

Making Sendmail Work For You

by: David L Cantrell Jr. <david@burdell.org>

LUG@GT Presentation, 19-Feb-2003



sendmail is one of the most popular SMTP servers available. Other popular ones include qmail and postfix. People stand behind their choice of mail server like their choice of editor. I have used qmail and sendmail extensively, but definitely understand

sendmail to the greatest degree. This presentation is aimed at someone who is interested in setting up sendmail, but doesn't understand how it works. I hope we can get some volunteers to do presentations on other mail servers in the future.

I plan to explain in general how the UNIX mail system is designed to work, how you can get sendmail up and running, a walk through the configuration files, and lastly some examples of sendmail configurations. Please stop me if you have a question and I'll try to answer it. I am by no means a sendmail expert, but I understand enough of it to get it working in several scenarios.

MTA, MUA, and MDA

The UNIX mail system follows the basic UNIX design principle, that is, each program really only does one task. The basic components of the mail system are:

- **MTA:** mail transport agent (e.g., sendmail)
- **MUA:** mail user agent (e.g., mutt)
- **MDA:** mail delivery agent (e.g., procmail)

When you compose a message, you typically do so from within your MUA. When you send the message, the MUA hands the message to the MTA. The MTA reads the envelope and directs it to the appropriate system where it is handled by the MDA for delivery. The MTA is a very important part of the mail system, so this document is mostly about configuring a popular MTA: sendmail.

sendmail

sendmail is one of the most popular MTA packages available. Other popular ones include qmail and postfix. Great wars have been fought over which one is really the best one to use, but in the end it doesn't really matter. This document covers sendmail,

but I hope future presentations cover the other MTA choices.

What and Where

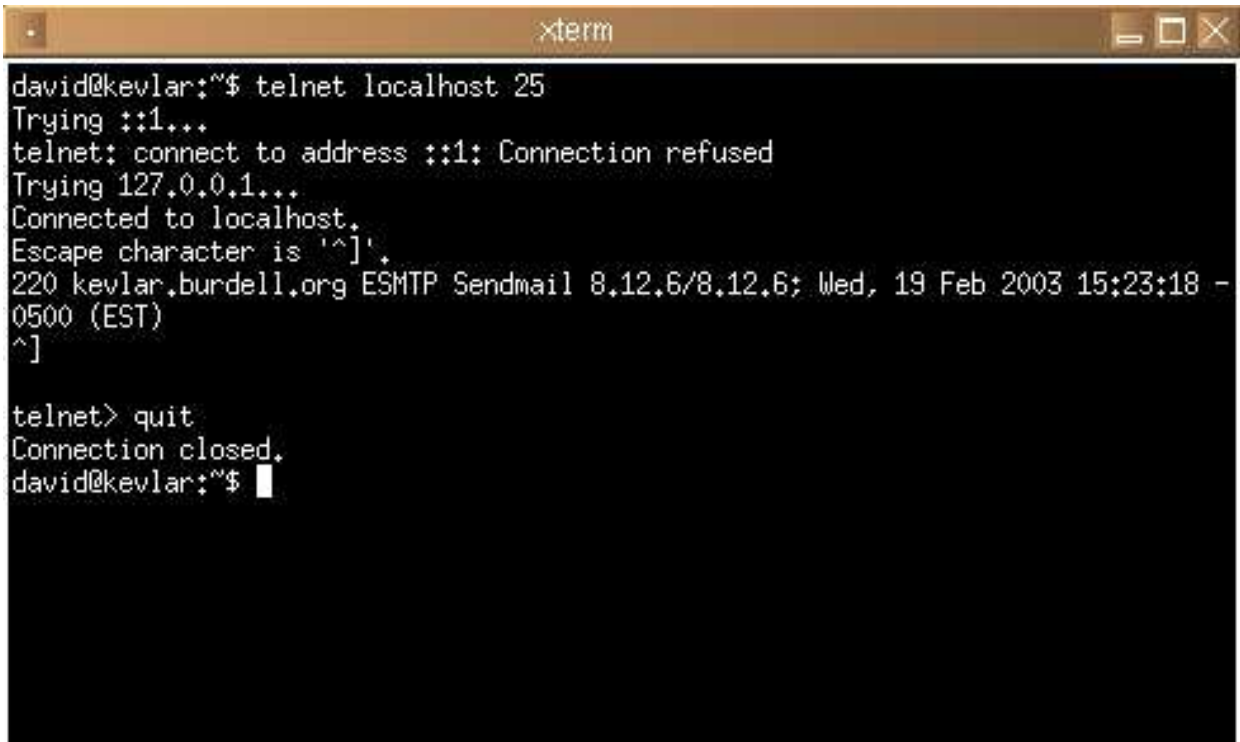
sendmail was written by Eric Allman at UCB for the BSD UNIX operating system. It has been ported to almost every platform in existence. Most Linux distributions and commercial UNIX operating systems include sendmail in one form or another. Linux distributions generally use the latest releases from sendmail.org, while commercial distributions lag behind since they like maintaining their own forks of the sendmail source tree. Whatever system you're using, if you plan on using sendmail, check to make sure it is the latest available. If it isn't, head over to www.sendmail.org and download the latest version. Documentation is included for compiling it on your system.

