
Mac OS X Server
Command-Line Administration
For Version 10.5 Leopard

Apple
1 Infinite Loop
Cupertino CA 95014-2084
408-996-1010
www.apple.com

019-0947/2007-11-01

Contents

About This Guide

This guide describes Mac OS X Server command-line tools and commands, including the syntax, purpose, and parameters, and provides examples of usage and output.

Command-Line Administration is written for system administrators familiar with administering and managing servers, storage, and networks.

Beneath the interface of Mac OS X is a core operating system known as Darwin. Darwin integrates a number of technologies, most importantly Mach 3.0, operating-system services based on Berkeley Software Distribution (BSD) release 4.4 high-performance networking facilities, and support for multiple integrated file systems.

Darwin maintains most of the functionality of BSD 4.4 commands. Although some commands are modified, most commands are kept as is, or their functionality has been extended to support Apple-specific technologies.

This guide focuses on commands developed by Apple to allow administrators to perform functions available in the graphical interface from the command line. The guide also highlights BSD commands that have been modified or extended to support Apple-specific functionality. Finally, the guide describes important commands commonly used by UNIX system administrators.

Note: Because Apple periodically releases new versions and updates to its software, images shown in this book may be different from what you see on your screen.

Using This Guide

This guide describes commands that perform functions used to configure and manage Mac OS X computers. Chapters in this guide describe sets of commands that work for specific aspects of the operating system.

Use this guide to:

- Learn which commands are available for specific tasks
- Learn how the commands work, and how to execute them
- Review examples of command usage

Understanding Notation Conventions

The following conventions are used throughout this book.

Summary

Notation	Indicates
`monospaced` font	A command or other text typed in a Terminal window
`$`	A shell prompt
`[text_in_brackets]`	An optional parameter
`(one|other)`	Alternative parameters (use one or the other)
`italicized`	A parameter you must replace with a value
`[...]`	A parameter that can be repeated
`<angle brackets>`	A displayed value that depends on your server configuration

Commands and Other Terminal Text

Commands or command parameters that you enter, along with other text that appears in a Terminal window, are shown in `this` font. For example:

You can use the `doit` command to get things done.

When a command is shown on a line by itself in this manual, it is preceded by a dollar sign and a space that represent the shell prompt. For example:

```
$ doit
```

To use this command, enter it without the dollar sign and the space in a Terminal window, and then press Return. (Terminal is found in /Applications/Utilities/.)

Command Parameters and Options

Most commands require parameters to specify command options or the item to which the command is applied to.

Parameters You Must Enter as Shown

If you must enter a parameter as shown, it appears following the command in the same font. For example:

```
$ doit -w later -t 12:30
```

To use the command in this example, enter the entire line as shown (without the `$` and space).

Parameter Values You Provide

If you must provide a value, its placeholder is italicized and has a name that indicates what you need to provide. For example:

```
$ doit -w later -t hh:mm
```

In this example, you replace *hh* with the hour and *mm* with the minute, as shown in the previous example.

Optional Parameters

If a parameter is not required, it appears in square brackets. For example:

```
$ doit [-w later]
```

To use the command in this example, enter `doit` or `doit -w later`. The result might vary, but you perform the command either way.

Alternative Parameters

If you must enter one of a number of parameters, they're separated by a vertical line and grouped within parentheses (|). For example:

```
$ doit -w (now|later)
```

To perform this command, enter `doit -w now` or `doit -w later`.

Default Settings

Descriptions of server settings usually include the default value for each setting. When this default value depends on your configuration (such as the name or IP address of your server), it's enclosed in angle brackets.

For example, the default value for the IMAP mail server is the host name of your server. This is indicated by `mail:imap:servername = "<hostname>."`

Commands Requiring Root Privileges

Throughout this manual, commands that require root privileges begin with `sudo`.
See "Commands Requiring Root Privileges" on page 26.

Mac OS X Server Administration Guides

Getting Started covers installation and setup for standard and workgroup configurations of Mac OS X Server. For advanced configurations, *Server Administration* covers planning, installation, setup, and general server administration. A suite of additional guides, listed below, covers advanced planning, setup, and management of individual services. You can get these guides in PDF format from the Mac OS X Server documentation website:

www.apple.com/server/documentation

This guide ...	tells you how to:
Getting Started and *Mac OS X Server Worksheet*	Install Mac OS X Server and set it up for the first time.
Command-Line Administration	Install, set up, and manage Mac OS X Server using UNIX command-line tools and configuration files.
File Services Administration	Share selected server volumes or folders among server clients using the AFP, NFS, FTP, and SMB protocols.
iCal Service Administration	Set up and manage iCal shared calendar service.
iChat Service Administration	Set up and manage iChat instant messaging service.
Mac OS X Security Configuration	Make Mac OS X computers (clients) more secure, as required by enterprise and government customers.
Mac OS X Server Security Configuration	Make Mac OS X Server and the computer it's installed on more secure, as required by enterprise and government customers.
Mail Service Administration	Set up and manage IMAP, POP, and SMTP mail services on the server.
Network Services Administration	Set up, configure, and administer DHCP, DNS, VPN, NTP, IP firewall, NAT, and RADIUS services on the server.
Open Directory Administration	Set up and manage directory and authentication services, and configure clients to access directory services.
Podcast Producer Administration	Set up and manage Podcast Producer service to record, process, and distribute podcasts.
Print Service Administration	Host shared printers and manage their associated queues and print jobs.
QuickTime Streaming and Broadcasting Administration	Capture and encode QuickTime content. Set up and manage QuickTime streaming service to deliver media streams live or on demand.
Server Administration	Perform advanced installation and setup of server software, and manage options that apply to multiple services or to the server as a whole.
System Imaging and Software Update Administration	Use NetBoot, NetInstall, and Software Update to automate the management of operating system and other software used by client computers.
Upgrading and Migrating	Use data and service settings from an earlier version of Mac OS X Server or Windows NT.

This guide ...	tells you how to:
User Management	Create and manage user accounts, groups, and computers. Set up managed preferences for Mac OS X clients.
Web Technologies Administration	Set up and manage web technologies, including web, blog, webmail, wiki, MySQL, PHP, Ruby on Rails, and WebDAV.
Xgrid Administration and High Performance Computing	Set up and manage computational clusters of Xserve systems and Mac computers.
Mac OS X Server Glossary	Learn about terms used for server and storage products.

Viewing PDF Guides Onscreen

While reading the PDF version of a guide onscreen:

- Show bookmarks to see the guide's outline, and click a bookmark to jump to the corresponding section.
- Search for a word or phrase to see a list of places where it appears in the document. Click a listed place to see the page where it occurs.
- Click a cross-reference to jump to the referenced section. Click a web link to visit the website in your browser.

Printing PDF Guides

If you want to print a guide, you can take these steps to save paper and ink:

- Save ink or toner by not printing the cover page.
- Save color ink on a color printer by looking in the panes of the Print dialog for an option to print in grays or black and white.
- Reduce the bulk of the printed document and save paper by printing more than one page per sheet of paper. In the Print dialog, change Scale to 115% (155% for *Getting Started*). Then choose Layout from the untitled pop-up menu. If your printer supports two-sided (duplex) printing, select one of the Two-Sided options. Otherwise, choose 2 from the Pages per Sheet pop-up menu, and optionally choose Single Hairline from the Border menu. (If you're using Mac OS X v10.4 or earlier, the Scale setting is in the Page Setup dialog and the Layout settings are in the Print dialog.)

You may want to enlarge the printed pages even if you don't print double sided, because the PDF page size is smaller than standard printer paper. In the Print dialog or Page Setup dialog, try changing Scale to 115% (155% for *Getting Started*, which has CD-size pages).

Getting Documentation Updates

Periodically, Apple posts revised help pages and new editions of guides. Some revised help pages update the latest editions of the guides.

- To view new onscreen help topics for a server application, make sure your server or administrator computer is connected to the Internet and click "Latest help topics" or "Staying current" in the main help page for the application.
- To download the latest guides in PDF format, go to the Mac OS X Server documentation website:

  www.apple.com/server/documentation

Getting Additional Information

For more information, consult these resources:

- *Read Me documents*—important updates and special information. Look for them on the server discs.
- *Mac OS X Server website* (www.apple.com/server/macosx)—gateway to extensive product and technology information.
- *Mac OS X Server Support website* (www.apple.com/support/macosxserver)—access to hundreds of articles from Apple's support organization.
- *Apple Training website (*www.apple.com/training)—instructor-led and self-paced courses for honing your server administration skills.
- *Apple Discussions website* (discussions.apple.com)—a way to share questions, knowledge, and advice with other administrators.
- *Apple Mailing Lists website* (www.lists.apple.com)—subscribe to mailing lists so you can communicate with other administrators using email.
- *Man pages* (developer.apple.com/documentation/Darwin/Reference/ManPages)—The Apple Developer Connection (ADC) Reference Library contains man pages for many BSD and POSIX functions and applications included with Mac OS X.
- *The public source website* (developer.apple.com/darwin)—Access to Darwin source code, developer information, and FAQs.

Executing Commands

1

Use this chapter to learn how to execute commands and to view online information about commands and tools.

A command-line interface is a way for you to manipulate your computer in situations where a graphical approach is not available. The Terminal application is the Mac OS X gateway to the BSD command-line interface (UNIX shell command prompt).

Each window in Terminal contains an execution context, called a *shell*, that is separate from all other execution contexts. The shell is an interactive programming language interpreter, with a specialized syntax for executing commands and writing structured programs called shell scripts.

Different shells feature slightly different capabilities and programming syntax. Although you can use any shell, the examples in this book assume that you are using `bash`, the standard Mac OS X shell.

UNIX 03 Certification

Mac OS X Server v10.5 is now an "Open Brand UNIX 03 Registered Product," conforming to the SUSv3 and POSIX 1003.1 specifications for the C API, Shell Utilities, and Threads.

Because Mac OS X Server v10.5 can compile and run your existing UNIX 03-compliant code, you can deploy it in environments that demand full conformance.

At the same, Mac OS X Server v10.5 provides full compatibility with existing server and application software.

Opening Terminal

To enter shell commands or run server command-line tools, you need access to the UNIX shell prompt on the local server or on a remote server.

To open Terminal, click the Terminal icon in the dock or double-click the application icon in the Finder (in /Applications/Utilities/).

Terminal presents a prompt when it is ready to accept a command. The prompt you see depends on your Terminal and shell preferences, but it often includes the name of the host you're logged in to, your current working folder, your user name, and a prompt symbol.

For example, if you're using the default `bash` shell, the prompt appears as:

```
server1:~ anne$
```

where you are logged in to a computer named *server1* as the user named *anne*, and your current folder is anne's home folder (`~`).

Throughout this manual, where a command is shown, the prompt is abbreviated as `$`.

Specifying Files and Folders

Most commands operate on files and folders, the locations of which are identified by paths. The folder names that make up a path are separated by slash characters. For example, the path to the Terminal application is /Applications/Utilities/Terminal.app.

Standard shortcuts used to represent specific folders are shown in the following table. Because they are relative to the current folder, these shortcuts eliminate the need to enter full paths in many situations.

Path string	Description
.	A single period represents the current folder. This value is often used as a shortcut to eliminate the need to enter in a full path. For example, the string "./Test.c" represents the Test.c file in the current folder.
..	Two periods represent the parent folder of the current folder. This string is used for navigating up one level from the current folder through the folder hierarchy. For example, the string "../Test" represents a sibling folder (named Test) of the current folder.
~	The tilde character represents the home folder of the user logged in. In Mac OS X, this folder resides in the local /Users folder or on a network server. For example, to specify the Documents folder of the current user, you would specify ~/Documents.

File and folder names traditionally include letters, numbers, a period, or the underscore character. Avoid most other characters, including space characters. Although some Mac OS X file systems permit the use of these other characters, including spaces, you might need to add single or double quotation marks around pathnames that contain them.

For individual characters, you can also "escape" the character—that is, put a backslash character immediately before the character in your string. For example, the pathname My Disk is "My Disk" or My\ Disk.

Standard Pipes

Many commands can receive text input from the user and print text to the console. They do so using *standard pipes*, which are created by the shell and passed to the command.

Standard pipes include:

- `stdin`—The standard input pipe is the means through which data enters a command. By default, the user enters this from the command-line interface. You can also redirect the output from files or other commands to `stdin`.
- `stdout`—The standard output pipe is where the command output is sent. By default, command output is sent to the command line. You can also redirect the output from the command line to other commands and tools.
- `stderr`—The standard error pipe is where error messages are sent. By default, errors are displayed on the command line like standard output.

Redirecting Input and Output

From the command line, you can redirect input and output from a command to a file or another command.

Redirecting output lets you capture the results of running the command and store it in a file for later use. Similarly, providing an input file lets you provide a command with preset input data, instead of needing to enter that data.

You can use the following characters to redirect input and output:

Redirect	Description
>	Use the greater-than character to redirect command output to a file.
<	Use the less-than character to use the contents of a file as input to the command.
>>	Use a double greater-than to append output from a command to a file.

In addition to using file redirection, you can also redirect the output of one command to the input of another using the vertical bar character, or *pipe*. You can combine commands in this manner to implement more sophisticated versions of the same commands.

For example, the command `man bash | grep "commands"` passes the formatted contents of the `bash` man page to the `grep` tool, which searches those contents for lines containing the word "commands." The result is a listing of lines with the specified text, instead of the entire man page.

For more information about redirection, see the `bash` man page.

Using Environment Variables

Some commands require the use of environment variables for their execution. Environment variables are inherited by all commands executed in the shell's context. The shell uses environment variables to store information, such as the name of the current user, the name of the host computer, and the paths to any commands.

You can create environment variables and use them to control the behavior of your command without modifying the command itself. For example, you can use an environment variable to have your command print debug information to the console.

To set the value of an environment variable, use the appropriate shell command to associate a variable name with a value. For example, to set the variable `PATH` to the value `/bin:/sbin:/user/bin:/user/sbin:/system/Library/`, you would enter the following command in a Terminal window:

```
$ PATH=/bin:/sbin:/user/bin:/user/sbin:/system/Library/ export PATH
```

This modifies the environment variable `PATH` with the value assigned.

To view all environment variables, enter the following:

```
$ env
```

When you launch an application from a shell, the application inherits much of the shell's environment, including exported environment variables. This form of inheritance can be a useful way to configure the application dynamically. For example, your application can verify for the presence (or value) of an environment variable and change its behavior accordingly.

Different shells support different semantics for exporting environment variables, so see the man page for your preferred shell for further information.

Although child processes of a shell inherit the environment of that shell, shells are separate execution contexts that do not share environment information with one another. Thus, variables you set in one Terminal window are not set in other Terminal windows.

After you close a Terminal window, variables you set in that window are gone. If you want the value of a variable to persist between sessions and in all Terminal windows, you must set it in a shell startup script.

Another way to set environment variables in Mac OS X is with a special property list in your home folder. At login, the computer looks for the ~/.MacOSX/environment.plist file. If the file is present, the computer registers the environment variables in the property list file.

Executing Commands and Running Tools

To execute a command in the shell, enter the complete pathname of the tool's executable file, followed by arguments, and then press Return.

If a command is located in one of the shell's known folders, you can omit path information and enter the command name.

The list of known folders is stored in the shell's PATH environment variable and includes the folders containing most command-line tools.

For example, to run the `ls` command in the current user's home folder, you could enter the following at the command line and press Return:

```
host:~ anne$ ls
```

To run a command in the current user's home folder, you would precede it with the folder specifier. For example, to run MyCommandLineProg, you would use something like the following:

```
host:~ anne$ ./MyCommandLineProg
```

To launch a tool package, you can use the open command (open MyProg.app) or launch the tool by entering the pathname of the executable file inside the package, usually something like ./MyProg.app/Contents/MacOS/MyProg.

When entering commands, if you get the message `command not found`, check your spelling. Here is an example:

```
server:/ anne$ sudo serversetup -getHostname
serversetup: Command not found.
```

If the error recurs, the command you're trying to run might not be in your default search path. You can add the path before the command name, for example:

```
server:/ anne$ sudo /System/Library/ServerSetup/serversetup -getHostname
server.example.com
```

or change your working folder to the folder that contains the tool. For example:

```
server:/ anne$ cd /System/Library/ServerSetup
server:/System/Library/ServerSetup anne$ sudo ./serversetup -getHostname
server.example.com
```

or

```
server:/System/Library/ServerSetup anne$ cd /
server:/ anne$ PATH="$PATH:/System/Library/ServerSetup"
server:/ anne$ sudo serversetup -getHostname
server.example.com
```

Correcting Typing Errors

You can use the Left and Right Arrow keys to correct typing errors before you press Return to execute a command.

To correct a typing error:

1. Press Left Arrow or Right Arrow to skip over parts of the command you don't want to change.
2. Press Delete to remove characters.
3. Enter regular characters to insert them.
4. Press Return to execute the command.

To ignore what you entered and start again, press Control–U.

Repeating Commands

To repeat a command, press Up Arrow until you see the command, then make modifications and press Return.

Including Paths Using Drag and Drop

To include a fully qualified filename or folder path in a command, you can drag and drop the folder or file from a Finder window into the Terminal window.

Searching for Text in a File

To locate a string within a file, use the `grep` tool. The `grep` tool searches the named input files for lines containing a match to the given pattern. By default, `grep` prints the matching lines.

To search for a unique string in a file:

```
$ grep search_string filename
```

Replace *search_string* with the the string to search for and *filename* with the name of the file you want to search through.

Commands Requiring Root Privileges

Many commands used to manage a server must be executed by the root user. If you get a message such as `permission denied`, the command probably requires root privileges.

However, when logged in as a root user, be careful: you have sufficient privileges to make changes that can cause your server to stop working.

Important: Don't execute commands as the root user unless you know what you're doing. Instead, log in as an administrator user and selectively use `sudo`, which gives you root user privileges to execute one command. This helps you avoid making unintended changes when running other commands.

The `sudo` command gives root user privileges to users specified in the sudoers file. If you're logged in as an administrator user and your username is specified in the etc/sudoers file, you can use this command.

To execute a single command with root user privileges, begin the command with `sudo` (short for super user do). For example:

```
$ sudo serveradmin list
```

If you haven't used `sudo` recently, you're prompted for your administrator password.

To switch to the root user so you don't need to repeatedly enter `sudo`, use the `su` command:

```
$ su root
```

or simply:

```
$ su
```

You're prompted for the root user password and are then logged in as the root user until you log out or use the `su` command to switch to another user.

Note: The root user password is set to the administrator user password when you install Mac OS X Server.

Important: To avoid running commands as root, log out after you finish using the `su` command.

For more information about the `sudo` and `su` commands, see their man pages.

Terminating Commands

To terminate the currently running command, enter Control-C. This keyboard shortcut sends an abort signal to the command. In most cases this causes the command to terminate, although commands can install signal handlers to trap this signal and respond differently.

Scheduling Tasks

To schedule tasks to run at defined times, use the `cron` tool. This tool is a daemon that executes scheduled commands defined in crontab files.

The `cron` tool searches the /var/cron/tabs/ folder for crontab files that are named after accounts in /etc/passwd, and loads the files into memory. The `cron` tool also searches for crontab files in the /etc/crontab/ folder, which are in a different format. `cron` then cycles every minute, examining stored crontab files and checking each command to see if it should be run in the current minute.

When commands execute, output is mailed to the owner of the crontab file or to the user named in the MAILTO environment variable in the crontab file, if one exists.

If you modify a `crontab` file, you must restart `cron`.

You use `crontab` to install, deinstall, or list the tables used to drive the `cron` daemon. Users can have their own crontab file.

To configure your crontab file, use the `crontab -e` command. This displays an empty crontab file.

An example of a configured crontab file:

```
SHELL=/bin/sh
PATH=/bin:/sbin:/usr/bin:/usr/sbin
HOME=/var/log

#min hour mday month wday     command
30   18   *    *     1-5      diskutil repairPermissions /Volumes/MacHD
50   23   *    *     0        diskutil repairVolume /Volumes/MacHD
```

Listed below is an explanation of the crontab structure shown above.

The following crontab entry repairs disk permissions for the MacHD volume at 18:30 every day, Monday through Friday:

```
30   18   *    *     1-5      diskutil repairPermissions /Volumes/MacHD
```

The following crontab entry schedules a repair volume operation to run at 23:50 every Sunday:

```
50 23 * * 0 diskutil repairVolume /Volumes/MacHD
```

Sending Commands to a Remote Computer

You must connect to a remote computer before you can execute commands on it. You can send commands to a remote computer using:

- Secure Shell (SSH), a tool for logging in to a remote computer and for executing commands on a remote computer.
- Telnet, a tool for communicating with another computer using the TELNET protocol.

For information about sending commands to remote computers, see Chapter 2, “Connecting to Remote Computers,” on page 31.

Viewing Command Information

Most command-line documentation comes in the form of man pages. These formatted pages provide reference information for shell commands, tools, and high-level concepts.

You can also access command information using the `help` command, and sometimes information is displayed if you enter the command without parameters or options.

To access a man page:

```
$ man command
```

where *command* is the topic you want to find information about. The man page contains detailed information about the command, its options, parameters, and proper use.

For help using the man command, enter:

```
$ man man
```

If man pages are too long to fit on your screen, use the `more` or `less` command to paginate the file. This allows you to view the file faster by loading screens of the man page at a time, rather than the entire file:

```
$ man serveradmin | less
```

When you use `more` or `less`, an information bar appears at the bottom of the screen. When you see the bar, you can press the Space bar to go to the next page, the B key to go back a page, or the Return key to scroll the file forward one line at a time.

When you get to the end of a file, `more` returns you to the prompt and `less` waits for you to press the Q key to quit.

Several third-party Mac OS X applications are available for viewing formatted man pages in scrollable windows. You can find one by choosing Mac OS X Software from the Apple menu and then searching for "man page."

Note: Not all commands and tools have man pages. For a list of available man pages, look in /usr/share/man.

To access command help:

- Enter the command followed by the `-help`, `-h`, `--help`, or `help` parameter:

```
$ hdiutil help
$ dig -h
$ diff --help
```

To view a list of options and parameters you can use with the command:

- Enter the command without options or parameters:

```
$ sudo serveradmin
```

Note: Not all techniques work for all commands, and some commands don't have onscreen help.

Connecting to Remote Computers 2

Use this chapter to learn the commands to connect to remote computers.

Connecting to remote computers helps you manage and configure resources efficiently. This chapter covers using Secure Shell (SSH) and Telnet to connect to remote computers.

Understanding SSH

SSH lets you send secure, encrypted commands to a computer remotely, as if you were sitting at the computer. You use the `ssh` tool in Terminal to open a command-line connection to a remote computer. While the connection is open, commands you enter are performed on the remote computer.

Note: You can use any application that supports SSH to connect to a computer running Mac OS X or Mac OS X Server.

How SSH Works

SSH works by setting up encrypted tunnels using public and private keys. Here is a description of an SSH session:

1. The local and remote computers exchange public keys.

   If the local computer has never encountered a given public key, SSH and your web browser prompt you whether to accept the unknown key.
2. The two computers use the public keys to negotiate a session key used to encrypt subsequent session data.
3. The remote computer attempts to authenticate the local computer using RSA or DSA certificates. If this is not possible, the local computer is prompted for a standard user-name/password combination.
4. After successful authentication, the session begins and remote shell, a secure file transfer, a remote command, or other action is begun through the encrypted tunnel.

The following are SSH tools:

- `sshd`—Daemon that acts as a server to all other commands
- `ssh`—Primary user tool that includes a remote shell, remote command, and port-forwarding sessions
- `scp`—Secure copy, a tool for automated file transfers
- `sftp`—Secure FTP, a replacement for FTP

Generating Key Pairs for Key-Based SSH Connections

By default, SSH supports the use of password, key, and Kerberos authentication. The standard method of SSH authentication is to supply login credentials in the form of a user name and password. Identity key pair authentication enables you to log in to the server without supplying a password.

Key-based authentication is more secure than password authentication because it requires that you have the private key file and know the password that lets you access that key file. Password authentication can be compromised without a private key file.

This process works as follows:

1. A private and a public key are generated, each associated with a user name to establish that user's authenticity.
2. When you attempt to log in as that user, the user name is sent to the remote computer.
3. The remote computer looks in the user's .ssh/ folder for the user's public key.

   This folder is created after using SSH the first time.
4. A challenge is sent to the user based on his or her public key.
5. The user verifies his or her identity by using the private portion of the key pair to decode the challenge.
6. After the key is decoded, the user is logged in without the need for a password.

   This is especially useful when automating remote scripts.

Note: If the server uses FileVault to encrypt the home folder of the user you want to use SSH to connect as, you must be logged in on the server to use SSH. Alternatively, you can store the keys for the user in a location that is not protected by FileVault, but this is not secure.

To generate the identity key pair:

1. Enter the following command on the local computer:

```
$ ssh-keygen -t dsa
```

2. When prompted, enter a filename in the user's folder to save the keys in; then enter a password followed by password verification (empty for no password).

   For example:

```
Generating public/private dsa key pair.
     Enter file in which to save the key (/Users/anne/.ssh/id_dsa): frog
     Enter passphrase (empty for no passphrase):
     Enter same passphrase again:
     Your identification has been saved in frog.
     Your public key has been saved in frog.pub.
     The key fingerprint is:
     4a:5c:6e:9f:3e:35:8b:e5:c9:5a:ac:00:e6:b8:d7:96 annejohnson1@mac.com
```

   This creates two files. Your identification or private key is saved in one file (*frog* in our example) and your public key is saved in the other (*frog*.pub in our example).

   The key fingerprint, which is derived cryptographically from the public key value, also appears. This secures the public key, making it computationally infeasible for duplication.

3. Copy the resulting public file, which contains the local computer's public key, to the .ssh/authorized_keys file in the user's home folder on the remote computer (~/.ssh/authorized_keys).

   The next time you log in to the remote computer from the local computer you won't need to enter a password.

   Note: If you are using an Open Directory user account and have logged in using the account, you do not need to supply a password for SSH login. On Mac OS X Server computers, SSH uses Kerberos for single sign-on authentication with any user account that has an Open Directory password. (Kerberos must be running on the Open Directory server.) For more information, see *Open Directory Administration*.

Updating SSH Key Fingerprints

The first time you connect to a remote computer using SSH, the local computer prompts for permission to add the remote computer's fingerprint (or encrypted public key) to a list of known remote computers. You might see a message like this:

```
The authenticity of host "server1.example.com" can't be established.
RSA key fingerprint is a8:0d:27:63:74:f1:ad:bd:6a:e4:0d:a3:47:a8:f7.
Are you sure you want to continue connecting (yes/no)?
```

The first time you connect, you have no way of knowing whether this is the correct host key. Most people respond "yes." The host key is then inserted into the ~/.ssh/known_hosts file so it can be verified in later sessions.

Be sure this is the correct key before accepting it. If possible, provide users with the encryption key through FTP, mail, or a download from the web, so they can be sure of the identity of the server.

If you later see a warning message about a *man-in-the-middle* attack (see below) when you try to connect, it might be because the key on the remote computer no longer matches the key stored on the local computer. This can happen if you:

- Change your SSH configuration on the local or remote computer.
- Perform a clean installation of the server software on the computer you are attempting to log in to using SSH.
- Start up from a Mac OS X Server CD on the computer you are attempting to log in to using SSH.
- Attempt to use SSH to access a computer that has the same IP address as a computer that you used SSH with on another network.

To connect again, delete the entries corresponding to the remote computer (which can be stored by name and IP address) in the file ~/.ssh/known_hosts.

An SSH Man-in-the-Middle Attack

Sometimes an attacker can access your network and compromise routing information, so that packets intended for a remote computer are routed to the attacker, who then impersonates the remote computer to the local computer and the local computer to the remote computer.

Here's a typical scenario: A user connects to the remote computer using SSH. By means of spoofing techniques, the attacker poses as the remote computer and receives information from the local computer. The attacker then relays the information to the intended remote computer, receives a response, and then relays the remote computer's response to the local computer.

Throughout the process, the attacker is privy to all information that goes back and forth, and can modify it.

A sign that can indicate a man-in-the-middle attack is the following message that appears when connecting to the remote computer using SSH.

```
@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@
@    WARNING: REMOTE HOST IDENTIFICATION HAS CHANGED!     @
@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@
```

Protect for this type of attack by verifying that the host key sent back is the correct host key for the computer you are trying to reach. Be watchful for the warning message, and alert your users to its meaning.

Important: Removing an entry from the known_hosts file bypasses a security mechanism that would help you avoid imposters and man-in-the-middle attacks. Before you delete its entry from the known_hosts file, be sure you understand why the key on the remote computer has changed.

Controlling Access to SSH Service

You can use Server Admin to control which users can open a command-line connection using the `ssh` tool in Terminal. Users with administrator privileges can always open a connection using SSH. The `ssh` tool uses the SSH service.
For information about controlling access to the SSH service, see *Open Directory Administration*.

Connecting to a Remote Computer

You can connect to a remote computer using SSH (secure) or Telnet (nonsecure).

Using SSH

Use the `ssh` tool to create a secure shell connection to a remote computer.

To access a remote computer using ssh:

1. Open Terminal.
2. Log in to the remote computer by entering the following command:

   ```
   $ ssh -l username server
   ```

   Replace *username* with the name of an administrator user on the remote computer. Replace *server* with the name or IP address of the remote computer. For example:

   ```
   $ ssh -l anne 10.0.1.2
   ```

   If this is the first time you've connected to the remote computer, you're prompted to continue connecting after the remote computer's RSA fingerprint appears.
3. Enter `yes`.
4. When prompted, enter the user's password for the remote computer.

   The command prompt changes to show that you're connected to the remote computer. In the case of the previous example, the prompt might look like this:

   ```
   10.0.1.2:~ anne$
   ```

5 To send a command to the remote computer, enter the command.

6 To close a remote connection, enter `logout`.

You can authenticate and send a command using a single line by appending the command to execute to the basic `ssh` tool. For example, to delete a file you could use:

```
$ ssh -l anne server1.example.com rm /Users/anne/Documents/report
```

or

```
$ ssh -l anne@server1.example.com "rm /Users/anne/Documents/report"
```

You're prompted for the user's password.

Using Telnet

Use the `telnet` tool to create a Telnet connection to a remote computer.

Because `telnet` isn't as secure as SSH, Telnet access is disabled by default.

To enable Telnet access:

```
$ sudo service telnet start
```

To disable Telnet access:

```
$ sudo service telnet stop
```

You are strongly advised not to enable Telnet. When you log in using Telnet, your login information, user name, and password (as well as your entire Telnet session) are passed over the Internet in clear text.

Any person on the network running `tcpdump`, `ethereal`, or similar applications can sniff the network and take possession of your user name and password. If you run something as root during your Telnet session, your root user account is also compromised.

To access a remote computer using telnet:

```
$ telnet -l username server
```

Replace *username* with the name of an administrator user on the remote computer. Replace *server* with the name or IP address of the remote computer. For example:

```
$ telnet -l anne 10.0.1.2
```

After being connected, the remote computer prompts for a login name and password. Depending on the type of computer you are accessing, you may see a message of the form:

```
TERM = (vt100)
```

Press Enter to accept this default setting.

You may see a series of messages on the screen, followed by the remote computer's prompt. You are now logged in.

When you finish working, log out from the remote computer by entering `logout` or `exit` at the remote computer's prompt. The telnet client exits when you log out from the remote computer.

For more information, see the `telnet` man page.

Remotely Controlling the Xserve Front Panel

You can use the `ipmitool` command to remotely control an Xserve's front panel.

To display the list of supported *virtual* front panel commands:

```
$ ipmitool chassis bootdev
bootdev <device> [clear-cmos=yes|no]
  none  : Do not change boot device order
  pxe   : Force PXE boot (LOM: Force boot NetBoot server)
  disk  : Force boot from default Hard-drive
  safe  : Force boot from default Hard-drive, request Safe Mode (LOM: Not
     used)
  diag  : Force boot from Diagnostic Partition (LOM: Force boot diagnostic
     mode from NetBoot server)
  cdrom : Force boot from CD/DVD
  bios  : Force boot into BIOS Setup (LOM: Not used)
Lights-out Management additional options
  nvram : Force reset of NVRAM
  tdm   : Force boot into Target Disk Mode
  other : Skip current startup disk selection, and boot from other
```

Mac OS X Server v10.5 supports the following commands: `none`, `pxe`, `disk`, `diag`, `cdrom`, `nvram`, `tdm`, and `other`.

