created a template, you can choose whether or not to use it for each new master zone that you create. On the creation form (explained in the **Creating a new master zone** section) is a field labeled **Use zone template?**, which is set to **Yes** by default if there are any template records. Next to it is a field named **IP address for template records**, which used for entering the IP for records for which the **From form** option is selected. If you chose to use a template and if there are any records that do not have an IP address specified, then this field must be filled in.

Default zone settings

At the bottom of the **Zone Defaults** page you will find several options that apply to all existing domains, but can all be set or overridden on a per-zone basis as explained in the **Editing a master zone** section. You can control which clients are allowed to query the server, and what kind of checking is done for the records of various domain types. Being able to limit the allowed client hosts is particularly useful, so that you can stop non-internal clients using your DNS server. However, you should make sure that master Internet zones hosted by your server are accessible to everyone, so that other DNS servers on the Internet can look them up.

To change these global options, the steps to follow are:

1. On the module's main page, click on the **Zone Defaults** icon and scroll down to the **Default zone settings** section.
2. To control which hosts are allowed to query your DNS server, change the **Allow queries from** field to **Listed** and enter a list of IP addresses, IP networks (like 192.168.1.0/24) and ACL names into the text box below. Clients that do not match any entry on the list will be denied, unless they are requesting a record in a zone which has its own separate settings allowing them.
3. To control which hosts are allowed to perform zone transfers from your server, change the **Allow transfers from** field to **Listed** and fill in the text box below with a list of IP addresses, IP networks and ACL names. Only servers that are acting as secondaries for zones that this server hosts really need to be able to do transfers, so it is usually a good idea to enter just their IP addresses. If you are restricting queries, this field must be filled in so that hosts that cannot lookup records are not allowed to perform transfers either.
4. The fields **Check names in master zones?** and **Check names in slave zones?** control the checking of records in all zone files for master and slave zones respectively. The available options for each are:

Warn
If an invalid record is found, an error will be written to the system log file but processing of other records continues normally.

Fail
Invalid records cause the entire zone to be rejected, but other zones will still be processed normally.

Ignore
No checking is done at all.

Default
The default checking level is used, which is **Fail**.

5. To have BIND check responses that it receives from other DNS servers, set the **Check names in responses?** field to **Warn** or **Fail**. The default is simply to pass potentially erroneous responses on to clients.
6. The **Notify slaves of changes?** field determines whether BIND sends a notification to all slaves of master zones hosted by this server when they change. To turn this on, select **Yes -** otherwise, select **No** or **Default**. Enabling notification is a good idea, as it ensures that secondary servers are kept in sync with the master.
7. When done, click the **Save** button at the bottom of the page to update the BIND configuration file, and then the **Apply Changes** button on the module's main page to make the changes active. The new settings will apply to all zones that do not explicitly override them on their own options pages.

Configuring forwarding and transfers

BIND can be configured to forward all requests for zones that it is not the master or slave for to another DNS server. When doing this, it acts like a DNS client itself, accepting requests from real clients and then sending them off to another server or servers for resolution instead of carrying out the normal process of contacting the root zone servers and finding the correct server for the domain. This can be useful if your DNS server is unable to contact the rest of the Internet, but can still communicate with a DNS server that does have full network access. For example, it may be on an internal network behind a firewall that only allows connections to a limited set of destinations.

To set up forwarding, the steps to follow are:

1. On the module's main page, click on the **Forwarding and Transfers** icon.
2. In the form that appears, fill in the **Servers to forward queries to** field the IP addresses of DNS servers that requests should be sent to. BIND will try them in order until one returns a positive or negative a response. If the list is empty, the

server will revert to the normal method of looking up records by contacting the root servers and so on.

3. If you want your server to attempt to resolve a client's query directly when it cannot contact any of the forwarding servers, set the **Lookup directly if no response from forwarder** field to **Yes**. This is only useful if your server is actually capable of doing lookups.
4. Click the **Save** button at the bottom of the page, and then hit **Apply Changes** back on the main page to make the new setting active. Assuming the forwarding list was filled in, your server will now send all client queries to the listed servers.

