

# MORE OPEN-SOURCE SOFTWARE

---

# 15

This chapter is intended to make you aware of a number of additional software packages you can use on your Mac. The software we describe here was selected for its usefulness and its diversity, to reinforce just how many different kinds of things you can now do with Unix.

Before proceeding with the tasks in this chapter, make sure that you are quite comfortable working at the command line and that you have read chapters 13 (“Installing Software from Source Code”) and 14 (“Installing and Configuring Servers”) thoroughly. You should be familiar with downloading, unpacking, installing, and configuring software from the command line using the Fink tool (covered in Chapter 13), as well as manually without the Fink tool.

We suggest that you skim through this entire chapter and then pick one of the first five packages to start with (avoid the three that follow them for the time being). We’ve listed them in the order in which we think you might want to try them out, but they are independent of each other, so it doesn’t matter which one you install first.

Installing the final two packages in the book (the Majordomo email-list manager and the TWIG Web interface for email and group scheduling) requires that you have the assistance of an experienced Unix system administrator. They are included with the hope that you take what you have learned in this book and continue on to deeper levels of understanding.

## named (DNS Server)

In Chapter 14, “Installing and Configuring Servers,” we discussed the importance of having a fully qualified domain name (FQDN) associated with your computer’s IP address. If you are using your computer as a server on the Internet, then you probably want people to reach it with a name like `www.cool-mac.sf.ca.us` instead of an IP address like `66.47.70.91`.

Making an association between an IP address and a domain name is done through the Domain Name System (DNS). The standard Unix software for translating between IP addresses and names is the `named` (*name daemon*) program.

Mac OS X 10.2 comes with `named` installed but not activated by default. It is easy to activate `named` on Mac OS X. See “Activating `named`” later in this section.

The default configuration for `named` is something called a *caching-only name server*. A caching-only name server performs translations by asking other machines on the Internet for answers and then remembering

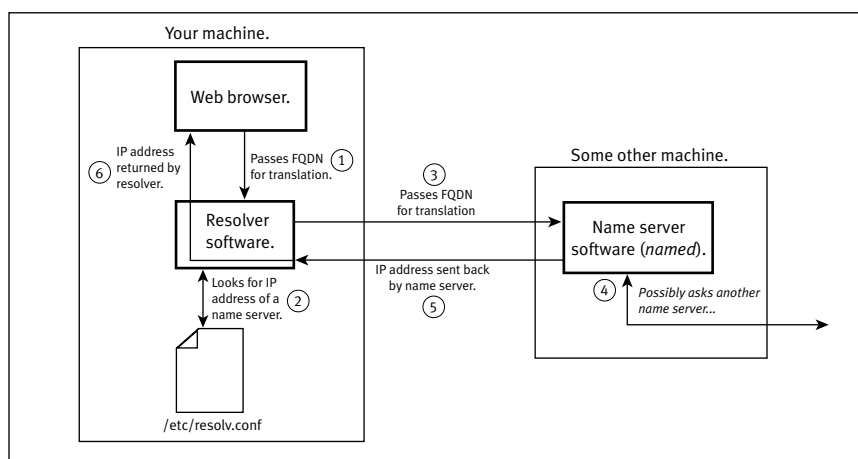
the answers for a period of time (determined by the machine that the answer comes from). A caching-only name server is not responsible for being an official source of information about any particular domain.

The Internet DNS is essentially a huge database spread out over the whole Internet (you may see it called a “distributed database”).

When `named` gets a request for a translation it hasn’t memorized, it asks another name server, which may ask another, and so on, until the question reaches the name server that knows the answer. The ultimate source of information about a particular domain name comes from the *primary name server* for that domain.

**Figure 15.1** is a simplified illustration of how the process works.

When a piece of software such as your Web browser is told to connect to another machine on the Internet, it is usually given an FQDN, like `www.peachpit.com`. The Web browser needs to translate the FQDN into an IP address in order to connect to another machine, so it calls upon *resolver* software on your machine (Step 1 in Figure 15.1).



**Figure 15.1** A simplified version of how the Domain Name System (DNS) works.

The resolver software passes the FQDN to a *name server* for translation to an IP address. The name server might be running on a different machine. How does the resolver know how to contact it?

The answer is that the resolver looks in the file `/etc/resolv.conf` to get the IP address of a name server to use (Step 2 in the diagram). The resolver then sends the FQDN to a name server (Step 3 in the diagram).

Now, in the diagram the name server is running on “some other machine,” but in fact, in the default Mac OS X configuration, it is

running on your own machine as a caching name server. But if your own *named* doesn’t know the answer to the query, it asks another name server (Step 4 in the diagram). That process repeats until the answer comes back (each name server remembers the answer for a while).

Eventually an answer gets passed back to the resolver (Step 5 in the diagram), and the resolver passes the IP address back to the Web browser (Step 6 in the diagram).

The Web browser (or whatever software needed the IP address) can now connect across the Internet and do its job.

### Learning More About DNS

The Internet Domain Name System is probably the largest distributed database in the world. Consisting of literally hundreds of thousands of servers on every continent, it is not only vast but also extremely reliable. Because each server holds only a portion of the whole system, it is virtually impossible to take the entire system down.

There are many parts to DNS—the server software, the resolver software, the databases maintained by the organizations that grant domain-name registrations (*registrars*), and the politics of how the whole system is administered.

Here are some resources to help you delve more deeply into the various aspects of DNS.

- ◆ DNS HOWTO is an online how-to guide for DNS and BIND (BIND is the “Berkeley

Internet Name Daemon,” the full name for the standard version of *named*) ([www.tldp.org/HOWTO/DNS-HOWTO.html](http://www.tldp.org/HOWTO/DNS-HOWTO.html)).

- ◆ The definitive book on DNS software is *DNS and BIND*, 2nd Edition, by Cricket Liu and Paul Albitz ([www.oreilly.com/catalog/dns2/](http://www.oreilly.com/catalog/dns2/)).
- ◆ The official Internet Engineering Task Force explanation of the Domain Name System is available at [www.ietf.org/rfc/rfc1035.txt](http://www.ietf.org/rfc/rfc1035.txt).
- ◆ There is a nice commercial Mac OS X Aqua application (free trial version available) from Men & Mice called Quick DNS ([www.menandmice.com/products/quickdnspro/](http://www.menandmice.com/products/quickdnspro/)).

The big question in all of this is: How does a machine get to be a name server for a domain?

- ◆ Whoever controls the top-level domain name must designate a particular IP address as the primary name server for that domain. So when the primary name server for matisse.net is 66.47.69.197, the name server running on 66.47.69.197 has the ultimate answer about translations for any FQDN ending in matisse.net, such as www.matisse.net or mail.clients.matisse.net.
- ◆ You must contact the person who handled the registration for a domain name in order to determine the primary name server.
- ◆ Besides a primary name server, each domain name must also have at least one secondary name server to act as a backup in case the primary is down. Secondary DNS servers fetch information from the primary at regular intervals (set by the primary).
- ◆ Name server software must be running on the machine designated as the primary name server.
- ◆ Finally, named must be properly configured to answer queries about FQDNs within the domain(s) it is responsible for.