For example, entering the following command and then restarting an Xserve system starts the system in Target Disk Mode:

```
$ ipmitool chassis bootdev tdm
```

After the system starts, the `ipmitool` command reverts to the default setting (none). Restarting the Xserve system without running the `ipmitool` command does not change the boot device order.

For more information about `ipmitool`, see its man page.

Installing Server Software and Finishing Basic Setup

3

Use this chapter to learn the commands to install, set up, and update Mac OS X Server software on local or remote computers.

This chapter explains the commands to perform software setup and installation tasks.

Some computers come with Mac OS X Server software installed. However, you might want to upgrade from a previous version, change a computer configuration, automate software installation, or refresh your server environment.

Installing Server Software

To install Mac OS X Server or other software on a computer, use the `/usr/sbin/installer` tool. You can use the `installer` tool locally or remotely.

The `installer` tool requires at least two arguments: the installation package and the destination of the installation package.

For a standard installation, your target would be the root drive. Here is an example installation command:

```
$ installer -pkg OSInstall.mpkg -target /
```

Other useful options include:

- `lang`—The operating system package requires that you choose a language. This flag allows you to do so from the command line. The argument is a two-character ISO language code. For English, it's `en`.
- `verbose`—Prints the details of the installation. It's useful for monitoring progress.

For more information, see the `installer` man page.

To use the installer to install Mac OS X Server software:

1. Start the target computer from the first installation CD or the installation DVD.

   The procedure you use depends on the target computer hardware:

   - If the target computer has a keyboard and an optical drive, insert the first installation disc into the optical drive; then hold down the C key on the keyboard while restarting the computer.
   - If the target computer is an Xserve with a built-in optical drive, start the computer using the first installation disc by following the instructions for starting from a system disc in the Xserve User's Guide.
   - If the target computer is an Xserve with no built-in optical drive, you can start it in target disk mode and insert the installation disc into the optical drive on your administrator computer. You can also use an external FireWire optical drive or an optical drive from another Xserve system to start the computer from the installation disc.

     Instructions for using target disk mode and external optical drives are in the *Quick Start* guide or *Xserve User's Guide* that came with your Xserve system.

2. If you're installing on a local computer, when Installer opens choose Utilities > Open Terminal to open the Terminal application.

   If you're installing on a remote computer, from Terminal on an administrator computer or from a UNIX workstation, establish an SSH session as the root user with the target computer, substituting *ip_address* with the target computer's actual IP address:

   ```
   $ ssh root@ip_address
   ```

   If you don't know the IP address, use the `sa_srchr` tool to identify computers, on the local subnet where you can install server software:

   ```
   $ /System/Library/Serversetup/sa_srchr 224.0.0.1
   mycomputer.example.com#PowerMac4,4#<ip address>#<mac address>#Mac OS X
        Server 10.5#RDY4PkgInstall#2.0#512
   ```

   You can also use Server Assistant to generate information for computers on the local subnet. To access the Destination pane and generate a list of computers awaiting installation in Open Server Assistant, select "Install software on a remote computer" and click Continue.

3. When prompted for a password, enter the first eight digits of the computer's built-in hardware serial number.

   To find a computer's serial number, look for a label on the computer. If the target computer is set up as a server, you'll also find the hardware serial number in /System/Library/ServerSetup/SerialNumber.

   If you're installing on an older computer that has no built-in hardware serial number, use 12345678 for the password.

Locating Computers for Installation

If you are installing software on a remote computer from Terminal, you must first establish an SSH session as the root user with the remote computer. To do so, you need the remote computer's IP address and serial number. You can find the serial number on a label on the computer.

Enter the serial number as the password when establishing the SSH session. If you are installing on an older computer that has no built-in hardware serial number, use 12345678 for the password.

To identify the IP address of each computer that's ready for installation on your subnet, use the `sa_srchr` tool.

Note: To locate remote computers, start up your computer from the installation CD.

To view computers on the local network:

```
$ /System/Library/ServerSetup/sa_srchr 224.0.0.1
```

The `sa_srchr` tool uses the broadcast address 224.0.0.1 to request a response (via `sa_rspndr`) from all computers ready for installation or setup. The response from a ready computer comes from `sa_rspndr` running on a computer started up from the Mac OS X Server installation CD.

The computer responds with output similar to the following:

```
localhost#unknown#<ip address>#<mac address>#Mac OS X Server
     10.5#RDY4PkgInstall#2.0#512
```

where `<ip_address>` is the working IP address and `<mac address>` is the unique MAC address of the network interface on a computer that is ready for installation.

Specifying the Target Computer Volume

To specify the target computer volume where you want to install the server software, use the `installer` tool.

To list volumes available for server software:

```
$ /usr/sbin/installer -volinfo -pkg /System/Installation/Packages/
     OSInstall.mpkg
```

To choose a network installation image you've created and mounted:

```
$ /usr/sbin/installer -volinfo -pkg /Volumes/ServerNetworkImage10.5/System/
     Installation/Packages/OSInstall.mpkg
```

The list displayed reflects your environment, but here's an example showing three available volumes:

```
/Volumes/Mount 01
/Volumes/Mount 1
/Volumes/Mount 02
```

Preparing the Target Volume for a Clean Installation

If the target volume has Mac OS X Server v10.3 or v10.4 installed, when you run `installer`, it upgrades the server to v10.5 and preserves user files.

If you're performing a clean installation, back up the user files you want to preserve, then use `diskutil` to erase the volume, format it, and enable journaling:

```
$ /usr/sbin/diskutil eraseVolume HFS+ "Mount 01" "/Volumes/Mount 01"
$ /usr/sbin/diskutil enableJournal "/Volumes/Mount 01"
```

You can also use case-sensitive Journaled HFS+ as a startup volume format, which is an available format for the Erase and Install option for local installations, but not for remotely controlled installations.

Important: Third-party applications might have problems with case-sensitive Journaled HFS+ format because of case mismatch. For example, when referencing the PlugIns folder, some third-party applications might use the term *PlugIns* while other parts might use the term *Plugins*. This works on HFS+ and Journaled HFS+, but not on case-sensitive Journaled HFS+.

You can also use `diskutil` to partition the volume and set up mirroring. For more information, see the `diskutil` man page or Chapter 7, "Working with Disks and Volumes," on page 85.

Important: Don't store data on the hard disk partition where the operating system is installed. If you must store additional software or data on the system partition, consider mirroring the drive. With this approach, you won't risk losing data if you reinstall or upgrade system software.

Restarting After Installation

When installation from the disc is complete, restart the computer by entering:

```
$ /sbin/reboot
```

or

```
$ /sbin/shutdown -r
```

Automating Server Setup

You can automate server setup by providing a configuration file that contains setup settings.

Normally when you install Mac OS X Server on a computer and restart, Server Assistant opens and prompts you for the basic information necessary to get the server running. This includes the user name and password of the administrator, the TCP/IP configuration information for the computer's network interfaces, and how the computer uses directory services.

Servers that have had Mac OS X Server v10.5 installed automatically detect the presence of the saved setup information and use it to complete initial server setup without user interaction.

You can define generic setup data that can be used to set up any computer.

For example, you can define generic setup data for a computer that's on order, or for 50 Xserve computers you want to be identically configured.

You can also save setup data that's specifically tailored for a computer.

Important: When you perform an upgrade, saved setup data is used and overwrites existing server settings. If you do not want saved server setup data to be used after an upgrade, rename the saved setup configuration file.

Creating a Configuration File

An easy way to prepare configuration files to automate the setup of a group of computers is to start with a file you save using Server Assistant.

You can save the file as the last step when you use Server Assistant to set up the first computer, or you can run Server Assistant later to create the file. You can then use that configuration file as a template for creating configuration files for other computers.

You can edit the file directly, or write scripts to create customized configuration files for computers that use similar hardware.

Note: If you intend to create a generic configuration file because you want to use the file to set up additional computers, don't specify network names (computer names or local hostnames), and make sure each network interface (port) is set to be configured using DHCP or using BootP.

To save a configuration file during server setup:

1. In the final pane of Server Assistant, after you review the settings, click Save As.
2. In the dialog that appears, choose Configuration File next to "Save As" and click OK:
   - If encryption is not required, don't select "Save in Encrypted Format."
   - To encrypt the file, select "Save in Encrypted Format" and enter and verify a passphrase. You must supply the passphrase before an encrypted setup file can be used by a target computer.
3. Navigate to the location where you want to save the configuration file, name the file using one of the following options, and click Save.

   Target computers search for names in the order listed:
   - *MAC-address-of-server*.plist (include leading zeros but omit colons)—for example, 0030654dbcef.plist
   - *IP-address-of-server*.plist—for example, 10.0.0.4.plist

- *partial-DNS-name-of-server*.plist—for example, myserver.plist
- *built-in-hardware-serial-number-of-server*.plist (first 8 characters only)—for example, ABCD1234.plist
- *fully-qualified-DNS-name-of-server*.plist—for example, myserver.example.com.plist
- *partial-IP-address-of-server*.plist—for example, 10.0.plist (matches 10.0.0.4 and 10.0.1.2)
- generic.plist—file that any server will recognize, used to set up servers that need the same setup values

Server Assistant uses the file to set up the computer with the matching address, name, or serial number. If Server Assistant cannot find a file named for a specific computer, it will use the file named generic.plist.

To create a configuration file after initial setup:

1. Open Server Assistant (located in /Applications/Server/).
2. In the Welcome pane, select "Save advanced setup information in a file or a directory record" and click Continue.
3. Enter settings in the remaining panes; then, after you review the settings in the final pane, click Save As.
4. In the dialog that appears, choose Configuration File next to Save As and click OK:
   - If encryption is not required, don't select "Save in Encrypted Format."
   - To encrypt the file, select "Save in Encrypted Format" and then enter and verify a passphrase. You must supply the passphrase before an encrypted setup file can be used by a target computer.
5. Navigate to the location where you want to save the configuration file, name the file using one of the following options, and click Save.

   Target computers search for names in the order listed here:
   - *MAC-address-of-server*.plist (include leading zeros but omit colons)—for example, 0030654dbcef.plist
   - *IP-address-of-server*.plist—for example, 10.0.0.4.plist
   - *partial-DNS-name-of-server*.plist—for example, myserver.plist
   - *built-in-hardware-serial-number-of-server*.plist (first 8 characters only)—for example, ABCD1234.plist
   - *fully-qualified-DNS-name-of-server*.plist—for example, myserver.example.com.plist
   - *partial-IP-address-of-server*.plist—for example, 10.0.plist (matches 10.0.0.4 and 10.0.1.2)
   - generic.plist—file that any computer will recognize, used to set up computers that need the same setup values.

   Server Assistant uses the file to set up the computer with the matching address, name, or serial number. If Server Assistant cannot find a file named for a computer, it uses the file named generic.plist.

Working with an Encrypted Configuration File

If the setup data in the configuration file is encrypted, make the passphrase available to target computers. You can supply the passphrase interactively using Server Assistant, or you can provide it in a text file.

To provide a passphrase in a file:

1. Create a text file and enter the passphrase for the saved setup file on the first line.
2. Save the file using one of the following names.

   Target computers search for names in the order listed here:

   - *MAC-address-of-server*.pass (include leading zeros but omit colons)—for example, 0030654dbcef.pass
   - *IP-address-of-server*.pass—for example, 10.0.0.4.pass
   - *partial-DNS-name-of-server*.pass—for example, myserver.pass
   - *built-in-hardware-serial-number-of-server*.pass (first 8 characters only)—for example, ABCD1234.pass
   - *fully-qualified-DNS-name-of-server*.pass—for example, myserver.example.com.pass
   - *partial-IP-address-of-server*.pass—for example, 10.0.pass (matches 10.0.0.4 and 10.0.1.2)
   - generic.pass—file that any computer will recognize
3. Put the passphrase file on a volume mounted locally on the target computer in /Volumes/*/Auto Server Setup/<pass-phrase-file>, where * is any device mounted under /Volumes.

To provide a passphrase interactively:

1. Use Server Assistant on an administrator computer that can connect to the target computer.
2. In the Welcome or Destination pane, choose File > Supply Passphrase.
3. In the dialog box, enter the target computer's IP address, password, and passphrase, then click Send.

Customizing a Configuration File

After you create a configuration file, you can modify it using a text editor, or you can write a script to generate custom configuration files for a group of computers.

The file uses XML format to encode the setup information. The name of an XML key indicates the setup parameter it contains.

The following sample configuration file shows the basic structure and contents of a configuration file for a computer with this configuration:

- An administrator user named "Administrator" (short name "admin") with a user ID of 501 and the password "secret"
- A computer name and host name of "server1.example.com"
- A single Ethernet network interface set to get its address from DHCP
- No server services set to start automatically

Note: Angle brackets used in XML format do not have the same usage as angle brackets used in Mac OS X Server commands.

Sample Configuration File

```
<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE plist PUBLIC "-//Apple//DTD PLIST 1.0//EN" "http://www.apple.com/
     DTDs/PropertyList-1.0.dtd">
<plist version="1.0">
<dict>
  <key>AdminUser</key>
  <dict>
     <key>exists</key>
     <false/>
     <key>name</key>
     <string>admin</string>
     <key>password</key>
     <string>secret</string>
     <key>realname</key>
     <string>admin</string>
     <key>uid</key>
     <string>501</string>
  </dict>
  <key>Bonjour</key>
  <dict>
     <key>BonjourEnabled</key>
     <true/>
     <key>BonjourName</key>
     <string>leopardserver</string>
  </dict>
  <key>ComputerName</key>
  <string>leopardserver</string>
  <key>DS</key>
  <dict>
     <key>DSType</key>
     <string>Standalone</string>
  </dict>
  <key>DefaultGroupName</key>
  <dict>
     <key>longname</key>
     <string>Work Group</string>
     <key>shortname</key>
```

```
    <string>workgroup</string>
</dict>
<key>HostName</key>
<string>leopardserver.example.com</string>
<key>InstallLanguage</key>
<string>English</string>
<key>Keyboard</key>
<dict>
    <key>DefaultFormat</key>
    <string>0</string>
    <key>DefaultScript</key>
    <string>0</string>
    <key>ResName</key>
    <string>U.S.</string>
    <key>ScriptID</key>
    <integer>0</integer>
    <key>kbResID</key>
    <integer>0</integer>
</dict>
<key>NetworkInterfaces</key>
<array>
    <dict>
        <key>ActiveAT</key>
        <false/>
        <key>ActiveTCPIP</key>
        <true/>
        <key>DNSServers</key>
        <array>
            <string>10.0.0.1</string>
        </array>
        <key>DeviceName</key>
        <string>en0</string>
        <key>EthernetAddress</key>
        <string>00:00:00:00:00:00</string>
        <key>IPv6</key>
        <dict>
            <key>IPv6Type</key>
            <string>3</string>
        </dict>
        <key>PortName</key>
        <string>Built-in Ethernet</string>
        <key>Settings</key>
        <dict>
            <key>IPAddress</key>
            <string>10.0.0.2</string>
            <key>Router</key>
            <string>10.0.0.1</string>
            <key>SubnetMask</key>
            <string>255.255.255.0</string>
            <key>Type</key>
            <string>Manual Configuration</string>
```

```
            </dict>
        </dict>
    </array>
    <key>PrimaryLanguage</key>
    <string>English</string>
    <key>SerialNumber</key>
    <string>XSVR-???-???-?-???-???-???-???-???-???-?|Registered_to|
        Organization</string>
    <key>ServiceNTP</key>
    <dict>
        <key>HostNTP</key>
        <false/>
        <key>HostNTPServer</key>
        <string>time.apple.com</string>
        <key>UseNTP</key>
        <true/>
    </dict>
    <key>TimeZone</key>
    <string>US/Pacific</string>
    <key>VersionNumber</key>
    <integer>3</integer>
</dict>
</plist>
```

Note: The contents of the configuration file depend on the hardware configuration of the computer it's created on, so you should customize a configuration file created on a computer similar to those you plan to set up.

Storing a Configuration File in an Accessible Location

Server Assistant looks for configuration files in the following location:

`/Volumes/`*`vol`*`/Auto Server Setup/`

where *vol* is a device volume mounted in /Volumes.

Devices you can use to provide configuration files include:

- A partition on a computer's hard disk
- An iPod
- An optical (CD or DVD) drive
- A USB or FireWire drive
- Any other portable storage device that mounts in the /Volumes folder

Configuring the Server Remotely from the Command Line

It's possible to configure the server remotely from the command line. Performing this task requires the following tools:

- `dscl`—Use to create, read, and manage directory service data. If invoked without commands, `dscl` runs interactively, reading commands from standard input.

  For more information about this command, see Chapter 8, "Managing User and Group Accounts."

- `systemsetup`—Use to set a number of system-wide preferences. If you used Server Assistant, you would need to select the proper keyboard and time zone. The `systemsetup` tool can configure these preferences, and more.

  For more information about this command, see Chapter 5, "Setting General System Preferences."

- `networksetup`—Use to configure anything that you can configure in the Network pane of System Preferences.

  For more information about this command, see Chapter 6, "Setting Network Preferences."

For more information about these tools, see their man pages. The man pages for `systemsetup` and `networksetup` are available only on Mac OS X Server.

Changing Server Settings

After initial setup, you can use a variety of commands to view or change Mac OS X Server configuration settings and services.

Using the serversetup Tool

The `serversetup` tool is located in /System/Library/ServerSetup/. To run it, you can enter the full path:

```
$ /System/Library/ServerSetup/serversetup -getHostname
```

To use the tool to perform several commands, change your working folder and enter a shorter command:

```
$ cd /System/Library/ServerSetup
$ ./serversetup -getHostname
$ ./serversetup -getComputername
```

Or, add the folder to your search path for this session and enter an even shorter command:

```
$ PATH="$PATH:/System/Library/ServerSetup"
$ serversetup -getHostname
```

To permanently add the folder to your search path, add the path to the file /etc/profile.

Using the serveradmin Tool

You use the `serveradmin` tool to administer service-related tasks. Some services must be restarted after you change specific settings.

If you make a change using a service's `writeSettings` tool that requires you to restart the service, the output from the command includes the setting `<svc>:needsRecycleOrRestart` with a value of `yes`.

Important: The `needsRecycleOrRestart` setting appears only if you use the `serveradmin` *`svc`*`:command = writeSettings` command to change settings. You won't see it if you use the `serveradmin settings` command.

Other chapters in this guide provide information about using `serveradmin` to administer specific services.

Notes on Communication Security and the servermgrd Tool

- When you run the `serveradmin` tool, you're communicating with a local or remote `servermgrd` process.
- By default, port 687, which allows cleartext connections with `servermgrd`, is disabled. You can enable this port by changing the `listenForRegularConnections` parameter or key to `yes` in the /Library/Preferences/com.apple.servermgrd.plist file.
- For encryption and client authentication, `servermgrd` uses SSL, but not for user authentication. User authentication uses Open Directory services.
- `servermgrd` uses a self-signed (test) SSL certificate installed by default, located in /etc/servermgrd/ssl.crt/. You can replace this with an actual certificate.

  To create and manage certificates, use Certificate Manager in Server Admin. For more information, see *Mail Service Administration*.
- The default certificate format for SSLeay/OpenSSL is PEM. PEM format can contain private keys (RSA and DSA), public keys (RSA and DSA), and (x509) certificates. It stores data in Base64-encoded DER format with ASCII header and footer lines, which makes it suitable for text-made transfers between computers.

  For some tools, you need the certificate in plain DER format. You can convert a PEM file (cert.pem) into the corresponding DER file (cert.der) with the following command:

```
$ openssl x509 -in cert.pem -out cert.der -outform DER
```

- `servermgrd` checks the validity of the SSL certificate if the "Require valid digital signature" option is selected in Server Admin preferences. This option uses an SSL certificate installed on a remote server to ensure that the remote server is a valid server. If this option is enabled, the certificate must be valid and not expired, or Server Admin will refuse to connect.

  Before enabling this option, use the instructions in *Mail Service Administration* for generating a Certificate Signing Request (CSR), obtaining an SSL certificate from an issuing authority, and installing the certificate on each remote server.

  Instead of placing files in /etc/httpd/, place them in /etc/servermgrd/.

  You can also generate a self-signed certificate and install it on the remote server.

- You can change `servermgrd` SSL encryption options by editing the com.apple.servermgrd.plist configuration file located in /Library/Preferences/. Your SSL certificate (ssl.crt/server.crt) and keyfile (ssl.key/server.key) are located in /private/etc/servermgrd/.

General and Network Preferences

For information about changing general system preferences and network settings, see the following:

- Chapter 5, "Setting General System Preferences," on page 59
- Chapter 6, "Setting Network Preferences," on page 65

Viewing, Validating, and Setting the Software Serial Number

To view or set the server's software serial number or to validate a server software serial number, use the `serversetup` tool, located in /System/Library/ServerSetup/.

To view the server's software serial number:

```
$ sudo serversetup -getServerSerialNumber
```

To set the server software serial number:

```
$ sudo serversetup -setServerSerialNumber serialnumber watermarkinformation
```

where *`serialnumber`* is a valid Mac OS X Server software serial number, as found on the software packaging that comes with the software.

To validate a server software serial number:

```
$ sudo serversetup -verifyServerSerialNumber serialnumber
     watermarkinformation
```

This displays `0` if the serial number is valid, or `1` if the serial number is invalid.

Serial numbers generated for the server can be generated with watermarks so they can be tracked to a specific company, group, or individual. If a serial number has watermarking strings associated with it, it is necessary to supply the watermark information when setting or validating the serial number.

To verify that a serial number is site-licensed:

```
$ sudo serversetup -isSiteLicensedSerialNumber
```

Updating Server Software

You can use the `softwareupdate` tool to check for and install software updates over the Internet from Apple's website.

To check for available updates:

```
$ sudo softwareupdate --list
```

The output is similar to the following:

```
Software Update Tool
Copyright 2002-2005 Apple

Software Update found the following new or updated software:

  - WebObjects5.3.1ServerUpdate-5.3.1
     WebObjects5.3.1 Server Update (5.3.1), 29110K [recommended] [restart]
  * J2SE50Release3-3.0
     **PRERELEASE** J2SE 5.0 Release 3 (8M318) (3.0), 44020K [recommended]
  - AirPort-1.0
     AirPort Update 2005-001 (1.0), 1440K [restart]
```

To install an update:

```
$ sudo softwareupdate --install update-version
```

Parameter	Description
update-version	The hyphenated product version string that appears in the list of updates when you use the `--list` option

Some updates require that you agree to a license agreement. To work around this in an automated command-line environment, execute the following command before running `softwareupdate`:

```
$ command_line_install=1 export command_line_install
```

This creates an environment variable named `command_line_install` that automates update responses.

For more information, see the `softwareupdate` man page.

Moving a Server

Before setting a server up for the first time, try to place it in its final network location (subnet). If you're concerned about unauthorized or premature access, set up a firewall to protect the server while you're finishing its configuration.

If you must move a server after setup, you must change settings that are sensitive to network location before the server can be used. For example, the server's IP address and host name—stored in both folders and configuration files that reside on the server—must be updated.

When you move a server, consider these guidelines:

- Minimize the time the server is in its temporary location so the information you must change is limited.
- Don't configure services that depend on network settings until the server is in its final location. Such services include Open Directory replication, Apache settings (such as virtual hosts), DHCP, and other network infrastructure settings that other computers depend on.
- Wait to import final user accounts. Limit accounts to test accounts so you minimize the user-specific network information (such as home folder location) that must be changed after the move.
- After you move the server, use the `changeip` tool to change IP addresses, host names, and other data stored in Open Directory and LDAP folders on the server. See "Changing a Server's IP Address" on page 68. After using the tool, you may need to adjust network configurations, such as the local DNS database.
- Reconfigure the search policy of computers (such as user computers and DHCP servers) that have been configured to use the server in its original location. For information about configuring a computer's search policy, see *Open Directory Administration*.

Restarting or Shutting Down a Computer

4

Use this chapter to learn the commands to shut down or restart a local or remote computer.

This chapter covers the commands that shut down or restart a local or remote computer. Computers must be shut down or restarted, whether locally or remotely, when installing tools or making computer repairs.

Restarting a Computer

To restart a computer at a specific time, use the `reboot` or `shutdown -r` command. For more information, see the relevant man pages.

To restart the local computer:

```
$ shutdown -r now
```

To restart a remote computer immediately:

```
$ ssh -l root computer shutdown -r now
```

To restart a remote computer at a specific time:

```
$ ssh -l root computer shutdown -r hhmm
```

Parameter	Description
computer	The IP address or DNS name of the computer
hhmm	The hour and minute when the computer restarts

Automatic Restart

You can also use the `systemsetup` tool to set up the computer to start up after a power failure or system freeze. See "Viewing or Changing Automatic Restart Settings" on page 61.

Changing a Remote Computer's Startup Disk

You can change a remote computer's startup disk using SSH.

To change the startup disk:

Log in to the remote computer using SSH and enter:

```
$ bless -folder "/Volumes/disk/System/Library/CoreServices" -setBoot
```

Parameter	Description
disk	The name of the disk that contains the startup volume

For information about using SSH to log in to a remote computer, see "Sending Commands to a Remote Computer" on page 28.

Shutting Down a Computer

To shut down a computer at a specific time, use the `shutdown` tool. For more information, see the `shutdown` man page.

To shut down a remote computer immediately:

```
$ ssh -l root computer shutdown -h now
```

To shut down the local computer in 30 minutes:

```
$ shutdown -h +30
```

Parameter	Description
computer	The IP address or DNS name of the computer

Shutting Down While Leaving the Computer on and Powered

To support UPS restart after power failure, the `shutdown` tool provides the `-u` option. This option halts system shutdown before the `shutdown` tool instructs the power manager to turn off the power supply.

The `-u` option keeps the system halted and waits for 5 minutes before removing power so an external UPS can forcibly remove power.

Using the `-u` option simulates a dirty shutdown, which allows a later automatic power on. The operating system uses the `-u` option with supported UPS devices in emergency shutdowns.

Manipulating Open Firmware NVRAM Variables

To manipulate Open Firmware NVRAM variables, use the `nvram` tool. If you modify a value with `nvram`, the value is saved only if the computer cleanly restarts or shuts down. For more information, see the `nvram` man page.

To view NVRAM variables:

```
$ nvram -p
```

Monitoring and Restarting Critical Services

In earlier versions of Mac OS X, a daemon called `watchdog` monitored critical services and restarted them if they failed or quit unexpectedly after a computer restarted. The `watchdog` daemon relied on the configuration file watchdog.conf, located in /etc/.

In Mac OS X Server v10.4, `watchdog` was replaced by `launchd`. The `launchd` daemon manages other daemons, both for the computer and for users. You can configure the `launchd` daemon to launch other daemons on demand, based on criteria specified in their respective XML property lists.

During system startup, `launchd` is the first process invoked by the kernel to run and set up the computer. In Mac OS X Server, it is preferable to have your daemon started by `launchd`.

Note: Some system administrators must modify the boot process to insert a script or implement a change in the default system configuration. System administrators are encouraged to work with `launchd` to implement changes, and avoid modifying `rc` or creating a SystemStarter Startup Item. The `rc` command script might be phased out in the future.

The configuration files are in the following folders:

Folder	Usage
/System/Library/LaunchAgents/	Configuration for the system
/System/Library/LaunchDaemons/	Configuration for the daemons
~/Library/LaunchAgents/	Configuration per user

Setting General System Preferences

5

Use this chapter to learn the commands to set system preferences.

You can use Mac OS X Server to manage the work environment of Mac OS X users by defining preferences. Preferences are settings that customize and control a user's computer experience.

Viewing or Changing the Computer Name

You can use the `systemsetup` tool to view or change a computer name (the name used to browse for AFP share points on the server), which would otherwise be set using the Sharing pane of System Preferences.

To display the computer name:

```
$ sudo systemsetup -getcomputername
```

or

```
$ sudo networksetup -getcomputername
```

To change the computer name:

```
$ sudo systemsetup -setcomputername computername
```

or

```
$ sudo networksetup -setcomputername computername
```

Viewing or Changing the Date and Time

You can use the `systemsetup` or `serversetup` tool to view or change a computer's system date, time, and time zone. In addition, you can use the `systemsetup` tool to view or change whether a server uses a network time server.

You can also change these settings using the Date & Time pane of System Preferences.

Viewing or Changing the System Date

To view the system date

```
$ sudo systemsetup -getdate
```

or

```
$ serversetup -getDate
```

To set the system date:

```
$ sudo systemsetup -setdate mm:dd:yy
```

or

```
$ sudo serversetup -setDate mm/dd/yy
```

Viewing or Changing the System Time

To view the system time:

```
$ sudo systemsetup -gettime
```

or

```
$ serversetup -getTime
```

To change the system time:

```
$ sudo systemsetup -settime hh:mm:ss
```

or

```
$ sudo serversetup -setTime hh:mm:ss
```

Viewing or Changing the System Time Zone

To view the time zone:

```
$ sudo systemsetup -gettimezone
```

or

```
$ serversetup -getTimeZone
```

To view available time zones:

```
$ sudo systemsetup -listtimezones
```

To change the system time zone:

```
$ sudo systemsetup -settimezone timezone
```

or

```
$ sudo serversetup -setTimeZone timezone
```

Viewing or Changing Network Time Server Usage

To see if a network time server is being used:

```
$ sudo systemsetup -getusingnetworktime
```

To enable or disable a network time server:

```
$ sudo systemsetup -setusingnetworktime (on|off)
```

To view the network time server:

```
$ sudo systemsetup -getnetworktimeserver
```

To specify a network time server:

```
$ sudo systemsetup -setnetworktimeserver timeserver
```

Viewing or Changing Energy Saver Settings

To view or change a server's energy saver settings, use the `systemsetup` tool (or the Energy Saver pane of System Preferences).

Viewing or Changing Sleep Settings

To view the idle time before sleep:

```
$ sudo systemsetup -getsleep
```

To set the idle time before sleep:

```
$ sudo systemsetup -setsleep minutes
```

To see if the system is set to wake for modem activity:

```
$ sudo systemsetup -getwakeonmodem
```

To set the system to wake for modem activity:

```
$ sudo systemsetup -setwakeonmodem (on|off)
```

To see if the system is set to wake for network access:

```
$ sudo systemsetup -getwakeonnetworkaccess
```

To set the system to wake for network access:

```
$ sudo systemsetup -setwakeonnetworkaccess (on|off)
```

Viewing or Changing Automatic Restart Settings

To see if the system is set to restart after a power failure:

```
$ sudo systemsetup -getrestartpowerfailure
```

To set the system to restart after a power failure:

```
$ sudo systemsetup -setrestartpowerfailure (on|off)
```

To see how long the system waits to restart after a power failure:

```
$ sudo systemsetup -getwaitforstartupafterpowerfailure
```

To set how long the system waits to restart after a power failure:

```
$ sudo systemsetup -setwaitforstartupafterpowerfailure seconds
```

Parameter	Description
seconds	Must be a multiple of 30 seconds

To see if the system is set to restart after a system freeze:

```
$ sudo systemsetup -getrestartfreeze
```

To set the system to restart after a system freeze:

```
$ sudo systemsetup -setrestartfreeze (on|off)
```

Changing Power Management Settings

You can use the `pmset` tool to change power management settings, including:

- Display dim timer
- System sleep timer
- Wake on network activity
- Wake on modem activity
- Restart after power failure
- Dynamic processor speed change
- Reduce processor speed
- Sleep computer on power button press

You configure settings for power modes using `pmset`. There are four `pmset` flags:

Flag	Description
`-a`	Applies the power settings to all.
`-b`	Applies the power settings to battery operation.
`-c`	Applies the power settings to the charger (wall power).
`-u`	Applies the power settings to the Uninterruptible Power Supply (UPS).

