

11

SENDMAIL



Sendmail is the de facto standard mail transfer agent, or *MTA*, in use on the Internet today. While there are now several worthy contenders for the title of best or most popular *MTA*, including Postfix and QMail (both of which have very good Webmin modules, and Postfix is documented in the preceding chapter), more mail probably passes through Sendmail than any other single *MTA*.

An *MTA* is the software that provides mail services for a network. A client mail user agent, or *MUA*, sends email, usually via the Simple Mail Transport Protocol, or *SMTP*, to the *MTA*. The *MTA* uses one of several transport protocols, most often via *SMTP*, to deliver it either directly to the recipient (if the address is served by the same server) or to the mail server for the user. Clients then access the mail on the server using either *POP3* or *IMAP*. So, Sendmail will operate on your server and provide those intermediary services, both sending and receiving mail, for clients and other *MTAs* on the Internet.

Configuring Sendmail

Sendmail has a reputation, not entirely undeserved, for being extremely obtuse and confusing to configure. The famously terse `sendmail.cf` file was designed to be easy and quick for the computer to parse, not for humans to be able to read and edit. Relatively recently, attempts have been made to remedy this problem, and the solution now provided with Sendmail is an “m4” macro-based configuration file, called `sendmail.mc` by default, that allows you to use much more human comprehensible configuration constructs. This configuration file is also supported by Webmin in the **Sendmail M4 Configuration** page.

Sendmail also uses a few other configuration files to dictate certain other behaviors. These include `aliases`, `mailertable`, `access`, and `domaintable`. These files are quite readable by mere mortal humans, usually containing a few (or more than a few in large networks) names, hosts, or domains, and an option or permission that applies to the name, host, or domain. Webmin provides a nice interface for these files as well, so you won't have to deal with them directly. However, it is good to know about them and what they're used for. You'll learn about them in more detail as the relevant Webmin sections are discussed.

The Sendmail Module

The Sendmail module in Webmin is thoroughly comprehensive, and it provides one-to-one access to nearly all, if not all, of Sendmail's important options and features. Opening the main Sendmail module page provides you with a number of icons that represent each type of option in the Sendmail configuration files (Figure 11-1). From here, you can edit the various global options, edit aliases, configure user mail settings, restrict access to your mail server, and even control the mail queue and read mail.



Figure 11-1: Sendmail module

Sendmail is configured in a number of files. The first, and most intimidating, is the `sendmail.cf` file. This configures all of the various limits and behaviors of Sendmail. The rest are related to users, hosts, domains, and aliases. They dictate primarily to whom mail is sent, and who and what hosts or networks have permission to send and receive mail from the server. The `sendmail.cf` file is configured on the **Options** page, discussed in the next section, while each of the other option files have their own page which are discussed in the Other Files section.

Options

The Sendmail Options page provides access to most of the relevant `sendmail.cf` directives. These options are usually “set and forget”-type options. Unless you have a problem with load, or memory, or untimely failed message delivery, you will have little reason to alter these options after first setting up your Sendmail system.

Send outgoing mail via host

This option sets whether outgoing mail will be sent directly or via another mail server. If it is to go through another mail server, it is entered here. If you are not on a permanently attached network (i.e., with a permanent IP and a domain name), then you should relay through the mail server at your ISP, as many mail servers refuse mail from hosts that cannot be resolved. This option edits the *Smart Host* macro named `S` in `sendmail.cf`. More on this and other common macros can be found in the macros sections of the *Administrative Details* [<http://www.sendmail.org/m4/admin.html>] documentation page at the Sendmail home page.

Forward unqualified usernames to host

If an email is sent by a local user that is unqualified (i.e., *joe* instead of *joe@swelltech.com*) it will by default attempt to deliver the mail locally (i.e., on the same machine that Sendmail runs on). However, if this is set, mail will be forwarded to the host selected. This is useful if you have a large organization with many Sendmail servers running, but you’d like all local mail to be delivered to a single host. This option edits the `R` macro, which refers to the *Relay* (for unqualified domains) configuration.

Forward mail for local users to host

Email that is received from anywhere that is destined for a user on the local host will be delivered locally, unless this option is set. This allows all mail to be collected on a single host. This option edits the `R` macro, which refers to the *Relay* (for unqualified domains) configuration.