The same form also contains fields for configuring BIND's behavior when doing zone transfers. You can control how long it will wait for a transfer to complete, the protocol used for transfers and the number that can be active at the same time. To edit these settings, follow these steps:

1. On the module's main page, click on the **Forwarding and Transfers** icon.
2. By default, BIND will wait 120 minutes (2 hours) for a zone transfer from a master to complete. To change this, enter a different number of minutes into the *Maximum zone transfer time* field. This can also be set or overridden on a per-slave zone basis.
3. BIND versions before 8.1 only support the transfer of a single zone at a time. Because this can be slow when transferring many zones from the same master server, the *Zone transfer format* field can be set to **Many** to use a new format that combines multiple domains into the same transfer. If **One at a time** or **Default** is chosen, then each zone will be transferred separately. This is the best choice unless you are sure that all slave servers are running BIND 8.1 or above.
4. By default, your nameserver will not carry out more than 2 concurrent zone transfers from the same master server. To increase this limit, change the *Maximum concurrent zone transfers* field to something higher. This can speed up the process of transferring a large number of domains, but at the expense of putting a higher load on the master server.
5. Click the **Save** button when you are done making changes, and then **Apply Changes** on the main page to activate them. The new settings will apply to all subsequent zone transfers.

Editing access control lists

An access control list (or ACL) is list of IP addresses, IP networks or other ACLs that are grouped together under a single name. The ACL name can then be used when specifying the list of clients allowed to query, update or perform zone transfers from a zone. This can make be used to reduce the amount of duplication in your BIND configuration, and to make it clearer. For example, the ACL *corpnet* might match the IP networks *192.168.1.0/24*, *192.168.2.0/24* and *1.2.3.0/24*, which are all part of your company's network. When configuring who can query a zone, you could just enter *corpnet* instead of that list of network addresses. To view and edit ACLs in Webmin, the steps to follow are :

1. On the module's main page, click on the **Access Control Lists** icon. This will take you to a page listing existing ACLs and allowing the addition of one more. If you want to add more than one ACL, you will need to save the form and re-edit it to force the display of a new blank row.
2. To add a new ACL, find the blank row at the bottom of the table and enter a short name consisting of only letters and numbers in the **ACL Name** column. Then in the field under *Matching addresses, networks and ACLs*, enter a list of IP addresses, IP networks and other ACL names that this new ACL will contain. IP addresses must be entered in their standard format like *192.168.1.1*, but hostnames are not allowed. IP networks must be entered in network/prefix format like *192.168.1.0/24* or *192.168.1/24*. You can also precede any address, network

or ACL name with a ! to negate it, so for example the entry *!192.168.1.0/24* would match all hosts outside the_ 192.168.1 _network.

3. Existing entries in the list can be edited by changing their fields in the table, and ACLs can be deleted by clearing out the field containing their names.
4. When you are done adding and editing ACLs, click the **Save** button. To activate the changes, hit **Apply Changes** back on the main page. As soon as an ACL is created, it can be used in other query, transfer and update restrictions of other zones.

BIND has four built-in ACLs that can be used in all the same places that user-defined ACLs can. They are:

any

Matches any client address.

none

Matches nothing.

localhost

Matches the IP addresses of all network interfaces on your system. Even though it is called localhost, it doesn't just match 127.0.0.1.

localnets

Matches all clients on all networks that your system is directly connected to. BIND works this out by looking at the IP addresses and netmasks of all network interfaces.

Setting up partial reverse delegation

Partial reverse zone delegation is method for transferring the management of a small set of reverse IP addresses to another DNS server. Normally, reverse zones cover an entire class C network containing 256 addresses. However, many organizations have networks much smaller than this, containing maybe 16 or 32 addresses. Normally, this would make it impossible for the organization to manage its own reverse address mappings, as the addresses come from a network that is owned by an ISP or hosting company.