To set the disk sleep timer for all modes of operation:

```
$ sudo pmset -u disksleep minutes
```

Parameter	Description
minutes	Must be a multiple of 30 seconds

To display the settings in use:

```
$ sudo pmset -g
```

For more information, see the `pmset` man page.

Viewing or Changing Startup Disk Settings

To view or change a computer's startup disk, use the `systemsetup` tool (or the Startup Disk pane of System Preferences).

To view the startup disk:

```
$ sudo systemsetup -getstartupdisk
```

To view available startup disks:

```
$ sudo systemsetup -liststartupdisks
```

To change the startup disk:

```
$ sudo systemsetup -setstartupdisk path
```

Viewing or Changing Sharing Settings

To view or change Sharing settings, use the `systemsetup` tool (or the Sharing pane of System Preferences).

Viewing or Changing Remote Login Settings

You can use SSH to log in to a remote server if remote login is enabled.

To see if the system is set to allow remote login:

```
$ sudo systemsetup -getremotelogin
```

To enable or disable remote login:

```
$ sudo systemsetup -setremotelogin (on|off)
```

or

```
$ serversetup -enableSSH
```

By default, Telnet access is disabled because it isn't as secure as SSH. However, you can enable Telnet access. See "Using Telnet" on page 36.

Viewing or Changing Apple Event Response

To see if the system is set to respond to remote events:

```
$ sudo systemsetup -getremoteappleevents
```

To set the server to respond to remote events:

```
$ sudo systemsetup -setremoteappleevents (on|off)
```

Creating the Groups Share Point

To create the Groups share point:

```
$ serversetup -createGroupsSharePoint
```

Viewing or Changing Language and Keyboard Settings

To view or change language settings, use the `serversetup` tool (or the International pane of System Preferences).

To view the primary language:

```
$ serversetup -getPrimaryLanguage
```

To view the installed language:

```
$ serversetup -getInstallLanguage
```

To set the installation language:

```
$ sudo serversetup -setInstallLanguage language
```

To select a keyboard:

```
$ sudo serversetup -setKeyboardSelection ScripID(0) kbResID(0) ResName(U.S.)
```

To select a keyboard:

```
$ sudo serversetup --setNewPrimaryLanguage adminshortname primaryLanguage
     installLanguage
```

To view the script setting:

```
$ serversetup -getPrimaryScriptCode
```

Viewing and Changing Login Settings

You can enable or disable the Restart and Shutdown buttons that appear in the login dialog.

To disable or enable the Restart and Shutdown buttons in the login dialog:

```
$ sudo serversetup -setDisableRestartShutdown (0|1)
```

`0` disables the buttons and `1` enables the buttons.

To view the current setting:

```
$ serversetup -getDisableRestartShutdown
```

Setting Network Preferences

6

Use this chapter to learn the commands to change network settings on a server.

Mac OS X Server provides command-line control to manage servers in a mixed-platform environment and to configure, deploy, and manage powerful network services. These tools make it easy to configure and maintain core network services, while providing the advanced features and functionality required by experienced IT professionals.

Configuring Network Interfaces

To configure network interfaces, Mac OS X Server provides `networksetup` and `serversetup`. Although `ifconfig` (the standard UNIX tool for configuring networks) is available, it's better to use `networksetup` and `serversetup` because if you use `ifconfig`, your computer will be out of sync and will revert to using the contents of preferences.plist after a restart.

You can still use `ifconfig` to view the network interface configuration. This is particularly beneficial when your computer is using an autonegotiated Ethernet connection.

For more information, see the `networksetup` and `serversetup` man pages.

Managing Network Interface Information

This section describes commands you address to a specific hardware device (for example, `en0`) or port (for example, `Built-in Ethernet`).

If you prefer to work with network port configurations following the approach used in the Network preferences pane of System Preferences, see the commands in "Managing Network Port Configurations" on page 67.

Viewing Port Names and Hardware Addresses

To list all port names with their Ethernet (MAC) addresses:

```
$ sudo networksetup -listallhardwareports
```

To list hardware port information by port configuration:

```
$ sudo networksetup -listallnetworkservices
```

An asterisk (*) in the results marks an inactive configuration.

To view the default (en0) Ethernet (MAC) address of the server:

```
$ serversetup -getMacAddress
```

To view the Ethernet (MAC) address of a port:

```
$ sudo networksetup -getmacaddress (devicename|"portname")
```

To scan for new hardware ports:

```
$ sudo networksetup -detectnewhardware
```

This command checks the computer for new network hardware and creates a default configuration for each new port.

Viewing or Changing MTU Values

All data transmitted over a network travels in data packets. The size of a packet is called a maximum transmission unit (MTU), which if too large or too small will affect performance. To change the MTU size for a port, use the `networksetup` tool.

To view the MTU value for a hardware port:

```
$ sudo networksetup -getMTU (devicename|"portname")
```

To list valid MTU values for a hardware port:

```
$ sudo networksetup -listvalidMTUrange (devicename|"portname")
```

To change the MTU value for a hardware port:

```
$ sudo networksetup -setMTU (devicename|"portname")
```

Viewing or Changing Media Settings

To view media settings for a port:

```
$ sudo networksetup -getMedia (devicename|"portname")
```

To list valid media settings for a port:

```
$ sudo networksetup -listValidMedia (devicename|"portname")
```

To change media settings for a port:

```
$ sudo networksetup -setMedia (devicename|"portname") subtype [option1]
     [option2] [...]
```

Managing Network Port Configurations

Network port configurations are sets of network preferences that can be assigned to a network interface and then enabled or disabled. The Network pane of System Preferences stores and displays network settings as port configurations.

Creating or Deleting Port Configurations

To list a port configuration:

```
$ sudo networksetup -listallnetworkservices
```

To create a port configuration:

```
$ sudo networksetup -createnetworkservice configuration hardwareport
```

To duplicate a port configuration:

```
$ sudo networksetup -duplicatenetworkservice configuration newconfig
```

To rename a port configuration:

```
$ sudo networksetup -renamenetworkservice configuration newname
```

To delete a port configuration:

```
$ sudo networksetup -removenetworkservice configuration
```

Activating Port Configurations

To see if a port configuration is on:

```
$ sudo networksetup -getnetworkserviceenabled configuration
```

To enable or disable a port configuration:

```
$ sudo networksetup -setnetworkserviceenabled configuration (on|off)
```

Changing Configuration Precedence

To list the configuration order:

```
$ sudo networksetup -listnetworkserviceorder
```

The configurations are listed in the order that they're tried when a network connection is established. An asterisk (*) marks an inactive configuration.

To change the order of port configurations:

```
$ sudo networksetup -ordernetworkservices config1 config2 [config3] [...]
```

Managing TCP/IP Settings

TCP/IP is a set of layered protocols that allow communication between computers on a high-speed network. You can use the following commands to change the TCP/IP settings of a server.

Changing a Server's IP Address

The server's setup must reflect the network settings of the server's primary interface. The primary interface is the topmost active connection in the Network pane of System Preferences.

When using your server as a gateway to the Internet, the server uses the primary interface to connect to the Internet. Therefore, during server setup, you configure the primary interface to use the server's public IP address and DNS information.

The server setup program uses this information to configure other server components (such as Open Directory, Kerberos, and Password Server). As such, the IP address and the DNS settings of the primary interface and these other components must always match.

If at some point you change the IP address or DNS name of the primary interface, the system will run the `changeip` command within a minute or two. If not, you must register the IP address change with the server setup program.

The `changeip` command makes all necessary changes at once, updating the settings of all components configured during server setup, including Open Directory, Kerberos, and Password Server.

The `changeip` command is a python script that runs tools from the /usr/libexec/changeip folder. Three tools are available: `changeip_ds`, `changeip_jabber`, and `changeip_mail`.

The `changeip_ds` tool updates the following local configuration files:

- /Library/Preferences/DirectoryService/DSLDAPv3PlugInConfig.plist
- /etc/openldap/slapd_macosxserver.conf
- /etc/hostconfig (if there is a static hostname)
- /etc/smb.conf

The `changeip_ds` tool also updates the following records in the local directory domain, as well as a parent directory domain, if specified:

- AuthAuthority and HomeDirectory in user records
- Addresses and hostname in machine records
- Addresses and hostname in computer records
- Mount paths and addresses in mount records
- Addresses in LDAP and Password Server config records

The `changeip_jabber` tool updates the jabber configuration using `serveradmin`.

The `changeip_mail` tool updates the mailman, postfix, and imap configurations using `serveradmin`.

To change a server's IP address:

1 Run the `changeip` tool:

```
$ sudo changeip [(directory|-)] old-ip new-ip [old-hostname new-hostname]
```

Parameter	Description
directory	If the server is an Open Directory master or replica, or is connected to a folder system, include the path to the folder domain (folder directory domain). For a standalone server, enter "-" instead.
old-ip	The current IP address.
new-ip	The new IP address.
old-hostname	(Optional) The current *fully qualified* DNS host name of the server.
new-hostname	(Optional) The new *fully qualified* DNS host name of the server.

For more information, see the `changeip` man page.

Important: If you change your IP address and computer name using `changeip` while you are connected to a directory server, you must disconnect and reconnect to the directory server to update the directory with the new computer name and IP address. If you do not disconnect and reconnect to the directory server, the directory is not updated and continues to use the old computer name and IP address.

2 To change the server's IP address, use the `networksetup` or `serversetup` tool (or the Network pane of System Preferences).

3 Restart the server.

To change the IP address of a computer hosting an LDAP master:

```
$ sudo changeip /LDAPv3/127.0.0.1 192.0.0.12 192.0.1.10 oldhost.example.com
     newhost.example.com
```

It might be necessary to change the configuration of computers pointing to this master.

To change the IP address of a standalone server:

```
$ sudo changeip - 192.0.0.12 192.0.1.10 oldhost.example.com
     newhost.example.com
```

Viewing or Changing the IP Address, Subnet Mask, or Router Address

To change a computer's TCP/IP settings, use the `serversetup` and `networksetup` tools.

Important: Changing a computer's IP address isn't as simple as changing the TCP/IP settings. You must first run the `changeip` tool to make sure necessary changes are made throughout the system. See "Changing a Server's IP Address" on page 68.

To list TCP/IP settings for a configuration:

```
$ sudo networksetup -getinfo "configuration"
```

For example, for built-in Ethernet, the computer responds with the following output:

```
$ networksetup -getinfo "Built-In Ethernet"
Manual Configuration
IP Address: 192.168.10.12
Subnet mask: 255.255.0.0
Router: 192.18.10.1
Ethernet Address: 1a:2b:3c:4d:5e:6f
```

To view TCP/IP settings for a port or device:

```
$ serversetup -getInfo (devicename|"portname")
```

To change TCP/IP settings for a port or device:

```
$ sudo serversetup -setInfo (devicename|"portname") ipaddress subnetmask
     router
```

To set manual TCP/IP information for a configuration:

```
$ sudo networksetup -setmanual "configuration" ipaddress subnetmask router
```

To validate an IP address:

```
$ serversetup -isValidIPAddress ipaddress
```

Displays `0` if the address is valid, `1` if it isn't.

To validate a subnet mask:

```
$ serversetup -isValidSubnetMask subnetmask
```

To set a configuration to use DHCP:

```
$ sudo networksetup -setdhcp "configuration" [clientID]
```

To set a configuration to use DHCP with a manual IP address:

```
$ sudo networksetup -setmanualwithdhcprouter "configuration" ipaddress
```

To set a configuration to use BootP:

```
$ sudo networksetup -setbootp "configuration"
```

Viewing or Changing DNS Servers

To view and modify DNS settings, use the `serversetup` tool.

To view DNS servers for port en0:

```
$ serversetup -getDefaultDNSServer (devicename|"portname")
```

To change DNS servers for port en0:

```
$ sudo serversetup -setDefaultDNSServer (devicename|"portname") server1
     [server2] [...]
```

To view DNS servers for a port or device:

```
$ serversetup -getDNSServer (devicename|"portname")
```

To change DNS servers for a port or device:

```
$ sudo serversetup -setDNSServer (devicename|"portname") server1 [server2]
     [...]
```

To list DNS servers for a configuration:

```
$ sudo networksetup -getdnsservers "configuration"
```

To view DNS search domains for port en0:

```
$ serversetup -getDefaultDNSDomain (devicename|"portname")
```

To change DNS search domains for port en0:

```
$ sudo serversetup -setDefaultDNSDomain (devicename|"portname") domain1
     [domain2] [...]
```

To view DNS search domains for a port or device:

```
$ serversetup -getDNSDomain (devicename|"portname")
```

To change DNS search domains for a port or device:

```
$ sudo serversetup -setDNSDomain (devicename|"portname") domain1 [domain2]
     [...]
```

To list DNS search domains for a configuration:

```
$ sudo networksetup -getsearchdomains "configuration"
```

To set DNS servers for a configuration:

```
$ sudo networksetup -setdnsservers "configuration" dns1 [dns2] [...]
```

To set search domains for a configuration:

```
$ sudo networksetup -setsearchdomains "configuration" domain1 [domain2]
     [...]
```

To validate a DNS server:

```
$ serversetup -verifyDNSServer server1 [server2] [...]
```

To validate DNS search domains:

```
$ serversetup -verifyDNSDomain domain1 [domain2] [...]
```

Enabling TCP/IP

To enable or disable TCP/IP on a computer, use the `serversetup` tool.

To enable TCP/IP on a port:

```
$ serversetup -EnableTCPIP [(devicename|"portname")]
```

If you don't provide an interface, en0 is assumed.

To disable TCP/IP on a port:

```
$ serversetup -DisableTCPIP [(devicename|"portname")]
```

If you don't provide an interface, en0 is assumed.

Statically Configuring Ethernet Interfaces

You can configure your server to define an IPv4 address on an interface that does not have a live link.

To define an IPv4 address on an interface that does not have a live link:

1. Edit the network preferences file located at /Library/Preferences/SystemConfiguration/preferences.plist.

   In the preferences.plist, navigate to the block that defines the relevant interface (say, en1), look for the IPv4 configuration block, and add the `IgnoreLinkStatus` key.

   Here is an example:

```
<key>IPv4</key>
<dict>
  <key>Addresses</key>
  <array>
     <string>10.12.0.7</string>
  </array>
  <key>ConfigMethod</key>
  <string>Manual</string>
  <key>IgnoreLinkStatus</key>
  <true/>
  <key>Router</key>
  <string>10.12.0.1</string>
  <key>SubnetMasks</key>
  <array>
     <string>255.255.0.0</string>
  </array>
</dict>
```

2. Save the /Library/Preferences/SystemConfiguration/preferences.plist file.
3. To activate the modified preference, restart your system or use `scselect` to reselect the current service (typically named `Automatic`, for example, `scselect Automatic`).

Creating, Deleting, and Viewing VLANs

A virtual local area network (VLAN) connects devices that may be on separate physical LANs to perform and communicate as if they were on the same physical LAN. Use the `networksetup` tool to configure and modify a VLAN.

To create a VLAN:

```
$ networksetup -createVLAN name parentdevice tag
```

To delete a VLAN:

```
$ networksetup -deleteVLAN name parentdevice tag
```

To list available VLANs:

```
$ networksetup -listVLANs
```

To list devices that support VLANs:

```
$ networksetup -listdevicesthatsupportVLAN
```

IEEE 802.3ad Ethernet Link Aggregation

IEEE 802.3ad provides increased bandwidth and automatic failover for the server environment.

Apple introduced the implementation of the IEEE 802.3ad Ethernet Link Aggregation standard as part of the `ifconfig` tool. IEEE 802.3ad is a standard for bonding or aggregating multiple Ethernet ports into one virtual interface.

The aggregated ports appear as a single IP address internally to your computer and tools and externally to other clients on the Internet. Any tool or server that relies on your IP address will continue to work seamlessly without modifications.

The advantage of aggregation is that the virtual interface provides increased bandwidth by merging the bandwidth of individual ports. The TCP connection load is then balanced across the ports.

In addition to load balancing, IEEE 802.3ad provides automatic failover in the event a port or cable fails. Traffic that was routed over the failed port is rerouted to a remaining port. This failover is transparent to the software using the connection.

Configuring a Network Interface

You can configure a network interface for TCP/IP using `ifconfig`. This tool is used to bring the interface up or down and set the interface IP address and subnet mask.

To add an Ethernet interface to a bond virtual device (pseudo device):

```
$ ifconfig bond_interface_name bondev physical_interface
```

The *`bond_interface_name`* parameter is the name of the pseudo device and the *`physical_interface`* parameter is the Ethernet interface you want to associate with the pseudo device (for example, en0).

If this is the first physical interface to be associated with the bond interface, the bond interface inherits the Ethernet address from the physical interface.

Physical interfaces that are added to the bond interface have their Ethernet address reprogrammed so members of the bond have the same Ethernet address.

If the physical interface is subsequently removed from the bond, a new Ethernet address is chosen from the remaining interfaces, and interfaces are reprogrammed with the new Ethernet address. If no remaining interfaces exist, the bond interface's Ethernet address is cleared.

To remove an Ethernet interface from a bond virtual device (pseudo device):

```
$ ifconfig bond_interface_name -bondev physical_interface
```

The link status of the bond interface depends on the state of link aggregation. If no active partner is detected, the link status remains inactive. To monitor the IEEE 802.3ad Link Aggregation state, use the `-b` option.

For more information, see the `ifconfig` man page.

Configuring Ethernet Link Aggregation

You can also use `networksetup` to configure Ethernet Link Aggregation. The following commands are supported.

To see if a device can be added to a bond:

```
$ sudo networksetup -isBondSupported device
```

To create a bond and add devices to it:

```
$ sudo networksetup -createBond name [device1] [device2] [...]
```

To delete a bond:

```
$ sudo networksetup -deleteBond bond
```

To add a device to a bond:

```
$ sudo networksetup -addDeviceToBond device bond
```

To remove a device from a bond:

```
$ sudo networksetup -removeDeviceFromBond device bond
```

To list available bonds:

```
$ sudo networksetup -listBonds
```

To display a bond status:

```
$ sudo networksetup -showBondStatus bond
```

Managing AppleTalk Settings

AppleTalk is a suite of protocols developed to implement file sharing, mail service, and printing between Apple computers. To enable or disable AppleTalk, use the `serversetup` tool.

To enable AppleTalk on a port:

```
$ serversetup -EnableAT [(devicename|"portname")]
```

If you don't provide an interface, en0 is assumed.

To disable AppleTalk on a port:

```
$ serversetup -DisableAT [(devicename|"portname")]
```

If you don't provide an interface, en0 is assumed.

To enable AppleTalk on en0:

```
$ serversetup -EnableDefaultAT
```

To disable AppleTalk on en0:

```
$ serversetup -DisableDefaultAT
```

To make AppleTalk active or inactive for a configuration:

```
$ sudo networksetup -setappletalk "configuration" (on|off)
```

To verify the AppleTalk state on en0:

```
$ serversetup -getDefaultATActive
```

To see if AppleTalk is active for a configuration:

```
$ sudo networksetup -getappletalk
```

Managing SNMP Settings

Simple Network Management Protocol (SNMP) is a set of standard protocols used to manage and monitor multiplatform computer network devices.

SNMP relies on a manager/agent design where the agent provides the interface between the manager and the physical device being managed. SNMP uses five basic messages (GET, GET-NEXT, GET-RESPONSE, SET, and TRAP) to communicate between manager and agent.

Mac OS X Server v10.5 includes NET-SNMP v5.4.1.

Setting Up SNMP

To set up SNMP beyond the default configuration:

```
$ snmpconf -g basic_setup
```

This command shows you a set of configuration questions and stores the configuration information in a set of configuration files in /etc/snmp/.

You can download additional documentation from the NET-SNMP Project Home Page (www.net-snmp.org) to learn how to further customize the SNMP configuration files for your site.

> ***WARNING:*** When SNMP is active, anyone with a route to the SNMP host can collect SNMP data from it.

The default configuration of the SNMP agent (`snmpd`) uses privileged port 161. For this reason and others, you must run the agent with root privileges or by using `setuid`.

You should use `setuid` with root privileges only if you understand the ramifications. If you do not, seek assistance or additional information.

Starting SNMP

You can start SNMP in one of the following ways:

- Using Server Admin
- Using the `launchctl` command

Both methods modify Net-SNMP's `launchd` property list (/System/Library/LaunchDaemons/org.net-snmp.snmpd.plist) and start the daemon (`snmpd`) immediately and for the next reboot.

To start SNMP using Server Admin:

1. In Server Admin, select your server.
2. Click General.
3. Enable SNMP by selecting Network Management Server (SNMP).

To start SNMP using the launchctl command:

```
$ sudo launchctl load -w /System/Library/LaunchDaemons/org.net-
     snmp.snmpd.plist
```

Configuring SNMP

The configuration (conf) file for `snmpd` is typically in the /etc/snmp/ folder and the default configuration file is /etc/snmp/snmpd.conf.

You can customize the configuration file while the daemon is running. After the configuration is complete, restart the daemon.

To customize the /etc/snmp/snmpd.conf file, use the `/usr/bin/snmpconf` command. For more information about this command, see its man page.

To customize snmpd data:

1. Add an snmpd.conf file by entering:

```
$ sudo /usr/bin/snmpconf -i
```

This command asks you a series of questions.

2. Provide the appropriate answers.
3. Restart `snmpd`.

Because `snmpd` reads its configuration files at startup, you must restart `snmpd` for your configuration changes to take effect.

To restart snmpd:

```
$ sudo killall snmpd
```

The `launchd` daemon restarts `snmpd`.

Collecting SNMP Information from the Host

To get the SNMP information you just added, enter this command from a host that has the SNMP tools installed:

```
$ snmpget -c community_string hostname system.sysLocation.0
```

Replace *community_string* with the string provided during basic setup. The default community string (or password) is `public`. Also, replace *hostname* with the name of the target host, which could be localhost.

After running the command, you should the location you provided during basic setup, for example:

```
system.sysLocation.0 = server_room
```

The other options defined during basic setup include:

```
$ snmpget -c community_string hostname system.sysContact.0
$ snmpget -c community_string hostname system.sysServices.0
```

The final `.0` indicates you are looking for the index object.

For more information, see the tutorials at net-snmp.sourceforge.net.

Another way to retrieve SNMP information is by retrieving a subtree of management values using the `snmpwalk` tool.

To gather SNMP information in bulk:

```
$ snmpwalk -c community_string localhost system
```

This lists multiple entries of SNMP data similar to the following output, where system name and location are defined in the snmp.conf file.

```
SNMPv2-MIB::sysName.0  -  system name
SNMPv2-MIB::sysLocation.0 - system location
SNMPv2-MIB::sysUpTime.0 - time in 1/100ths of a second since the last system
     start
```

To display all management values:

```
$ snmpwalk -c community_string localhost .1
```

Note: This command generates several thousand lines of output.

To view the system name:

```
$ snmpget -c community_string localhost system.sysName.0
SNMPv2-MIB::sysName.0 = STRING: xlabxs06.example.com
```

To view the system location:

```
$ snmpget -c community_string localhost system.sysLocation.0
SNMPv2-MIB::sysLocation.0 = STRING: "server_room"
```

To view the system uptime:

```
$ snmpget -c community_string localhost system.sysUptime.0
SNMPv2-MIB::sysUpTime.0 = Timeticks: (72239) 0:12:02.39
```

To view a list of snmp man pages:

```
$ man -k snmp
```

Managing Proxy Settings

The proxy server is a component of Mac OS X Server that functions as a relay between a client and the server. This proxy server protects the network from unauthorized users and provides a more secure environment. To view or change the proxy settings, use the `networksetup` tool.

Viewing or Changing FTP Proxy Settings

To view FTP proxy information for a configuration:

```
$ sudo networksetup -getftpproxy "configuration"
```

To set FTP proxy information for a configuration:

```
$ sudo networksetup -setftpproxy "configuration" domain portnumber
```

To view the FTP passive setting for a configuration:

```
$ sudo networksetup -getpassiveftp "configuration"
```

To enable or disable FTP passive mode for a configuration:

```
$ sudo networksetup -setpassiveftp "configuration" (on|off)
```

To enable or disable the FTP proxy for a configuration:

```
$ sudo networksetup -setftpproxystate "configuration" (on|off)
```

Viewing or Changing Web Proxy Settings

To view web proxy information for a configuration:

```
$ sudo networksetup -getwebproxy "configuration"
```

To set web proxy information for a configuration:

```
$ sudo networksetup -setwebproxy "configuration" domain portnumber
```

To enable or disable the web proxy for a configuration:

```
$ sudo networksetup -setwebproxystate "configuration" (on|off)
```

Viewing or Changing Secure Web Proxy Settings

To view secure web proxy information for a configuration:

```
$ sudo networksetup -getsecurewebproxy "configuration"
```

To set secure web proxy information for a configuration:

```
$ sudo networksetup -setsecurewebproxy "configuration" domain portnumber
```

To enable or disable the secure web proxy for a configuration:

```
$ sudo networksetup -setsecurewebproxystate "configuration" (on|off)
```

Viewing or Changing Streaming Proxy Settings

To view streaming proxy information for a configuration:

```
$ sudo networksetup -getstreamingproxy "configuration"
```

To set streaming proxy information for a configuration:

```
$ sudo networksetup -setstreamingproxy "configuration" domain portnumber
```

To enable or disable the streaming proxy for a configuration:

```
$ sudo networksetup -setstreamingproxystate "configuration" (on|off)
```

Viewing or Changing Gopher Proxy Setting

To view gopher proxy information for a configuration:

```
$ sudo networksetup -getgopherproxy "configuration"
```

To set gopher proxy information for a configuration:

```
$ sudo networksetup -setgopherproxy "configuration" domain portnumber
```

To enable or disable the gopher proxy for a configuration:

```
$ sudo networksetup -setgopherproxystate "configuration" (on|off)
```

Viewing or Changing SOCKS Firewall Proxy Settings

To view SOCKS firewall proxy information for a configuration:

```
$ sudo networksetup -getsocksfirewallproxy "configuration"
```

To set SOCKS firewall proxy information for a configuration:

```
$ sudo networksetup -setsocksfirewallproxy "configuration" domain portnumber
```

To enable or disable the SOCKS firewall proxy for a configuration:

```
$ sudo networksetup -setsocksfirewallproxystate "configuration" (on|off)
```

Viewing or Changing Proxy Bypass Domains

To list proxy bypass domains for a configuration:

```
$ sudo networksetup -getproxybypassdomains "configuration"
```

To set proxy bypass domains for a configuration:

```
$ sudo networksetup -setproxybypassdomains "configuration" [domain1] domain2
     [...]
```

Managing AirPort Settings

AirPort uses wireless local area network (WLAN) technology to provide wireless communication between computers. To view or change AirPort settings, use the `networksetup` tool.

To see if AirPort power is on or off:

```
$ sudo networksetup -getairportpower
```

To turn AirPort power on or off:

```
$ sudo networksetup -setairportpower (on|off)
```

To display the name of the AirPort network:

```
$ sudo networksetup -getairportnetwork
```

To join an AirPort network:

```
$ sudo networksetup -setairportnetwork network [password]
```

Managing Computer, Host, and Bonjour Names

These names are used by networking applications to identify a computer and are explained in the following sections.

Computer Name

The computer name is the local name of a computer. This name is typically assigned to the computer when the operating system is installed. To view or modify the computer name, use the `serversetup` tool.

To display the computer name:

```
$ sudo systemsetup -getcomputername
```

or

```
$ sudo networksetup -getcomputername
```

or

```
$ serversetup -getComputername
```

To change the computer name:

```
$ sudo systemsetup -setcomputername computername
```

or

```
$ sudo networksetup -setcomputername computername
```

or

```
$ sudo serversetup -setComputername computername
```

To validate a computer name:

```
$ serversetup -verifyComputername computername
```

Hostname

The host name is a unique name that corresponds to a unique hardware MAC address. It is the name the network uses to identify a device attached to the network. To view or modify the host name, use the `serversetup` tool.

To display the server's local host name:

```
$ serversetup -getHostname
```

To change the server's local host name:

```
$ sudo serversetup -setHostname hostname
```

Note: You can also set and get the host name using `snmpd` and `scutil`.

Bonjour Name

Bonjour, also known as zero-configuration networking, enables automatic discovery of computers, devices, and services on IP networks. Bonjour uses industry-standard IP protocols to allow devices to discover each other without the need to enter IP addresses or configure DNS servers.

Specifically, Bonjour enables automatic IP address assignment without a DHCP server, name-to-address translation without a DNS server, and service discovery without a directory server.

To view or change the Bonjour name, use the `serversetup` tool.

To display the server's Bonjour name

```
$ serversetup -getBonjourname
```

To change the server's Bonjour name:

```
$ sudo serversetup -setBonjourname bonjourname
```

If the name was changed, the command displays `0`.

Note: If you use Server Admin to connect to a server using its Bonjour name and change the server's Bonjour name, you must reconnect to the server the next time you open the Server Admin application.

Managing Preference Files and the Configuration Daemon

The sets of configuration information a user creates at different locations, whether in System Preferences or through the command line, are stored in the preference.plist file located in /Library/Preferences/SystemConfiguration/.

Network configuration is handled by `configd`, the configuration daemon. `configd` reads the network configuration and stores it with the current state of the computer's networking information.

Storage is in the form of key-value pairs. The key is a description of what is being stored, and the value is the value of the information being stored.

You can view the values stored by `configd` at run time and monitor them using the `scutil` tool. This can be especially valuable when you are debugging your network configuration from the command line.

Invoked with no options, `scutil` provides a command-line interface to the data that is maintained by `configd`. For a list of commands you can use with `scutil`, enter `help` at the `scutil` prompt.

To start a `scutil` session (interactive mode):

```
$ scutil
> open
```

This opens a session with `configd`. After the session is open, you can list all keys in the data store for `configd`:

```
> list
```

Each item on the list is a piece of information stored by `configd`, sorted by type. *Setup* indicates information that has been read from a configuration file. *State* indicates information that represents the state of the computer. *File* indicates stored information as of the last time the configuration file was updated.

To view data in the keys, use `scutil`. First you get the data; then you show the data. For example:

```
> get State:/Network/Interface/en0/IPv4
> d.show
```

`scutil` stores the information from the `get` command in a local dictionary variable called `d`. You can also watch or monitor a variable so that if its state changes `scutil` alerts you.

To quit the `scutil` session, enter `quit` at the prompt.

```
> quit
```

You can also manage system configuration parameters `scutil` using the `--get` and `--set` options. These provide a means of reporting and updating a group of persistent system preferences, including ComputerName, LocalHostName, or HostName.

To set the hostname of a system:

```
$ sudo scutil --set HostName mycomputer.mac.com
```

Parameter	Description
mycomputer.mac.com	The new hostname value you want to set

To get the hostname of a system:

```
$ scutil --get HostName
mycomputer.mac.com
```

For more information, see the `scutil` man page or enter `help` at the `scutil` prompt.

Changing Network Locations

A network location contains all network configuration settings for a specific network, such as Ethernet, AirPort, FireWire, or Bluetooth®. Each location has a separate set of network settings.

Mobile users who switch between networks have multiple locations set up on their computer and might need to switch between locations quickly. `scselect` allows you to access these configuration sets or locations.

To view locations:

```
$ scselect
```

The computer responds with output similar to the following:

```
Defined sets include: (* == current set)
     * 0    (Automatic)
       1    (AirPort)
       2    (Home Office)
```

To change the location, enter the number of the location to switch to:

```
$ scselect 1
```

In this example, the network location switches to AirPort.

Working with Disks and Volumes

7

Use this chapter to learn the commands to initialize and test disks and volumes.

This chapter covers the commands used to manage, configure, initialize, and test disks and volumes.

Understanding Disks, Partitions, and the File System

Like UNIX, Mac OS X uses special files called device files, located in /dev, to keep track of the devices (disks, keyboards, monitors, network connections, and so on) attached to the computer.

Device files for a disk are named /dev/disk*n*, where *n* is the number of the disk. For example, a computer with one drive would have a device file called /dev/disk0. If the computer has a second drive, the computer creates a second device file called /dev/disk1, and so on.