## Activating named

The `named` daemon is installed but not turned on in the default Mac OS X 10.2 configuration. You activate it by making a change to `/etc/hostconfig` and running the appropriate StartupItems script.

### To activate named:

#### 1. Edit `/etc/hostconfig`.

You need to change the `DNSSERVER` entry in `/etc/hostconfig` from “-NO-” to “-YES-”

```
sudo vi /etc/hostconfig
```

Then change the `DNSSERVER` entry and save your changes.

⚠ Note: In Mac OS X 10.2 and 10.2.1 the entry in `/etc/hostconfig` actually needs to be `DNSSSERVER` with three S's. That is because there is a mistake in the StartupItem script (see next step) where the variable is spelled incorrectly. We don't know when Apple will fix this, so have a look at the StartupItem script mentioned in the next step and use the same spelling in `/etc/hostconfig`.

#### 2. Start `named`.

The technically correct way to do this is:

```
sudo /System/Library/StartupItems/  
→ BIND/BIND start
```

BIND stands for “Berkeley Internet Name Daemon” and is the full name for the most common implementation of DNS software.

In Mac OS X 10.2.1 the `BIND` script actually ignores any arguments and always tries to start `named` if the `DNSERVER` entry (or “`DNSSERVER`”, see step 1 above) in `/etc/hostconfig` is “-YES-”.

## Configuring `named`

The `named` program is a server (name daemon) that listens on port 53 and responds to requests for translation between domain names and IP addresses. The `named` server configuration file is `/etc/named.conf`, and `named` is started by `/System/Library/StartupItems/BIND/BIND`.

The default configuration in Mac OS X is to act as a “caching-only” name server.

You can also configure your Mac to act as a primary or secondary name server for specific domains.

## To configure your Mac to be a primary name server for a domain:

1. Contact the person who set up the domain name for which you want your computer to be the primary server.  
In order for this configuration to work, your Mac’s IP address *must* be designated as a name server by the person who issued the domain name in the first place. Contact that person to find out who the *registrar* for the domain name is and how to contact him or her. You will need to contact the registrar once everything else is ready.
2. Read the man page for `named.conf`.  
You will be editing that file later on.  
Also consider the various DNS resources listed in the sidebar “Learning More About DNS.”
3. Add the appropriate entry to `/etc/named.conf`.

You’ll need to be root to do this. **Figure 15.2** is an example of `/etc/named.conf`

*continues on next page*

```
/*
   This is a comment. Comments are good.
   /etc/named.conf - configuration file for named
   see    man named.conf
*/

options {
    directory "/var/named";
};

// We are the primary name server for the domain xxxxxx.yyy
zone "xxxxxx.yyy" in {
    type master;
    file "xxxxxx.yyy.zone";
};
```

**Figure 15.2** Example of the `named` configuration file, `/etc/named.conf`. The highlighted entry configures `named` to be a primary name server.

with a highlighted new entry you would add in order to be the primary name server for the domain xxxxxx.yyy (we'll use this fake domain xxxxxx.yyy as an example throughout the following text).

**4.** Create the *zone file* for xxxxxx.yyy.

The zone file contains the master copy of all the information for a domain, including entries (called *A records*) that associate IP addresses with FQDNs.

This file is called xxxxxx.yyy.zone and goes in the directory listed in Figure 15.2.

**Figure 15.3** is a sample file with the highlighted text demonstrating the following assumptions:

- ◆ The domain in question is xxxxxx.yyy
- ◆ Your Mac's FQDN is aaaaaa.bbbbbb.ccc.
- ◆ "Postmaster@bbbbbb.ccc" is a valid email address.
- ◆ The serial number for the file is based on the date you create or modify it, in the format YYYYMMDD##, where YYYY is the four-digit year, MM is the two-digit month, DD is the two-digit day, and ## is a two-digit number showing how many changes were made on that day. The serial number 2002112503 would mean the third change on November 25, 2002.
- ◆ The secondary name server for xxxxxx.yyy is aaa.bbb.ddd.
- ◆ Email for the domain xxxxxx.yyy will be handled by the machine mail.xxxxxx.yyy.
- ◆ If xxxxxx.yyy is translated as an FQDN, it will translate to the IP address 192.168.5.1 (that is a made-up IP address; you would use a different one!).

```
$ORIGIN yyy.
$TTL 3D ; Default length of time that answers are good for
xxxxxx IN SOA aaaaaa.bbbbbb.ccc. postmaster.bbbbbb.ccc. (
    2002112500 ; serial number YYYYMMDD##
    1D ; Secondary name server fetch interval (D=days, H=hours)
    1H ; Secondary name server retry interval if fetch fails
    7D ; Secondary never keeps answer longer than this
    3D ; How long should "can't find it" answers last?
)
    7D IN NS aaaaaa.bbbbbb.ccc.
    7D IN NS aaa.bbb.ddd.
    7D IN MX 5 mail.xxxxxx.yyy.
    1D IN A 192.168.5.1

$ORIGIN xxxxxx.yyy.
www 1D IN A 192.168.23.17
mail 1D IN CNAME www
```

**Figure 15.3** Example of a zone file: the file containing the master information for a domain. The highlighted portions should be replaced with the appropriate data for your configuration.

- ◆ The FQDN `www.xxxxxx.yyy` is supposed to translate to the IP address `192.168.23.17` (also a made-up IP address; use whatever is correct for you). This is an “A record.”
  - ◆ The FQDN `mail.xxxxxx.yyy` is another name (an *alias*) for `www.xxxxxx.yyy`.
5. Restart `named`:  

```
sudo /System/Library/StartupItems
→ /Named/Named restart
```
  6. Get your Mac’s IP address designated as a primary name server for the domain in question.  
 This means contacting the registrar you discovered in step 1. He or she will need your Mac’s IP address and FQDN.  
 If the address for the secondary server(s) is also changing, you must notify the registrar about that as well.  
 This step can take several days to be completed from the time you notify the registrar of the change.

7. Notify the person who runs the secondary name server that your Mac is now the primary server. He or she will need your Mac’s IP address. This is required so that the secondary server knows which IP address to fetch data from.

### ✓ Tips

- You can use the `whois` command to determine the registrar for a domain. See `man whois`. Also, the Network Utility application can perform the `whois` function from within Aqua (Applications > Utilities > Network Utility).
- Make sure you don’t have firewall software blocking port 53.

### To configure your Mac to be a secondary name server for a domain:

1. Create the directory:  
`/var/named/secondary`.
2. Make sure the directory is owned and writable by root.
3. Add an entry to `/etc/named.conf`.

*continues on next page*

4. **Figure 15.4** is an example of the `named.conf` file with a highlighted entry added so that this machine becomes a secondary name server for the domain `zzzzzz.xxx`, assuming that the primary name server for that domain is `192.168.23.17`.
5. Restart `named`.
6. Notify the registrar for the domain about the new secondary name server.  
The registrar needs your Mac's IP address and FQDN.