Fortunately, there is a solution - the ISP can set up Name Alias (CNAME) records in the reverse zone for the parent network that point to Reverse Address records in a special zone on the organization's DNS server. The parent zone must also contain a Name Server (NS) record for the special sub-zone that points to the customer's server, so that other DNS clients know where to look when resolving the Name Alias records.

An example may make this clearer - imagine for example than an ISP had granted addresses in the range 192.168.1.100 to 192.168.1.110 to Example Corporation, which owns the example.com domain. The company already runs its own DNS server to host the example.com zone, but wants to control reverse address resolution for its IP range as well. The ISP would create Name Alias records in the 192.168.1 zone pointing to the special sub-zone 192.168.1.100-110, which will contain the actual Reverse Address records named like 192.168.1.100-100.101. The ISP also needs to create a Name Server record for 192.168.1.100-110 which tells other servers that Example Corporation's DNS server should be used to find records under that zone.

Webmin handles reverse address delegation well, and automatically converts special network zones like 192.168.1.100-110 to and from the real zone names used by BIND such as 100-110.1.168.192.in-addr.arpa. The exact steps to follow on both the server that hosts the actual class C network zone and the server that a subset of it is being delegated to are :

1. Decide on the range of addresses that are being delegated, such as *192.168.1.100* to *192.168.1.110*. Typically, the sub-zone name is based on the range of addresses being delegated, but this does not have to be the case as long as it is under the parent network domain.
2. On the server that hosts the class C network zone, add a Name Server record for *192.168.1.100-110* with the server set to the IP address or name of the sub-zone's DNS server.

3. For each address in the range, add a Name Alias record to the reverse zone named like *101.1.168.192.in-addr.arpa.* with the **Real Name** set like *101.100-110.1.168.192.in-addr.arpa*. As you can see, the alias points to a record inside the zone for the sub-network.
4. When all of the Name Alias records have been created, everything that needs to be done on this server is finished and you can hit **Apply Changes**.
5. On the DNS server for the sub-network, create a new master zone for the reverse network *192.168.1.100-110*. Webmin will automatically convert this to the correct in-addr.arpa format for you.
6. Add Reverse Address records to the new zone as normal for IP addresses like *192.168.1.100-110.101*. Adding a record for the IP 192.168.1.101 will not work.
7. When you are done creating reverse records, click the *Apply Changes* button to make them active. You should now be able to look them up using a command like nslookup on the server for the parent network zone.

The instructions above can be used to delegate multiple ranges from a single class C network to several different DNS servers. There is no limit on the size of ranges, nor any requirement that they follow normal network block boundaries - however, for routing reasons most IP allocation is done in power-of-two sized blocks (like 4, 8, 16 and so on), which means that any sub-zone ranges will be the same size.

The only problem with reverse address delegation when using Webmin is that Reverse Address are not automatically created and updated when Address records are. This means that you will have to create all such records manually on the sub-zone server, as in the steps above.

One inconvenience in setting up partial reverse delegation is the number of similar Name Alias records that must be created on the parent network zone server. Fortunately, there is a simpler alternative - record generators. A generator is a special BIND configuration entry that creates multiple similar records using an incrementing counter. This module allows you to created and edit generators, by following these steps :

1. On the module's main page, click on the icon for the reverse zone that you want to create records in. This will typically be a class C network domain that is going to have a range of addresses delegated to some other Server.
2. Click on the **Record Generators** icon. This takes you to a page containing a table of existing generators, with a blank row for adding a new one.
3. In the empty row, select **CNAME** from the menu under the **Type** column.
4. Under the **Range** column, enter numbers for the start and end of the address range into the first two fields, such as *100* and *110*. The third field is for entering the gap between each step, and should be left blank. If you were to enter 2, then the range would go *100*, *102*, *104* and so on.
5. In the **Address pattern** field, enter _$_ (a single dollar sign). When the records are created, the $ will be replaced with the number of each record, which will in turn resolve to an IP address in the range. You could also enter *$.1.168.192.in-addr.arpa.*, which makes things more obvious but is longer to type.
6. In the **Hostname pattern** field, enter *$.100-110*. Similarly, the $ will be replace with the number of each record, which will resolve to something like *101.100-110. 1.168.192.in-addr.arpa.*.
7. If you like, enter a comment that describes what this generator is for into the **Comment** field.
8. Click the **Save** button, return to the module's main page and click on **Apply Changes**.