Each drive that is divided into multiple partitions has a device file for each partition. The first partition on disk 0 is called /dev/disk0s1, the second partition is /dev/disk0s2, and so on.

Although Mac OS X Server assigns a device name to each device, the files on a device are not accessed in this way. A virtual file system is created where all files on all devices appear to exist in a single hierarchy. This sets one root folder, and every file existing on the computer is under that folder. This is known as the Hierarchical File System (HFS+). The root folder can exist anywhere on a network as a shared resource.

Mounting and Unmounting Volumes

To gain access to files on a different device, you must first mount the device. This process informs the operating system where in the folder tree you want those files to appear. The folder identified to the operating system is the mount point. Different volumes on a computer can have different file systems.

Mounting Volumes

You can use the `mount` tool with parameters appropriate to the type of file system you want to mount, or use one of these file-system–specific mount commands:

- For Apple File Protocol (AppleShare) volumes: `mount_afp`
- For ISO 9660 volumes: `mount_cd9660`
- For CD Digital Audio format (CDDA) volumes: `mount_cddafs`
- For Apple Hierarchical File System (HFS) volumes: `mount_hfs`
- For PC MS-DOS volumes: `mount_msdos`
- For Network File System (NFS) volumes: `mount_nfs`
- For Server Message Block (SMB) volumes: `mount_smbfs`
- For Universal Disk Format (UDF) volumes: `mount_udf`
- For Web-based Distributed Authoring and Versioning (WebDAV) volumes: `mount_webdav`

`mount` prepares and grafts a special device or the remote node (rhost:path) to the file system tree at the point node. For more information, see the related man pages.

To view a list of mounted file systems:

```
$ sudo mount
```

To mount a network folder:

```
$ mount /dev/
```

If the mount succeeded, `mount` returns the value `0`.

Unmounting Volumes

You can use the `umount` tool to unmount a volume. `umount` removes a special device or the remote node (rhost:path) from the file system tree at the point node.

To unmount a volume:

```
$ umount
```

If the umount succeeded, `umount` returns the value `0`. For more information, see the `umount` man page.

Displaying Disk Information

Use the `df` tool in /bin to view free disk space and to identify:

- What your current disk partitions are
- How much space each partition uses
- Which block each partition starts on
- Which device file is associated with each partition
- Where each partition is mounted

To view disk information:

```
$ df
```

The computer responds with output similar to the following:

```
Filesystem              512-blocks     Used     Avail Capacity  Mounted on
/dev/disk0s3             156039264 26138984 129388280    17%    /
devfs                          193      193         0   100%    /dev
fdesc                            2        2         0   100%    /dev
<volfs>                       1024     1024         0   100%    /.vol
automount -nsl [170]             0        0         0   100%    /Network
automount -fstab [174]           0        0         0   100%    /automount/
     Servers
automount -static [174]          0        0         0   100%    /automount/
     static
```

The `-l` option restricts reporting to local drives only. The `-k` option displays sizes in kilobyte format.

Each line in the output refers to a different partition:

- The first column tells you the device file associated with that partition.
- The second column displays the capacity of the partition followed by used and available space on the volume.
- The last column tells you where the partition is mounted.

Monitoring Disk Space

You can monitor the amount of free space on disks and take predefined actions when thresholds are exceeded.

When you need more vigilant monitoring of disk space than the log rolling scripts provide, you can use the `diskspacemonitor` tool. It lets you monitor disk space and take action more frequently than once a day when disk space is critically low, and gives you the opportunity to provide your own action scripts. By default, `diskspacemonitor` is disabled.

To enable diskspacemonitor:

```
$ sudo diskspacemonitor on.
```

You might be prompted for your password.

For more information, see the `diskspacemonitor` man page.

When enabled, `diskspacemonitor` uses information in a configuration file to determine when to execute alert and recovery scripts for reclaiming disk space.

The configuration file is /etc/diskspacemonitor/diskspacemonitor.conf. You can specify how often you want to monitor disk space, and the thresholds to use for determining when to take the actions in the scripts.

By default, disks are checked every 10 minutes, an alert script is executed when disks are 75% full, and a recovery script is executed when disks are 85% full.

To edit the configuration file, log in to the server as an administrator and use a text editor to open the file. For additional information, see the comments in the file.

By default, two predefined action scripts are executed when the thresholds are reached.

The default alert script is /etc/diskspacemonitor/action/alert. It runs in accord with instructions in the configuration file /etc/diskspacemonitor/alert.conf. It sends mail to recipients you specify.

The default recovery script is /etc/diskspacemonitor/action/recover. It runs in accord with instructions in the configuration file /etc/diskspacemonitor/recover.conf.

For more information, see the comments in the script and configuration files.

To provide your own alert and recovery scripts, put your alert script in /etc/diskspacemonitor/action/alert.local and your recovery script in /etc/diskspacemonitor/action/recovery.local. Your scripts are executed before the default scripts when the thresholds are reached.

To configure the scripts on a server from a remote Mac OS X computer, open a Terminal window and log in to the remote computer using SSH.

Reclaiming Disk Space Using Log-Rolling Scripts

The following scripts are executed to reclaim space used on your server:

- The script /etc/periodic/daily/600.daily.server runs daily. Its configuration file is /etc/diskspacemonitor/daily.server.conf.
- The script /etc/periodic/weekly/600.weekly.server runs weekly, but is empty. Its configuration file is /etc/diskspacemonitor/weekly.server.conf.
- The script /etc/periodic/monthly/600.monthly.server runs monthly, but is empty. Its configuration file is /etc/diskspacemonitor/monthly.server.conf.

These scripts reclaim space used by log files generated by the following services:

- Apple file service
- Windows service
- Web service
- Web performance cache
- Mail service
- Print service

As configured, the scripts specify actions that complement the log file management performed by the services listed above, so don't modify them. Log in as an administrator and use a text editor to define thresholds in the configuration files that determine when actions are taken. Thresholds include:

- The number of megabytes a log file must contain before its space is reclaimed.
- The number of days since a log file's last modification that need to pass before its space is reclaimed.

Specify one or both thresholds. The actions are taken when either threshold is exceeded.

You can specify several additional parameters. For information about the parameters and how to set them, see comments in the configuration files.

The scripts ignore log files except those for which at least one threshold is present in the configuration file.

To configure the scripts on a server from a remote Mac OS X computer, open a Terminal window and log in to the remote server using SSH. Then, open a text editor and edit the scripts.

You can also use the `diskspacemonitor` tool to reclaim disk space.

Using the diskutil Tool

You can use `diskutil` to erase, modify, verify, and repair disks. This command provides functionality that overlaps the functionality of `pdisk`, `newfs_hfs`, and `disktool`.

For example, you can use `diskutil` and `pdisk` to partition a disk. However, unlike `pdisk`, which lets you partition tables at their most basic level by setting the base address and partition length in blocks, `diskutil` lets you partition a disk automatically by calculating the base address and the partition length in blocks based on the partition size you specify.

The `diskutil` tool allows you to perform the following actions on a disk:

To list the disks known and available on the computer:

```
$ diskutil list
```

If your system is an Xserve computer, you can use this command to determine which drive is in which bay.

To erase and repartition a disk:

```
$ diskutil partitionDisk disk numberOfPartitions <part1Format part1Name
    part1Size> <part2Format part2Name part2Size> ...
```

Parameter	Description
disk	Device name (such as `disk0`).
numberOfPartitions	Number of partitions.
part1Format	The format of the volume. The valid formats or filesystem names available in Disk Utility are: • "Journaled HFS+"—corresponds to Mac OS Extended (Journaled) and is the default and recommended startup volume format. • HFS+—corresponds to Mac OS Extended. • "Case-sensitive Journaled HFS+"—corresponds to Mac OS Extended (Case-sensitive, Journaled). This format is available for the "erase and install" option for local installations, is *not* available for remotely controlled installations, and might have issues with third-party applications. • "Case-sensitive HFS+"—corresponds to Mac OS Extended (Case-sensitive). • "MS-DOS FAT32"—corresponds to MS-DOS (FAT). • Swap—corresponds to Free Space. • ZFS—corresponds to Zettabyte File System (ZFS). Other valid formats are HFS, "MS-DOS FAT16", MS-DOS, "MS-DOS FAT12", Linux, and UFS. UFS is not a supported boot volume format. The available formats for erasing, partitioning, and creating RAID sets are specified in a plist file for each filesystem (/System/Library/Filesystems/*fs_name*.fs/Contents/Info.plist, where *fs_name* is an acronym in lower case representing the filesystem).
part1Name	The name of the partition.
part1Size	The size of the partition in bytes (such as 98187445B), kilobytes (such as 810240K), megabytes (such as 4024M), gigabytes (such as 4G), or terabytes (such as 1T).

Because HFS+ is case preserving but not case sensitive, there might be times when you would want to set the file system to be case sensitive. Use the `diskutil` tool to format a drive for case-sensitive HFS+.

To mount a volume:

```
$ diskutil mountDisk diskvol
```

Parameter	Description
diskvol	Device name

To get mount info about a partition:

```
$ diskutil info diskvol
```

Parameter	Description
diskvol	Device name (for example, `disk0s9`) for the partition

This command tells you the device file that corresponds to the mounted partition (or device name) you specify.

To format a Mac OS Extended volume as case-sensitive HFS+:

```
$ sudo diskutil eraseVolume "Case-sensitive HFS+" newvolname volume
```

Parameter	Description
newvolname	The name given to the reformatted, case-sensitive volume
volume	The path to the existing volume to be reformatted For example: `/Volumes/HFSPlus`

For more options and information about repairing and modifying disks, see the `diskutil` man page.

Using the pdisk, disklabel, and newfs Tools

Disk partitions are subdivisions of a disk that you apply operating-system-specific formatting to.

Partitioning a Disk

You can use `pdisk`, located in /usr/sbin, to initialize the disk, create partitions, and delete partitions. The `pdisk` tool is menu-driven, which means that when it is launched, you are prompted to enter a `pdisk` command. You can find the commands by entering `?` at the pdisk prompt.

The following are some of the more useful commands:

Command	Description
`L`	Lists the partition maps of all drives. `pdisk` lists all partitions for a disk—even the unmountable partitions, such as the partition containing the partition map.
`e`	Edits the partition map of the named device. To edit a partition map, use the raw device file as the argument.

When you start editing a device, the `pdisk` options change. Enter `?` at the pdisk prompt to see the editing commands. The following are some of the more important ones:

Command	Description
`p`	Prints the partition map for the current device.
`i`	Initializes the partition map for the current device.
`C`	Creates a partition. There are two partition types: Apple_HFS and Apple_UFS.
`w`	Writes the modifications to the partition map on-disk. Before that, edits and modifications are only in memory and are not yet implemented.

`pdisk` does not support the Intel/DOS partitioning scheme supported by `fdisk`. For more information about DOS partitions, see the `fdisk` man page.

After a partition is created on a device, the partition must be formatted before the computer can store data on the device. Formatting a disk partition creates the volume and sets the file system.

Labeling a Disk

After a disk is formatted, it must be labeled. The `disklabel` tool manipulates Apple Label partition metadata. Apple Label partitions allow for a disk device to have a consistent name, ownership, and permissions across reboots, even though it uses a dynamic pseudo file system for /dev.

The Apple Label partition uses a set of metadata (as a plist) in a reserved area of the partition. This metadata describes the owner, name, and so forth.

To create a disk label for a device with 1 MB of metadata area, owned by Anne, with a device name of Fred, and writable by Anne:

```
$ disklabel -create /dev/rdisk1s1 -msize=1M owner-uid=anne dev-devname=anne
     name=anne owner-mode=0644
```

The following example prints the key-value pairs from the previous example:

```
$ disklabel -properties /dev/rdisk1s1
```

For more information about creating disk labels, see the `disklabel` man page.

Formatting a Disk

To create a volume, use `newfs`, located in /sbin. `newfs` builds a file system on the specified special device, basing its defaults on the information in the disk label.

There are many parameters you can set when formatting disks, such as block and clump size, b-tree attribute, and catalog node sizes.

Important: Take extreme care to ensure a successful format when modifying the settings beyond the default.

Before running `newfs`, label the disk using the `disklabel` tool.

To format a disk:

```
$ newfs
```

For more information, see the `newfs` man page.

To format a disk to HFS+:

- Use the `newfs_hfs` tool in /sbin:

```
$ newfs_hfs
```

For more information, see the `newfs_hfs` man page.

Troubleshooting Disk Problems

To verify the physical condition and file system integrity of a volume, use the `diskutil` or `fsck` tool (`fsck_hfs` for HFS volumes). For more information, see the related man pages.

Managing Disk Journaling

A robust file system journaling feature is available to enhance the availability and fault tolerance of servers and server-attached storage devices.

Journaling protects the integrity of the Mac OS Extended (HFS+) file system in the event of an unplanned shutdown or power failure, and maximizes uptime by expediting repairs to the affected volumes when the computer restarts.

Determining if Journaling Is Enabled

To see if journaling is enabled on a volume, use the `mount` tool.

To see if journaling is enabled:

```
$ mount
```

Look for `journaled` in the attributes in parentheses following a volume. For example:

```
/dev/disk0s9 on / (local, journaled)
```

Enabling Journaling for a Volume

To enable journaling on a volume without affecting files on the volume, use the `diskutil` tool.

Important: Always check the volume for disk errors using the `fsck_hfs` tool before you enable journaling.

To enable journaling:

```
$ diskutil enableJournal volume
```

Parameter	Description
volume	The volume name or device name of the volume

The following example shows journaling being enabled on volume /dev/disk0s10.

```
$ mount
/dev/disk0s9 on / (local, journaled)
/dev/disk0s10 on /Volumes/OS 9.2.2 (local)
$ sudo fsck_hfs /dev/disk0s10/
** /dev/rdisk0s10
** Checking HFS plus volume.
** Checking extents overflow file.
** Checking Catalog file.
** Checking Catalog hierarchy.
** Checking volume bitmap.
** Checking volume information.
** The volume OS 9.2.2 appears to be OK.
$ diskutil enableJournal /dev/disk0s10
Allocated 8192K for journal file.
Journaling has been enabled on /dev/disk0s10
$ mount
/dev/disk0s9 on / (local, journaled)
/dev/disk0s10 on /Volumes/OS 9.2.2 (local, journaled)
```

Enabling Journaling When You Erase a Disk

To set up and enable journaling when you erase a disk, use the `newfs_hfs` tool.

To enable journaling when erasing a disk:

```
$ newfs_hfs -J -v volname device
```

Parameter	Description
volname	The name you want the new disk volume to have
device	The device name of the disk

Disabling Journaling

To disable journaling:

```
$ diskutil disableJournal volume
```

Parameter	Description
volume	The volume name or device name of the volume

Understanding Spotlight Technology

Spotlight is a desktop search technology that combines metadata-indexing with content-indexing that's optimized for Mac OS X.

When a file is added, moved, deleted, or modified, the file system notifies the Spotlight engine. The Spotlight engine then updates its index, known as the Spotlight store. The Spotlight engine then updates applications that use Spotlight, and changes are reflected dynamically to the user.

The Spotlight store retains information in two indexes, one for metadata and the other for content. Each index is created on a per-volume basis, which means each disk or partition carries its own set of indexes for the information about that volume.

Enabling and Disabling Spotlight

By default, the value of the *spotlight* parameter in the /etc/hostconfig file is set to `-YES-`, which means Spotlight is enabled on your Mac OS X Server computer.

To disable Spotlight on your server:

1. Open the /etc/hostconfig file for editing with root privileges using your favorite editor.

   For example:

   ```
   $ sudo pico /etc/hostconfig
   ```

2. Change the value of the *spotlight* parameter to `-NO-`.

   You can set the value of the *spotlight* parameter to `-NO-` as follows:

   ```
   $ sudo /System/Library/ServerSetup/serversetup -setAutoStartSpotlight 0
   ```

3. Restart your server.

To enable Spotlight on your server:

1. Open `/etc/hostconfig` for editing with root privileges.
2. Change the value of the *spotlight* parameter to `-YES-`.

   You can set the value of the `SPOTLIGHT` parameter to `-YES-` as follows:

   ```
   $ sudo /System/Library/ServerSetup/serversetup -setAutoStartSpotlight 1
   ```

3. Restart your server.

Performing Spotlight Searches

Mac OS X provides the ability to view the metadata of a file and perform Spotlight searches from the command line.

To view a file's Spotlight metadata, use the `mdls` tool. This tool, similar to the `ls` tool, lists metadata attributes for a file.

To view the metadata of a file:

```
$ mdls filename
```

The computer responds with something similar to the following output:

```
<filename> -------------
kMDItemAttributeChangeDate = 1970-01-01 00:43:07 -0600
kMDItemFSContentChangeDate = 2005-10-03 22:04:19 -0500
kMDItemFSCreationDate      = 2005-10-03 22:04:19 -0500
kMDItemFSCreatorCode       = 0
kMDItemFSFinderFlags       = 16384
kMDItemFSInvisible         = 1
kMDItemFSIsExtensionHidden = 0
kMDItemFSLabel             = 0
kMDItemFSName              = "filename"
kMDItemFSNodeCount         = 0
kMDItemFSOwnerGroupID      = 0
kMDItemFSOwnerUserID       = 0
kMDItemFSSize              = 4330232
kMDItemFSTypeCode          = 0
kMDItemID                  = 634516
kMDItemLastUsedDate        = 2005-10-03 21:04:19 -0500
kMDItemUsedDates           = (2005-10-03 21:04:19 -0500)
```

To perform a Spotlight search using the mdfind tool:

```
$ mdfind "kMDItemAcquisitionModel ='Canon Powershot S45'"
/Users/anne/Documents/vacation1.jpg
/Users/anne/Documents/vacation2.jpg
/Users/anne/Documents/vacation3.jpg
/Users/anne/Documents/vacation4.jpg
```

Controlling Spotlight Indexing

By default, indexing of volumes in Mac OS X Server is disabled. However, you can use the `mdutil` tool to enable or disable indexing on a volume.

To enable indexing on a volume:

Run the `mdutil` tool with root privileges and set the indexing status to `on`.

```
$ sudo mdutil -i on volume
```

To disable indexing on a volume:

Run the `mdutil` tool with root privileges and set the indexing status to `off`.

```
$ sudo mdutil -i off volume
```

For more information, see the `mdutil` man page.

Managing RAID Volumes

In addition to standard drive management options, you can use `diskutil` to manage software RAID volumes.

To create a RAID set:

```
$ diskutil createRAID type setName volType disks
```

Parameter	Description
type	Mirror or stripe
setName	Name of the new RAID volume
volType	HFS, HFS+, UFS, or BootableHFS
disks	List of device names for members of the RAID set

To get a list of disks available to add to a RAID set:

```
$ diskutil list
```

Similarly, you can remove a RAID set with the `diskutil destroyRAID` command.

To view a list of available RAID sets:

```
$ diskutil checkRAID device
```

Parameter	Description
device	Device file

To create an unpaired mirrored RAID set from a single file system disk:

```
$ diskutil enableRAID mirror device
```

Parameter	Description
mirror	Name of the mirror RAID set
device	Device file

To repair a failed mirror:

```
$ diskutil repairMirror device slicenumber fromDisk toDisk
```

Parameter	Description
device	Device file
slicenumber	The slice number to replace
fromDisk	The mirror source
toDisk	The repaired mirror destination

Note: Xsan RAID volumes have their own commands, described in an appendix of the *Xsan Administrators* guide. For information about the `megaraid` tool (used for managing a PCI RAID card), see the appendix.

Imaging and Cloning Volumes Using ASR

You can use Apple Software Restore (ASR) to copy a disk image onto a volume or to prepare disk images with checksum information for faster copies. ASR can perform file copies, in which individual files are restored to a volume unless an identical file exists there, and block copies, which restores entire disk images.

The `asr` tool doesn't create the disk images. You use `hdiutil` to create disk images from volumes or folders.

You must run ASR with root privileges. You cannot use ASR on read or write disk images.

To image a boot volume:

1. Install and configure Mac OS X on the volume.
2. Restart from a different volume.
3. Make sure the volume you're imaging has permissions enabled.

   Use the following to verify permissions:

   ```
   $ diskutil verifyPermissions [mount point|disk identifier|device node]
   ```

4. Use `hditutil` to make a read-write disk image of the volume.

   See "Using hdiutil with System Images" on page 183.

5. Mount the disk image.
6. Remove cache files, host-specific preferences, and virtual memory files.

   For examples of what files to remove, see the `asr` man page.

7. Unmount the volume and convert the read-write image to a read-only compressed image:

   ```
   $ hdiutil convert -format UDZO pathtoimage -o compressedimage
   ```

8. Prepare the image for duplication by adding checksum information:

   ```
   $ sudo asr -imagescan compressedimage
   ```

To restore a volume from an image:

```
$ sudo asr -source compressedimage -target targetvolume -erase
```

For more information, see the `asr` man page.

Managing User and Group Accounts

8

Use this chapter to learn the commands to set up and manage user and group accounts.

With Mac OS X Server, you can quickly create and administer accounts for users and groups. Several command-line tools are available to facilitate working with the directory domains that hold these accounts.

User, Group, Computer, and Computer Group Accounts

You set up four kinds of accounts with Workgroup Manager: user accounts, group accounts, computer accounts, and computer group accounts.

When you define a user's account, you specify the information needed to prove the user's identity: user name, password, and user identification number (user ID). Other information in a user's account is needed by various services to determine what the user is authorized to do and to personalize the user's environment.

Along with accounts you create, Mac OS X Server has predefined user and group accounts, some of which are reserved for use by Mac OS X.

Most users have an individual account used to authenticate them and control their access to services. When you want to personalize a user's environment, you define user, group, or computer preferences for that user.

The term *managed client* or *managed user* designates a user who has administrator-controlled preferences associated with his or her account. When a managed user logs in, the preferences that take effect are a combination of the user's preferences and preferences set up for any workgroup or computer list he or she belongs to.

Administering and Creating User Accounts

This section describes how to administer user accounts stored in directory domains.

A user account stores data that Mac OS X Server needs to validate the user's identity and provide services for the user.

User and group accounts, as well as computer and computer group accounts, can be stored in any Open Directory domain accessible from any Mac OS X computer. A directory domain can reside on a Mac OS X computer (for example, the LDAP folder of an Open Directory master or another read/write directory domain) or it can reside on a non-Apple server (for example, a non-Apple LDAP or Active Directory server).

Creating a Local Administrator User Account for a Server

Users with server or directory domain administration privileges are known as administrators. An administrator can be a server administrator, domain administrator, or both. Server administrator privileges determine whether a user can view information about or change the settings of a specific server.

Domain administrator privileges determine the extent to which the user can view or change account settings for users, groups, computers, and computer groups in the directory domain.

To create local administrator users for a server, use the `serversetup` tool. The `serversetup` tool is located in /System/Library/ServerSetup/ and is not in the local path, so you must provide the path to it. You must also run it with root privileges.

To create nonadministrator users, see "Creating a Nonadministrator User Account" on page 102.

To create administrator users in a network directory domain, see "Creating a Domain Administrator User Account" on page 101.

To create a local administrator user account:

```
$ sudo /System/Library/ServerSetup/serversetup -createUser fullname
     shortname password
```

Enter the name, short name, and password in the order shown. If the full name includes spaces, enter it in quotes.

The command displays a `0` if successful, or a `1` if the full name or short name is already in use.

To create a local administrator user with a specific UID:

```
$ sudo /System/Library/ServerSetup/serversetup -createUserWithID fullname
     shortname password uid
```

Enter the name, short name, password, and UID in the order shown. If the full name includes spaces, enter it in quotes.

The command displays a `0` if successful, or a `1` if the full name, short name, or UID is already in use or if the UID you specified is less than 100.

To create a local administrator user with a specific UID and home folder:

```
$ sudo /System/Library/ServerSetup/serversetup -createUserWithIDIP fullname
     shortname password uid homedirpath
```

Enter the name, short name, password, and UID in the order shown. If the full name includes spaces, enter it in quotes.

The command displays a `0` if successful, or a `1` if the full name, short name, or UID is already in use or if the UID you specified is less than 100.

Creating a Domain Administrator User Account

To create a domain administrator user account for a networked directory, you must have a domain administrator user account.

Before starting, you should have a nonadministrator user account that you want to give domain administrator privileges to. For instructions on creating nonadministrator user accounts, see "Creating a Nonadministrator User Account" on page 102.

To create a domain administrator user account:

1. Start the `dscl` tool in interactive mode, specifying the computer you are using as the source of directory service data.

   Use the `dscl` tool to create a domain administrator user account.

   ```
   $ dscl localhost
   >
   ```

   In interactive mode, the `dscl` tool displays the current folder in the directory domain (not the current folder in the file system) and a ">" character as a prompt.

2. After you connect to the directory, choose the directory domain and change the current folder to LDAPv3/*ipaddress*/Groups:

   ```
   > cd LDAPv3/ipaddress/Groups
   ```

   Replace *ipaddress* with the IP address of your directory server.

3 Authenticate as an administrator by entering the following command, replacing *adminusername* with your administrator user name, and entering your administrator password when prompted:

```
> auth adminusername
```

4 Create an administrator user.

```
>append admin Member adminusername
```

This command creates an administrator user, but it doesn't add the globally unique identifier (GUID) of the administrator user to the group account.

5 Add the administrator user to the group.

```
> append admin GroupMembers guid
```

Replace *guid* with the globally unique identifier.

6 Quit the `dscl` tool.

```
>quit
```

To find the GUID of the administrator user:

```
> cd /LDAPv3/ipaddress/Users
> read adminusername GeneratedUID
```

Verifying a User's Administrator Privileges

To verify the administrator privileges of a user, use the `serversetup` tool.

To see if a user is a server administrator:

```
$ sudo /System/Library/ServerSetup/serversetup -isAdministrator shortname
```

The command displays a `0` if the user is an administrator, or a `1` if the user is not an administrator.

Creating a Nonadministrator User Account

You can create user accounts by using `dscl` and other tools.

When you create a user account from the command line, you must also set values for basic attributes of the user account, such as the short name, long name, user ID, and home folder location.

To create a nonadministrator user account:

1 Identify an unused user ID by using the `dscl` tool to display lists of assigned user IDs and group IDs.

```
$ dscl /LDAPv3/ipaddress -list /Users UniqueID| awk '{print $2}' | sort -n
```

Replace */LDAPv3/ipaddress* with the location of your directory domain (the way it appears in the search path in Directory Access).

After you enter the command, the `dscl` tool displays a list of assigned user ID numbers, similar to the following output. These user IDs are for computer accounts that are included with Mac OS X Server:

```
-2
0
1
99
25
26
27
70
71
75
76
77
78
79
501
```

Important: Select a user ID that isn't in the list of assigned user ID numbers created when you install Mac OS X Server.

2 Start the `dscl` tool in interactive mode, specifying the computer you are using as the source of directory service data; and use the `dscl` tool to create a nonadministrator user account.

```
$ dscl localhost
>
```

In interactive mode, the `dscl` tool displays the current folder in the directory domain (not the current folder in the file system) and a ">" character as a prompt.

3 Change the current folder to /LDAPv3/*ipaddress*/Users by entering the path at the prompt:

```
> cd /LDAPv3/ipaddress/Users
```

Replace *ipaddress* with the IP address of your directory server.

4 Authenticate as an administrator by entering the following command, replacing *adminusername* with your administrator user name, and entering your administrator password when prompted:

```
> auth adminusername
```

5 Create a user account, replacing *ajohnson* with the new user account's short name and specifying the path to the new user's home folder in /Users/:

```
> create ajohnson HomeDirectory "<home_dir><url>afp://sp.apple.com/Users
    </url><path>ajohnson</path></home_dir>"
> create ajohnson NFSHomeDirectory /Network/Servers/sp.apple.com/Users/
    ajohnson
```

Replace `sp.apple.com` with your home folder server's location.

6 Specify the new user's default UNIX shell:

```
> create ajohnson UserShell /bin/bash
```

7 Specify the user ID, replacing *1234* with the new user's ID:

```
> create ajohnson UniqueID 1234
```

8 Specify the long name for the new user account, replacing *Anne Johnson* with the actual long name:

```
> create ajohnson RealName "Anne Johnson"
```

9 Review the settings of your new user account by entering the following command, replacing *ajohnson* with the new user account's short name as before:

```
> read ajohnson
```

`dscl` displays the settings for your new user account, similar to the following output:

```
dsAttrTypeNative:apple-generateduid:1B2A3456-E7C8-9EC1-2345-678D912E3456
dsAttrTypeNative:cn: anne johnson
dsAttrTypeNative:gidNumber: 99
dsAttrTypeNative:HomeDirectory: /LDAPv3/ipaddress/Users/ajohnson
dsAttrTypeNative:loginShell: /bin/bash
dsAttrTypeNative:objectClass: inetOrgPerson posixAccount shadowAccount
     apple-user extensible object organizationalPerson top person
dsAttrTypeNative:sn: ajohnson
dsAttrTypeNative:uid: ajohnson
dsAttrTypeNative:uidNumber: 1234
AppleMetaNodeLocation: /LDAPv3/ipaddress
GeneratedUID:1B2A3456-E7C8-9EC1-2345-678D912E3456
LastName: johnson
NFSHomeDirectory: /LDAPv3/ipaddress/Users/ajohnson
PasswordPlus:********
PrimaryGroupID: 99
RealName: Anne Johnson
RecordName: ajohnson anne
RecordType: dsRecTypeStandard:Users
UniqueID: 1234
UserShell: /bin/bash
```

10 Assign a password to the account by entering the following command, replacing *ajohnson* with the new account's short name:

```
> passwd ajohnson
```

11 Quit `dscl` by entering:

```
> quit
```

The `dscl` tool displays `Goodbye`, and then the standard shell prompt appears.

12 Use the `ssh` tool to connect to the server where you are hosting home folders:

```
$ ssh -l username server
```

Replace *username* with the name of an administrator user on the remote server and replace *server* with the name or IP address of the server.

13 Create the home folder for the new user.

Use the `-s` option if you are using a network directory domain or the `-c` option if you are using a local directory domain. You must run the command to create the home folder with root privileges.

```
$ sudo createhomedir -s -u ajohnson
```

To create a group account for the user, see "Creating a Group Account" on page 111 before doing this step.

The user account is now complete and can be used for logging in. For more information, see the `dscl` man page.

Retrieving a User's GUID

When a user account is created, the computer generates a 128-bit integer called a GUID. This is stored in the LDAP directory.

The GUID is used for permissions and for associating users with group memberships. In command-line tools, you might see a GUID referred to as a GeneratedUID.

To retrieve a user's GUID:

1 Start the `dscl` tool in interactive mode, specifying the computer you are using as the source of directory service data:

```
$ dscl localhost
>
```

2 Change the current folder to /LDAPv3/*ipaddress*/Users by entering the path at the prompt:

```
> cd /LDAPv3/ipaddress/Users
```

Replace *ipaddress* with the IP address of your directory server.

3 Authenticate as an administrator by entering the following command, replacing *adminusername* with an administrator's user name, and entering an administrator's password when prompted:

```
> auth adminusername
```

4 Review the GUID for a user.

```
> read username GeneratedUID
```

5 Quit `dscl` by entering:

```
> quit
```

Removing a User Account

You can remove a user account by using the `dscl` tool. This does not remove the user's home folder and the data that may be stored there. You can use the Finder to drag the deleted user's home folder to the Trash.

To remove a user account:

1. Start the `dscl` tool in interactive mode, specifying the computer you are using as the source of directory service data:

```
$ dscl localhost
>
```

2. Change the current folder to /LDAPv3/*ipaddress*/Users by entering the path at the prompt:

```
> cd /LDAPv3/ipaddress/Users
```

Replace *ipaddress* with the IP address of your directory server.

3. Authenticate as an administrator by entering the following command, replacing *adminusername* with an administrator's user name, and entering that administrator's password when prompted:

```
> auth adminusername
```

4. Delete the user account by entering the following command, replacing *ajohnson* with the user account's short name:

```
> delete ajohnson
```

5. Quit `dscl` by entering:

```
> quit
```

A user account usually has a matching group of the same name. For information about deleting this group, see "Removing a Group Account" on page 112.