How sendmail should run

In the ideal environment, sendmail runs on each machine handling local mail transport and delivery, as well as talking to a main SMTP node. A lot of people like this setup, since it allows for mail composition from almost anywhere. But, unless properly configured, your mail spool ends up on several machines. The common way to overcome the multiple mail spool problem is to keep your mail spool on an NFS mount. In my opinion, this is totally unnecessary. Most people agree and end up with...

How most people run sendmail

Designate a system as your mail server. Configure sendmail to deliver mail on this system. Mail composition and reading can be handled through [hear](#). If remote access is desired, allow POP3 or IMAP access to mail spools. This configuration is much more desirable than the other scenario.



```
xterm
david@kevlar:~$ telnet localhost 25
Trying ::1...
telnet: connect to address ::1: Connection refused
Trying 127.0.0.1...
Connected to localhost.
Escape character is '^]'.
220 kevlar.burdell.org ESMTSP Sendmail 8.12.6/8.12.6; Wed, 19 Feb 2003 15:23:18 -
0500 (EST)
^]

telnet> quit
Connection closed.
david@kevlar:~$
```

Check your sendmail version

Sendmail is usually installed as `/usr/sbin/sendmail` or `/usr/lib/sendmail`. Some bizarre systems may even install it as `/usr/etc/sendmail`. If sendmail is running on your system, check the version by connecting to the SMTP port:

You can hit the escape sequence to drop back to the shell. Or simply type 'quit' and the server will close the connection.

Installation

Sendmail is pretty easy to build and install by hand. Most all Linux distributions include it, except Debian. Check and see if you have it already or if there is a package available. It will save time.

If you must compile and install it by hand, follow these steps:

1. Download the source from [ftp.sendmail.org](ftp://ftp.sendmail.org). I reference version 8.12.7 here, which is the latest version available. Be sure to use the latest version of sendmail available.

```
wget
ftp://ftp.sendmail.org/pub/sendmail/sendmail.8.12.7.tar.gz
```

2. Extract

```
gzip -dc sendmail.8.12.7.tar.gz | tar -xvf -
```

3. Compile

```
cd sendmail-8.12.7
# Follow the steps in the INSTALL file, which walk you through
compiling
# sendmail and setting up your configuration files.
```

Ties with procmail

The sendmail software can be combined with procmail which results in a nice system for mail delivery on top of sendmail. Procmail is a topic for an entirely different presentation. It's worth noting here because if you are installing sendmail from scratch, be sure to get procmail. You will most certainly want it. Moshe can answer your procmail questions at all hours of the night. That's why he has a pager.

The configuration files

Sendmail reads several different configuration files to figure out what it should be doing. These files are explained below. This list is not complete, but covers the most common files you're likely to encounter.

- **sendmail.mc**

The sendmail.mc file is your main sendmail configuration file. Technically the program reads sendmail.cf and not sendmail.mc. Since sendmail.cf is not modifiable by humans, we write the mc file and run it through m4 to generate the sendmail.cf file. The configuration elements you put in your mc are actually m4 macros that get expanded to the real configuration elements for sendmail.