A generator can replace the Name Alias records that the first set of instructions in this section tell you to create, so there is no need for them anymore. Note that the automatically generated replacements will not be visible or editable in the normal way, only through the Record Generators page.

Using BIND views

BIND version 9 introduced the concept of views, which are groups of zones that are visible only to certain DNS clients. Views can be used to hide internal zones from the Internet, to present the same zone in two different ways, or to stop non-local clients resolving non-hosted domains through your server. Every view has a unique name, and a list of matching IPs addresses and IP networks that determines which clients and servers it is visible to.

When it detects that you are running BIND 9, several additional features are available in the module. You can create views, move zones from one view to another, and choose which view zones are created in. On the main page, each current view is represented by an icon under **Existing Client Views** heading, and each zone icon has a label that indicates which view it is in.

If any views exist, then every zone must be in a view. Only if none are defined will Webmin allow the creation of zones outside views, as this is not supported by BIND. This includes the root zone, which must be available to a client for DNS requests for records in domains not hosted by this server to succeed. For this reason, it often makes sense to put the root zone in a view that is available to all clients.

To add a new view to your BIND configuration, the steps to follow are:

1. On the module's main page, click on the **Create a new view** link in the **Existing Client Views** section. This will take you to a form for entering its details.
2. Enter a short alphanumeric name for the view (such as *internal* or *everyone*) into the **View name** field. Each view must have a unique name.
3. Leave the **DNS records class** field set to **Default**.
4. If this zones in this view are to be visible to everyone, set the **Apply this view to clients** field to **All clients**. Otherwise, choose **Selected addresses, networks and ACLs** and enter a list of IP addresses, IP networks and BIND ACL names into the text box below. Only clients that match one of the entries in this list will have access to the view.
5. Click the **Create** button at the bottom of the form. You will be returned to the main page, which will include an icon for your new view.
6. Move any existing zones that you want to be in this view into it. A zone can be moved by clicking on its icon, then on *Edit Zone Options*, and then selecting the new view from the menu next to the **Move to view** button before clicking it. If this is your first view, all existing zones must be moved into it (or another view) before the new configuration will be accepted by BIND.
7. When you are done moving zones, click the **Apply Changes** button on the main page.

Once a view has been created, you can change the list of addresses and networks that it matches by clicking on its icon on the main page and updating the **Apply this view to clients** field. Then hit the **Save** button followed by **Apply Changes** to make the new client list active.

A view can be deleted by clicking the **Delete** button on the same form. This will bring up a confirmation page that allows you to choose what should happen to the zones that it contains, if any. The available options are:

Delete totally

All zones in the view are deleted, along with their records files.

Move out of views

Zones in the view are moved out to the top level. This option should only be used when deleting the last view, for the reasons explained above.

Move to view

Zones are moved to a different existing view.

When one or more views have been defined on your system, you can choose which one to use when adding new zones. This is done using the **Create in view** field on the master, slave, forward and root zone creation forms, which allows you to select a view from its menu. Naturally, there is no option for creating a zone outside of any views as this is not allowed by BIND.

One common use of views is hiding internal zones from clients outside your internal network. This is a good way of hiding the structure of your network and the hosts on it from potential attackers. To set it up, the steps to follow are:

1. Create a new view called *internal* that matches clients on your internal LAN.
2. Create a second view called *everyone* that matches all clients.
3. Move any zones that are for internal use only into the *internal* view. Zones for Internet domains such as *example.com* must not be put in this view, as that would make them inaccessible to the rest of the world.
4. Move all other zones (including the root zone) to the *everyone* view.