Delivery mode

This option controls how messages will be scheduled for delivery. If `Background` is selected, Sendmail will deliver messages as soon as possible silently in the background. `Queue only` places mail into a queue to be delivered upon a manual or periodically scheduled flush of the queue. Interactive messages are delivered immediately synchronously. `Deferred` is like `Queue only`, except Sendmail will not attempt to resolve host names until the queue is flushed (ideal for a sporadic net connections, such as a dial-up). This option configures the `DeliveryMode` directive.

Max load average for sending

This option determines at what load average Sendmail will no longer continue to send messages. If this load average is crossed, Sendmail will queue messages for later delivery. This option configures the `QueueLA` directive and defaults to 8. If your system is becoming overloaded at times with delivering mail, it may be possible to tune this setting to help ease the load.

Max load average for receiving

This option determines at what load average Sendmail will stop accepting incoming messages via SMTP. This forces other mail servers to queue them for later delivery. While most mail servers will be polite and actually save the refused messages for later delivery, if assurance of mail service is important to your users, it is probably best to avoid refusing mail. This option configures the `RefuseLA` option and defaults to 12.

Max child processes

Controls how many child processes Sendmail will spawn in order to handle incoming mail. Limiting this allows you to control, somewhat, the memory footprint of Sendmail. This option configures the `MaxDaemonChildren` directive and defaults to 12.

Max connections/second

Configures the maximum number of new connections per second that Sendmail will accept. With this option, you may limit the CPU and memory usage of Sendmail on your system, or in high load environments allow Sendmail to receive a larger volume of mail. This option configures the `ConnectionRateThrottle` directive and defaults to 3.

Min time before retrying send

This sets the minimum amount of time mail will wait in the queue after a failed send attempt, before Sendmail attempts to re-send it. Values can be in seconds (e.g., 45s), minutes (30m), hours (2h), days (3d), or weeks (1w). This correlates to the `MinQueueAge` directive and defaults to 30m.

Maximum queue size

Determines the maximum number of queued jobs Sendmail will process in a single queue run. This correlates to the `MaxQueueRunSize` directive and defaults to 1000. This should remain as high as possible to avoid losing jobs that fall late in the queue.

Time before giving up

This is the amount of time that Sendmail will continue to try resending a failed message before giving up and considering it undeliverable. Non-permanent delivery failures can occur for a number of reasons, including network connectivity problems, DNS resolution failure, the recipient server not responding, and so on. When this limit is reached, a bounced message will be sent to the sender, and the message will be discarded. This option accepts times in the same format as discussed above, for seconds, minutes, hours, days, and weeks. This configures the `Timeout.queuereturn` directive and defaults to 5d.

Time before sending warning

In the event of a non-permanent delivery failure, as discussed in the previous option, this option configures how long Sendmail will wait before sending a warning to the sender of the message that a problem has occurred. Because these warnings usually resolve themselves shortly (either the network comes back up, DNS resolves again, the recipient server returns to service, and so on) it is often not necessary to trouble the sender with an error message until it begins to appear that a problem might become a permanent failure. This option configures the `Timeout.queewarn` directive and defaults to 4h.

Mail queue directory

This sets the location of your mail queue directory where Sendmail stores queued mail. This option correlates to the `QueueDirectory`, and often defaults to `/var/spool/mail`, though on some systems this may differ. There is rarely reason to change this.

Send error messages to

In the event of a problem, such as a delivery failure, error messages will be sent to some user on the system. This is usually `Postmaster`, which on many systems is aliased to `root`. This option correlates to the `PostMasterCopy` directive.

User forward files

This option dictates where Sendmail will search for forwarding information for users. This is a colon-separated list (much like the shell `PATH` environment variable). This option allows you to use variables to include certain values, such as username (`$u`), user's home directory (`$z`), and system host name (`$w`). So, for example, if I wanted to search first for `/var/forward/username` and then in `/home/joe/.forward`, I could enter `/var/forward/$u:$z/.forward`. This option configures the `ForwardPath` directive and usually defaults to `$z/.forward.$w:$z/.forward`.

Min free disk space

If the amount of free disk space is lower than this value, Sendmail will refuse to accept messages from other systems. Allows one to prevent Sendmail from filling the disk on which the queue resides. This option correlates to the `MinFreeBlocks` directive and often defaults to 100.

Max message size

This option sets the maximum size of a message that will be accepted by Sendmail. Any message over this size, either received from a local user or a remote mail server, will be bounced. This option configures the `MaxMessageSize` directive and defaults to 1000000.