#### ✓ Tips

- Your `named` should fetch a copy of the zone file from the primary name server as soon as you restart it. The copy is stored in the directory you created in step 1, so you can check to see if it is working.
- Make sure you don't have firewall software blocking port 53.

### NAMED (DNS SERVER)

```
/*
   This is a comment. Comments are good.
   /etc/named.conf - configuration file for named
   see    man named.conf
*/

options {
    directory "/var/named";
};

// We are the primary name server for the domain xxxxxx.yyy
zone "xxxxxx.yyy" in {
    type master;
    file "xxxxxx.yyy.zone";
};

// Entry to make us be a secondary name server for zzzzzz.xxx
zone "zzzzzz.xxx" in {
    type slave;
    file "secondary/datamassage.com.zone";
    masters{
        192.168.23.17;
    };
};
```

**Figure 15.4** Example of `/etc/named.conf`. The highlighted entry configures `named` to be a secondary name server.



## Lynx (Command-Line Web Browser)

Lynx dates from the early 1990s, when the Web was first invented and most of the people with Internet access were users on Unix machines (usually college and graduate students) using command-line interfaces.

Even with the widespread availability of graphical Web browsers, a text-only Web browser still has some value. For one thing, it is very fast. Another value is for people who use software that reads to them from their computer screen. A text-only Web browser could be less confusing to the reader software. Also, if you are already working at the command line, it's nice to be able to check a Web site without having to switch the modality of your interface—you can stick with the text-only universe. Of course, you get no pictures (we did say “text-only” three times already, right?).

One other interesting feature of Lynx: It can be compiled and installed for use in an “anonymous” mode, suitable for use in publicly accessible terminals where people's access needs to be restricted.

### Another Text-Only Web Browser: Links

There is another text-only Web browser similar to Lynx, only it's spelled *Links*. You can install Links using Fink, and there is also a version that supports Secure Sockets Layer (SSL). The Fink package is `links-ssl`, so you install it with

```
sudo fink install links-ssl
```

Lynx is easy to install using the instructions from Chapter 13, “Installing Software from Source Code.” Here is the abbreviated version.

### To install lynx:

1. Download the source code from <http://lynx.isc.org/> and save it in the directory `/usr/local/src/lynx/`
2. Unpack the compressed tar archive using `tar`.
3. Cd into the resulting source-code directory.
4. Read the read-me file.
5. Read the installation file.
6. `./configure`
7. `make`
8. `make install`

This installs Lynx as `/usr/local/bin/lynx`.

Lynx is very easy to use. It takes over the entire Terminal window (like `vi`), and you use the arrow keys and `[Return]` to navigate around. You can get help by pressing `[H]`.

### To use Lynx:

- ◆ `lynx url`  
Where *url* is the URL you want to visit. Some examples:  
`lynx http://www.peachpit.com/`  
`lynx ftp://ftp.loc.gov/pub/american.  
→memory/`  
`lynx gopher://gopher.well.sf.ca.us/`

## The Pine Email Client and Three Mailbox Servers

This section covers several pieces of software that come in a single source-code package. You can use either one separately or both of them together.

One program is a command-line email client called Pine. There are also three email server programs included with the Pine package. One is `imapd`, which uses IMAP (Internet Mail Access Protocol). The other two are `pop2d` and `pop3d`, which use POP (Post Office Protocol), versions 2 and 3, respectively. We briefly mentioned IMAP and POP in the Sendmail section of Chapter 14, “Installing and Configuring Servers.” The Pine package also provides the `pilot` directory-browsing program and the `pico` text editor (Mac OS X already has `pico` installed in `/usr/bin/pico`).

### About Pine

Pine is a command-line email client developed in 1989 at the University of Washington as an easy-to-use email program. This was back in the days before graphical user interfaces were common for Internet software. A popular but complex email client at that time was the Elm program, and the name Pine came from “Pine Is Not Elm.” (These days, the folks at the University of Washington say that the name Pine is based on the boring but more descriptive “Program for Internet News and Email.”)

A big reason for using Pine in this day of graphical email tools is the ability to check your email without leaving the command line (trust us, once you starting using the command line, you often want to stay there, without reaching for the mouse). If you log in to your machine from somewhere else on the Internet, you can fire up Pine and be able to handle most email tasks.

Pine is available in source code from the University of Washington’s Pine Information Center ([www.washington.edu/pine/](http://www.washington.edu/pine/)), where you will also find documentation, history, email lists, and other help resources.

### About the IMAP and POP servers

The IMAP and POP servers all allow users to connect and read their email using a variety of email clients—the Mac OS X Mail application, Pine, Eudora, Microsoft Outlook, Netscape, Mozilla, and many others.

The POP servers use the simple but effective POP (POP3, the protocol used by `pop3d`, is almost certainly what you want; POP2 is more or less obsolete). All graphical email clients support POP3.

IMAP is supported by most, but not all, email clients and provides features not found in POP3. The most important feature of IMAP: It makes reading your email from more than one computer easy.

### IMAP and POP Server Security Alert

All three of these servers normally send and receive information over unencrypted connections, making them vulnerable to sniffer attacks (as described in Chapter 12, “Security”). It is possible, however, to build any of these servers with support for encrypted connections using the SSL protocol. To do this, you will need to download the source code (from [www.washington.edu/imap/](http://www.washington.edu/imap/)) and build the servers manually. The (required) OpenSSL software is already installed on Mac OS X. For instructions on building these servers with SSL, see the University of Washington’s “UW IMAP Server Documentation” page ([www.washington.edu/imap/documentation/SSLBUILD.html](http://www.washington.edu/imap/documentation/SSLBUILD.html)).

### Another IMAP Server

Another IMAP server is Cyrus (<http://asg.web.cmu.edu/cyrus/imapd/>), intended for use on machines that are used only as email servers where users cannot log in on the command line. This design allows the Cyrus server to handle more email and users than would otherwise be the case, with the tradeoff that a machine must be dedicated for the purpose.

With an IMAP server, such as `imapd`, each user's email is stored in a collection of folders on the server, and they can access these folders, create new ones, and move messages from folder to folder. Each time users connect, whether from their home, office, or elsewhere, they see the same collection of folders and messages.

### Installing Pine and `imapd`

The easiest way to install both Pine and `imapd` is to use Fink. You can also download the source code from the Pine Information Center ([www.washington.edu/pine/getpine/unix.html](http://www.washington.edu/pine/getpine/unix.html)) and compile and install the software manually. Here's how to do it using Fink.

#### To install Pine and `imapd`:

- ◆ `sudo fink install pine`  
Enter your password if prompted.  
Fink downloads the source code, and compiles and installs several programs, including Pine and `imapd`. See **Table 15.1** for a list.