Views can also be used to prevent clients outside your network looking up non-hosted domains on your server, as follows:

1. Create a new view called *internal* that matches clients on your internal LAN.
2. Create a second view called *everyone* that matches all clients.
3. Move the root zone to the *internal* view, which will prevent the server from looking up records for non-local clients that require contact with the root servers.
4. Move all other zones to the *everyone* view.

Module access control

Like others, the BIND DNS Server module allows you to control which of its features are available to a particular Webmin user or group. This can be useful for giving people the rights to manage only records in their own zones and nobody else's. Even though this would normally require root access to the records files, with Webmin it can be granted to people without giving them level of power that a root login would allow.

Once you have created a user with access to the module as explained on WebminUsers, the steps to limit his access to only certain zones are:

1. Click on the BIND DNS Server next to the name of the user in the Webmin Users module. This will being up a page of access control options.
2. Change the **Can edit module configuration?** field to **No**, so that the user is not allowed to change the paths that the module uses to named.conf and other files.
3. For the *Domains this user can edit *field, choose *Selected zones* and select the ones that you want him to have access to from the list to its right. If you want him to be able to edit almost all zones, it may be better to choose **All except selected** and select only those that he should not be allowed to manage records in. If your DNS server uses views, you can use the *Zones in view* options to allow or deny access to all zones in a view as well.
4. Change the fields **Can create master zones?**, *Can create slave/stub zones?*, **Can create forward zones?** and *Can edit global options?* to **No**.
5. If you want Reverse Address records in zones that the user does not have access to to be updated by changes to Address records in zones that he does, set the *Can update reverse addresses in any domain?* field to **Yes**. This may not be a good idea from a security point of view though, as he would be able to change almost any existing Reverse Address record on your system. For that reason, I suggest that this field be set to **No**.
6. To stop the user creating more than one Address record with the same IP, set the *Can multiple addresses have the same IP? **field to *No**. Even though creating

multiple records is harmless, you may want to set this to **No** to prevent the user allocating the same IP twice.

7. Leave the **Read-only access mode?** field set to **No**. If it is changed to **Yes**, the user will only be able to view zones and records using the module, and not change anything. This might be useful for creating a different kind of restricted user though - one who can see all settings, but not edit them.
8. Leave the **Can apply changes?** field set to **Yes**, so that he can use the **Apply Changes** button to make his additions and modifications active.
9. Unless you want the user to be able to edit his records file manually, change the **Can edit records file?** field to **No**. Most un-trusted users are not smart enough to perform manual editing.
10. The **Can edit zone parameters?** field determines if the user can see and use the **Edit Zone Parameters** icon for his domains. Setting this to **Yes** is quite safe, as the user can only harm his own zones by setting the parameters to silly values.
11. Similarly, the **Can edit zone options?** field determines if the **Edit Zone Options** icon is visible or not. You should set this to **No**, as it is possible for a user to create a syntax error in named.conf by improper use of the zone options form.
12. Unless you want the user to be able to delete his own domains, change the **Can delete zones?** field to **No**. Users should contact the master administrator instead if they can to delete zones.
13. The **Can edit record generators?** field can be left set to **Yes**, as it simply allows the creation of multiple records at once. However, some users may get confused by this feature so it might be a good idea to change the field to **No**.
14. The **Can lookup WHOIS information?** And *Can search for free IP numbers?* fields can also be left on **Yes**, as those features mere display information to the user.
15. Change the **Can create and edit views?** field to **No**, so that the user cannot manage BIND 9 views. If the user is allowed to create zones, you can use the *Views this user can edit and add zones to* field to limit those that he can create zones in.
16. **Can create slave zones on remote servers?** should be set to **No**, but this doesn't really matter as the user is not going to be allowed to create master or slave zones anyway.
17. Finally, click the **Save** button to make the new restrictions for the user active.

See also:

- Resolution for Virtual Hosts