Preventing a User from Logging In

Sometimes it is necessary to revoke a user's ability to access the computer. This involves preventing the user from logging in and then terminating the user's processes.

The latter can be done by forcing the user to log out and then killing remaining processes, or by just killing the user's processes.

To prevent a user from logging in:

- Disable the user account by entering the following command:

```
$ pwpolicy -a diradmin -u ajohnson -setpolicy "isDisabled=1"
```

Replace *ajohnson* with the short name of the user account and replace *diradmin* with the short name of your domain administrator account.

Note: The `pwpolicy` command only works for LDAP/Password server users. For a local user, use Workgroup Manager or the Accounts pane of System Preferences.

To terminate a user's processes:
After disabling the user account, you need to kill the user's active processes that are running on the directory server.

> ***WARNING:*** Unconditionally killing a user's processes causes the user to lose unsaved data.

1. Make all processes clean up and exit by entering the following command, replacing *ajohnson* with the user name:

```
$ sudo killall -TERM -u ajohnson
```

2. Wait a few seconds to allow the previous command to execute; then, to terminate the user's processes, enter the following command, replacing *ajohnson* with the user name:

```
$ sudo killall -9 -u ajohnson
```

For more information about terminating processes, see the `killall` man page.

To reenable a disabled user account:

- Enable the user account by entering the following command.

```
$ pwpolicy -a diradmin -u ajohnson -setpolicy "isDisabled=0"
```

Replace *ajohnson* with the short name of the user account and replace *diradmin* with the short name of your domain administrator account.

Verifying a Server User's Name, UID, or Password

To verify the name, UID, or password of a user in the server's local directory domain, use the following commands.

Note: These tasks apply only to the local directory domain on the server.

To see if a full name is in use:

```
$ sudo /System/Library/ServerSetup/serversetup -verifyRealName "longname"
```

The command displays a `1` if the name is in use, or a `0` if it isn't.

To see if a short name is in use:

```
$ sudo /System/Library/ServerSetup/serversetup -verifyName shortname
```

The command displays a `1` if the name is in use, or a `0` if it isn't.

To see if a UID is in use:

```
$ sudo /System/Library/ServerSetup/serversetup -verifyUID uid
```

The command displays a `1` if the UID is in use, or a `0` if it isn't.

To test a user's password:

```
$ sudo /System/Library/ServerSetup/serversetup -verifyNamePassword shortname
    password
```

The command displays a `1` if the password is good, or a `0` if it isn't.

To view names associated with a UID:

```
$ sudo /System/Library/ServerSetup/serversetup -getNamesByID uid
```

If you don't receive a response, the UID is not valid.

To get the default UNIX short name for a user long name:

```
$ sudo /System/Library/ServerSetup/serversetup -getUNIXName "longname"
```

Note: Mac OS X Server provides the `net` tool, which is essentially a clone of the Windows `net` command. The `net` tool enables administrators to perform advanced customization of the Primary Domain Controller (PDC) and mapping domain privileges to UNIX groups. For more information, see the `net` man page.

Modifying a User Account

You can change the value of an attribute in a user account by using `dscl`.

You can set or modify the following user account attributes using `dscl`:

Attribute	Description
apple-GeneratedUID	User ID generated by the system
cn	User's common name
homeDirectory	Location of the user's Home folder
loginShell	User's Terminal shell
sn	User's surname name
LastName	User's last name
NFSHomeDirectory	Location of the user's Home folder
PasswordPlus	User's password
PrimaryGroupID	User's primary group ID
RealName	User's name
UserShell	User's Terminal shell

To change a user account attribute to a new value:

1. Start the `dscl` tool in interactive mode, specifying the computer you are using as the source of directory service data:

```
$ dscl localhost
>
```

2. Change the current folder to /LDAPv3/*ipaddress*/Users by entering the path at the prompt:

```
> cd /LDAPv3/ipaddress/Users
```

Replace *ipaddress* with the IP address of your directory server.

3. Authenticate as an administrator by entering the following command, replacing *adminusername* with your administrator user name, and entering your administrator password when prompted:

```
> auth adminusername
```

4. Set the user attribute to the desired value by entering the following command:

```
> create ajohnson attribute newvalue
```

Replace *ajohnson* with the user account's short name, *attribute* with the name of the attribute whose value you want to change, and *newvalue* with the value.

5. Quit `dscl` by entering:

```
> quit
```

Managing Home Folders

A Home folder is a folder where a user's files and preferences are stored. Other users can see a user's Home folder and read files in its Public folder, but they can't (by default) access anything else in that folder. This is true only for other users whose Home folders reside on the same server or share point.

When you create a user account in a directory domain on the network, you specify the location of the user's Home folder on the network. The location is stored in the user account and used by various services, including the login window and Mac OS X managed client services.

Creating a User's Home Folder

Normally, you can create a user's Home folder by clicking the Create Home Now button on the Homes pane of Workgroup Manager. You can also create Home folders using the `createhomedir` tool. Otherwise, Mac OS X Server creates the user's Home folder when the user logs in for the first time.

You can use `createhomedir` to create:

- A Home folder for a specific user (`-u` option)
- Home folders for all users in a directory domain (`-l` or `-n` option)
- Home folders for all users in all domains in the folder search path (`-a` option)

For more information, see the `createhomedir` man page.

In all cases, Home folders are created on the server where you run the tool.

To create a Home folder for a user:

```
$ sudo createhomedir -u uid
```

In addition to the *uid*, you can also use the user's short name.

To create a Home folder for users in the local domain:

```
$ sudo createhomedir [(-a|-l|-n domain)] -u uid
```

You can also create a user's Home folder using the `serversetup` tool.

To create a Home folder for a user:

```
$ sudo /System/Library/ServerSetup/serversetup -createHomedir uid
```

The command displays a `1` if the user ID you specify doesn't exist.

Mounting a User's Home Folder

To mount a user's Home folder, use `mnthome`. The `mnthome` tool unmounts the AFP (AppleShare) Home folder that was automounted as guest, and remounts it with the correct privileges by logging into the AFP server using the current user name and password.

To mount a user's shared Home folder on an AFP server:

```
$ mnthome -p password
```

For more information, see the `mnthome` man page.

Administering Group Accounts

A group is a collection of users who have similar needs. For example, you can add all users with a task to one group and give the group permission to access certain files or folders on a volume.

Groups simplify the administration of shared resources. Instead of granting access to resources to each individual who needs them, you can add the users to a group and then grant access to the group. Information in group accounts helps control user access to folders and files. Individual users can belong to multiple groups, depending on their access needs.

A group can be nested within another group. A group that contains another group is called a parent group, and the group that is contained is called a nested group. Nested groups are useful for inheriting access permissions at login time.

Creating a Group Account

You can create a group account by using `dscl` and other tools. When you create a group account via the command line, you must also set values for basic attributes of a group account, such as short name and group ID.

To add a group account:

1. Identify an unused group ID by entering the following command to display a list of assigned group IDs.

```
$ dscl /LDAPv3/ipaddress -list /Groups PrimaryGroupID | awk '{print $2}' |
     sort -n
```

Replace *ipaddress* with the location of your directory domain (the way it appears in the search path in Directory Access).

After you enter the command, the `dscl` tool displays a list of assigned IDs similar to the following output:

```
-2
0
1
25
78
79
501
```

Important: In this example, select an ID that isn't on the list, and that is greater than 501.

2. Start the `dscl` tool in interactive mode, specifying the computer you are using as the source of directory service data:

```
$ dscl localhost
>
```

3. Change the current folder to /LDAPv3/*ipaddress*/Groups by entering the path at the prompt:

```
> cd /LDAPv3/ipaddress/Groups
```

Replace *ipaddress* with the IP address of your directory server.

4. Authenticate as an administrator by entering the following command, replacing *adminusername* with your administrator user name, and entering your administrator password when prompted:

```
> auth adminusername
```

5. Create a group, replacing *officegroup* with the new group account's short name, and specify the group ID, replacing *600* with the primary group ID.

```
> create officegroup PrimaryGroupID 600
```

6 Review the settings of your group by entering the following command, replacing *officegroup* with the group account's short name.

```
> read officegroup
```

`dscl` displays the settings for your group account, similar to the following output:

```
dsAttrTypeNative:apple-generateduid:4B3A5678-E9C1-2EC3-4567-891D234E5678
dsAttrTypeNative:cn: officegroup
dsAttrTypeNative:gidNumber: 600
dsAttrTypeNative:objectClass: posixGroup apple-group extensibleObject top
AppleMetaNodeLocation: /LDAPv3/ipaddress
GeneratedUID:4B3A5678-E9C1-2EC3-4567-891D234E5678
PasswordPlus:********
PrimaryGroupID: 600
RecordName: officegroup
RecordType: dsRecTypeStandard:Groups
```

7 Quit the `dscl` tool.

```
>quit
```

For more information, see the `dscl` man page.

Removing a Group Account

You can remove group accounts by using the `dscl` tool.

To remove a group account:

1 Start the `dscl` tool in interactive mode, specifying the computer you are using as the source of directory service data:

```
$ dscl localhost
>
```

2 Change the current folder to /LDAPv3/*ipaddress*/Groups by entering the path at the prompt:

```
> cd /LDAPv3/ipaddress/Groups
```

Replace *ipaddress* with the IP address of your directory server.

3 Authenticate as an administrator by entering the following command, replacing *adminusername* with your administrator user name, and entering your administrator password when prompted:

```
> auth adminusername
```

4 Remove the group by entering the following command, replacing *officegroup* with the group account's short name:

```
> delete officegroup
```

5 Quit `dscl` by entering:

```
> quit
```

Adding a User to a Group

You can add users to a group using the `dscl` tool.

To add a user to a group:

1. Start the `dscl` tool in interactive mode, specifying the computer you are using as the source of directory service data:

```
$ dscl localhost
>
```

2. Change the current folder to /LDAPv3/*ipaddress*/Groups by entering the path at the prompt:

```
> cd /LDAPv3/ipaddress/Groups
```

   Replace *ipaddress* with the IP address of your directory server.

3. Authenticate as an administrator by entering the following command, replacing *adminusername* with your administrator user name, and entering your administrator password when prompted:

```
> auth adminusername
```

4. Add the user to the group by entering the following command, replacing *groupPath* with the group's path relative to the current folder, and *userName* with the user's short name:

```
> append groupPath GroupMembership userName
```

   For example, if the group's folder is in the /Groups folder, replace *groupPath* with the group's short name. However, if the group's folder is in the /Groups/building1/ folder, replace *groupPath* with building1/*shortName*, where *shortName* is the group's short name.

5. Review the settings of the group by entering the following command, replacing *groupShortName* with the group account's short name:

```
> read groupShortName
```

   `dscl` displays the settings for the group account, similar to the following output:

```
dsAttrTypeNative:apple-generateduid:4B3A5678-E9C1-2EC3-4567-891D234E5678
dsAttrTypeNative:cn: officegroup
dsAttrTypeNative:gidNumber: 600
dsAttrTypeNative:memberUid: mchen ajohnson bmiller
dsAttrTypeNative:objectClass: posixGroup apple-group extensibleObject top
AppleMetaNodeLocation: /LDAPv3/ipaddress
GeneratedUID:4B3A5678-E9C1-2EC3-4567-891D234E5678
GroupMembership: mchen ajohnson bmiller
Member: mchen ajohnson bmiller
PasswordPlus:********
PrimaryGroupID: 600
RecordName: officegroup
RecordType: dsRecTypeStandard:Groups
```

6 Quit `dscl` by entering:

```
> quit
```

To find the GUID of the administrator user admin on the local host:

```
$ dscl localhost
> cd /LDAPv3/127.0.0.1/Users
> read admin GeneratedUID
```

Removing a User from a Group

You can remove users from a group by using the `dscl` tool.

To remove a user from a group:

1 Start the `dscl` tool in interactive mode, specifying the computer you are using as the source of directory service data:

```
$ dscl localhost
>
```

2 Change the current folder to /LDAPv3/*ipaddress*/Groups by entering the path at the prompt:

```
> cd /LDAPv3/ipaddress/Groups
```

Replace *ipaddress* with the IP address of your directory server.

3 Authenticate as an administrator by entering the following command, replacing *adminusername* with your administrator user name, and entering your administrator password when prompted:

```
> auth adminusername
```

4 View the current members of the group by entering the following (replacing *officegroup* with the group account's short name):

```
> read officegroup
```

`dscl` displays the settings for the group account, similar to the following output, where the group named `officegroup` has users `mchen`, `ajohnson`, and `bmiller` as members:

```
dsAttrTypeNative:apple-generateduid:4B3A5678-E9C1-2EC3-4567-891D234E5678
dsAttrTypeNative:cn: officegroup
dsAttrTypeNative:gidNumber: 600
dsAttrTypeNative:MemberUid: mchen ajohnson bmiller
dsAttrTypeNative:objectClass: posixGroup apple-group extensibleObject top
AppleMetaNodeLocation: /LDAPv3/ipaddress
GeneratedUID:4B3A5678-E9C1-2EC3-4567-891D234E5678
GroupMembers:2B3A4567-E8C9-9EC2-3456-789D123E4567 1B2A3456-E7C8-9EC1-2345-
     678D912E3456 8B9A1234-E5C6-7EC8-9123-456D78E9123
GroupMembership: mchen ajohnson bmiller
Member: mchen ajohnson bmiller
PasswordPlus:********
PrimaryGroupID: 600
RecordName: officegroup
RecordType: dsRecTypeStandard:Groups
```

5 Remove the user by entering the following command, replacing *ajohnson* with the short name of the user account, *ajguid* with ajohnson's GUID, and *officegroup* with the short name of the group account:

```
> delete officegroup GroupMembership ajohnson
> delete officegroup GroupMembership ajguid
```

6 Review the new settings of the group:

```
> read officegroup
```

`dscl` displays the settings for the group, showing that the user you removed is no longer a group member, similar to the following output:

```
dsAttrTypeNative:apple-generateduid:4B3A5678-E9C1-2EC3-4567-891D234E5678
dsAttrTypeNative:cn: officegroup
dsAttrTypeNative:gidNumber: 600
dsAttrTypeNative:MemberUid: mchen bmiller
dsAttrTypeNative:objectClass: posixGroup apple-group extensibleObject top
AppleMetaNodeLocation: /LDAPv3/ipaddress
GeneratedUID:4B3A5678-E9C1-2EC3-4567-891D234E5678
GroupMembers:2B3A4567-E8C9-9EC2-3456-789D123E4567 8B9A1234-E5C6-7EC8-9123-
     456D78E9123
GroupMembership: mchen bmiller
Member: mchen bmiller
PasswordPlus:********
PrimaryGroupID: 600
RecordName: officegroup
RecordType: dsRecTypeStandard:Groups
```

7 Quit `dscl` by entering:

```
> quit
```

Creating and Deleting a Nested Group

Nested groups allow for one group (the child) to be a member of a second group (the parent), inheriting the permissions and attributes of the parent group. Members of a nested group become child members of the parent group.

You can create a nested group by using the `dseditgroup` tool with the `-a` option, which adds the group record to the parent group.

To create a nested group:

```
$ dseditgroup -o edit [-a childgroup] [-t group] [-u username] [-P password]
     [-n /LDAPv3/ipaddress] parentgroup
```

Parameter	Description
childgroup	The name of the child group you are adding to the parent group
username	The short name of a user with LDAP directory service access
password	The user password
ipaddress	The IP address of your directory server
parentgroup	The name of the parent group that the child group is being added to

To verify a nested group:

1. Start the `dscl` tool in interactive mode, specifying the computer you are using as the source of directory service data:

   ```
   $ dscl localhost
   >
   ```

2. Change the current folder to /LDAPv3/*ipaddress*/Groups by entering the path at the prompt:

   ```
   > cd /LDAPv3/ipaddress/Groups
   ```

   Replace *ipaddress* with the IP address of your directory server.

3. Authenticate as an administrator by entering the following command, replacing *adminusername* with your administrator user name, and entering your administrator password when prompted:

   ```
   > auth adminusername
   ```

4. View the members of the group by entering the following (replacing *parentgroup* with the group account's short name):

   ```
   > read parentgroup
   ```

   `dscl` displays the settings for the group account, similar to the following output where the group named parentgroup is shown as nested:

   ```
   dsAttrTypeNative:apple-generateduid:4B3A5678-E9C1-2EC3-4567-891D234E5678
   dsAttrTypeNative:apple-group-nestedgroup:1A2B3456-C7D8-9EF1-2345-
        678G912H3456
   dsAttrTypeNative:cn: parentgroup
   dsAttrTypeNative:gidNumber: 700
   dsAttrTypeNative:objectClass: posixGroup apple-group extensibleObject top
   AppleMetaNodeLocation: /LDAPv3/ipaddress
   GeneratedUID:4B3A5678-E9C1-2EC3-4567-891D234E5678
   NestedGroups:1A2B3456-C7D8-9EF1-2345-678G912H3456
   PasswordPlus:********
   PrimaryGroupID: 700
   RecordName: parentgroup
   RecordType: dsRecTypeStandard:Groups
   ```

After a nested group is established, it can be unnested by using the `dseditgroup` tool with the `-d` option, which deletes the group record but leaves the group intact.

To unnest a group:

```
$ dseditgroup -o edit [-d childgroup] [-t group] [-u username] [-P password]
     [-n /LDAPv3/ipaddress] parentgroup
```

Parameter	Description
childgroup	The name of the child group you are adding to the parent group
group	The type of account you are changing (in this case, group)
username	The short name of a user with LDAP directory service access
password	The user password
ipaddress	The IP address of your directory server
parentgroup	The name of the parent group that the child group is being added to

Editing Group Records

To add, remove, or edit group records in the local directory service, use `dsEditGroup`.

To display group information:

```
$ dseditgroup officegroup
```

To delete a group:

```
$ dseditgroup -o delete -p -n /LDAPv3/ipaddress -u diradmin groupname
```

Replace *ipaddress* with the IP address of the DNS name of the LDAPv3 server, *diradmin* with the name of the directory administrator, and *groupname* with the name of the group you want to delete.

The `-p` option prompts you for your diradmin password, which is more secure than putting the password in the command you are sending.

For more information, see the `dseditgroup` man page.

Creating a Group Folder

A group folder facilitates the sharing of files between members of a group. After you set up a group folder in Workgroup Manager, use the `CreateGroupFolder` tool to create the group folder. You should create group folders on the server that hosts these folders.

To create a group folder:

```
$ sudo /usr/bin/CreateGroupFolder
```

For more information, see the `CreateGroupFolder` man page.

Viewing the Workgroup a User Selects at Login

When you define preferences for a group, it is known as a workgroup. A workgroup provides you with a way to manage the working environment of group members.

Preferences you define for a Mac OS X workgroup are stored in the group account. When a user selects a workgroup at login, a property list (plist) file stores the short name of the workgroup in its workgroup key.

Important: You can only view the workgroup a user selects at login on the client computer.

To view the workgroup a user selects at login:

1. Connect to the client computer using an account with administrator privileges.

   ```
   $ ssh admin@computer.name
   ```

   Replace *admin* with the short name of the client computer's administrator and *computer.name* with the IP address or the DNS name of the client computer.

2. Convert the binary com.apple.MCX.plist file to XML format.

   ```
   $ sudo plutil -convert xml1 /Library/Managed Preferences/shortname/
       com.apple.MCX.plist
   ```

   Replace *shortname* with the short name of the logged-in client account.

3. View the workgroup key in /Library/Managed Preferences/shortname/com.apple.MCX.plist file.

   ```
   $ cat /Library/Managed Preferences/shortname/com.apple.MCX.plist
   ```

   Replace *shortname* with the short name of the logged-in client account.

Working with Managed Preferences

To control managed preferences, use MCX extensions with the `dscl` command. You can also use the `mcxquery` command to view effective managed preferences for users, workgroups, and computer groups.

Using MCX Extensions

Although you can use other `dscl` commands to control managed preferences, using MCX command extensions with `dscl` provides an easier way. You can use these extensions in interactive or command-line modes.

The `dscl` command provides the following MCX extensions:

Extension	Description
`-mcxread`	Displays the existing values of an MCX preference key.
`-mcxset`	Sets the value of an MCX preference key.
`-mcxedit`	Updates the value of an MCX preference key.

Extension	Description
`-mcxdelete`	Removes management for the specified MCX preference keys.
`-mcxexport`	Same functionality as the `-mcxread` command, but stores the output in the specified file using the specified format. The resulting file can later be imported using the `-mcximport` command.
`-mcximport`	Imports the keys and values previously exported using the `-mcxexport` command.
`-mcxhelp`	Displays help information for MCX extensions.

Syntax

These command extensions have the following syntax:

```
-mcxread    recordPath [-v mcxVersion]
     [-o filePath] [-format {xml | plist | text}] [appDomain [keyName]]
-mcxset     recordPath [-v mcxVersion] appDomain keyName [mcxDomain
     [keyValue [UPK]]]
-mcxedit    recordPath [-v mcxVersion] appDomain keyPath [keyValue]
-mcxdelete  recordPath [-v mcxVersion] [appDomain [keyName]]
-mcxexport  recordPath [-o filePath] [-format {xml | plist | text}]
     [appDomain [keyName]]
-mcximport  recordPath [-d] filePath
-mcxhelp
```

Parameter	Description
recordPath	The record in the service directory node to be accessed (for example, `/LDAPv3/127.0.0.1/Users/sam`). This parameter is always required, but if you are in interactive mode, you can use a period to represent the current directory.
mcxVersion	The version of the key to be retrieved. If you omit this parameter, the command searches for version 1 keys.
-format	The format of the output file (XML, plist, or text).
optArgs	(Optional) One or more options.
appDomain	(Optional) An application's domain. For example, the application domain for the Dock is `com.apple.dock`.
keyName	(Optional) The name of the managed preference (for example, `familyControlsEnabled`, `mcx_emailAddress`, and `mcx_defaultWebBrowser`).
mcxDomain	(Optional) The type of management applied to the key. Legal values are: • `none` (not managed) • `always` • `once` • `often` • `unset`

Parameter	Description
keyValue	(Optional) The new value to be used for a key. You can specify this parameter using the same syntax as that of the `defaults` command. For more information, see the man page of the `defaults` command. When specifying plist or xml values, enclose the parameter in single quotes (for example, `'(authenticate, eject)'` and `'<real>64.0</real>'`).
UPK	(Optional) The value for the Union Policy Key (UPK). If present, the UPK *must* be specified as a dictionary. The valid keys for the dictionary include: • `mcx_input_key_names` or `input` (single string or array of strings) • `mcx_output_key_names` or `output` (single string) • `mcx_remove_duplicates` (boolean) • `mcx_union_as_dictionary` (boolean) • `mcx_replace` (boolean) If `mcx_input_key_names` or `mcx_output_key_name` is omitted, the value of *keyName* is used instead.
keyPath	(Optional) The path to a sub-plist in an existing key value. For example, `'mount-controls:dvd:1'` means the second element the array with the key `'dvd'` the key `'mount-controls.'`
filePath	(Optional) The location of the output or input file.
`-d`	The keys found in the import file from the record that should be deleted. This is equivalent to calling `-mcxdelete` for every key in the import file. The value of the key in the import file is ignored.

Example

The following command sets the `autohide` key in the `com.apple.dock` domain to a value of `TRUE` with `always` for management.

```
$ dscl -mcxset /LDAPv3/127.0.0.1/Users/sam com.apple.dock autohide always
     -bool 1
```

The following command removes preference management for the `autohide` key in the `com.apple.dock` domain for the current record:

```
$ dscl
> cd /LDAPv3/127.0.0.1/Users/sam
/LDAPv3/127.0.0.1/Users/sam > mcxset . com.apple.dock autohide none
```

The following command displays, in plist format, all keys for all application domains for the current record:

```
$ dscl
> cd /LDAPv3/127.0.0.1/Users/sam
/LDAPv3/127.0.0.1/Users/sam > mcxread . -format plist = =
```

The following command changes the autohide key to `TRUE`, preserving the current management setting:

```
$ dscl
> cd /LDAPv3/127.0.0.1/Users/sam
/LDAPv3/127.0.0.1/Users/sam > mcxedit . com.apple.dock autohide -boot 1
```

The following command causes the `autohide` Dock key to no longer be managed:

```
$ dscl
> cd /LDAPv3/127.0.0.1/Users/sam
/LDAPv3/127.0.0.1/Users/sam > mcxdelete . com.apple.dock autohide
```

The following command exports the keys in the `com.apple.dock` domain for the current record to the /tmp/export.plist file:

```
$ dscl
> cd /LDAPv3/127.0.0.1/Users/sam
/LDAPv3/127.0.0.1/Users/sam > mcxexport . -o /tmp/export.plist
     com.apple.dock
```

The following command imports the keys in the /tmp/export.plist file into the current directory:

```
$ dscl
> cd /LDAPv3/127.0.0.1/Users/sam
/LDAPv3/127.0.0.1/Users/sam > mcximport . /tmp/export.plist
```

For more examples, use the `mcxhelp` extension.

Determining Effective Managed Preferences

Workgroup Manager allows you to configure managed preferences at the user, workgroup, and computer level. Determining the effective managed preferences that determine a user's computer experience is not easy, especially if the managed user is a member of many managed workgroups, and each workgroup is a member of a different computer group.

To simplify the process of determining effective managed preferences, Mac OS X Server provides the `mcxquery` command. You can use this command to determine the effective managed preferences for user, workgroup, or computer group records.

Syntax

```
$ mcxquery options -user userName -group groupName -computer computerName
```

Parameter	Description
options	(Optional) Two options for specifying the name and format of the file where the results of the query (the effective managed preferences) are stored: • `-o fileName`: The name of the output file (including the path) where the results of running this command are stored. • `-format {space | tab | xml}`: The format of the output, which can be space-delimited, tab-delimited, or XML.
userName	(Optional) The short name of a user. If you do not provide the short name for this option or use the equal sign (=), this command uses the short name of the logged in console user.

Parameter	Description
groupName	(Optional) The short name of a workgroup. A value of = indicates the workgroup (if any) chosen for the current login session.
computerName	(Optional) The short name of the computer group or the MAC address of a computer. If you do not provide a value for this option or use the equal sign (=), this command uses the MAC address of the current computer.

Examples

The following example displays the managed preferences for Sam and stores the results in XML format in the samPrefs.out file:

```
$ mcxquery -o samPrefs.out -user sam
```

The following example displays the managed preferences for Jane, who is logged in using the science workgroup from a computer that is a member of the lab1_12 computer group:

```
$ mcxquery -user jane -group science -computer lab1_12
```

The following example displays the managed preferences for Jane, who is logged in using the science workgroup from the computer whose Ethernet MAC address is 11:22:33:44:55:66:

```
$ mcxquery -user jane -group science -computer 11:22:33:44:55:66
```

Importing Users and Groups

To import user and group accounts into a folder, use `dsimport`. The `dsimport` tool permits logging at three levels with the `-l` switch. You can use the `dsimport` tool to import records from a flexible text–delimited file.

For more information, see the `dsimport` man page. For a list of record types and attributes, see *Open Directory Administration*. This guide also describes how to edit permitted attributes for each record type for use in an LDAP folder.

The `dsimport` tool is located in /usr/bin/.

For information about the formats of the files you can import, see "Creating a Character-Delimited User Import File" on page 123.

```
$ dsimport (-g|-s|-p) filepath DSNodePath (O|M|I|A|N) -u user -p password
     [options]
```

Parameter	Description
`-g|-s|-p`	Specify one of these to indicate the type of file you're importing: `-g` for a character-delimited file `-s` for an XML file exported from Users & Groups in Mac OS X Server v10.1.x `-p` for an XML file exported from AppleShare IP v6.x
filepath	The path of the file to import.

Parameter	Description
DSNodePath	The path to the Open Directory server node where the imported records will be added.
O|M|I|A|N	Specifies how user data is handled if a record for an imported user exists in the folder: • O: Overwrite the matching record. • M: Merge the records. Empty attributes in the folder and assume values from the imported record. • I: Ignore imported record and leave the record unchanged. • A: Append data from an import record to an existing record. • N: Do not check for duplicates.
user	The name of the Open Directory domain administrator.
password	The password of the Open Directory domain administrator.
options	Additional command options. To see available options, execute the `dsimport` command with no parameters.

To import users and groups:

1. Create a file containing the accounts to import, and place it in a location accessible from the importing server.

   You can export this file from an earlier version of Mac OS X Server or AppleShare IP 6.3, or create your own character-delimited file. See "Creating a Character-Delimited User Import File" on page 123.

   Open Directory supports up to 200,000 records.

2. Log in as the administrator of the directory domain you want to import accounts into.

3. Use the `dsimport` tool to import users and groups.

   For example, to import a file generated by Workgroup Manager named "sample" and export it into the LDAPv3 directory located at 192.168.2.2, use the following command:

   ```
   $ dsimport -g sample /LDAPv3/192.168.2.2 -O -u diradmin
   ```

   Replace *diradmin* with the short name of the directory administrator. When two records match, the import file overwrites the matching record.

4. To create home folders for imported users, use `createhomedir`.

   See "Creating a User's Home Folder" on page 109.

Creating a Character-Delimited User Import File

You can create a character-delimited file by using Workgroup Manager or `dsimport` to export accounts in the LDAP directory of an Open Directory master. You can also create a character-delimited file by hand, using a script, or by using a database or spreadsheet application.

The first record in the file, the record description, describes the format of each account record in the file. There are three options for the record description:

- Write a full record description
- Use the shorthand `StandardUserRecord`
- Use the shorthand `StandardGroupRecord`

The other records in the file describe user or group accounts, encoded in the format described by the record description. A line in a character-delimited file that begins with # is ignored during importing.

Writing a Record Description

The record description specifies the fields in each record in the character-delimited file, specifies the delimiting characters, and specifies the escape character that precedes special characters in a record.

Encode the record description using the following elements in the order specified, separating them with a space:

- End-of-record indicator (in hex notation)
- Escape character (in hex notation)
- Field separator (in hex notation)
- Value separator (in hex notation)
- Type of accounts in the file (`dsRecTypeStandard:Users` or `dsRecTypeStandard:Groups`)
- Number of attributes in each account record
- List of attributes

For user accounts, the list of attributes must include the following, although you can omit UID and PrimaryGroupID if you specify a starting UID and a default primary group ID when you import the file:

- RecordName (the user's short name)
- Password
- UniqueID (the UID)
- PrimaryGroupID
- RealName (the user's full name)

In addition, you can include:

- UserShell (the default shell)
- NFSHomeDirectory (the path to the user's home folder)
- Other user data types, described in Open Directory Administration.

For group accounts, the list of attributes must include:

- RecordName (the group name)
- PrimaryGroupID (the group ID)
- GroupMembership

The following is an example of a record description:

```
0x0A 0x5C 0x3A 0x2C dsRecTypeStandard:Users 7
RecordName Password UniqueID PrimaryGroupID
RealName NFSHomeDirectory UserShell
```

The following is an example of a record encoded using the previous description:

```
anne:Adl47E$:408:20:A. Johnsons, M.D.:/Network/Servers/somemac/Homes/anne:/
     bin/csh
```

The record consists of values, delimited by colons. Use a double-colon (`::`) to indicate that a value is missing.

The following is another example, which shows a record description and user records for users whose passwords are to be validated using the Password Server. The record description should include a field named `dsAttrTypeStandard:AuthMethod`, and the value of this field for each record should be `dsAuthMethodStandard:dsAuthClearText`:

```
0x0A 0x5C 0x3A 0x2C dsRecTypeStandard:Users 8
dsAttrTypeStandard:RecordName dsAttrTypeStandard:AuthMethod
dsAttrTypeStandard:Password dsAttrTypeStandard:UniqueID
dsAttrTypeStandard:PrimaryGroupID dsAttrTypeStandard:Comment
dsAttrTypeStandard:RealName dsAttrTypeStandard:UserShell
skater:dsAuthMethodStandard\:dsAuthClearText:pword1:374:11:comment:
Tony Hawk:/bin/csh
mattm:dsAuthMethodStandard\:dsAuthClearText:pword2:453:161::
Matt Mitchell:/bin/tcsh
```

As these examples illustrate, you can use the prefix `dsAttrTypeStandard:` when referring to an attribute, or you can omit the prefix. When you use Workgroup Manager to export character-delimited files, it uses the prefix in the generated file.