*continues on next page*

**Table 15.1**

The Pine Software Collection	
<i>Fink installs all these in <code>/sw/bin</code>.</i>	
PROGRAM NAME	WHAT IT IS
<code>imapd</code>	IMAP protocol mailbox server.
<code>ipop2d</code>	Mailbox server that uses the old POP2 protocol.
<code>ipop3d</code>	Mailbox server using the POP3 protocol.
<i>See the IMAP Information Center (<a href="http://www.washington.edu/imap/">www.washington.edu/imap/</a>) for documentation for all three of the above.</i>	
<code>mtest</code>	A testing program intended for programmers creating new email clients written in the C language. You don't need to use <code>mtest</code> .
<code>pico</code>	An easy-to-use command-line text editor. Far less powerful than <code>vi</code> but quite useful for editing simple files. Used inside the Pine program as the text editor for composing email. (Pico means <i>Pine Composer</i> .) Mac OS X comes with <code>pico</code> in <code>/usr/bin</code> . Fink will install another copy in <code>/sw/bin</code> . Has a man page.
<code>pilot</code>	A command-line directory-browsing program. Sort of a command-line version of the Finder. Used inside Pine but can also be used by itself. Has a man page.
<code>pine</code>	The Pine email client. Has a man page and built-in help. Also see the tutorial at the Pine Information Center ( <a href="http://www.washington.edu/pine/tutorial.4/">www.washington.edu/pine/tutorial.4/</a> ).

**Figure 15.5** shows a Terminal window after starting Pine. You use the arrow keys to move the selection (highlighted text) and press **Return** to choose a selection. You can also choose a selection by typing the character next to it. For example, type a question mark to get help.

### ✓ Tip

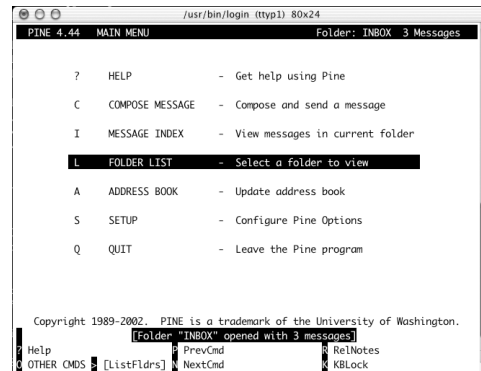
- Documentation for the `imapd` server is included in the source code that Fink downloads, but it is not installed as man pages (unfortunately). You can get to the documentation, though—Fink saves the compressed `tar` file in the `/sw/src` directory, where it is named something like `pine4.44.tar.Z`. You can unpack that file using `tar`, then look in the `imap/docs` subdirectory of the directory that `tar` created. The same documentation (although perhaps not as up-to-date) can also be found online at the University of Washington's IMAP Information Center ([www.washington.edu/imap/documentation/](http://www.washington.edu/imap/documentation/)).

## Using Pine

Pine comes with a man page and built-in help. It's fairly easy to use. To be able to send email with Pine, make sure you have set up Sendmail as described in Chapter 14, "Installing and Configuring Servers."

### To start Pine:

- ◆ `pine`  
That starts the Pine program. Your terminal window will look something like that in Figure 15.5. Press **[?]** for help.  
Also, see the tutorial at the Pine Information Center ([www.washington.edu/pine/tutorial.4/](http://www.washington.edu/pine/tutorial.4/)).



**Figure 15.5** A Terminal window showing Pine right after starting it.

## Learning About IMAP and POP

Here are resources where you can get more information about the IMAP and POP protocols, and about configuring and managing the servers:

- ◆ The IMAP Connection: "What is IMAP?" ([www.imap.org/about/whatisIMAP.html](http://www.imap.org/about/whatisIMAP.html))
- ◆ The IMAP Connection: "Comparing Two Approaches to Remote Mailbox Access: IMAP vs. POP" ([www.imap.org/papers/imap.vs.pop.brief.html](http://www.imap.org/papers/imap.vs.pop.brief.html))
- ◆ The University of Washington: "UW IMAP Server Source Distribution Documentation" ([www.washington.edu/imap/documentation/](http://www.washington.edu/imap/documentation/))

## Using `imapd`, `pop2d`, and `pop3d`

The `imapd`, `pop2d`, and `pop3d` servers are designed to be run using `xinetd`, which we described in the section about the FTP servers in Chapter 14, “Installing and Configuring Servers.”

### To activate `imapd`, `pop2d`, or `pop3d`:

1. Add the appropriate entries to the NetInfo database. Add the entries for only the server(s) you will actually use. The commands must be performed as root.

The entries for `imapd`:

```
niutil -create . /services/imap
niutil -createprop . /services/imap port 143
niutil -createprop . /services/imap protocol tcp
```

For `pop2d`:

```
niutil -create . /services/pop2
niutil -createprop . /services/pop2 port 109
niutil -createprop . /services/pop2 protocol tcp
```

For `pop3d`:

```
niutil -create . /services/pop3
niutil -createprop . /services/pop3 port 110
niutil -createprop . /services/pop3 protocol tcp
```

*continues on next page*

## Chapter 15

2. Create appropriate files in `/etc/xinetd.d`.

**Figures 15.6, 15.7, and 15.8** show the files for all three servers, but you should create only the one(s) for the server(s) you want to use. Each file should be saved inside the directory `/etc/xinetd.d/`.

## 3. Restart xinetd:

You need to obtain the *process IdD* (*pid*) number for xinetd.

It is supposed to be saved in `/var/run/xinetd.pid`, but in Mac OS X 10.2.1 that was not working. Instead, run `sudo kill -HUP `ps -auxc | grep xinetd | awk '{print $2}'``

Enter your password if prompted.

Once you have one or more of the mail servers running, you will need to configure each user's email client software to use your machine as its POP or IMAP server.

```
service imap
{
    disable = no
    socket_type = stream
    wait = no
    user = root
    groups = yes
    server = /sw/sbin/imapd
}
```

**Figure 15.6** Code listing of the file you would save as `/etc/xinetd.d/imap` for use with the `imapd` server.

```
service pop
{
    disable = no
    socket_type = stream
    wait = no
    user = root
    groups = yes
    server = /sw/sbin/pop2d
}
```

**Figure 15.7** Code listing of the file you would save as `/etc/xinetd.d/pop` for use with the `ipop2d` server.

```
service pop3
{
    disable = no
    socket_type = stream
    wait = no
    user = root
    groups = yes
    server = /sw/sbin/ipop3d
}
```

**Figure 15.8** Code listing of the file you would save as `/etc/xinetd.d/pop3` for use with the `ipop3d` server.

## X Windows

X Windows is a collection of software for providing a graphical user interface (primarily on Unix systems) and is the basis for the graphical interface available on virtually all Unix systems in use today. There is one exception: Mac OS X uses a completely different collection of software to provide a graphical interface—Aqua.

A great deal of Unix software uses the X Windows system for its graphical interface. Programs written to use X Windows for their graphics display do not work on Mac OS X's Aqua interface unless you install X Windows.