Log level

This option sets the logging behavior of Sendmail. Logging levels 0-10 are, by convention, used for useful information that is probably worth logging on any system. The default logging level is 9 and is a good middle ground, wherein Sendmail usually only logs things that an administrator would want to be aware of. Higher log levels between 10 and 64 will provide much more information, while levels over 64 are reserved for extremely verbose debugging output. The normal log levels are documented in the *The System Log* [<http://www.sendmail.org/~ca/email/doc8.10/op-sh-2.html#sh-2.1>] section of the *Installation and Operation Guide* [<http://www.sendmail.org/~ca/email/doc8.10/op.html>]. This option configures the `LogLevel` directive.

MIME-encode bounce messages?

This option configures whether Sendmail will encode bounce messages in multi-part MIME format or as a plain-text message. Most mail clients today support MIME encoded messages, but if your client base has problems with this you may turn it off. This correlates to the `SendMimeErrors` directive and defaults to `True`. More on MIME can be found in *RFC 2045* [<http://www.ietf.org/rfc/rfc2045.txt?number=2045>] and *RFC 1344* [<http://www.ietf.org/rfc/rfc1344.txt?number=1344>].

File security options

In order to avoid cracking attempts, Sendmail checks most of its support files. If these files are in group writable directories, or some other risky configuration, Sendmail will ordinarily refuse to run. This option allows you to turn off this checking in the ways described by the options available. This option configures the `DontBlameSendmail` directive and defaults to `Safe`. For obvious reasons, it is strongly suggested that you solve the permissions problem(s) you may have, rather than turning off any of these checks.

Other Support Files

The other side of configuring Sendmail is setting up how it will deliver mail and who it will allow to use its services. The rest of the Sendmail module is devoted to these options, and you are likely to spend more time on these pages than on the Options page. These pages configure all of the other files that Sendmail relies on to tell it how to do its job, including the `aliases`, `access`, `domaintable`, `mailertable`, `relay-domains`, and `virtusertable` files.

Mail Aliases

Sendmail provides a means to direct mail to a given recipient under an *alias*. For example, it is possible to have mail sent to `Postmaster` delivered to `root`. It is also possible to direct mail, to several addresses, into a file, to feed it to a program, or provide an automatic reply. These aliases are stored in a file called `aliases` that is usually located in `/etc`.

Address

This is simply the address that will be the alias. When mail is sent to this address, the action defined in `Alias to` option below will be performed. The alias does not contain the domain name. For example, `joesalias` instead of `joe-salias@swelltech.com`.

Enabled

Here you may mark an alias as enabled or disabled. A disabled address will be preceded by a `#` in the `aliases` and will appear in *italics* in the list of aliases in the Webmin display.

Alias to

Here you define what Sendmail does when it receives a message for this aliased address. There are several options for this, and they are selected from the drop-down list. `Email address` is simply another email address to deliver the mail to. `Addresses in a file` causes the mail to be sent to every address named in the file provided in the text entry field — this allows you to more easily allow users to create their own aliases without giving them access to the `/etc/aliases` file. `Write to file` will cause Sendmail to write every mail sent to the address to a file chosen in the text entry field. `Feed to program` is interesting, in that it allows you to direct mail to any program on your system, thus you could write a script or a program (or find one already written) to provide any number of services based on the email received. Or it could file your mail in a database, or customer service system, or any number of other useful things. Finally, `Autoreply from file` simply sends a mail automatically back to the sender containing whatever is in the file listed in the text field.

The rest of the page is devoted to a listing of existing aliases. As mentioned above, enabled entries are in plain text, while disabled entries are in *italics*. Clicking an alias allows you to edit, delete, or add destinations to an alias. Clicking `Manually edit /etc/aliases` provides a simple text editor field wherein you can

manually edit or view your aliases file. Be careful, as the format and entries will not be checked by Webmin for correctness. If you make a mistake your Sendmail may complain loudly (in the logs) and stop working.

Local Domains

This page configures the `sendmail.cf` file and allows you to choose what domains Sendmail will accept local mail delivery for. By default Sendmail only accepts delivery for the local host. In order to accept mail for a whole domain, or a number of domains, they must be listed here. Also, the domains must have a DNS MX record that points to the server where your Sendmail is running. Sendmail can handle mail for any number of domains; however, setting up virtual hosting with Sendmail is a little tricky. A good document that describes the technique can be found on the *Virtual Hosting With Sendmail* [<http://www.sendmail.org/virtual-hosting.html>] page. Virtual hosting is also discussed briefly in the tutorial section of this chapter.