When importing user passwords, you can insert the following in the list of attributes to set the user's password type to Open Directory:

```
dsAttrTypeStandard:AuthMethod
```

Then, insert the following in the formatted record (in this example, the user 's password is "password"):

```
dsAuthMethodStandard\:dsAuthClearText:password
```

Note: In this example, the colon (:) is the field separator. Because there is a colon in the description for this attribute, the escape character must be used to indicate that the colon should not be treated as a delimiter. The backslash (\) is the escape character in this example. If the field separator is anything other than the colon, the escape character is not needed.

The method for setting an imported user's password type to Open Directory requires that the imported data has a password value. If the password value is missing for a user, the corresponding user record is created with a password type of Crypt or Shadow Password.

Before importing user accounts, remember to manually set passwords or set default passwords to a known value. After importing user records, you can set up a password policy that requires users to change their password at first login.

Note: Importing passwords generally works only if the password is a plain text string in the import file. Additionally, you need to set the AuthMethod attribute so that `dsimport` can import the password. Encrypted passwords that are in hash format in the import file cannot be recovered. Also, passwords cannot be exported using Workgroup Manager or any other method.

Using StandardUserRecord Shorthand

When the first record in a character-delimited import file contains `StandardUserRecord`, the following record description is assumed:

```
0x0A 0x5C 0x3A 0x2C dsRecTypeStandard:Users 7
RecordName Password UniqueID PrimaryGroupID
RealName NFSHomeDirectory UserShell
```

An example user account looks like this:

```
anne:Ad147E$:408:20:A. Lo, M.D.:/Network/Servers/somemac/Homes/anne:/bin/csh
```

Using StandardGroupRecord Shorthand

When the first record in a character-delimited import file contains `StandardGroupRecord`, the following record description is assumed:

```
0x0A 0x5C 0x3A 0x2C dsRecTypeStandard:Groups 4
RecordName Password PrimaryGroupID GroupMembership
```

The following is an example of a record encoded using the description:

```
students:Ad147:88:johnson,miller,clark,chen,wong
```

Exporting Users and Groups

To export records from Open Directory use `dsexport`.

The `dsexport` tool is in the /usr/bin/ folder.

```
$ dsexport filePath DSNodePath recordType options DSProxy
```

Parameter	Description
filepath	The name (including the path) of the file to export.
DSNodePath	The path to the Open Directory server node to export records from.
recordType	(Optional) The type of record to be exported from the Open Directory server node.
options	Additional command options. To see available options, execute the `dsexport` command with no parameters. Also, see the command's man page.
DSProxy	(Optional) A set of options for connecting to a proxy system. All options are needed. If you do not specify the password as an argument, the tool prompts you for it. These options are: • `-a proxyAddress`: The address of the proxy machine the user wants to use. • `-u proxyUser`: The username to use for the proxy connection. • `-p proxyPassword`: The password to use for the proxy connection.

For example, use the following to export user records from the local Open Directory server node and store the exported data in the exportedUserRecords.out file:

```
$ dsexport exportedUserRecords.out /Local/Default dsRecTypeStandard:Users
```

Use the following to export group records for admin and staff from the LDAPv3 node on the proxy system (proxy.machine.com) to the exportedGroupRecords.out file:

```
$ dsexport exportedGroupRecords.out /LDAPv3/127.0.0.1
     dsRecTypeStandard:Groups -r admin, staff -a proxy.machine.com
     -u diradmin -p pass
```

Setting Permissions

To control access to your information, Mac OS X sets permissions for disks, folders, and files. You can only change permissions to items that you own.

Be sure that the default permissions are appropriate. For most purposes, files should be accessible to other members of your group. If you have private or confidential information, the default permissions of the files may allow others to see it.

To prevent others from accessing personal information, create a folder and set its permissions to "owner"; then place your confidential files into it. No other users are allowed to access the folder.

Mac OS X provides distinct permissions for these types of users:

- The owner of the item, who is usually the person who created the item
- Any member of the group assigned to the item by Mac OS X
- Any other user with access to the computer

These are the levels of permission:

- *Read & Write*, which allows a user to open the item to see its contents and change it.
- *Read Only*, which allows a user to open the item to see its contents, but not change or copy the contents.
- *Write Only*, which makes a folder into a drop box. Users can copy items to the drop box but cannot open the drop box to see its contents. Only the owner of the drop box can open it to access items.
- *No Access*, which blocks access to the item so users can't open the item, change its contents, or copy its contents.

Viewing Permissions

Each security group is assigned a code that controls that group's permissions:

- r (read) allows the user to see the item but not make changes.
- w (write) allows the user to see and make changes to the item.
- x (execute) allows the user to run scripts or programs.
- - (access) means access is turned off.

To view permissions for files and folders, enter the `ls -l` command. For each file or folder listed, you see the permissions, owner and group name, and file or folder name.

Examples of permission settings:

Following are examples of permission settings:

- The following file (-) displays read, write, and executable permissions for owner (rwx), group (rwx) and all others (rwx):

  ```
  -rwxrwxrwx
  ```

- The following file (-) displays read, write, and executable permissions for owner (rwx), and group (rwx), but no permissions for others (---):

  ```
  -rwxrwx---
  ```

- The following file (-) displays read, write, and executable permissions for owner (rwx), but no permissions for group (---) or others (---):

  ```
  -rwx------
  ```

- The following file (-) displays read and write, but no executable permissions for owner (rw-), group (rw-), and others (rw-):

  ```
  -rw-rw-rw-
  ```

- The following file (-) displays read, write, and executable permissions for owner (rwx), but only read and executable for group (r-x) and others (r-x):

```
-rwxr-xr-x
```

- The following file (-) displays read, write, and executable permissions for owner (rwx), but only read for group (r--) and others (r--):

```
-rwxr--r--
```

For more information, see the `ls` man page.

Setting the umask Setting for a User

The global umask setting determines the permissions of files and folders created by a local user:

```
$ sudo defaults write -g NSUmask -int value
```

Use one of the following values to set the permission level:

Value	Permission Level
63 (octal equivalent 077)	Only the user can read files.
23 (octal equivalent 027)	The user and members of the user's default group can read files.
18 (octal equivalent 022)	All users can read newly created files.

The default umask setting, 022, removes group and world write permissions but allows group and world read permissions.

With a umask setting of 027, files and folders created by a user are not readable by other users on the computer, but they are readable by members of the user's assigned group. To make a file or folder accessible to others, the owner can by change the permissions in the Finder's Get Info window or use the `chmod` tool.

To set the umask settings for local users to octal 027 (decimal equivalent 23):

```
$ sudo defaults write /Library/Preferences/.GlobalPreferences NSUmask 23
```

Note: The path above refers to the .GlobalPreferences defaults domain, not to the file .GlobalPreferences.plist, which might accidentally be filled in while using the shell autocomplete feature.

This command affects the permissions on files and folders created by programs that respect the Mac OS X NSUmask settings. Programs should follow the value set for umask, but there is no guarantee that they will. Also, users can override their own umask setting at any time. The changes to the umask settings take effect at next login.

WARNING: Setting permissions to group, or all, allows private or confidential information in these folders to be visible to others. To prevent private files from being accessed, the user should create a folder and restrict the permissions.

Changing Permissions

To change permissions for an item, use the `chmod` tool.

```
$ chmod securitygroup changetype permission fileorfolder
```

Parameter	Description
securitygroup	The person or group whose permission you are changing. Can be the following: • u—user • g—group • o—other • all—all
changetype	Type of change. To add or subtract the permission, use: • "+" —add permission • "-"—subtract permission
permission	The permission you are changing: • r—read • w—write • x—execute
fileorfolder	The name of the file or folder to change.

To remove the write access permission for group and other from the file myfile:

```
$ chmod go-w myfile
```

To add read and write access permissions for group and other to files myfile1 and myfile2:

```
$ chmod go+rw myfile1 myfile2
```

To add read, write, and execute permissions for all to myfile1:

```
$ chmod ugo+rwx myfile1
```

For more information, see the `chmod` man page.

Changing the Owner

To change the owner of a file or folder, use the `chown` tool.

```
$ chown username fileorfolder
```

Parameter	Description
username	The user who will become the owner of the file.
fileorfolder	The name of the file or folder to change.

To change the owner of file1 to the user jdoe:

```
$ chown jdoe file1
```

For more information, see the `chown` man page.

Changing the Group

To change the group of a file or folder, use the `chgrp` tool.

```
$ chgrp groupname fileorfolder
```

Parameter	Description
groupname	The group that will become associated with the file or folder.
fileorfolder	The name of the file or folder to change.

To change the group of file1 and file2 to the group ateam:

```
$ chgrp ateam file1 file2
```

For more information, see the `chgrp` man page.

Securing System Accounts

The following sections cover security settings for user accounts.

Securing Initial System Accounts

Two accounts on the computer require attention before further configuration:

- The permissions on the home folder of the initial administrator account should be changed.
- Necessary modifications to the root account should be performed.

To secure initial system accounts, the permissions on the home folder of the initial administrator account should be changed to allow only administrator access.

The permissions on the home folder of the just-created administrator account allow any user who logs in to the computer to browse its contents.

To change permissions on the administrator's home folder:

```
$ chmod 700 /Users/adminname
```

Replace *adminname* with the name of the account. The 700 permission setting allows only the administrator to read and browse files in the administrator's home folder.

Securing the Root Account

Mac OS X Server includes a root account like other UNIX-based systems. Initially, its password is set to that of the first administrator account.

Direct root login should not be allowed, because the logs cannot identify which administrator logged in. Instead, accounts with administrator privileges should be used for logging in, and then the `sudo` tool should be used to perform actions with root privileges.

The computer uses a file called /etc/sudoers to determine which users have the authority to use the `sudo` program. This file initially specifies that all accounts with administrator privileges can use `sudo`.

To disable root login:

1. Start the `dscl` tool in interactive mode, specifying the computer you are using as the source of directory service data:

```
$ dscl localhost
>
```

2. Change the current folder to /Local/Users by entering the path at the prompt:

```
> cd /Local/Users
```

3. Authenticate as an administrator by entering the following command, replacing *adminusername* with your administrator user name, and entering your administrator password when prompted:

```
> auth adminusername
```

4. Use the following commands to disable the root login by removing the `AuthenticationAuthority` property and its value, and modifying the root password property.

```
> delete root AuthenticationAuthority ;ShadowHash;
> delete root AuthenticationAuthority
```

Any user with administrative privileges can reenable root login by entering `passwd root` in a Terminal window.

Restricting Use of the sudo Tool

Limit the list of administrators allowed to use the `sudo` tool to those administrators who require the ability to run commands with root user privileges.

To change the /etc/sudoers file:

1. Edit the /etc/sudoers file using the `visudo` tool, which allows for safe editing of the file.

   Run the following command with root user privileges:

```
$ sudo visudo
```

2. When prompted, enter your administrator password.

   There is a timeout value associated with the `sudo` tool. This value indicates the number of minutes until the `sudo` tool prompts for a password again.

   The default value is 5, which means that after issuing the `sudo` command and entering the correct password, additional `sudo` commands can be entered for 5 minutes without re-entering the password. This value is set in the /etc/sudoers file.

   For more information, see the `sudo` and `sudoers` man pages.

3. In the Defaults specification section of the file, add the following line:

```
Defaults timestamp_timeout=0
```

4 Restrict which administrators are allowed to run the `sudo` tool by removing the line that begins with `%admin` and adding the following entry for each user, substituting the user's short name for the word *user*:

```
user ALL=(ALL) ALL
```

Doing this means that any time an administrator is added to a system, the administrator must be added to the /etc/sudoers file as described above if that administrator needs to use the `sudo` tool.

5 Save and quit `visudo`.

For more information, see the `vi` and `visudo` man pages.

Securing Single-User Boot

On Apple computers running Mac OS X, Open Firmware is the software executed immediately after the computer is powered on. This boot firmware is analogous to the BIOS on an x86-based PC.

To prevent users from obtaining root access by booting into single user mode or booting from other disks, alter the Open Firmware settings. For desktop computers, the Open Firmware security mode should be set to command. To configure the Open Firmware settings, use the `nvram` tool.

To set the variable security mode:

```
$ nvram security-mode="command"
```

In command mode, the computer boots from the boot device specified in the computer's boot device variable and disallows users from providing boot arguments.

To verify that the computer is in command mode as recommended:

1 Close all applications and choose Restart from the Apple menu.

A confirmation window appears. Restart the computer by clicking the Restart button.

2 Hold down the key combination Command-S while the computer boots.

If the command mode has been set correctly, the computer displays the Mac OS X login window. Normally, holding down the Command-S key combination while starting up causes the computer to start up in single-user mode.

3 If the computer started up in single-user mode, restart the computer by issuing the command `reboot`; then repeat the previous steps for putting the computer into command mode.

Open Firmware protection can be violated if the user has physical access to the computer or if the user changes the physical memory configuration of the computer and then resets the PRAM 3 times (holding down Option-P-R during boot). This disables the Open Firmware password.

Note: An Open Firmware password provides some protection, but it can be reset if a user has physical access to the computer and can change the physical memory configuration of the computer.

To set the Open Firmware password for increased security:

1. Boot the computer while holding Command-Option-O-F (all four keys at the same time) to enter the Open Firmware command prompt.
2. At the prompt, enter the command:

   ```
   > password
   ```

3. Enter and verify the password to be used as the Open Firmware password.

   This password is limited to eight characters. Choose a strong password. In this instance, a computer-generated random password is a good choice.

   This password should be recorded and secured in the same location as the Master FileVault password.

   This password is not needed except when the computer must be booted from an alternate disk, such as if the startup disk fails or its file system needs of repair.

4. To restart the computer and enable the settings, enter the command:

   ```
   > reset-all
   ```

   The computer should restart and display the login window.

Setting Password Policy

To adjust the password policies of your users, us the `pwpolicy` tool. You can use this tool to:

- View or set global password policies that force users to change passwords
- Limit the number and type of characters in a password
- Limit the length of time before passwords can be reused
- Limit when passwords must be changed

For secure passwords, you should require every password to have a minimum of 5 characters. You can use a higher number of characters if you want a more secure password. It is also good to have users change passwords frequently.

For more information, see the `pwpolicy` man page.

To change a user's password:

```
$ pwpolicy -n /LDAPv3/ipaddress -a adminusername -u usertochange
     -setpassword newpassword
```

Parameter	Description
ipaddress	Location of the LDAP directory
adminusername	User name of an administrator
usertochange	Name of the user whose password is changing
newpassword	Password the user is changing to

To view the global password policy:

```
$ pwpolicy -getglobalpolicy
```

To set the minimum password length to 5 characters:

```
$ pwpolicy -n /LDAPv3/ipaddress -a adminusername -setglobalpolicy
     "minChars=5"
```

Parameter	Description
ipaddress	Location of the LDAP directory
adminusername	User name of an administrator
minChars	Minimum number of characters in the password

To set a more secure global password policy:

```
$ pwpolicy -n /LDAPv3/ipaddress -a adminusername -setglobalpolicy
     "minChars=6 usingHistory=4 requiresNumeric=1
     maxMinutesUntilChangePassword=43200"
```

This sets the global password policy for users and requires the following:

- The password must have a minimum of six characters.
- The users cannot reuse a password from the previous four passwords.
- The password must contain at least one number.
- The password must be changed every 30 days.

Parameter	Description
ipaddress	Location of the LDAP directory
adminusername	User name of an administrator
minChars	Minimum number of characters in the password
usingHistory	Number of previous passwords the user cannot reuse
requiresNumeric	Number of numeric characters that must be in the password
maxMinutesUntilChangePassword	Number of minutes until a password must be changed

To set the password policy of a user to require that they change their password:

```
$ pwpolicy -n /LDAPv3/ldap.apple.com -a adminusername -p adminpassword
     -u usertochange -setpolicy "newPasswordRequired=1"
```

Parameter	Description
ldap.apple.com	Location of the LDAP directory.
adminusername	User name of an administrator.
adminpassword	Administrator password. (Omit to prompt for the password.)
usertochange	User name of the user whose password is changing.
newPasswordRequired	A value of 1 prompts the user to enter a new password.

Finding User Account Information

Use the `dscacheutil` tool to gather information and statistics by querying the Directory Service cache. You can also interactively use it to find out user account information.

To view a user's account information:

```
$ dscacheutil -q user -a name jdoe
name: jdoe
password: ********
uid: 501
gid: 501
dir: /Users/jdoe
shell: /bin/csh
gecos: John Doe
```

To view all user accounts:

```
$ dscacheutil -q user
```

For more information about `dscacheutil`, see its man page.

Working with File Services

9

Use this chapter to learn the commands to create share points and manage file services.

This chapter covers the commands used to configure and manage these file services.

Mac OS X Server allows you to set up central network storage that is accessible to clients throughout your organization. Using native protocols, it delivers the following file services to heterogeneous clients on your network:

- Apple Filing Protocol (AFP) for Mac
- Network File System (NFS) for UNIX and Linux
- Server Message Block (SMB) for Windows
- WebDAV and FTP for Internet clients

For more information about file services, see *File Services Administration*.

Managing Share Points

A share point is a folder, hard disk, hard disk partition, CD, or DVD that users can access over the network to share information. Users with access privileges, which are assigned, view share points as mounted volumes.

Mac OS X Server supports Microsoft Windows file sharing of any defined share point, not just Shared and Public folders in a user's Home folder. It also supports Windows Internet Naming Service (WINS), which allows Windows clients across multiple subnets to perform name/address resolution.

To list, create, modify, and disable share points, use the `sharing` tool described in the following sections.

To set space quotas for share points, use the `edquota` command.

For more information, see the `sharing` and `edquota` man pages.

Listing Share Points

To list share points:

```
$ sudo sharing -l
```

In the resulting list is a section of properties similar to the following for each share point defined on the server (`1` = yes, true, or enabled; `0` = false, no, or disabled).

```
name:           Share1
path:           /Volumes/100GB
        afp:    {
                name:   Share1
                shared: 1
                guest access:   0
                inherit perms:  0
        }
        ftp:    {
                name:   Share1
                shared: 1
                guest access:   1
        }
        smb:    {
                name:   Share1
                shared: 1
                guest access:   1
                inherit perms:  0
                oplocks:        0
                strict locking: 0
                directory mask: 493
                create mask:    420 }
```

Creating a Share Point

To create a share point:

```
$ sudo sharing -a path [-n customname] [-A afpname] [-F ftpname]
     [-S smbname] [-s shareflags] [-g guestflags] [-i inheritflags]
     [-c creationmask] [-d directorymask] [-o oplockflag]
     [-t strictlockingflag]
```

Parameter	Description
path	The full path to the folder you want to share.
customname	The name of the share point. If you don't specify the custom name, it's set to the name of the folder, the last name in *path*.
afpname	The share point name shown to and used by AFP clients. This name is not the same as the share point name.
ftpname	The share point name shown to and used by FTP clients.
smbname	The share point name shown to and used by SMB clients.
shareflags	A three-digit binary number indicating the protocols used to share the folder. The digits represent, from left to right, AFP, FTP, and SMB. `1`=shared, `0`=not shared.

Parameter	Description
guestflags	A group of flags indicating which protocols allow guest access. The flags are written as a three-digit binary number with the digits representing, from left to right, AFP, FTP, and SMB. `1`=guests allowed, `0`=guests not allowed.
inheritflags	A group of flags indicating whether new items in AFP or SMB share points inherit the ownership and access permissions of the parent folder. The flags are written as a two-digit binary number with the digits representing, from left to right, AFP and SMB. `1`=inherit, `0`=don't inherit.
creationmask	The SMB creation mask. Default=`0644`.
directorymask	The SMB folder mask. Default=`0755`.
oplockflag	A parameter that specifies whether opportunistic locking is allowed for an SMB share point. 1=enable oplocks, 0=disable oplocks. For more information about oplocks, see *File Services Administration*.
strictlockingflag	A parameter that specifies whether strict locking is used on an SMB share point. `1`=enable strict locking, `0`=disable. For more information about strict locking, see *File Services Administration*.

To create a share point that uses AFP, FTP, and SMB:
Enter the following command, replacing *100GB* with the name of the volume containing the share point and *Archive* with the share point name:

```
$ sudo sharing -a /Volumes/100GB/Archive
```

To create a share point that appears differently for different users:
Enter the following command, replacing *100GB* with the name of the volume containing the share point and *Windows* with the share point name so that it appears as WinDocs for server management purposes, and Documents for SMB file service users:

```
$ sudo sharing -a /Volumes/100GB/Windows\ Docs -n WinDocs -S Documents
     -s 001 -o 1
```

This share point is shared using only SMB with oplocks enabled.

Modifying a Share Point

To change share point settings:

```
$ sudo sharing -e sharepointname [-n customname] [-A afpname] [-F ftpname]
     [-S smbname] [-s shareflags] [-g guestflags] [-i inheritflags]
     [-c creationmask] [-d directorymask] [-o oplockflag]
     [-t strictlockingflag]
```

Parameter	Description
sharepointname	The current name of the share point.
Other parameters	See the parameter descriptions in "Creating a Share Point" on page 138.

Disabling a Share Point

To disable a share point:

```
$ sudo sharing -r sharepointname
```

Parameter	Description
sharepointname	The current name of the share point.

Setting Disk Quotas

You can use the `edquota` command to set disk quotas for users and groups.

For more information about this command, see its man page.

To set disk quotas for users on a share point:

```
$ sudo edquota -u -p proto-username username ...
```

Parameter	Description
proto-username	The user whose disk quota will be duplicated to other users.
username	The user whose disk quota should be set to the same quota as *proto-username*.

To set disk quotas for groups on a share point:

```
$ sudo edquota -u -p proto-groupname groupname ...
```

Parameter	Description
proto-groupname	The group whose disk quota will be applied to other groups.
groupname	The group whose disk quota should be set to the same quota as *proto-groupname*.

To set the grace period for enforcing disk quotas for users:

```
$ sudo edquota -t -u
```

You specify the default grace period in /usr/include/sys/quota.h. For a user, you specify the grace period in the file .quota.ops.user located at the root of the user's mounted file system.

To set the grace period for enforcing disk quotas for groups:

```
$ sudo edquota -t -g
```

For a group, you specify the grace period in the file .quota.ops.group located at the root of the group's mounted file system.

Managing AFP Service

AFP allows any Mac OS X computer to access shared folders on the server. Mac OS X Server uses Bonjour to provide automatic discovery of AFP file services, and to prevent shared disks from unmounting after extended periods of inactivity.

Starting and Stopping AFP Service

To start AFP service:

```
$ sudo serveradmin start afp
```

To stop AFP service:

```
$ sudo serveradmin stop afp
```

Viewing AFP Service Status

To see if AFP service is running:

```
$ sudo serveradmin status afp
```

To see complete AFP status:

```
$ sudo serveradmin fullstatus afp
```

To list a setting:

```
$ sudo serveradmin settings afp setting
```

Parameter	Description
setting	Any AFP service setting. For a complete list of settings, enter `$ sudo serveradmin settings afp` or see "Available AFP Settings" on page 142.

To view a group of settings:

You can view a group of settings that have part of their names in common by entering as much of the name as you want, stopping at a colon (`:`), and entering an asterisk (`*`) as a wildcard for the remaining parts of the name. For example:

```
$ sudo serveradmin settings afp:loggingAttributes:*
```

Viewing all AFP Settings

To view all AFP service settings:

```
$ sudo serveradmin settings afp
```

Changing AFP Settings

You can change AFP service settings using the `serveradmin` tool.

To change a setting:

```
$ sudo serveradmin settings afp:setting = value
```

Parameter	Description
setting	An AFP service setting. To see a list of available settings, enter `$ sudo serveradmin settings afp` or see "Available AFP Settings" on page 142.
value	An appropriate value for the setting. Enclose text strings in double quotes (for example, `"text string"`).

To change several settings:

```
$ sudo serveradmin settings
afp:setting = value
afp:setting = value
afp:setting = value
[...]
Control-D
```

Available AFP Settings

The following table lists AFP settings as they appear using `serveradmin`.

Parameter (`afp:`)	Description
`activityLog`	Turn activity logging on or off. Default = `no`
`activityLogPath`	Location of the activity log file. Default = `/Library/Logs/AppleFileService/AppleFileServiceAccess.log`
`activityLogSize`	Rollover size (in kilobytes) for the activity log. Used only if `activityLogTime` isn't specified. Default = `1000`
`activityLogTime`	Rollover time (in days) for the activity log. Default = `7`
`admin31GetsSp`	Set to yes to force administrator users on Mac OS X to see share points instead of volumes. Default = `yes`
`adminGetsSp`	Set to yes to force administrator users on Mac OS 9 to see share points instead of volumes. Default = `no`
`afpServerEncoding`	Encoding used with Mac OS 9 clients. Default = `0`
`afpTCPPort`	TCP port used by AFP on server. Default = `548`

Parameter (`afp:`)	Description
`allowRootLogin`	Allow user to log in as root. Default = `no`
`attemptAdminAuth`	Allow administrator user to masquerade as another user. Default = `yes`

`authenticationMode`	Authentication mode. Can be:
	`standard`
	`kerberos`
	`standard_and_kerberos`
	Default = `"standard_and_kerberos"`

`autoRestart`	Allow the AFP service to restart automatically when abnormally terminated. Default = `yes`
`clientSleepOnOff`	Allow client computers to sleep. Default = `yes`
`clientSleepTime`	Time (in hours) that clients are allowed to sleep. Default = `24`
`createHomeDir`	Create home folders. Default = `yes`
`errorLogPath`	Location of the error log. Default = `/Library/Logs/AppleFileService/AppleFileServiceError.log`
`errorLogSize`	Rollover size (in kilobytes) for the error log. Use only if `errorLogTime` isn't specified. Default = `1000`
`errorLogTime`	Rollover time (in days) for the error log. Default = `0`
`guestAccess`	Allow guest users access to the server. Default = `yes`
`idleDisconnectFlag: adminUsers`	Enforce idle disconnect for administrator users. Default = `yes`
`idleDisconnectFlag: guestUsers`	Enforce idle disconnect for guest users. Default = `yes`
`idleDisconnectFlag: registeredUsers`	Enforce idle disconnect for registered users. Default = `yes`
`idleDisconnectFlag: usersWithOpenFiles`	Enforce idle disconnect for users with open files. Default = `yes`
`idleDisconnectMsg`	Idle disconnect message. Default = `""`
`idleDisconnectOnOff`	Enable idle disconnect. Default = `no`

Parameter (afp:)	Description
idleDisconnectTime	Idle time (in minutes) allowed before disconnect. Default = 10
kerberosPrincipal	Kerberos server principal name. Default = "afpserver"
loggingAttributes: logCreateDir	Record folder creations in the activity log. Default = yes
loggingAttributes: logCreateFile	Record file creations in the activity log. Default = yes
loggingAttributes: logDelete	Record file deletions in the activity log. Default = yes
loggingAttributes: logLogin	Record user logins in the activity log. Default = yes
loggingAttributes: logLogout	Log user logouts in the activity log. Default = yes
loggingAttributes: logOpenFork	Log file opens in the activity log. Default = yes
loginGreeting	Login greeting message. Default = ""
loginGreetingTime	Last time the login greeting was set or updated.
maxConnections	Maximum simultaneous user sessions allowed by the server. Default = -1 (unlimited)
maxGuests	Maximum simultaneous guest users allowed. Default = -1 (unlimited)
maxThreads	Maximum AFP threads. (Must be specified at startup.) Default = 40
noNetworkUsers	Indication to client that all users are users on the server. Default = no
permissionsModel	How permissions are enforced. Can be set to: • classic_permissions • unix_with_classic_admin_permissions • unix_permissions Default = "classic_permissions"
recon1SrvrKeyTTLHrs	Time-to-live (in hours) for the server key used to generate reconnect tokens. Default = 168
recon1TokenTTLMins	Time-to-live (in minutes) for a reconnect token. Default = 10080

Parameter (afp:)	Description
reconnectFlag	Allow reconnect options. Can be set to: • none • all • no_admin_kills Default = "all"
reconnectTTLInMin	Time-to-live (in minutes) for a disconnected session waiting reconnection. Default = 1440
registerAppleTalk	Advertise the server using AppleTalk NBP. Default = yes
registerNSL	Advertise the server using Bonjour. Default = yes
sendGreetingOnce	Send the login greeting only once. Default = no
shutdownThreshold	Don't modify. Internal use only.
specialAdminPrivs	Grant administrator users root user read/write privileges. Default = no
SSHTunnel	Allow SSH tunneling. Default = yes
TCPQuantum	TCP message quantum. Default = 262144
tickleTime	Frequency of tickles sent to client. Default = 30
updateHomeDirQuota	Enforce quotas on the user's volume. Default = yes
useAppleTalk	Don't modify. Internal use only.

Available AFP serveradmin Commands

In addition to the standard start, stop, status, and settings commands, you can use serveradmin to execute the following service-specific AFP commands. For details on how to use these commands, see the examples in the following sections.

Command (afp:command=)	Description
cancelDisconnect	Cancel a pending user disconnect. See "Canceling a User Disconnect" on page 148.
disconnectUsers	Disconnect AFP users. See "Disconnecting AFP Users" on page 147.
getConnectedUsers	List settings for connected users. See "Viewing Connected Users" on this page.
getHistory	View a periodic record of file data throughput or number of user connections. See "Viewing AFP Service Statistics" on page 150.

Command (`afp:command=`)	Description
`getLogPaths`	Display the locations of the AFP service activity and error logs. See "Viewing AFP Log Files" on page 149.
`sendMessage`	Send a text message to connected AFP users. See "Sending a Message to AFP Users" on page 147.
`syncSharePoints`	Update share point information after changing settings.
`writeSettings`	Equivalent to the standard `serveradmin settings` command, but also returns a setting indicating whether the service needs to be restarted. See "Using the serveradmin Tool" on page 50.

Viewing Connected Users

To retrieve information about connected AFP users, use the `getConnectedUsers` command with the `serveradmin` tool. You can use this command to retrieve session IDs you need to disconnect or to send messages to users.

To view connected users:

```
$ sudo serveradmin command afp:command = getConnectedUsers
```

The computer responds with the following settings displayed for each connected user:

```
afp:usersArray:_array_index:i:disconnectID = <disconnectID>
afp:usersArray:_array_index:i:flags = <flags>
afp:usersArray:_array_index:i:ipAddress = <ipAddress>
afp:usersArray:_array_index:i:lastUseElapsedTime = <lastUseElapsed>
afp:usersArray:_array_index:i:loginElapsedTime = <loginElapsedTime>
afp:usersArray:_array_index:i:minsToDisconnect = <minsToDisconnect>
afp:usersArray:_array_index:i:name = <name>
afp:usersArray:_array_index:i:serviceType = <serviceType>
afp:usersArray:_array_index:i:sessionID = <sessionID>
afp:usersArray:_array_index:i:sessionType = <sessionType>
afp:usersArray:_array_index:i:state = <state>
```

Value returned by `getConnectedUsers` (`afp:usersArray:_array_index:`*n*`:`)	Description
`<disconnectID>`	An integer that identifies this disconnect. This appears after a disconnect is issued.
`<flags>`	Indicates the type of user. • 1-session belongs to the administrator. • 2-session belongs to a guest. • 4-session is sleeping.
`<ipAddress>`	User's IP address.
`<lastUseElapsed>`	Time since the command was last run.
`<login-elapsed-time>`	Elapsed time since the user connected.
`<minsToDisconnect>`	Number of minutes between the time the command is issued and the user is disconnected.
`<name>`	User's name.

Value returned by `getConnectedUsers` (`afp:usersArray:_array_index:<n>:`)	Description
`<serviceType>`	Share point the user is accessing.
`<sessionID>`	Integer that identifies the user session.
`<state>`	State of the service.

Sending a Message to AFP Users

To send a text message to connected AFP users, use the `sendMessage` command with the `serveradmin tool`. Users are specified by session ID.