Besides enabling you to run existing Unix X Windows software on your machine, X Windows offers another interesting feature. The X Windows approach to providing a graphical interface is designed to work across networks, using a “client and server” approach. The X Windows idea of client and server will at first glance seem backward

from the way you usually think of client and server software.

In X Windows, the client software, such as a spreadsheet program, word processor, or Web browser, is the program you actually want to use. The X Windows server software is a separate piece of software that takes keyboard and mouse input and provides the graphics display for the user interface.

When an X Windows client program is running, it sends its display output not to a video card but instead to another piece of software, called an X Windows server. The X Windows server software must be compiled for a specific hardware/operating system combination (such as Darwin on the PowerPC chip, or Linux on the Intel 686). It is the X Windows server software that communicates with the keyboard, mouse, and video card.

An X Windows client program usually uses an X Windows server running on the same machine as itself, but it doesn't have to. What does this mean in practice?

### X on X?

In the Unix world, X Windows is often referred to simply as “X”—that is, the letter X, not the roman numeral for ten, as Apple does.

It is confusing.

X Windows began to be developed at the Massachusetts Institute of Technology in 1984 as a set of tools for providing graphical interfaces, primarily for Unix systems. (The X Windows Web site is now [www.x.org](http://www.x.org).)

When Apple announced the name “Mac OS X,” many Unix users groaned, wondering about the confusion of names between the venerable X Windows graphics system and the new operating-system name.

Most Mac users, of course, need never tackle X Windows, so for them the double use of X is not a problem.

For Macintosh users delving into Unix (that's you!), and for Unix programmers working to get X Windows to run on Mac OS X, the confusion becomes an opportunity for a cute name. The main version of X Windows that runs on Mac OS X is called, of course, XonX. The XonX home page is <http://sourceforge.net/projects/xonx/>.

Apple has a new kind of machine, which it calls Xserve. If you have an Xserve, then you can get some **TRIPLE XXX ACTION** by running XonX on an Xserve ([www.apple.com/xserve/](http://www.apple.com/xserve/)). (Last pun, we promise.)

It works like this:

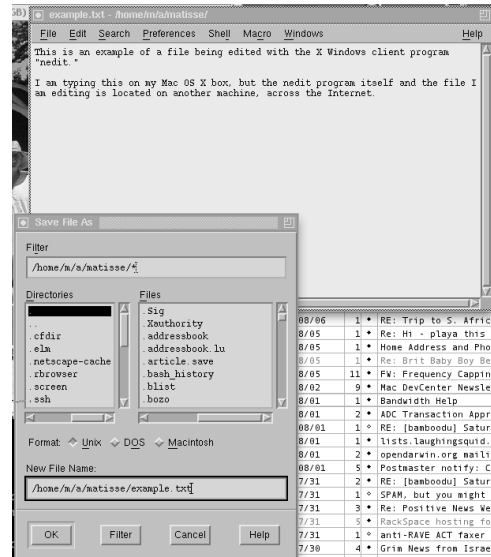
- ◆ Assume you are logged in to your Mac OS X machine, and have installed and are running the XDarwin program, which is an X Windows server.
- ◆ You set your DISPLAY environment variable to be :0.0 (this is an environment variable used by the X Windows software).
- ◆ You then connect to your friend Mikey's Linux box using ssh (with the -X option, which is required to allow X Windows software on the remote machine to communicate with the X Windows server on your machine through your ssh connection; see the ssh man page).
- ◆ You want to edit a file in your home directory on Mikey's machine. Mikey has installed the X Windows text editor called nedit. So to run nedit on Mikey's machine, you would use the following command line:

```
nedit example.txt
```

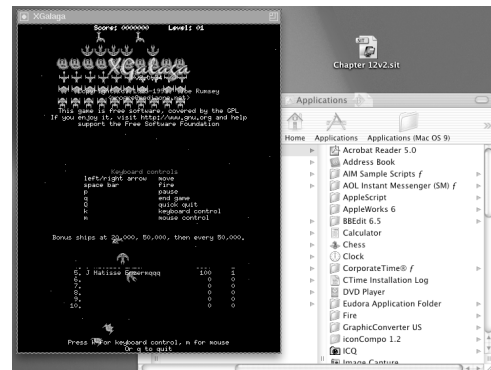
and voilà, you see the graphical interface for nedit on your Mac. **Figure 15.9** shows what X Windows looks like in use—a file is being edited using the nedit application. It is important to note that both the file and the nedit program are running on a remote machine. Only the display is running on the local (Mac OS X) machine. There isn't anything obvious in the display that indicates which machine nedit is running on—you have to know where you started it.

In this situation, your Mac is the server, and the remote machine is the client. Kind of backward from the way you would normally think of “client and server.” The key thing to grasp is that the X Windows server software provides graphic-display services for the client program.

Another example of X Windows, just for fun: **Figure 15.10** shows the classic video game Galaga in its X Windows version, running on Mac OS X.



**Figure 15.9** The Save File As dialog box from the nedit program. What is not obvious is that both the nedit application and the file being edited are on another machine, across the Internet.



**Figure 15.10** The XGalaga X Windows version of the classic video game.



### More About X Windows on Mac OS X

The best place we've seen to start learning more about X Windows on Mac OS X is the OroborOSX Web site—in particular the “Introduction to X11 on OS X” page. X11 is the most recent standard version of the core X Windows software (<http://oroborosx.sourceforge.net/X11onOSX.html>).

That Web page includes tips on some tricky subjects, such as using copy-and-paste with X Windows, as well as links to further reading.

OroborOSX, an application that makes X Windows easier to use, is covered in this section (“Installing OroborOSX”).

### Installing XonX (X Windows for Mac OS X)

The version of X Windows you will want to install on your Mac is called XonX.

There are three ways (at least) to get XonX for your Mac:

- ◆ If you have installed Fink, you can install the package called xfree86-rootless.
- ◆ You can use a Web browser to visit the SourceForge Web site (<http://sourceforge.net/projects/xonx/>) and follow the “files” link to a page where you can download a GUI installer package, as well as packages to perform updates on older versions.
- ◆ You can download the source code from the XFree86 Project, Inc. ([www.xfree86.org](http://www.xfree86.org)), and compile and install it manually. (This involves more than 60 MB of downloads and an hour or more of compiling.)

The X Windows software is large—easily more than 50 MB. Allow time for downloading.

#### To install XonX using Fink:

1. `sudo fink install xfree86-rootless`  
Enter your password if prompted for it. This downloads and installs a complete set of X Windows software, including an X Windows server called XDarwin. XFree86 is the name of the most widely used collection of X Windows software. It gets its name from its origin as a project to create an open-source version of X Windows that would run on Intel x86 series chips. The XFree86 team now has many subteams, and the XFree86 software is available for many hardware platforms.

*continues on next page*

## 2. Launch the XDarwin application.

Fink will have installed the XDarwin application in your `/Applications` directory. You can double-click it from the Finder or launch it from the command line with `open -a XDarwin`

You get the XDarwin splash startup screen (**Figure 15.11**), which shows two buttons: “Full screen” and “Rootless.” These are two different modes available for running the XDarwin server.