NOTE *For virtual hosting in Sendmail, you will also need to perform some configuration in **Address Mapping** and possibly in **Outgoing Addresses (generics)**.*

Domain Masquerading

The Domain Masquerading page provides access to the domain masquerading features of Sendmail. This allows you to make all outgoing messages appear to be from the same domain. When a domain masquerading rule is in place, Sendmail will replace the From address of all outgoing mail to appear to come from the domain to be masqueraded as. Also see the **Local Domains** page for more as well as a helpful link regarding virtual hosting in Sendmail.

Trusted Users

Ordinarily, Sendmail will not allow a user to claim to be a different user. However, if listed here, users will be trusted to claim they are another user or from another domain. Care should be taken when using this option, as it is one of the safeguards against *spoofed* email addresses.

Address Mapping

An Address mapping is similar to an alias, except they are able to handle domain information, as needed by virtual domains. To create an address mapping, all you must do is enter the address or domain to act upon. And a send to action to perform on each message as it is received by Sendmail. For example, if I host the domain `penguinfeet.org` on my `swelltech.com` server, then I must set up a method for mail sent to `sysadmin@penguinfeet.org` to make it into my mailbox at `joe`. So, I would create a map wherein the **Mail for** address is `sysadmin@penguinfeet.org` and the **Send to** address is `joe`.

Domain Routing

This option provides a special type of gateway in which your server accepts mail for a domain or host, but then passes it on to another specific mail server. This can be of use in environments where one or more subdomains have their own mail server that is behind a firewall and cannot directly deliver or receive mail on its own. Also, this allows Sendmail to provide gateway/proxy/translation services, if the other mail server does not support common transports and protocols. This use is in decline, as the vast majority of mail servers (even the few not running Sendmail) now speak the common protocols. These domains should not be listed in **Local Domains**, as then the server would accept the mail for local delivery, which is not what is desired in this case. You should still have a DNS MX record that points to the Sendmail server for each of the domains listed here, so that mail will be sent first to this system, where it then will handle it in whatever way is defined.

Mail for

This field allows you to enter a host or domain for which Sendmail will accept mail. It will not deliver the mail locally, but will instead pass it on to another server.

Delivery

Here you select how Sendmail will deliver the mail for the selected domain. The most common method is SMTP; however, Sendmail supports a wide range of delivery methods.

Send to

This should be the mail server where mail for this domain should be forwarded to. Checking the **Ignore MX for SMTP delivery** box will cause Sendmail to ignore MX entries in the DNS server for the domain and send to an explicitly selected server.

Outgoing Addresses (Generics)

Here you define mappings that Sendmail will use to modify the From addresses of outgoing mail (either from local users or from other hosts for delivery to other servers). This can be useful if you host multiple domains on the same mail server and want mail from some users to be addressed as though coming from those other domains. You must also include any domains to be remapped on the **Outgoing Domains** page, before Sendmail will perform any remappings on an address. Also, this mapping will not affect mail delivered to local users unless your .cf contains support for FEATURE('always_add_domain').

NOTE *This option is not enabled by default in most Linux distributions and other operating systems, nor in a default Sendmail installation. You must add the genericstable feature to your sendmail.mc file and regenerate the .cf file. The procedure for regenerating a .cf is documented later in this guide.*

Mail from

Here you enter a username or full email address for a user to remap.

Change to

Here you enter the address to change the above address into.

Manually edit /etc/mail/genericstable

Clicking this provides a simple text entry field, where all genericstable mappings are listed. You may edit them manually here. Take care, however, as manually edited entries are not checked by Webmin for grammatical correctness.

Outgoing Domains

By default Sendmail only performs **Outgoing Address** translations on mail from local users (users in the same domain as the Sendmail server). Any outside domains to be remapped must be entered here.

Domain Mapping

This feature allows you to remap all To and From addresses for a domain to another domain. This is useful if, for example, your company changes names and you'd like all mail to be changed from `mailuser@oldname.com` to `mailuser@newname.com`. This change will affect all mail that is delivered to, relayed through, or sent out from your server. Use of this should be limited to *your domains*.

Spam Control

On this page, you may configure any number of rules, with the purpose of preventing spammers from using your system and network resources for their evil purposes. Though this is just the tip of the iceberg for the spam control features provided by recent Sendmail versions, it does provide you with a very good means of preventing spam on your network. The first and primary goal is to prevent anyone from outside of your network from using your server as a relay for spam. Luckily, in recent versions of Sendmail, the default is to refuse to relay from any host not on your local network. This makes your job a little easier, because all you must do is allow mail relaying from your trusted hosts and domains. Here also, you can add rules to explicitly disallow mail from some known spammers. For example, if your users began receiving large batches of unsolicited commercial email (*UCE*, affectionately known as spam) from the bigdumbspammers.com (not a real domain at the time of this writing) and the administrators of this domain either don't care or are active participants in the spamming, you could simply block them from sending mail to any of our clients. You would enter the domain name and select Reject or provide an error message by using the Error code.