Why is it done this way? Well, sendmail was designed to be somewhat like a machine. It doesn't really know what to do except load sendmail.cf and "execute" it. So think of sendmail.cf as an embedded language that drives sendmail. It may seem silly, but that's because it is. Sendmail came out of an era where computing resources were much more scarce, so saving time and making the most of what you had was important. I don't care at all about my sendmail.cf file. I only edit my mc file and have it automatically generate the cf file. I think the mc file should be named .cf, but it isn't, so we get this layer of confusion.

- **aliases**

Ever emailed a webmaster@something email address? It's fairly common. Try to make a user account with the name 'webmaster'. It won't happen, usernames are limited to 8 characters. The way we get the webmaster address is by using an alias. The aliases file maps email address aliases to something, usually a real user account.

Sendmail doesn't directly read the aliases file, it reads the aliases.db file. This is a BerkDB format of aliases. Again, the historical reasoning comes in to play here. Each time you modify the aliases file, you need to run `newaliases` to update the BerkDB version.

- **access**

This is a plain text file listing host access rights to the server. The default policy of sendmail is to accept mail locally or for domains that it is specifically configured for. However, if you get a steady stream of spam from a specific host, consider listing that host in the access file. For example, the entry:

```
co.kr ERROR:"550 Korea is a gigantic spam house;  
go away"
```

Denies all mail coming from servers on the co.kr domain. The error message defined is returned by sendmail and typically written to the log files on their end.

- **genericstable**

This provides the outbound name to virtual address mapping, that is, the reverse of what the virtusertable does. For a proper virtual domain configuration, you will need to configure this file as well as the virtusertable (described below).

This is a file that must be compiled to BerkDB format before sendmail can read it.

- **mailertable**

This file contains custom domain routing information. You may wish to specifically route all email to addresses on the gatech.edu domain through a different SMTP server. This is the file where you define that.

This file must be compiled in to a BerkDB file that sendmail actually reads.

- **relay-domains**

This is a plain text file that lists individual hosts or ranges of hosts that are allowed to relay mail off your server. You'll need this if you want to be able to use your mail server as an SMTP server when configuring a program like Evolution.

- **virtusertable**

This file maps usernames from one hostname to a real user or another hostname. This file is used to set up virtual domains and virtual addresses.

This is another file that must be converted to BerkDB format before sendmail can read it.

- **local-host-names**

This file lists the domain names that you are delivering mail on. If you own headnut.org and want to accept AND deliver mail for that domain, you need to put headnut.org in the local-host-names file.

This file sometimes differs in name across various distributions. Red Hat used to (or still does) call it sendmail.cw. Some simply call it locals. The format and purpose are the same, but the name may be different.

Inconsistencies

You may have noticed that among the sendmail configuration files there are several inconsistencies. Namely, some files are plain text, some are special format files (BerkDB), and some are macro processed files. This is one of the things that bothers me about sendmail, but it's not the end of the world.

Message Submission Program

If you wish to have users remotely connect to your SMTP to send outgoing email, you should consider using the message submission program with sendmail. I won't go in to the details here, but here is a link:

<http://www.sendmail.org/m4/msp.html>

With MSP, you can configure sendmail to require user logins and passwords to connect to the SMTP server.

Writing a sendmail configuration file

sendmail.mc authoring

The format of the mc file is fairly simple. It is a list of m4 macros and accompanying options. Generally there is one macro per line. Let's make a basic configuration file. We'll start with the basic settings:

```
divert(0)
VERSIONID('My very own sendmail.mc')
OSTYPE(linux)
DOMAIN(generic)
```

We now have some basic settings, but we should add some features to the mail server. First note the use of capital letters for VERSIONID, OSTYPE, and DOMAIN. These are the m4 macro names. The values in the parentheses are the options for that macro. Please take a note to ask me about m4 quoting if I haven't already explained it.

Features are added using the FEATURE macro. I'm going to add some common features to my mc file:

```
FEATURE(access_db, 'hash -o -T<TMPF> /etc/mail/access')
FEATURE(local_lmtp)
```

We should define a couple of settings for the configuration file. Namely:

```
define('confCW_FILE', '-o /etc/mail/local-host-names')
```

Lastly, we need to set some mailers for our system:

```
MAILER(local)
```

```
MAILER( smtp)
```

This configures sendmail for local mail operation and SMTP mail operation. Now, there are a ton of other settings available for sendmail. Rather than buying the Bat Book, I recommend you refer to the documentation on [sendmail.org](http://www.sendmail.org):

```
http://www.sendmail.org/m4/readme.html
```

The Sendmail Consortium provides a nice HTML browsable copy of the documentation that ships with sendmail. This is a very good reference.