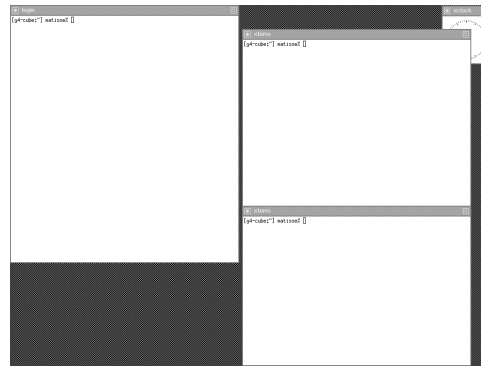
- ◆ If you choose full-screen mode, then the X Windows server takes over your entire screen and you see only X Windows applications. But you can switch between X Windows and Aqua with `Cmd Option A`. **Figure 15.12** shows what this looks like.
- ◆ If you choose rootless mode, then windows from X Windows applications will be mixed in with windows from Aqua applications (this is shown in Figures 15.9 and 15.10). **Figure 15.13** shows what you see upon starting XDarwin in rootless mode.

You can use X Windows applications as soon as you have installed XonX, and in fact when you launch XDarwin, two X Windows client applications are also launched: `xclock` and `xterm`. The `xclock` application is an analog clock, and `xterm` is the X Windows version of the Terminal program, providing windows with a command-line interface. XDarwin launches three `xterm` windows when it starts up.

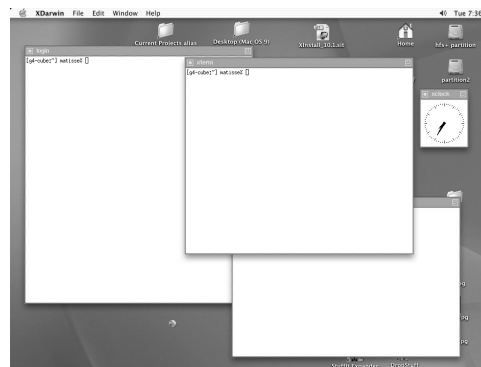
The windows that XDarwin creates are very basic—they have no scroll bars and cannot easily be resized. By adding one more application, you can improve the quality of your X Windows experience.



**Figure 15.11** The XDarwin startup screen, showing the buttons to choose between full-screen or rootless modes.



**Figure 15.12** XDarwin in full-screen mode.



**Figure 15.13** XDarwin right after startup in rootless mode.

## Installing OroborOSX

OroborOSX is an X Windows *window manager*. A window manager is an X Windows application that improves X Windows's ease-of-use with such features as adding scroll bars to windows, and making it easier to resize, minimize, and expand windows. You must have an X Windows server installed before using an X Windows window manager.

The features of a window-manager application in X Windows are the kind of thing we Mac users take for granted—the Mac graphical interface has always provided such features, but with X Windows you need another application to add them.

OroborOSX is a window manager designed to work with XonX. OroborOSX makes windows from X Windows applications look almost identical to native Mac OS X (that is, Aqua) windows. It allows windows from X Windows applications to be “interleaved” with native Aqua windows (if you are using XDarwin without OroborOSX, then all the X Windows windows come to the foreground together if any of them are selected.)

You can download the OroborOSX software, and find documentation and other helpful resources, at the OroborOSX Web site (<http://oroborosx.sourceforge.net>).

### To install OroborOSX:

1. Download the software from the OroborOSX Web site.

The software is supplied as a compressed **tar** archive. If you have downloaded the archive using a Web browser, it was probably decompressed automatically. Otherwise, you can unpack it as you would any other compressed **tar** file.

2. Move the uncompressed OroborOSX folder into your Applications folder.

Launch OroborOSX by double-clicking the OroborOSX application in the Finder.

If you want to launch it from the command line, you need to use its full path (as of the version 0.8 beta):

```
open /Applications/OroborOSX-Folder/OroborOSX-v0.8b.app
```

OroborOSX launches its own version of XDarwin, which runs only in rootless mode.

**Figure 15.14** shows the **nedit** application in use (compare with Figure 15.9, where OroborOSX was not installed). This time, the **nedit** windows look more like the native Mac OS X (Aqua) windows.



**Figure 15.14** Nedit being used under OroborOSX. Note the Aqua-like appearance of the nedit windows.

## Darwin Streaming Server

The Darwin Streaming Server enables you to run your own Internet radio or television station. It is the open-source version of Apple's QuickTime Streaming Server. In fact, in many places in the software you will see the name "QuickTime Streaming Server"—the Darwin Streaming Server is the same code under a different name.

This server enables you to serve movies (in QuickTime format, also known as MPEG-4) and audio files (MP3 files). You can create playlists and set them to run repeatedly and/or randomly, and you can set your server to feed into another streaming server (to take advantage of another server's greater bandwidth). The Darwin Streaming Server listens on ports 554, 7070, 8000, and 8001. Only one of these needs to be available for people to reach the server. The software comes with a separate Web server, which is used to administer the streaming server on port 1220.

The Darwin Streaming Server can be downloaded as a Macintosh Disk Image file (a .dmg file) or as source code. To download the source code, you must be a member of the Apple Developer Connection, which is free.

If you download the Disk Image version, then installation is a simple matter of double-clicking the QTSS.pkg file in the disk image. You are then taken through a GUI installer process.

### To install the streaming server using the GUI installer:

1. Download the compiled binary version. You will find links for downloading the software on Apple's "Streaming Server" page (<http://developer.apple.com/darwin/projects/streaming/>). You will need to register with Apple. This is free and very easy. Look for the section entitled "Download Release 4.1.1 to get the compiled binary versions. The file you are downloading is an Apple disk-image file (a .dmg file). In version 4.1 of the Darwin Streaming Server, the file is named QTSS\_4.1.dmg (QTSS for "QuickTime Streaming Server")—it's the same software.
  - ◆ If your Web browser automatically launches the .dmg file, the file is mounted as if it is a new disk partition.
  - ◆ If your Web browser doesn't launch the .dmg file, just double-click the file. That launches the Disk Copy application, which mounts the file as if it is a disk partition.
2. Inside the newly mounted disk partition is a package file, DarwinStreamingServer.pkg. Double-click the package file. You are taken through a standard Macintosh software installation (you need to enter an admin password to install the software).

A special consideration when going through the install process: At one point you are asked which partition the software should be installed on. *Make sure you select the partition that your / directory is on.* **Figure 15.15** shows this stage in the installation process.

The installer puts its files in the /Applications directory, in /Library, in /System/Library/StartupItems, in /usr/bin, and in /usr/sbin. If you choose a partition other than the one that / is mounted on, the installer creates those directories on the selected partition (so if you select a partition that is mounted as /Volumes/partition2, the installer creates /Volumes/partition2/Applications, /Volumes/partition2/Library, and so forth).



**Figure 15.15** The Darwin Streaming Server installation screen where you select the partition on which to install the software.