For more on spam control topics in Sendmail, the *Allowing controlled SMTP relaying in Sendmail* [<http://www.sendmail.org/tips/relaying.html>] page provides more documentation for several of the new Sendmail features to prevent spam, such as using *The Realtime Blackhole List* [<http://mail-abuse.org/rbl/>], which

provides a blacklist of known spammers and open relays. Also, it may be worthwhile to visit *The Mail Abuse Prevention System* [<http://mail-abuse.org/>] for more on ways to fight spam.

Relay Domains

Any local domains that you would like to allow relaying to should be listed here. Any incoming mail that is not for a local user and not for one of these listed domains will be rejected by Sendmail. If your Sendmail is providing mail service for several domains in a virtual hosting fashion, those domains should be listed here also.

Mail Queue

This page provides access to the mail queue. Depending on your configuration, your queue may be a fleeting thing, or it may fill until it is flushed periodically or manually. Usually, if you have full-time Net access and a full-time DNS server, you will use the background mode of delivering mail. In this case, mail will only be in the queue for a few seconds or minutes before being sent out to the recipient servers. However, in the event of a transient delivery failure (a non-permanent error), the message will remain in the queue until the message is discarded (due to permanent error) or successfully sent. Any message in the queue may be viewed by clicking the message ID. Also, messages may be deleted from the queue. Finally, it is possible to manually flush the queue, and Sendmail will attempt to deliver all messages in the queue immediately.

User Mailboxes

A great feature of the Webmin Sendmail module is the ability to read mail via a Web interface. While not a full-featured mail client (even as web-based clients go), it is a quick and easy way to check messages for accounts that ordinarily do not get checked by a user. For example, on my system, I receive daily backup reports from my backup system, so I check them periodically via the Webmin interface just to look out for problems.

To check mail for root simply click the name. From there you'll be presented with a list of all of that user's emails. Clicking the message will display it. From the **User Email** page it is also possible to delete messages and to compose a new message (Figure 11-2).

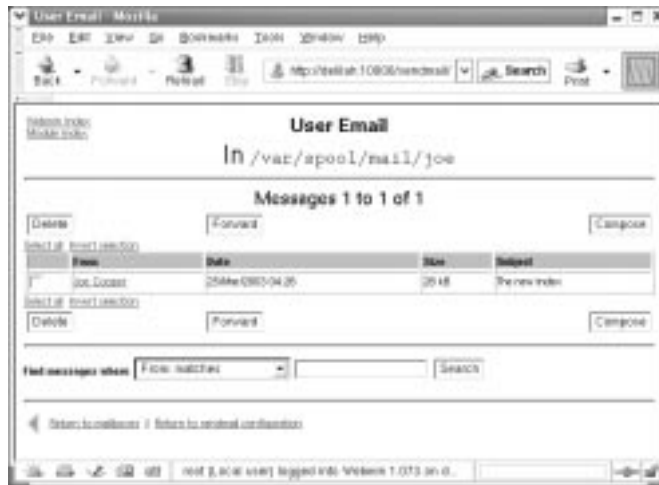


Figure 11-2: User Email

Editing the m4 Configuration File

Sendmail on your system probably has a pretty good feature set included in the default `sendmail.cf`. But as has been discussed, some features are often left out, such as `genericstable`. Usually, these features are not needed, but if they are, you must add them to your configuration file. Manually editing the `sendmail.cf` is generally regarded as not being an option for mere mortals like you and I (Eric Allman, the creator of Sendmail, *might* be able to do it). However, recent versions of Sendmail provide a novel method of adding features to the `sendmail.cf`, which uses a macro file named `sendmail.mc` and the **m4** macro processor.

Adding a Feature

Unlike directly editing the `.cf`, adding a feature using the **m4** macro file is actually pretty easy.

After opening the **M4 Configuration** page, you'll see a file that looks something like the page shown in Figure 11-3.