Generating sendmail.cf

Generating the cf file for sendmail uses the m4 macro processor. Some distributions provide a Makefile in `/etc/mail` that automatically generates the configuration file. If you don't have this, you'll need to run the m4 command by hand. You run m4 and pass it the path to the sendmail macro directory and the path to the main sendmail configuration file macro and your new mc file. That's a mouthful. Here's what you do:

```
m4 -D_CF_DIR=/usr/share/sendmail/cf /usr/share/sendmail/cf/m4/cf.m4  
\  
    sendmail.mc > sendmail.cf
```

This assumes your sendmail m4 directory is in `/usr/share/sendmail/cf`. It may be in a different location on your system. Sometimes you'll find it in `/usr/src/sendmail`. Once you run the above command, you'll have a ready-to-use `sendmail.cf` file.

Starting and Stopping sendmail

The queue runner and sendmail

To start the server, I run these commands:

```
/usr/sbin/sendmail -L sm-mta -bd -q25m  
/usr/sbin/sendmail -L sm-msp-queue -Ac -q25m
```

This starts the sendmail MTA as well as the queue runner. Your distribution probably includes some form of a script that runs the above two commands in a dozen lines or so. If a script is provided in `/etc/init.d`, you should use that.

To stop sendmail, I issue this command:

```
/sbin/killall sendmail
```

Sendmail reacts to signals in a normal manner and when it is sent SIGTERM it will shut itself down.

Examples

Acting as a primary mail server

Acting as a primary mail server means we want to accept and deliver mail for a specific domain. Assuming you have the proper MX records configured according to Moshe's BIND presentation, you're ready to set up sendmail as your primary nameserver.

In this case, all we need to do is add the domain to the `/etc/mail/local-host-names` configuration file and restart sendmail. Mail will be accepted for that domain and sendmail will deliver it on the system.

Acting as a backup mail server for your best friend

As Moshe stated in his BIND presentation, sendmail is smart enough to know if it is a backup MX server. If it determines it is the backup server and the domain is not listed in `local-host-names`, sendmail will spool the mail and try to pass it on to the primary MX at a later time.

You can run `sendmail -qf` to give sendmail a swift kick to deliver any messages it has waiting for a primary MX.

Masquerading as another server

This idea of masquerading as another server is useful if you access the Internet through a dialup connection. You can run sendmail locally and compose messages locally, but sendmail will be configured to pass all outbound messages to the server it is masquerading as. What you generally don't want is for sendmail to pass all mail, even local mail, to the other server. The options below can be added to your `sendmail.mc` file to enable masquerading:

```
FEATURE(`remote_mode')
define(`SMART_HOST', `mail.example.com')
FEATURE(`masquerade_envelope')
FEATURE(`genericstable', `hash -o
/etc/mail/genericstable.db')
GENERIC_DOMAIN(`localhost mybox.example.com')
```

You should add `localhost` and the hostname you are masquerading as to the `local-host-names` file.

The `genericstable` file can be used to map local user names to the remote address name.

Running mail for a virtual domain

With a virtual domain configuration, you simply populate the `virtusertable`

and genericstable with the mappings for the virtual address to the real user name. Be sure to set up the hostname and usernames first.

Resources

There is a wealth of great sendmail information on the Interweb and probably on your own computer.

- **man pages**

The commands and files associated with the sendmail system ship with good man pages. Be sure to consult these when you're looking for an answer.

- **<http://www.sendmail.org/>**

The Sendmail Consortium (not to be confused with Sendmail, Inc.) supports the open source releases of sendmail. They provide a lot of documentation, FAQs, security notices, and links to other resources.

- **<http://www.procmail.org/>**

Not specifically part of this presentation, but it's worth noting that procmail ties in very well with the sendmail system. If you will be using sendmail, consider using it with procmail to get a nice mail system. Moshe is available to answer all of your procmail questions.

- **Don't Blame Sendmail**

<http://www.sendmail.org/tips/DontBlameSendmail.html>

This page talks about problems people encounter and blame sendmail for. Worth a look if you plan on running your own server.

Fin