The miniature Web server for administering the streaming server is running (on port 1220) when you finish the installation. In fact, the installer tries to launch a Web browser for you, giving it the URL <http://127.0.0.1:1220/>.

You use this Web server to configure the streaming server. See the task “To start and administer the streaming server,” below.

If you download the source-code version of the server, the installation process is similar to, but a little different from, that for other servers we have described. The source-code version is installed in various subdirectories of /usr/local.

### To install the Darwin Streaming Server from source code:

1. Download the compressed tar file.
2. Unpack the tar file.
3. Cd into the new directory.
4. `./Buildit`  
This compiles the software.
5. `sudo ./Install`  
The installation itself must be done as root.

## To start and administer the streaming server:

1. Use the supplied Web server.

The Darwin Streaming Server comes with a tiny Web server (written in Perl) that is used to administer the streaming server. This admin server should have started up when you installed the streaming server, but if it didn't, it is easy enough to start:

```
sudo /System/Library/StartupItems
→ /QuickTimeStreamingServer/QuickTime
→ StreamingServer start
```

Enter your password if prompted for it.

When the server starts up (which may take several seconds), you see

Starting QuickTime Streaming Server  
→ Admin

The admin server listens on port 1220.

2. Use a Web browser to connect to <http://127.0.0.1:1220/>.

You can also use the URL <http://localhost:1220/>.

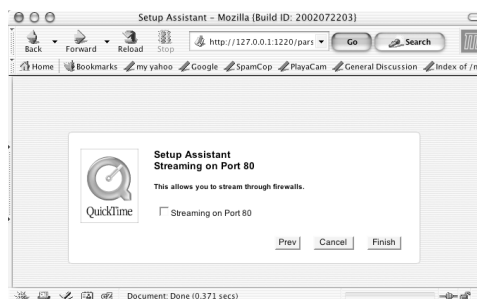
The first time you connect to the admin server, you are presented with a series of Setup Assistant pages, which take you through the process of configuring the streaming server.

The key entry you have to make is the location (Media Folder) of your media files. **Figure 15.16** shows the Web page where you make this selection. You can change it later, so don't panic, but if there aren't any movies or MP3 files in this folder, you'll have nothing to serve.

One other important choice to make in the Setup Assistant is deciding if you want to activate "Streaming on Port 80" (**Figure 15.17**). If you do this, then people who are behind many firewalls can reach your streaming server on port 80.



**Figure 15.16** The Setup Assistant Web interface showing the page where you select the Media Folder where you store the media files you wish to serve.



**Figure 15.17** The Setup Assistant page where you can choose to have the streaming server listen on port 80. This will only work if no other servers are listening on port 80.

Port 80 is the standard port for Web servers, so it is often not blocked by firewalls. You can only run one server on port 80, so if you are already running a Web server on port 80 (for example, if you have activated Web sharing in the Sharing panel of System Preferences), you cannot also run the streaming server on port 80.

3. Use the Web interface to administer the streaming server.

The Web interface allows you to start and stop the streaming server, control how much bandwidth it uses, create playlists, and view a list of current connections.

The Apple QuickTime Streaming Help Web site covers the features available through the admin interface (<http://help.qt.apple.com/qtssWebAdminHelpR3/qtssWebAdmin.help/English.lproj/QTSSHelp.htm>).

#### ✓ Tip

- The admin Web server is a Perl script that you may wish to examine. It is installed as `/usr/sbin/streamingadminserver.pl`.
- The configuration file for the streaming server is an XML file: `/Library/QuickTimeStreaming/Config/streamingserver.xml`.

All the settings you make using the Web interface are saved in this file, plus there are many other settings that are not shown in the Web interface. You can examine that file and make changes, but make a copy first!

## Mailman Email List Management

Mailman is an open source email list-management system that provides a nice Web interface for managing email lists and viewing the archives of email lists. Mailman is available from [www.list.org](http://www.list.org).

Email lists are used for recreation, customer support, software development, announcements, and many other purposes.

Your machine needs to have Fully Qualified Domain Name (FQDN) to properly use Mailman. See Chapter 14, “Installing and Configuring Servers,” for more about FQDN’s.

Here are some Mac OS X-specific notes to have handy alongside the Mailman INSTALL file:

- ◆ Configure Sendmail as we described in Chapter 14.  
The Mailman software comes with an INSTALL file that describes the installation and configuration process in a good, step-by-step fashion.
- ◆ Mailman uses the Python programming language and Python interpreter, which are already installed in Mac OS X, so you don’t have to worry about installing them.
- ◆ You need to add a new user, called *mailman*. Do this using the Accounts tool in System Preferences. The new user’s home directory (`/Users/mailman`) is where the Mailman software gets installed. Do *not* give this new user the ability to administer your machine.
- ◆ You need to create a new Unix group, called *mailman*, and add the new user *mailman* to the new group *mailman*. See chapter 11 for instructions.
- ◆ Throughout the Mailman installation, you see references to “\$prefix.” In your case, this is `/Users/mailman`. After you have created the new *mailman* group, you need to change the group ownership and permissions on that directory:  

```
sudo chgrp -R mailman /Users/mailman
sudo chmod 2775 /Users/mailman
```
- ◆ Download and install the Mailman software itself from [www.list.org](http://www.list.org). Your installation process will go more smoothly if you do the download, configuration and installation as the new *mailman* user.  

```
su - mailman
```

to become the user *mailman*. Now you’re ready to download and install the software, but only after reading the following notes.
- ◆ When you run the `configure` script:  
First set the environment variable `WWWHOST` to your FQDN; for example:  

```
setenv WWWHOST mac.politburo.com
```

(see Chapter 7 for more on setting environment variables). Then run the `configure` script:  

```
./configure --prefix=/Users/mailman
→ --with-mail-gid=smmosp
```

The two options there tell the `configure` script where you are installing Mailman, as well as the group name that Sendmail runs as.
- ◆ The Mailman installation instructions tell you to make various changes to your Web server configuration. In your case we suggest you do these all in the file `/etc/httpd/users/mailman.conf`, which was created when you created the new *mailman* user using the Accounts tool in System Preferences. **Figure 15.18** is an example of this file after making the changes.



```
# For the mailman logo icons
#
Alias "/mailman/icons/" "/Users/mailman/icons/"
<Directory "/Users/mailman/icons/">
    AllowOverride None
    Order allow,deny
    Allow from all
</Directory>

# For the mailman CGI scripts. Make sure this comes after the
# /mailman/icons/ Alias
#
ScriptAlias "/mailman/" "/Users/mailman/cgi-bin/"
<Directory "/Users/mailman/cgi-bin/">
    Options ExecCGI
    AllowOverride None
    Order allow,deny
    Allow from all
</Directory>

# Alias for the mailman list archives.
#
Alias "/pipermail/" "/Users/mailman/archives/public/"
<Directory "/Users/mailman/archives/public/">
    Options FollowSymLinks
    AllowOverride None
    Order allow,deny
    Allow from all
</Directory>

# The stuff below this comment was the standard
# /etc/httpd/users/mailman.conf file
#
<Directory "/Users/mailman/Sites/">
    Options Indexes MultiViews
    AllowOverride None
    Order allow,deny
    Allow from all
</Directory>
```