Figure 11-3: M4 configuration file

Each line, with the exception of the `divert(-1)` comment lines and the `divert(-1)` line, are of the form `macro-type(value list)`. In this example, you're going to add a new feature called `genericstable`. So you'll insert a line like this into the `FEATURE` list:

```
FEATURE(`genericstable', `hash -o /etc/mail/genericstable')
```

To do this, select `Feature` from the drop-down list at the bottom of the page, and click the **Add new entry of type** button. Then, select `genericstable` (Outgoing Addresses). Next in the parameters field you specify the type and location of the table file, `hash -o /etc/mail/genericstable`. The single quote marks are not required, as Webmin will insert them for you. For later convenience it is probably wise to use the arrow buttons on the right of the page to raise the new entry to be just below the other `FEATURE` lines in your file. It isn't strictly necessary, but it is nice to have neat configuration files, even if Webmin hides them from you most of the time.

After saving the changed file, you will generate the new `sendmail.cf` (don't forget to back up the old one to another file if you've already set up your `Sendmail`). To create a new `sendmail.cf` based on your `.mc` file, click the **Rebuild Sendmail Configuration** button. You'll then be able to open the **Outgoing Domains** page, and create the new `genericstable` file and edit it normally. A restart of `Sendmail` will be required to apply the changes you've made.

Tutorial: Setting Up Sendmail

When first installed Sendmail will only need a few small changes in order to begin providing service for sending and receiving mail. The first step is to specify for whom mail will be accepted, which you will specify in the **Local Domains** page, while the second step will be to permit local network users to send, or relay, email through the server, which will be specified in the **Spam Control** page.

NOTE *This tutorial assumes you have already configured DNS service for your network, including an MX record for your domain. If you haven't already done so, refer back to the BIND chapter, and configure name resolution before attempting the steps in this tutorial.*

Configuring Domains to Receive Mail For

By default, Sendmail is not configured to receive mail for any host or network other than the machine on which it is running. So you must first configure Sendmail to permit anyone to send mail for delivery to your domain through your server. Open the **Local Domains** page, and enter the domains for which your server will accept mail. In my case, I would enter `swelltech.com`. Any number of domains can be entered here, as can host names, so I could also enter `www.swelltech.com` if ever I expected mail to be delivered to that address.

Click the Save button to update the `sendmail.cf` file. This will add new Cw lines to include your specified domains.

Permitting Local Users to Relay

The next step to achieving a simple mail server is to permit your local users to send mail through your server. Click the **Spam Control** icon, and create one or more rules matching your local networks. To create a new rule, first select a **Mail source** of Network, and specify the IP of the network you'd like to relay for. For example, on a local network using private IP addresses, one might enter `192.168.1` to specify all of the hosts in the `192.168.1.0/24` network. Then, select Allow relaying, and click **Create** to add the new rule to the access file.

Finally, return to the primary Sendmail page, and click the **Start Sendmail** button. It is usually useful to keep an eye on the logs when starting a daemon so that problems will be immediately obvious. Sendmail logs to the `maillog` on most systems, which is likely located in `/var/log` directory. You can use the Webmin **System Logs** module to view this log.

Tutorial: Virtual Hosting Email with Sendmail

Virtual hosting is a rather broad term applied to many network services to specify that the server in question provides service to two or more network domains with some degree of separation. Specifically, in the case of a mail server, it means that the mail server will deliver to a unique local user based on the username *and* the domain in the to field of the received email. For example, an email to `joe@swelltech.com` would be treated differently from an email sent to `joe@notswelltech.com` and would be delivered to a different mailbox.

As with most Open Source software there are many ways to accomplish our goal, but here you'll learn the simplest method provided by Sendmail. Configuring Sendmail for virtual mail hosting is a three-step process. First, DNS must be appropriately configured for each domain being served including an MX record, as documented in the BIND chapter of this book. Second, the new domain is added to the **Local Domains** table. Finally, one or more entries are added to the **Address Mapping** table. As DNS has its own chapter, and adding an entry to the **Local Domains** table was covered in the preceding tutorial, you'll only learn the final step here.

Adding Address Mapping Entries

Click on the **Address Mapping** icon, and create new mappings as appropriate for your environment. To create a new entry, select **Address** and fill in the address on which mail will be received in the **Mail for** field. This will include the name and domain name of the recipient, so for example, I might enter `joe@virtualhost.com` in this field. Next, select the **Address** option and enter the destination mailbox for this user in the, which needs to be an existing user, into the **Send to** field. For example, I might enter a username of `virtualhost-joe` here. The username must be created on the system, as well, which can be done using the section called "Users and Groups" in Chapter 5.

Click the **Create** button, and test your work by sending mail to your newly created virtual user.