To send a message:

```
$ sudo serveradmin command
afp:command = sendMessage
afp:message = "message-text"
afp:sessionIDsArray:_array_index:0 = sessionid1
afp:sessionIDsArray:_array_index:1 = sessionid2
afp:sessionIDsArray:_array_index:2 = sessionid3
[...]
Control-D
```

Parameter	Description
message-text	Message that appears on client computers.
sessionidn	Session ID of the user you want to receive the message. To list the session IDs of connected users, use the `getConnectedUsers` command. See "Viewing Connected Users" on page 146.

Disconnecting AFP Users

To disconnect AFP users, use the `disconnectUsers` command with the `serveradmin` tool. You can specify a delay time before a disconnect and include a warning message.

To disconnect users:

```
$ sudo serveradmin command
afp:command = disconnectUsers
afp:message = "message-text"
afp:minutes = minutes-until
afp:sessionIDsArray:_array_index:0 = sessionid1
afp:sessionIDsArray:_array_index:1 = sessionid2
afp:sessionIDsArray:_array_index:2 = sessionid3
[...]
Control-D
```

Parameter	Description
message-text	The message that appears on client computers in the disconnect announcement dialog.

Parameter	Description
minutes-until	The number of minutes between the time the command is executed and the users are disconnected.
sessionidn	The session ID of a user you want to disconnect. To list the session IDs of connected users, use the `getConnectedUsers` command. See "Viewing Connected Users" on page 146.

The computer responds with the following output:

```
afp:command = "disconnectUsers"
afp:messageSent = "<message>"
afp:timeStamp = "<time>"
afp:timerID = <disconnectID>
<user listing>
afp:status = <status>
```

Value	Description
`<message>`	The message sent to users in the disconnect announcement dialog.
`<time>`	The time when the command was executed.
`<disconnectID>`	An integer that identifies this disconnect. To cancel the disconnect, use this ID with the `cancelDisconnect` command.
`<user listing>`	A standard array of user settings for each user scheduled for disconnect. For a description of these settings, see "Viewing Connected Users" on page 146.
`<status>`	A command status code. `0` = command successful.

Canceling a User Disconnect

To cancel a `disconnectUsers` command, use the `cancelDisconnect` command with the `serveradmin` tool. Users receive an announcement that they're no longer scheduled to be disconnected.

To cancel a user disconnect:

```
$ sudo serveradmin command
afp:command = cancelDisconnect
afp:timerID = timerID
Control-D
```

Parameter	Description
timerID	The integer value of the `afp:timerID` parameter output when you executed the `disconnectUsers` command. You can also find this number by listing a user scheduled to be disconnected and looking at the value of the `disconnectID` setting for the user.

The computer responds with the following output:

```
afp:command = "cancelDisconnect"
afp:timeStamp = "<time>"
afp:status = <status>
```

Value	Description
`<time>`	The time the command was executed.
`<status>`	A command status code: `0` = command successful.

Viewing AFP Log Files

To view the contents of the AFP service logs, use `tail` or another file listing tool.

To view the latest entries in a log:

```
$ tail log-file
```

To see where the current AFP error and activity logs are located, use the `getLogPaths` command with the `serveradmin` tool.

To view the log paths:

```
$ sudo serveradmin command afp:command = getLogPaths
```

The computer responds with the following output:

```
afp:accesslog = <access-log>
afp:errorlog = <error-log>
```

Value	Description
`<access-log>`	The location of the AFP service access log. Default = `/Library/Logs/AppleFileService/AppleFileServiceAccess.log`.
`<error-log>`	The location of the AFP service error log. Default = `/Library/Logs/AppleFileService/AppleFileServiceError.log`.

Viewing AFP Service Statistics

To view a log of periodic samples of the number of connections and the data throughput, use the `serveradmin getHistory` command. Samples are taken once each minute.

To view service statistic samples:

```
$ sudo serveradmin command
afp:command = getHistory
afp:variant = statistic
afp:timeScale = scale
Control-D
```

Parameter	Description
statistic	The value you want to display valid values: • `v1` = number of connected users (average during sampling period). • `v2` = throughput (bytes/sec).
scale	The length of time in seconds, ending with the current time, you want to see samples for. For example, to see 30 minutes of data, you would specify `afp:timeScale = 1800`.

The computer responds with the following output:

```
afp:nbSamples = <samples>
afp:samplesArray:_array_index:0:vn = <sample>
afp:samplesArray:_array_index:0:t = <time>
afp:samplesArray:_array_index:1:vn = <sample>
afp:samplesArray:_array_index:1:t = <time>
[...]
afp:samplesArray:_array_index:i:vn = <sample>
afp:samplesArray:_array_index:i:t = <time>
afp:vnLegend = "<legend>"
afp:currentServerTime = <servertime>
```

Value displayed by `getHistory`	Description
`<samples>`	The total number of samples listed.
`<legend>`	A textual description of the selected statistic: • `"CONNECTIONS"` for v1. • `"THROUGHPUT"` for v2.
`<sample>`	The numerical value of the sample: • For connections (v1), this is an integer average number of users. • For throughput, (v2), this is an integer bytes per second.
`<time>`	The time the sample was measured. A standard UNIX time (number of seconds since September 1, 1970). Samples are taken every 60 seconds.

Managing NFS Service

NFS is a file service used to provide file sharing to UNIX and Linux systems. With NFS, Mac OS X Server can host data for UNIX application servers and provide integration with enterprise UNIX storage devices. Support for NFS file locking prevents overwriting files while others are accessing them.

NFS service can be used to mount NFS volumes and reshare them over AFP with Mac OS X and Mac OS 9 clients. This allows client computers to access NFS volumes using the secure authentication and service discovery provided by AFP service.

Starting and Stopping NFS Service

NFS service starts when a share point is exported using NFS. The NFS daemons that satisfy client requests continue to run until there are no more NFS exports and the server is restarted.

Viewing NFS Service Status

To see if the service and related processes are running:

```
$ sudo serveradmin status nfs
```

To see complete status:

```
$ sudo serveradmin fullstatus nfs
```

Viewing NFS Service Settings

To list a setting:

```
$ sudo serveradmin settings nfs:setting
```

To list all settings:

```
$ sudo serveradmin settings nfs
```

Changing NFS Service Settings

To change settings for the NFS service, use the following parameters with the `serveradmin` tool.

Parameter (`nfs:`)	Description
`nbDaemons`	To reduce the number of daemons, restart the server after changing this value. Default = `6`.
`useTCP`	Restart the server after changing this value. Default = `yes`.
`useUDP`	Restart the server after changing this value. Default = `yes`.

Managing FTP Service

Mac OS X Server features a robust FTP file service for Internet file sharing from any platform. FTP provides the broadest compatibility across platforms, making it ideal for anonymous downloads or sharing files that are too large to be sent over mail.

Mac OS X Server improves the security of FTP service with Kerberos authentication. It also supports automatic resumption of disconnected FTP file transfers.

Starting FTP Service

To start the service:

```
$ sudo serveradmin start ftp
```

Stopping FTP Service

To stop the service:

```
$ sudo serveradmin stop ftp
```

Viewing FTP Service Status

To see if the service is running:

```
$ sudo serveradmin status ftp
```

To see complete status:

```
$ sudo serveradmin fullstatus ftp
```

Viewing FTP Service Settings

To view a setting:

```
$ sudo serveradmin settings ftp:setting
```

To view a group of settings:

You can view a group of settings that have part of their names in common by entering as much of the name as you want, stopping at a colon (:), and entering an asterisk (*) as a wildcard for the remaining parts of the name. For example:

```
$ sudo serveradmin settings ftp:logCommands:*
```

To view all settings:

```
$ sudo serveradmin settings ftp
```

Changing FTP Service Settings

To change FTP service settings, use the `serveradmin` tool.

To change a setting:

```
$ sudo serveradmin settings ftp:setting = value
```

Parameter	Description
setting	An FTP service setting. To see a list of available settings, enter `$ sudo serveradmin settings ftp` or see "Available FTP Service Settings" below.
value	An appropriate value for the setting.

To change several settings:

```
$ sudo serveradmin settings
ftp:setting = value
ftp:setting = value
ftp:setting = value
[...]
Control-D
```

Available FTP Service Settings

To change settings for the FTP service, use the following parameters with the `serveradmin` tool.

Parameter (`ftp:`)	Description
`administratorEmailAddress`	Sets the administrator mail address. Default = `"user@hostname"`
`anonymous-root`	Sets the anonymous root directory. Default = `"/Library/FTPServer/FTPRoot"`
`anonymousAccessPermitted`	Allows anonymous access to FTP if you change the default setting to yes. Default = `no`
`authLevel`	Sets the authentication method. "`KERBEROS`" and "`ANY METHOD`" are the other possible values. Default = `"STANDARD"`

Parameter (`ftp:`)	Description

`bannerMessage`	Displays a banner message that appears when you are prompted to log in to FTP. Customize to your own preferences.
	Default =
	`"----------`
	`This is the "Banner" message for the Mac OS X Server's FTP server process.`
	
	`FTP clients will receive this message immediately before being prompted for a name and password.`
	
	`PLEASE NOTE: Some FTP clients may exhibit problems if you make this file too long.`
	
	`----------"`

`chrootType`	Default = `"STANDARD"`
`enableMacBinAndDmgAutoConversion`	Default = `yes`
`ftpRoot`	The directory where the FTP content is stored. Default = `"/Library/FTPServer/FTPRoot"`
`logCommands:anonymous`	Default = `no`
`logCommands:guest`	Default = `no`
`logCommands:real`	Default = `no`
`loginFailuresPermitted`	Default = `3`
`logSecurity:anonymous`	Default = `no`
`logSecurity:guest`	Default = `no`
`logSecurity:real`	Default = `no`
`logToSyslog`	Default = `no`
`logTransfers:anonymous:inbound`	Default = `yes`
`logTransfers:anonymous:outbound`	Default = `yes`
`logTransfers:guest:inbound`	Default = `no`
`logTransfers:guest:outbound`	Default = `no`
`logTransfers:real:inbound`	Default = `yes`
`logTransfers:real:outbound`	Default = `yes`
`maxAnonymousUsers`	Default = `50`
`maxRealUsers`	Default = `50`
`showBannerMessage`	Default = `yes`

Parameter (ftp:)	Description
showWelcomeMessage	Default = yes
welcomeMessage	Displays a welcome message that appears after you log in to FTP. Customize to your own preferences. Default = "---------- This is the "Welcome" message for the Mac OS X Server's FTP server process. FTP clients will receive this message right after a successful log in. ----------"

Available FTP serveradmin Commands

To manage FTP service, use the following commands with `serveradmin`. For details on how to use these commands, see the examples in the following sections.

Command (ftp:command=)	Description
getConnectedUsers	View connected users. See "Viewing for Connected FTP Users" on page 155.
getLogPaths	Show location of the FTP transfer log file. See "Viewing the FTP Transfer Log" on page 155.
writeSettings	Equivalent to the standard `serveradmin settings` command, but also returns a setting indicating whether the service must be restarted. See "Using the serveradmin Tool" on page 50.

Viewing the FTP Transfer Log

You can use `tail` or another file-listing tool to view the contents of the FTP transfer log.

To view the latest entries in the transfer log:

```
$ tail log-file
```

By default the *log-file* is located in /Library/Logs/FTP.transfer.log. To see where the current transfer log is located, use the `serveradmin getLogPaths` command.

To view the log path:

```
$ sudo serveradmin command ftp:command = getLogPaths
```

Viewing for Connected FTP Users

To see how many FTP users are connected:

```
$ ftpcount
```

or

```
$ sudo serveradmin command ftp:command = getConnectedUsers
```

Managing SMB Service

Mac OS X Server includes Samba 3, a popular open-source project that delivers high-performance SMB file and print services and Microsoft Windows NT domain services for Microsoft Windows clients.

Support for native service discovery protocols means that Mac OS X Server computers appear in the My Network Places window (Windows XP and 2000) or the Network Neighborhood window (Windows 95, 98, or ME) like a Windows server. This enables Windows clients to browse folders and share files without installing additional software.

Starting and Stopping SMB Service

To start the service:

```
$ sudo serveradmin start smb
```

To stop the service:

```
$ sudo serveradmin stop smb
```

Viewing SMB Service Status

To see if the service is running:

```
$ sudo serveradmin status smb
```

To see the complete status:

```
$ sudo serveradmin fullstatus smb
```

Viewing SMB Service Settings

To view a setting:

```
$ sudo serveradmin settings smb:setting
```

Parameter	Description
setting	An SMB service setting. To view a list of available settings, enter `$ sudo serveradmin settings smb` or see "Available SMB Service Settings" on page 157.

To view a group of settings:

You can view a group of settings that have part of their names in common by entering as much of the name as you want, stopping at a colon (`:`), and entering an asterisk (`*`) as a wildcard for the remaining parts of the name. For example:

```
$ sudo serveradmin settings smb:adminCommands:*
```

To view all service settings:

```
$ sudo serveradmin settings smb
```

Changing SMB Service Settings

You can change SMB service settings using the `serveradmin` tool.

To change a setting:

```
$ sudo serveradmin settings smb:setting = value
```

Parameter	Description
setting	An SMB service setting. To view a list of available settings, enter `$ sudo serveradmin settings smb` or see "Available SMB Service Settings" on page 157.
value	A value for the setting. For a list of values that correspond to GUI controls in the Server Admin application, see "Available SMB Service Settings" on page 157.

To change several settings:

```
$ sudo serveradmin settings
smb:setting = value
smb:setting = value
smb:setting = value
[...]
Control-D
```

Available SMB Service Settings

To change settings for the SMB service, use the following parameters with the `serveradmin` tool.

Parameter (`smb:`)	Description
`adminCommands:homes`	Whether Home folders are mounted when Windows users log in so you don't need to set up share points for each user. Can be set to: `yes` | `no` This corresponds to the "Enable virtual share points" checkbox in the Advanced pane of Window service settings in the Server Admin application.
`adminCommands:serverRole`	The authentication role played by the server. Can be set to: • `"standalone"` • `"domainmember"` • `"primarydomaincontroller"` • `"backupdomaincontroller"` This corresponds to the Role pop-up menu in the General pane of Windows service settings in the Server Admin application.

Parameter (smb:)	Description
domain master	Whether the server is providing Windows domain master browser service. Can be set to: yes | no This corresponds to the Domain Master Browser checkbox in the Advanced pane of Window service settings in the Server Admin application.
dos charset	The code page being used. Can be set to: • 437 (Latin US) • 737 (Greek) • 775 (Baltic) • 850 (Latin1) • 852 (Latin2) • 861 (Icelandic) • 866 (Cyrillic) • 932 (Japanese SJIS) • 936 (Simplified Chinese) • 949 (Korean Hangul) • 950 (Traditional Chinese) • 1251 (Windows Cyrillic) This corresponds to the Code Page pop-up menu on the Advanced pane of Windows service settings in the Server Admin application.
local master	Whether the server is providing Windows workgroup master browser service. Can be set to: yes | no This corresponds to the Workgroup Master Browser checkbox in the Advanced pane of Window service settings in the Server Admin application.
log level	The amount of detail written to the service logs. Can be set to: • 0 (Low: errors and warnings only) • 1 (Medium: service start and stop, authentication failures, browser name registrations, and errors and warnings) • 2 (High: service start and stop, authentication failures, browser name registration events, log file access, and errors and warnings) This corresponds to the Log Detail pop-up menu in the Logging pane of Window service settings in the Server Admin application.
map to guest	Whether guest access is allowed. Can be set to: • "Never" (No guest access) • "Bad User" (Allow guest access) This corresponds to the "Allow Guest access" checkbox in the Access pane of Window service settings in the Server Admin application.

Parameter (smb:)	Description
max smbd processes	The maximum allowed number of smbd server processes. Each connection uses its own smbd process, so this is the same as specifying the maximum number of SMB connections. 0 means unlimited. This corresponds to the "maximum" client connections field in the Access pane of the Windows service settings in the Server Admin application.
netbios name	The server's NetBIOS name. Can be set to a maximum of 15 bytes of UTF-8 characters. This corresponds to the Computer Name field in the General pane of the Windows service settings in the Server Admin application.
server string	Text that helps identify the server in the network browsers of client computers. Can be set to a maximum of 15 bytes of UTF-8 characters. This corresponds to the Description field in the General pane of the Windows service settings in the Server Admin application.
wins support	Whether the server provides WINS support. Can be set to: yes | no This corresponds to the WINS Registration "Off" and "Enable WINS" server options in the Advanced pane of the Windows service settings in the Server Admin application.
wins server	The name of the WINS server used by the server. This corresponds to the WINS Registration "Register with WINS server" option and field in the Advanced pane of the Windows service settings in the Server Admin application.
workgroup	The server's workgroup. Can be set to a maximum of 15 bytes of UTF-8 characters. This corresponds to the Workgroup field in the General pane of the Windows service settings in the Server Admin application.

Available SMB serveradmin Commands

To manage SMB service, use the following commands with the serveradmin tool. For details on how to use these commands, see the examples in the following sections.

Command (smb:command=)	Description
disconnectUsers	Disconnect SMB users. See "Disconnecting SMB Users" on page 161.
getConnectedUsers	List users connected to an SMB service. See "Viewing SMB User Information" on page 160.
getHistory	List connection statistics. See "Listing SMB Service Statistics" on page 161.
getLogPaths	Show location of service log files. See "Viewing SMB Service Logs" on page 162.

Command (smb:command=)	Description
syncPrefs	Update the service to recognize changes in share points. See "Updating Share Point Information" on page 162.
writeSettings	Equivalent to the standard serveradmin settings command, but also returns a setting indicating whether the service must be restarted. See "Using the serveradmin Tool" on page 50.

Viewing SMB User Information

To retrieve information about connected SMB users, use the serveradmin getConnectedUsers command. For example, you can use this command to retrieve the session IDs you need to disconnect users.

To view connected user information:

```
$ sudo serveradmin command smb:command = getConnectedUsers
```

The computer responds with the following array of settings for each connected user:

```
smb:usersArray:_array_index:i:loginElapsedTime = <login-elapsed-time>
smb:usersArray:_array_index:i:service = <service>
smb:usersArray:_array_index:i:connectAt = <connect-time>
smb:usersArray:_array_index:i:name = "<name>"
smb:usersArray:_array_index:i:ipAddress = "<ip-address>"
smb:usersArray:_array_index:i:sessionID = <sessionID>
```

Value returned by getConnectedUsers (smb:usersArray:_array_index:*n*:)	Description
<login-elapsed-time>	The elapsed time since the user connected.
<service>	The share point the user is accessing.
<connect-time>	The date and time the user connected to the server.
<name>	The user's name.
<ip-address>	The user's IP address.
<sessionID>	An integer that identifies the user session.

Disconnecting SMB Users

To disconnect SMB users, use the `serveradmin disconnectUsers` command. Users are specified by session ID.

To disconnect users:

```
$ sudo serveradmin command
smb:command = disconnectUsers
smb:sessionIDsArray:_array_index:0 = sessionid1
smb:sessionIDsArray:_array_index:1 = sessionid2
smb:sessionIDsArray:_array_index:2 = sessionid3
[...]
Control-D
```

Parameter	Description
sessionidn	The session ID of a user you want to disconnect. To list the session IDs of connected users, use the `getConnectedUsers` command. See "Viewing SMB User Information" on page 160.

The computer responds with the following output:

```
smb:command = "disconnectUsers"
smb:status = <status>
```

Value	Description
`<status>`	A command status code. `0` = command successful

Listing SMB Service Statistics

To display a list of the number of SMB connections, use the `smbstatus` command.

To list connections:

```
$ smbstatus
```

The computer responds with the following output:

```
Samba version 3.0.10
PID       Username       Group          Machine
-------------------------------------------------------------------
8287      ajohnson       officegroup    mycomputer     (123.123.12.12)

Service   pid       machine        Connected at
-------------------------------------------------------------------
IPC$      8287      mycomputer     Fri Jan 13 06:06:15 2007
No Locked Files
```

Updating Share Point Information

After you make a change to an SMB share point using the `sharing` tool, you must update the SMB service information.

To update share point information:

```
$ sudo serveradmin command smb:command = syncPrefs
```

Viewing SMB Service Logs

To view the contents of the SMB service logs, use `tail` or another file-listing tool.

To view the latest entries in a log:

```
$ tail log-file
```

To see where the SMB logs are located, use the `serveradmin getLogPaths` command.

To display log paths:

```
$ sudo serveradmin command smb:command = getLogPaths
```

The computer responds with the following output:

```
smb:fileServiceLog = <smb-log>
smb:nameServiceLog = <name-log>
```

Value	Description
`<smb-log>`	The location of the SMB service log. Default = `/var/log/samba/log.smbd`
`<name-log>`	The location of the name service log. Default = `/var/log/samba/log.nmbd`

Managing ACLs

For greater flexibility in configuring and managing file permissions, Mac OS X Server implements access control lists (ACLs). An ACL is a list of access control entries (ACEs), each specifying the permissions to be granted or denied to a group or user, and how these permissions are propagated through a folder hierarchy.

ACLs in Mac OS X Server let you set file and folder access permissions for multiple users and groups, in addition to standard POSIX permissions. This makes it easy to set up collaborative environments with smooth file sharing and uninterrupted workflows, without compromising security.

Mac OS X Server has implemented file system ACLs that are fully compatible with Microsoft Windows Server 2003 and Windows XP.

For more about ACLs and how they compare to POSIX permissions, see the Overview chapter of *File Services Administration*.

Using chmod to Modify ACLs

Using `chmod`, you can add and delete ACEs for a file or a folder. The following parameters can be used with ACLs:

Parameter	Description
`+a`	Adds an entry to the ACL.
`+ai`	Adds an inherited entry.
`-a`	Removes an entry from the ACL.

The following are common permissions you can assign to files:

Permission	Description
`delete`	Grants permission to delete the item.
`readattr`	Reads an object's basic attributes.
`read`	Reads the object.
`write`	Writes to the object.
`writeattr`	Writes an object's basic attributes.
`readextattr`	Reads extended attributes.
`writeextattr`	Writes extended attributes.
`readsecurity`	Reads an object's extended security information (ACL).
`writesecurity`	Writes an object's security information (ACL).
`chown`	Changes an object's ownership.

The following permissions are applicable to folders:

Permission	Description
`list`	Lists entries.
`add_file`	Adds a file.
`add_sudirectory`	Adds a subfolder.
`delete_child`	Deletes an object.

To grant a user write permission for a file:

Enter the following command, replacing *user1* with the name of the user you are granting permission to and *file1* with the name of the file:

```
$ chmod +a "user1 allow write" file1
```

To deny a guest read permission for a file:

Enter the following command, replacing *file1* with the name of the file:

```
$ chmod +a "guest deny read" file1
```

To view the ACL of a file:

Enter the following command, replacing *file1* with the name of the file:

```
$ ls -le file1
```

The output should look like the following:

```
-rw-r--r--+ 1 juser  wheel  0 Apr 28 14:06 file1
owner: juser
0: guest deny read
1: user1 allow write
```

For more information, see the `ls` man page.

Using fsaclctl to Enable and Disable ACL Support

By default, ACL is enabled at the volume level. However, you can use the `fsaclctl` command to disable or enable ACL support on any volume. In addition, you can use this command to determine whether ACL support is enabled on a given volume.

After enabling or disabling ACL support using the `fsaclctl` command, restart your server or remount the volume.

To enable ACL support on a volume:
Enter the following command:

```
$ sudo fsaclctl -p path -e enable
```

Parameter	Description
path	The path to the volume.

To disable ACL support on a volume:
Enter the following command:

```
$ sudo fsaclctl -p path -d disable
```

To enable ACL support on all mounted volumes:
Enter the following command:

```
$ sudo fsaclctl -a -e enable
```

To disable ACL support on all mounted volumes:
Enter the following command:

```
$ sudo fsaclctl -a -d disable
```

To display ACL support status for a volume:
Enter the following command:

```
$ fsaclctl -p path
```

The output is similar to the following:

```
Access control lists are supported on /Volumes/Data HD.
```

To display ACL support status for all mounted volumes:
Enter the following command:

```
$ fsaclctl -a
```

The output is similar to the following:

```
ProcessVolume: processing /
Access control lists are supported on /.
ProcessVolume: processing /Volumes/Data HD
Access control lists are supported on /Volumes/Data HD.
```

Working with the Print Service

10

Use this chapter to learn the commands to configure and manage the Print service.

This chapter covers the commands needed to view, modify, or change Print service settings.

Print service in Mac OS X Server lets you share network and direct-connect printers among clients on your network. Print service also includes support for managing print queues, monitoring print jobs, extensive logging, and using print quotas.

For more information, see *Print Service Administration*.

Understanding the Print Process

Apple's printing infrastructure is built on the Common UNIX Printing System (CUPS). CUPS uses open standards, such as Internet Printing Protocol (IPP) and PostScript Printer Description (PPD) files. Tools derived from the old LPD and LP systems are fully integrated with the printing system.

You can add a print queue with Printer Setup Utility or from the command line, and print to it from a Mac OS X application or the command line. CUPS allows Mac OS X to support all printers that other UNIX systems support.

The CUPS daemon is `/usr/sbin/cupsd`. Mac OS X applications and tools communicate with the daemon using IPP. IPP uses UDP and HTTP for transport over IP. Some configuration files that affect the behavior of `cupsd` reside in /etc/cups/. When you make a change to printer sharing or to the printer list using Mac OS X applications or tools, you modify cupsd.conf or printers.conf, respectively.

To prepare files for printing, `cupsd` invokes other tools called filters and backends. These reside in subfolders of /usr/libexec/cups/.

CUPS has its own URL, 127.0.0.1:631, which you can access with a web browser. The URL is independent of the Apache web server, so you do not need to enable web sharing to use it. You can find the CUPS documentation at www.cups.org.

CUPS includes System V (`lp`) and Berkeley (`lpr`) printing commands. CUPS supports many different file formats, including PostScript and image files, so you can print most files from the command line.

The CUPS log files, located in /var/log/cups/, include the following:

- `access_log`, which contains all HTTP requests processed by CUPS server
- `error_log`, which contains messages from the scheduler (errors, warnings, and so on)
- `page_log`, which contains a summary of each page sent to a printer

To add a print queue, use the `lpadmin` tool or the CUPS web interface. When you add a printer or create a printer pool, you create a CUPS print queue. A PPD file, which defines the attributes of that queue, is placed in /etc/cups/ppd/. The name of the PPD file corresponds with the name of the queue (either the name of a printer or the name of a class). CUPS uses PPD files for non-PostScript printers as well.

The PPD file is copied from another folder on your computer. The standard CUPS location for PPD files is /usr/share/cups/model/ and its subfolders. The standard location is in the following folders: /Library/Printers/PPDs/Contents/Resources/ and /System/Library/Printers/PPDs/Contents/Resources/. The `lpadmin` tool can use only PPD files in /usr/share/cups/model/ and its subfolders.

When you initiate a print job, you generate a CUPS spool file and an IPP attributes file in /var/spool/cups/. The `lp` or `lpr` tool generates an IPP attributes file and spool file. The spool file is a copy of the original document, so its format is the same as that of the original file. If the tools do not support a file's format, you get an error message.

After the file is copied to /var/spool/cups/, `cupsd` begins preparing the file for printing.

For more information about CUPS and tools specific to CUPS, see the documentation at www.cups.org/documentation.php.

You can also see the man pages for the following CUPS commands: `accept`, `backend`, `cancel`, `filter`, `lp`, `lpadmin`, `lpinfo`, `lpoptions`, `lpq`, `lpr`, `lpstat`, and `reject`.

Note: For information about configuring Kerberos support for Print service IPP shared queues, see *Print Service Administration*.

Performing Print Service Tasks

To perform print service tasks, use the `serveradmin` tool and commands that interact with CUPS.

Starting and Stopping Print Service

To start Print service:

```
$ sudo serveradmin start print
```

To stop Print service:

```
$ sudo serveradmin stop print
```

Viewing the Status of Print Service

To see summary status of Print service:

```
$ sudo serveradmin status print
```

To see detailed status of Print service:

```
$ sudo serveradmin fullstatus print
```

Viewing Print Service Settings

To view a setting:

```
$ sudo serveradmin settings print:setting
```

To view a group of settings:

You can view a group of settings that have part of their names in common by entering as much of the name as you want, stopping at a colon (`:`), and entering an asterisk (`*`) as a wildcard for the remaining parts of the name. For example, to see all settings for a particular print queue:

```
$ sudo serveradmin settings print:queuesArray:_array_id:queue-id:*
```

Parameter	Description
queue-id	CUPS queue ID (for example, `<id>` or `_192_216_3_45`).

To list all settings:

```
$ sudo serveradmin settings print
```

Changing Print Service Settings

To change a setting:

```
$ sudo serveradmin settings print:setting = value
```

Parameter	Description
setting	A print service setting. To see a list of available settings, enter `$ sudo serveradmin settings print` or see "Available Print Service Settings" on page 170.
value	An appropriate value for the setting.

To change several settings:

```
$ sudo serveradmin settings
print:setting = value
print:setting = value
print:setting = value
[...]
Control-D
```

Available Print Service Settings

To change settings for the print service, use the following parameters with the `serveradmin` tool.

Parameter (`print:`)	Description
`serverLogArchiveEnable`	Default = `no`; `yes` enforces log size limits
`<queue arrays>`	See "Queue Data Array" on page 171
`serverLogArchiveSizeMB`	Default = `1`; maximum log size Range = 1–512 MB
`logLevel`	Default = `info`; for details, see CUPS doc
`logLevelNames`	Read-only list of valid log level names
`defaultLprQueue`	Queue-ID of selected default LPR-shared queue
`lprQueues`	Read-only list of available LPR-shared queues
`useRemoteQueues`	Default = `yes`; `no` = suppress inclusion of remote queues in queue list
`maxClients`	Default = `500`
`maxClientsPerHost`	Default = `100`

The log size limits apply to all CUPS logs:

- /var/log/cups/error_log (CUPS general message log)
- /var/log/cups/access_log (CUPS access log)
- /var/log/cups/error_log (CUPS page log)

These limits also apply to the following log files:

- /Library/Logs/PrintService/PrintService.admin.log (Server Admin Print log: logs all Print administrative actions issued from Server Admin)
- /Library/Logs/atprintd/<queue-id>.spool.log (AppleTalk spool logs: 1 per shared AppleTalk queue)

The log level option filters the number of messages written to the following logs:

- /var/log/cups/error_log
- /Library/Logs/PrintService/PrintService.admin.log
- /Library/Logs/atprintd/<queue-id>.spool.log

Queue Data Array

Print service settings include an array of values for each print queue. The array is a set of parameters that define values for each queue.

The array of sharing services now includes IPP. This is the same service as Mac OS X v10.3 printer sharing, now integrated with Mac OS X Server v10.5.

Many of the following parameters are CUPS parameters. For more details about CUPS parameters, see the CUPS documentation.

`<id>` is a CUPS queue ID (for example, `<id>` or `_192_216_3_45`).

Parameter (`print:`)	Description
`queuesArray:_array_id:<id>:quotasEnforced`	Default = `no`; `yes` = enforce quota limits for queue.
`queuesArray:_array_id:<id>:sharingList:_array_index:0:service`	Service name for IPP (CUPS).
`queuesArray:_array_id:<id>:sharingList:_array_index:1:service`	Default = `"LPR"`; service name for UNIX Line Printer.
`queuesArray:_array_id:<id>:sharingList:_array_index:2:service`	Default = `"SMB"`; service name for Windows SMB.
`queuesArray:_array_id:<id>:sharingList:_array_index:3:service`	Default = `"AppleTalk"`; service name for AppleTalk.
`queuesArray:_array_id:<id>:shareable`	Cannot be changed. Default = `yes`.
`queuesArray:_array_id:<id>:printerName`	Cannot be changed using `serveradmin`. Default = `"<printer-name>"`
`queuesArray:_array_id:<id>:printerURI`	Format depends on type of printer. Cannot be changed using serveradmin. Default = `<uri>`; CUPS printer device info.
`queuesArray:_array_id:<id>:registerRendezvous`	Default = `yes`; `yes` = advertise printer over multicast DNS.
`queuesArray:_array_id:<id>:printerKind`	CUPS queue identifier. Cannot be changed using serveradmin.
`queuesArray:_array_id:<id>:sharingName`	Name used to advertise queue on network.
`queuesArray:_array_id:<id>:defaultCoverPage`	Name of assigned cover page.