**Figure 15.18** Example of `/etc/httpd/users/mailman.conf` after adding the directives for the Mailman software.

## Chapter 15

Note that we are suggesting you do *not* copy the Mailman logo icons from the Mailman directory to another directory, but instead tell the Web server where to find them in the mailman home directory. If you do as we suggest, you also need to add a line to the end of `/Users/mailman/Mailman/mm_cfg.py` that says:

```
IMAGE_LOGOS = '/Users/mailman/icons/'
```

- ◆ Remember that anywhere you see `$prefix` in the Mailman installation notes you should substitute `/Users/mailman`.
- ◆ In order for Mailman to work properly, you need to add a series of entries to the file `/etc/mail/aliases` (which you probably need to create). The Sendmail configuration must be changed to use this file. You do this by uncommenting the line:

```
#0 AliasFile=/etc/mail/aliases
in /etc/mail/sendmail.cf and then
restarting Sendmail. Better yet, go and re-
read the instructions in Chapter 14 "To
change the Sendmail configuration:" and
remove the line:
```

```
undefine(`ALIAS_FILE')
```

from the template file you are using and re-generate `/etc/mail/sendmail.cf`.

Note that when you run the `newlist` command (described in the Mailman installation notes), it gives you a set of entries to add to the `/etc/mail/aliases` file. Any time you change that file you must then run:

```
sudo newaliases
```

to get Sendmail to notice the changes. (Run that from a shell where you are not the mailman user, which should *not* have `sudo` privileges.)

- ◆ See Chapter 14 for instructions on restarting the Apache Web server.

If you have followed all the Mailman installation instructions, as modified by the notes above, you should be able to see the Mailman web interface at:

`http://localhost/mailman/admin/` (you can substitute your machine's FQDN for 'localhost' if you are viewing it from a machine other than the one where you installed Mailman).

## Majordomo Email-List Management

Majordomo is an open-source software package for creating, managing, and using email lists. Majordomo works in conjunction with your mail-server software (such as Sendmail) to allow users to subscribe, unsubscribe, and administer email lists. Except for the creation of new lists, all commands are handled via email.

Majordomo has been around for over ten years and is widely used. It is, however, more difficult to set up than Mailman (described earlier in this chapter) and so you are likely to use majordomo because someone else you are working with is already using it.

You should only install Majordomo with the assistance of an experienced Unix system administrator. The installation and configuration process requires that you make configuration changes to Sendmail and that you make some decisions about how to handle security. We *strongly* suggest that you discuss this with someone who has been through it before. Of course, there is a Majordomo email list you can subscribe to in order to learn more and seek out help. There is a Web-based interface to Majordomo (called Webcool) that you might also be interested in.

The Majordomo Web site is [www.greatcircle.com/majordomo/](http://www.greatcircle.com/majordomo/); the source code can be obtained there.

### Requirements for installing and configuring Majordomo

- ◆ Having Sendmail or another Mail Transport Agent running on your machine.
- ◆ Downloading and installing the Majordomo software itself.

- ◆ Deciding how to handle security for the directory in which Majordomo is installed. Adapted from the Majordomo installation notes:

“The default installation of Majordomo ... WILL NOT RESULT IN A SECURE INSTALLATION. In particular, the Majordomo home directory and the “wrapper” program are, by default, accessible to any user. These open privileges can be (mis)used to change list membership, list configuration details, forge email, perhaps even create and/or delete lists, and anything else that the Majordomo user has permissions to do.”

In order to create a secure installation of Majordomo, you need to change the default permissions on the Majordomo directory after installation, and then configure Sendmail to use the “wrapper” program that is supplied with Majordomo. Explaining how to do this is beyond the scope of this book, and you should seek the help of someone who has done it before.

### Where to get help with Majordomo

The Majordomo Web site has links to several helpful resources:

- ◆ A Frequently Asked Questions page ([www.greatcircle.com/majordomo/majordomo-faq.html](http://www.greatcircle.com/majordomo/majordomo-faq.html)).
- ◆ Archives of the Majordomo users email list and information on subscribing to the list ([www.greatcircle.com/lists/majordomo-users/](http://www.greatcircle.com/lists/majordomo-users/)).

Also, the Majordomo source code comes with a collection of documentation in the Doc subdirectory of the source-code directory.

## TWIG—Web Interface for Email

TWIG (“The Web Information Gateway”) is an open-source “groupware” software package.

The term *groupware* generally means software intended to facilitate work by groups of people, usually by providing a single interface for email, scheduling, discussion areas, and similar features.

We are including TWIG here primarily for its email interface, but TWIG also provides a Web interface for scheduling, Usenet newsgroups, to-do lists, and several other tools.

**Figure 15.19** shows the TWIG demonstration screen from <http://twig.screwdriver.net/demo/>.

That screen shot shows only a portion of TWIG’s features. For a complete list, see <http://twig.screwdriver.net/about.php3>.

### Requirements for installing and configuring TWIG

- ◆ Having Sendmail running on your machine.
- ◆ Having installed an IMAP email server (as described earlier in this chapter).
- ◆ Modifying the Apache Web server that comes with Mac OS X.

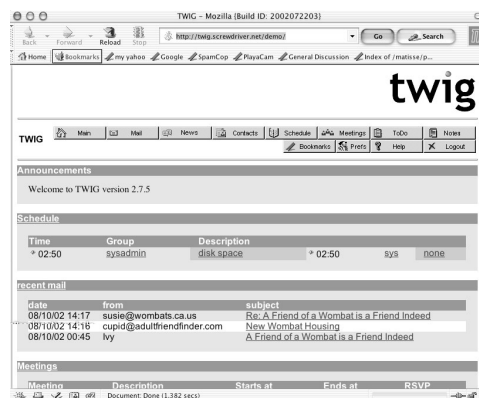
You need to download the source code for the Apache PHP module (from [www.php.net](http://www.php.net)) and compile it as a “Dynamic Shared Object” with support for IMAP. You then need to modify the Apache-configuration file to use the new PHP module you have built.

- ◆ TWIG requires a database.

It uses a simple one of its own (based entirely on text files), but for situations with heavy use you will want to install a SQL database such as MySQL (covered in Chapter 14, “Installing and Configuring Servers”).

### Where to get help with TWIG

- ◆ Download the TWIG source code (from <http://twig.screwdriver.net/download.php3>) and read the documentation that comes with it.  
That will help you decide what additional help you will need.
- ◆ Read the TWIG Frequently Asked Questions page (<http://twig.screwdriver.net/file.php3?file=FAQ&plaintext=1>).
- ◆ Subscribe to the TWIG email list (<http://twig.screwdriver.net/lists.php3>).



**Figure 15.19** The TWIG demonstration web site.