The following is an example of a queue array parameter block:

```
print:queuesArray:_array_id:my_printer:quotasEnforced = no
print:queuesArray:_array_id:my_printer:sharingList:_array_index:0:service =
     "LPR"
print:queuesArray:_array_id:my_printer:sharingList:_array_index:0:sharingEna
     ble = no
print:queuesArray:_array_id:my_printer:sharingList:_array_index:1:service =
     "SMB"
print:queuesArray:_array_id:my_printer:sharingList:_array_index:1:sharingEna
     ble = no
print:queuesArray:_array_id:my_printer:sharingList:_array_index:2:service =
     "AppleTalk"
print:queuesArray:_array_id:my_printer:sharingList:_array_index:2:sharingEna
     ble = no
print:queuesArray:_array_id:my_printer:shareable = yes
print:queuesArray:_array_id:my_printer:printerName = "Room 3 Printer"
print:queuesArray:_array_id:my_printer:printerURI = "pap://*/
     Room%203%20Printer/LaserWriter"
print:queuesArray:_array_id:my_printer:registerRendezvous = yes
print:queuesArray:_array_id:my_printer:printerKind = "Lexmark_Optra_E310"
print:queuesArray:_array_id:my_printer:sharingName = "Room 3 Printer"
```

Note: In the example above, "my_printer" refers to the CUPS queue id.

Managing Print Service

To modify and manage Print service, use the `serveradmin` tool and the following commands that interact with CUPS.

Command (`print:command=`)	Description
`getJobs`	Lists information about jobs in a queue. The name required for this command is the sharing name given to the queue by the administrator, as previously described. See "Listing Jobs and Job Information" on page 173.
`getLogPaths`	Finds the locations of Print service and job logs. See "Viewing Print Service Log Files and Log Paths" on page 175.
`getQueues`	Lists Print service queues. See "Listing Queues" on page 173.
`setJobState`	Holds or releases a job. The name required for this command is the sharing name given to the queue by the administrator, as previously described. See "Holding and Releasing a Job" on page 174.
`setQueueState`	Pauses or releases a queue. The queue name required for this command is the sharing name given to the queue by the administrator, not the original printer name or the CUPS queue identifier. See "Pausing and Releasing a Queue" on this page.
`writeSettings`	Equivalent to the standard `serveradmin settings` command, but also returns a setting indicating whether the service must be restarted. See "Using the serveradmin Tool" on page 50.

Listing Queues

To list print service queues, use the `serveradmin getQueues` command.

```
$ sudo serveradmin command print:command = getQueues
```

Pausing and Releasing a Queue

You can use the `serveradmin setQueueState` command to pause or release a queue.

To pause a queue:

```
$ sudo serveradmin command
print:command = setQueueState
print:state = PAUSED
print:namesArray:_array_index:0 = queue
Control-D
```

Parameter	Description
queue	The name of the queue. To find the name of the queue, use the `getQueues` command and look for the value of the `printer` setting. See "Listing Queues" on page 173.

To release a queue:

```
$ sudo serveradmin command
print:command = setQueueState
print:state = RESUMED
print:namesArray:_array_index:0 = queue
Control-D
```

Listing Jobs and Job Information

To list information about print jobs, use the `serveradmin getJobs` command.

```
$ sudo serveradmin command
print:command = getJobs
print:maxDisplayJobs = jobs
print:queueNamesArray:_array_index:0 = queue
Control-D
```

Parameter	Description
jobs	The maximum number of jobs to list.
queue	The name of the queue. To find the name of the queue, use the `getQueues` command and look for the value of the `printer` setting. See "Listing Queues" on page 173.

For each job, the command lists:

- Document name
- Document size
- Job ID
- Submitting user
- Submitting host
- Job name
- Job state
- Job priority

Holding and Releasing a Job

To hold or release a job, use the `serveradmin setJobState` command.

To hold a job:

```
$ sudo serveradmin command
print:command = setJobState
print:status = HOLD
print:jobsArray:_array_index:0:printer = queue
print:jobsArray:_array_index:0:idsArray:_array_index:0 = jobid
Control-D
```

Parameter	Description
queue	The name of the queue. To find the name of the queue, use the `getQueues` command and look for the value of the `printer` setting. See "Listing Queues" on page 173.
jobid	The ID of the job. To find the ID of the job, use the `getJobs` command and look for the value of the `jobId` setting. See "Listing Jobs and Job Information" on page 173.

To release the job for printing, change its state to `PENDING`.

To release a job:

```
$ sudo serveradmin command
print:command = setJobState
print:status = PENDING
print:jobsArray:_array_index:0:printer = queue
print:jobsArray:_array_index:0:idsArray:_array_index:0 = jobid
Control-D
```

Viewing Print Service Log Files and Log Paths

To view the contents of the Print service logs and to view log paths, use `tail` or another file-listing tool.

To view the latest entries in a log:

```
$ tail log-file
```

The following are log files for Print service:

- /var/log/cups/error_log (CUPS general message log)
- /var/log/cups/access_log (CUPS access log)
- /var/log/cups/page_log (CUPS page log)
- /Library/Logs/PrintService/PrintService.admin.log (Server Admin Print log: logs all Print administrative actions issued from Server Admin)
- /Library/Logs/atprintd/<queue-id>.spool.log (AppleTalk spool logs: 1 per shared AppleTalk queue)

To see where current logs are located, use the `serveradmin getLogPaths` command.

To view log paths:

```
$ sudo serveradmin command print:command = getLogPaths
```

The computer responds with the following output:

```
print:logPathsArray:_array_index:0:name = "Print Service Admin log"
print:logPathsArray:_array_index:0:path = "/Library/Logs/PrintService/
     PrintService_admin.log"
print:logPathsArray:_array_index:1:name = "CUPS: error_log"
print:logPathsArray:_array_index:1:path = "/var/log/cups/error_log"
print:logPathsArray:_array_index:2:name = "CUPS: access_log"
print:logPathsArray:_array_index:2:path = "/var/log/cups/access_log"
print:logPathsArray:_array_index:3:name = "CUPS: page_log"
print:logPathsArray:_array_index:3:path = "/var/log/cups/page_log"
```

Viewing Cover Pages

To obtain a list of available cover pages:

```
$ sudo serveradmin settings print:coverPageNames
```

This returns a read-only list of permitted values for this setting. The value "none" is not listed as a cover page name but is used to disable the cover page feature for the selected print queue.

Working with NetBoot Service and System Images

11

Use this chapter to learn the commands to configure and manage NetBoot Service and system images.

This chapter describes the commands used to configure and manage NetBoot service.

You can use NetBoot to host a standard operating system and application configuration on a server for all clients on a network.

Understanding NetBoot Service

NetBoot service in Mac OS X Server enables multiple Mac computers to boot from a single server-based disk image, instead of from their internal hard disks. This allows you to create a standard configuration and use it on all desktop computers on a network—or to host multiple images customized for different workgroups.

You can also create server configurations and run all servers from one image. Updating the disk image on the NetBoot server updates all computers when they restart. In addition, you can copy a directory server configuration to all clients using the same system image.

Starting and Stopping NetBoot Service

To start the service:

```
$ sudo serveradmin start netboot
```

If you get the following response, you have not yet enabled NetBoot on a network port:

```
$ netboot:state = "STOPPED"
$ netboot:status = 5000
```

To stop the service:

```
$ sudo serveradmin stop netboot
```

Viewing NetBoot Service Status

To see if the service is running:

```
$ sudo serveradmin status netboot
```

To see complete service status:

```
$ sudo serveradmin fullstatus netboot
```

Viewing NetBoot Settings

To list all service settings:

```
$ sudo serveradmin settings netboot
```

Changing NetBoot Settings

You can change NetBoot service settings using the `serveradmin` tool.

To change a setting:

```
$ sudo serveradmin settings netboot:setting = value
```

Parameter	Description
setting	A NetBoot service setting. To see a list of available settings, enter `$ sudo serveradmin settings netboot` or see "Changing General Netboot Service Settings" on this page.
value	An appropriate value for the setting.

To change several settings:

```
$ sudo serveradmin settings
netboot:setting = value
netboot:setting = value
netboot:setting = value
[...]
Control-D
```

Changing General Netboot Service Settings

NetBoot allows client computers to start up from an operating system image stored on your server.

To change settings for NetBoot service, use the following parameters with the `serveradmin` tool:

Parameter (`netboot:`)	Description
`filterEnabled`	A parameter that specifies whether client filtering is enabled. Default = `"no"`
`netBootStorageRecordsArray...`	An array of values for each server volume used to store boot or installation images. For a description, see "The Storage Record Array" on page 179.

Parameter (netboot:)	Description
netBootFiltersRecordsArray...	An array of values for each computer explicitly allowed or disallowed access to images. For a description, see "The Filters Record Array" on page 180.
netBootImagesRecordsArray...	An array of values for each boot or installation image stored on the server. For a description, see "The Image Record Array" on page 180.
netBootPortsRecordsArray...	An array of values for each server network port used to deliver boot or installation images. For a description, see "The Port Record Array" on page 181.

The Storage Record Array

A volume parameter array.

Parameter (netboot:)	Description
netBootStorageRecordsArray:_array_index:<n>:sharepoint	First parameter in an array describing a volume available to serve images. Default = "no"
netBootStorageRecordsArray:_array_index:<n>:clients	Default = "no"
netBootStorageRecordsArray:_array_index:<n>:ignorePrivs	Default = "false"
netBootStorageRecordsArray:_array_index:<n>:volType	Default = <voltype> Example: "hfs"
netBootStorageRecordsArray:_array_index:<n>:path	Default = "/"
netBootStorageRecordsArray:_array_index:<n>:volName	Default = <name>
netBootStorageRecordsArray:_array_index:<n>:volIcon	Default = <icon>
netBootStorageRecordsArray:_array_index:<n>:okToDeleteClients	Default = "yes"
netBootStorageRecordsArray:_array_index:<n>:okToDeleteSharepoint	Default = "yes"

The Filters Record Array

An array of the following values appears in NetBoot service settings for each computer explicitly allowed or denied access to images stored on the server.

Parameter (`netboot:`)	Description
`netBootFiltersRecordsArray:_array_index:<n>:hostName`	The host name of the filtered computer, if available.

`netBootFiltersRecordsArray:_array_index:<n>:filterType`	Whether the specified computer is allowed or denied access. Options:
	`"allow"`
	`"deny"`

`netBootFiltersRecordsArray:_array_index:<n>:hardwareAddress`	The Ethernet hardware (MAC) address of the filtered computer.

The Image Record Array

An array of the following values appears in NetBoot service settings for each image stored on the server.

Parameter (`netboot:`)	Description
`netBootImagesRecordsArray:_array_index:<n>:Name`	Name of the image as it appears in the Startup Disk control panel (Mac OS 9) or Preferences pane (Mac OS X).
`netBootImagesRecordsArray:_array_index:<n>:IsDefault`	`yes` specifies this image file as the default boot image on the subnet.
`netBootImagesRecordsArray:_array_index:<n>:RootPath`	The path to the .dmg file.
`netBootImagesRecordsArray:_array_index:<n>:isEdited`	Whether the image is edited.
`netBootImagesRecordsArray:_array_index:<n>:BootFile`	Name of the boot ROM file: `booter`.
`netBootImagesRecordsArray:_array_index:<n>:Description`	Arbitrary text describing the image.
`netBootImagesRecordsArray:_array_index:<n>:SupportsDiskless`	`yes` directs the NetBoot server to allocate space for shadow files needed by diskless clients.
`netBootImagesRecordsArray:_array_index:<n>:Type`	`NFS` or `HTTP`.
`netBootImagesRecordsArray:_array_index:<n>:pathToImage`	The path to the parameter list file in the .nbi folder on the server describing the image.
`netBootImagesRecordsArray:_array_index:<n>:Index`	`1–4095` indicates a local image unique to the server. `4096–65535` is a duplicate, identical image stored on multiple servers for load balancing.

Parameter (netboot:)	Description
netBootImagesRecordsArray:_array_index:<n>:IsEnabled	Sets whether the image is available to NetBoot (or Network Image) clients.
netBootImagesRecordsArray:_array_index:<n>:IsInstall	yes specifies a network installation image; no specifies a NetBoot image.

The Port Record Array

An array of the following items is included in the NetBoot service settings for each network port on the server set to deliver images.

Parameter (netboot:)	Description
netBootPortsRecordsArray:_array_index:<m>:isEnabledAtIndex	First parameter in an array describing a network interface available for responding to netboot requests. Default = "no"
netBootPortsRecordsArray:_array_index:<m>:nameAtIndex	Default = "<devname>" Example: "Built-in Ethernet"
netBootPortsRecordsArray:_array_index:<m>:deviceAtIndex	Default = "<dev>" Example: "en0"

Enabling NetBoot 1.0 for Older NetBoot Clients

If you want older computers, such as tray-loading iMac or Power Macintosh G3 (Blue and White) computers, to use NetBoot, you must enable NetBoot 1.0. You can do so by using the dscl tool.

Note: NetBoot 1.0 and 2.0 can run on the same network simultaneously.

To enable NetBoot:

```
$ sudo dscl . create /config/dhcp old_netboot_enabled port_list
$ sudo killall bootpd
```

Parameter	Description
port_list	List of ports you want to enable for NetBoot 1.0, formatted like: en0 en1 en2.

Working with System Images

A boot image is a file that acts like a mountable disk or volume. NetBoot boot images contain the system software needed to act as a startup disk for client computers on the network. An installation image is a special boot image that boots the client long enough to install software from the image, after which the client can start up from its own hard disk.

Both boot images and installation images are special kinds of disk images. Disk images are files that behave like disk volumes.

You can set up multiple boot or installation images to suit the needs of different groups of clients or to provide several copies of the same image to distribute the client startup load. By using NetBoot with Mac OS X client management services, you can provide a personalized work environment for each user.

Updating an Image

To update a package from the command line, use the `installer` tool the same way you would to install packages on your default installation volume.

To update an image:

```
$ installer -pkg pkg.mpkg -target image_path
```

Booting from an Image

To boot from an image, set the `nvram` environment variables by using the `nvram` tool or by booting into open firmware mode.

To boot from an image:

1. Boot into open firmware by pressing command-option-o-f as you boot.
2. At the prompt, enter the following:

```
> setenv boot-file enet:YourServerIPAddress,NetBoot\NetBootsSP*\<name of
     .nbi folder>\mach.macosx
> setenv boot-args rp=nfs: YourServerIPAddress:/private/tftpboot/NetBoot/
     NetBootSP*:<name of .nbi folder>/<Name of image>.dmg
> setenv boot-device enet: YourServerIPAddress,NetBoot\NetBootSP*\<name of
     .nbi folder>\booter
> mac-boot
```

Using hdiutil with System Images

To manipulate disk images, use the `hdiutil` tool. You can use this tool to perform many functions, such as creating, compressing, mounting, unmounting, and resizing images. You can also display image information and burn images onto CDs. For information about how to manipulate disk images, see the `hdiutil` man page.

The following examples provide basic `hdiutil` tool functions:

To verify an image by comparing it to its internal checksum:

```
$ hdiutil verify myimage.img
```

To split an image into three segments:

```
$ hdiutil segment -segmentSize 10m -o /tmp/aseg 30m.dmg
```

This creates three separate files: aseg.dmg, aseg.002.dmgpart, and aseg.003.dmgpart.

To convert an image to a CD export image with a .toast extension:

```
$ hdiutil convert master.dmg -format UDTO -o master
```

To burn an image onto the CD:

```
$ hdiutil burn myImage.dmg
```

To create an image from a folder:

```
$ hdiutil create -srcfolder mydir mydir.dmg
```

Using asr to Clone a Volume or to Restore System Images

Use the `asr` tool to restore a system image onto a volume or to clone volumes.

To clone a volume:

```
$ sudo asr -source /Volumes/Classic -target /Volumes/install
```

To restore a system image onto a volume:

```
$ sudo asr -source compressedimage -target <targetvol> -erase
```

Note: The target drive is erased.

Imaging Multiple Clients Using Multicast asr

You can enable a multicast image server using the Mac OS X Server Multicast `asr` command. Multicast `asr` can restore multiple clients simultaneously from one looping multicast of an asr disk image.

Each client can receive the restore image at any time during a multicast of the image, and the client continues receiving the first part of the next multicast until the client receives the complete restore image.

The server multicasts only one copy of the restore image at a time, and all clients receive this copy.

If the server finishes multicasting the restore image and a client is still requesting the image, the server multicasts the image again. Thus, using multicast `asr` to stream images to multiple clients doesn't congest the network nearly as much as Network Install with multiple clients.

To enable the image server, use the `asr` tool with the `-server` flag and a correctly built image and plist.

To start a multicast server for a specified image:

```
$ asr -source <compressedimage> -server <configuration.plist>
```

The image does not start multicasting on the network until a client attempts to start a restore. The server continues to multicast the image until the process is terminated.

To configure a client to receive a multicast stream:

```
$ sudo asr -source asr://<hostname> -target <targetvol> -erase
```

The client receives the multicast stream from `<hostname>` and saves it to a client. To overwrite an existing image, add `-erase`. Using `-erase` with `-target` indicates an existing image should be overwritten when doing a multicast.

Choosing a Boot Device Using systemsetup

To choose your boot device, use the `systemsetup` tool. When setting the startup disk, you must know the full path to core services. For example,
to boot from "Disk 2," which is now mounted in /Volumes, you would enter:

```
$ sudo systemsetup -setstartupdisk /Volumes/Disk\ 2/System/Library/
     CoreServices
```

Managing Mail Service

12

Use this chapter to learn the commands to manage Mail service.

Mac OS X Server provides a full complement of tools for setting up and managing Mail service for your users. You use the commands described in this chapter to control the components that make up Mail service.

Understanding Mail Service

Mail service in Mac OS X Server consists of the following components, all based on open standards with full support for Internet mail protocols:

- Postfix, the SMTP mail transfer agent
- Cyrus, which supports IMAP and POP
- Mailman, which provides mailing list management features

For more information, see *Mail Service Administration*.

Postfix Agent

Mac OS X Server uses Postfix as its SMTP mail transfer agent. Postfix is easy to administer. Its basic configuration can be managed through Server Admin, and therefore, it does not rely on editing the configuration file /etc/postfix/main.cf.

Postfix uses multiple layers of defense to protect the server computer from intruders. There is no direct path from the network to the security-sensitive local delivery tools. Postfix does not trust the contents of its own queue files, or the contents of its own IPC messages. Postfix filters sender-provided information before exporting it via environment variables. Nearly every Postfix application can run with fixed low privileges and no ability to change ID, run with root privileges, or run as any other user.

Postfix uses the configuration file main.cf in /etc/postfix/. When Server Admin modifies Postfix settings, it overwrites the main.cf file.

If you make a manual change to the configuration file of Postfix, Server Admin overwrites your changes the next time you use it to modify the Mail service configuration.

The spool files for Postfix are located in /var/spool/postfix/ and the log file is /var/log/mail.log. For more information about postfix, see www.postfix.org.

Cyrus

Cyrus was developed at Carnegie Mellon University to create a highly scalable enterprise mail system for use in small- to large-enterprise environments. Cyrus technologies can scale from independent use in small departments to a system centrally managed in a large enterprise.

Each message is stored as a separate file in a mail folder for each user. The mailbox database is stored in parts of the file system that are private to the Cyrus IMAP system. This design gives the server advantages in efficiency, scalability, and administration. All user access to mail is through software using the IMAP or POP3 protocol.

Cyrus uses the configuration file /etc/imapd.conf. Server Admin uses the defaults file /etc/imapd.conf.default. Cyrus logs its events in /etc/mailaccess.log. The Cyrus database is located in /var/imap/ and user folders are located in /var/spool/imap/.

In brief, Cyrus works as follows: The Cyrus delivery application receives mail from the Postfix delivery agent, updates the mailboxes database located at /var/imap/mailboxes.db, and stores the mail in user spool files located at /var/spool/imap/*username*/*folder*. The user can then use IMAP or POP to retrieve messages.

For more information about Cyrus, see asg.web.cmu.edu/cyrus.

Mailman

Mailman is a Mailing List service with support for built-in archiving, automatic bounce processing, content filtering, digest delivery, spam filters, and other features. Mailman provides a customizable web page for each mailing list. Users can subscribe and unsubscribe themselves, as well as change list preferences. List and site administrators can use the web interface for common tasks such as account management, approvals, moderation, and list configuration. The web interface requires that you have the Apache web server running.

You can access Mailman at www.yourdomain.com/mailman/listinfo.

Mailman receives mail from the local postfix process by configuring alias maps. Messages destined for a mail list are piped by the local process to Mailman processes. The mapping is provided in /var/mailman/data/aliases.

You can find more information about configuring and administering mail lists using Mailman at www.list.org and at /Library/Documentation/Services/mailman.

Managing Mail Service

Mac OS X Server ships with powerful tools to help you administer Mail service. The following sections describe basic Mail service functions.

Starting and Stopping Mail Service

To start Mail service:

```
$ sudo serveradmin start mail
```

To stop Mail service:

```
$ sudo serveradmin stop mail
```

Checking the Status of Mail Service

To see summary status of Mail service:

```
$ sudo serveradmin status mail
```

To see detailed status of Mail service:

```
$ sudo serveradmin fullstatus mail
```

Viewing Mail Service Settings

To view Mail service configuration settings:

```
$ sudo serveradmin settings mail
```

To view a setting:

```
$ sudo serveradmin settings mail:setting
```

To view a group of settings:

You can view a group of settings that have part of their names in common by entering as much of the name as you want, stopping at a colon (`:`), and entering an asterisk (`*`) as a wildcard for the remaining parts of the name. For example:

```
$ sudo serveradmin settings mail:imap:*
```

Changing Mail Service Settings

You can use `serveradmin` to modify your server's mail configuration. However, to work with Mail service from the command line, you'll probably find it more straightforward to work with the underlying Postfix and Cyrus agents.

For more information about these agents see the following:

- For information about Postfix, see www.postfix.org.
- For information about Cyrus IMAP/POP, see asg.web.cmu.edu/cyrus.

Mail Service Settings

Use the following parameters with the `serveradmin` tool to change settings for Mail service.

Parameter (`mail:`)	Description
`postfix:message_size_limit`	Default = `10240000`
`postfix:readme_directory`	Default = `no`
`postfix:double_bounce_sender`	Default = `"double-bounce"`
`postfix:default_recipient_limit`	Default = `10000`
`postfix:local_destination_recipient_limit`	Default = `1`
`postfix:queue_minfree`	Default = `0`
`postfix:show_user_unknown_table_name`	Default = `yes`
`postfix:default_process_limit`	Default = `100`
`postfix:export_environment`	Default = `"TZ MAIL_CONFIG"`
`postfix:smtp_line_length_limit`	Default = `990`
`postfix:smtp_rcpt_timeout`	Default = `"300s"`
`postfix:masquerade_domains`	Default = `""`
`postfix:soft_bounce`	Default = `no`
`postfix:pickup_service_name`	Default = `"pickup"`
`postfix:config_directory`	Default = `"/etc/postfix"`
`postfix:smtpd_soft_error_limit`	Default = `10`
`postfix:undisclosed_recipients_header`	Default = `"To: undisclosed-recipients:;"`
`postfix:lmtp_lhlo_timeout`	Default = `"300s"`
`postfix:smtpd_recipient_restrictions`	Default = `"permit_mynetworks,reject_unauth_destination"`
`postfix:unknown_local_recipient_reject_code`	Default = `450`
`postfix:error_notice_recipient`	Default = `"postmaster"`
`postfix:smtpd_sasl_local_domain`	Default = `no`
`postfix:strict_mime_encoding_domain`	Default = `no`
`postfix:unknown_relay_recipient_reject_code`	Default = `550`
`postfix:disable_vrfy_command`	Default = `no`
`postfix:unknown_virtual_mailbox_reject_code`	Default = `550`
`postfix:fast_flush_refresh_time`	Default = `"12h"`
`postfix:prepend_delivered_header`	Default = `"command, file, forward"`
`postfix:defer_service_name`	Default = `"defer"`
`postfix:sendmail_path`	Default = `"/usr/sbin/sendmail"`

Parameter (mail:)	Description
postfix:lmtp_sasl_password_maps	Default = no
postfix:smtp_sasl_password_maps	Default = no
postfix:qmgr_clog_warn_time	Default = "300s"
postfix:smtp_sasl_auth_enable	Default = no
postfix:smtp_skip_4xx_greeting	Default = yes
postfix:smtp_skip_5xx_greeting	Default = yes
postfix:stale_lock_time	Default = "500s"
postfix:strict_8bitmime_body	Default = no
postfix:disable_mime_input_processing	Default = no
postfix:smtpd_hard_error_limit	Default = 20
postfix:empty_address_recipient	Default = "MAILER-DAEMON"
postfix:forward_expansion_filter	Default = "1234567890!@%-_=+:,./ abcdefghijklmnopqrstuvwxyzA BCDEFGHIJKLMNOPQRSTUVWXYZ"
postfix:smtpd_expansion_filter	Default = "\t\40!"#$%&'()*+,-./ 0123456789:;<=>?@ABCDEFGHIJ KLMNOPQRSTUVWXYZ[\\]^_`abcd efghijklmnopqrstuvwxyz{|}~"
postfix:relayhost	Default = ""
postfix:defer_code	Default = 450
postfix:lmtp_rset_timeout	Default = "300s"
postfix:always_bcc	Default = ""
postfix:proxy_interfaces	Default = ""
postfix:maps_rbl_reject_code	Default = 554
postfix:line_length_limit	Default = 2048
postfix:mailbox_transport	Default = 0
postfix:deliver_lock_delay	Default = "1s"
postfix:best_mx_transport	Default = 0
postfix:notify_classes	Default = "resource,software"
postfix:mailbox_command	Default = ""
postfix:mydomain	Default = <domain>
postfix:mailbox_size_limit	Default = 51200000
postfix:default_verp_delimiters	Default = "+="
postfix:resolve_dequoted_address	Default = yes
postfix:cleanup_service_name	Default = "cleanup"
postfix:header_address_token_limit	Default = 10240

Parameter (mail:)	Description
postfix:lmtp_connect_timeout	Default = "0s"
postfix:strict_7bit_headers	Default = no
postfix:unknown_hostname_reject_code	Default = 450
postfix:virtual_alias_domains	Default = "$virtual_alias_maps"
postfix:lmtp_sasl_auth_enable	Default = no
postfix:queue_directory	Default = "/private/var/spool/postfix"
postfix:sample_directory	Default = "/usr/share/doc/postfix/examples"
postfix:fallback_relay	Default = 0
postfix:smtpd_use_pw_server	Default = "yes"
postfix:smtpd_sasl_auth_enable	Default = no
postfix:mail_owner	Default = "postfix"
postfix:command_time_limit	Default = "1000s"
postfix:verp_delimiter_filter	Default = "-=+"
postfix:qmqpd_authorized_clients	Default = 0
postfix:virtual_mailbox_base	Default = ""
postfix:permit_mx_backup_networks	Default = ""
postfix:queue_run_delay	Default = "1000s"
postfix:virtual_mailbox_domains	Default = "$virtual_mailbox_maps"
postfix:local_destination_concurrency_limit	Default = 2
postfix:daemon_timeout	Default = "18000s"
postfix:local_transport	Default = "local:$myhostname"
postfix:smtpd_helo_restrictions	Default = no
postfix:fork_delay	Default = "1s"
postfix:disable_mime_output_conversion	Default = no
postfix:mynetworks:_array_index:0	Default = "127.0.0.1/32"
postfix:smtp_never_send_ehlo	Default = no
postfix:lmtp_cache_connection	Default = yes
postfix:local_recipient_maps	Default = "proxy:unix:passwd.byname $alias_maps"
postfix:smtpd_timeout	Default = "300s"
postfix:require_home_directory	Default = no
postfix:smtpd_error_sleep_time	Default = "1s"
postfix:helpful_warnings	Default = yes

Parameter (mail:)	Description
postfix:mail_spool_directory	Default = "/var/mail"
postfix:mailbox_delivery_lock	Default = "flock"
postfix:disable_dns_lookups	Default = no
postfix:mailbox_command_maps	Default = ""
postfix:default_destination_concurrency_limit	Default = 20
postfix:2bounce_notice_recipient	Default = "postmaster"
postfix:virtual_alias_maps	Default = "$virtual_maps"
postfix:mailq_path	Default = "/usr/bin/mailq"
postfix:recipient_delimiter	Default = no
postfix:masquerade_exceptions	Default = ""
postfix:delay_notice_recipient	Default = "postmaster"
postfix:smtp_helo_name	Default = "$myhostname"
postfix:flush_service_name	Default = "flush"
postfix:service_throttle_time	Default = "60s"
postfix:import_environment	Default = "MAIL_CONFIG MAIL_DEBUG MAIL_LOGTAG TZ XAUTHORITY DISPLAY"
postfix:sun_mailtool_compatibility	Default = no
postfix:authorized_verp_clients	Default = "$mynetworks"
postfix:debug_peer_list	Default = ""
postfix:mime_boundary_length_limit	Default = 2048
postfix:initial_destination_concurrency	Default = 5
postfix:parent_domain_matches_subdomains	Default = "debug_peer_list,fast_flush_domains,mynetworks,permit_mx_backup_networks,qmqpd_authorized_clients,relay_domains,smtpd_access_maps"
postfix:setgid_group	Default = "postdrop"
postfix:mime_header_checks	Default = "$header_checks"
postfix:smtpd_etrn_restrictions	Default = ""
postfix:relay_transport	Default = "relay"
postfix:inet_interfaces	Default = "localhost"
postfix:smtpd_sender_restrictions	Default = ""
postfix:delay_warning_time	Default = "0h"
postfix:alias_maps	Default = "hash:/etc/aliases"
postfix:sender_canonical_maps	Default = ""

Parameter (mail:)	Description
postfix:trigger_timeout	Default = "10s"
postfix:newaliases_path	Default = "/usr/bin/newaliases"
postfix:default_rbl_reply	Default = "$rbl_code Service unavailable; $rbl_class [$rbl_what] blocked using rbl_domain{rbl_reason?; $rbl_reason}"
postfix:alias_database	Default = "hash:/etc/aliases"
postfix:qmgr_message_recipient_limit	Default = 20000
postfix:extract_recipient_limit	Default = 10240
postfix:header_checks	Default = 0
postfix:syslog_facility	Default = "mail"
postfix:luser_relay	Default = ""
postfix:maps_rbl_domains:_array_index:0	Default = ""
postfix:deliver_lock_attempts	Default = 20
postfix:smtpd_data_restrictions	Default = ""
postfix:smtpd_pw_server_security_options:_array_index:0	Default = "none"
postfix:ipc_idle	Default = "100s"
postfix:mail_version	Default = "2.0.7"
postfix:transport_retry_time	Default = "60s"
postfix:virtual_mailbox_limit	Default = 51200000
postfix:smtpd_noop_commands	Default = 0
postfix:mail_release_date	Default = "20030319"
postfix:append_at_myorigin	Default = yes
postfix:body_checks_size_limit	Default = 51200
postfix:qmgr_message_active_limit	Default = 20000
postfix:mail_name	Default = "Postfix"
postfix:masquerade_classes	Default = "envelope_sender, header_sender, header_recipient"
postfix:allow_min_user	Default = no
postfix:smtp_randomize_addresses	Default = yes
postfix:alternate_config_directories	Default = no
postfix:allow_percent_hack	Default = yes
postfix:process_id_directory	Default = "pid"
postfix:strict_rfc821_envelopes	Default = no

Parameter (mail:)	Description
postfix:fallback_transport	Default = 0
postfix:owner_request_special	Default = yes
postfix:default_transport	Default = "smtp"
postfix:biff	Default = yes
postfix:relay_domains_reject_code	Default = 554
postfix:smtpd_delay_reject	Default = yes
postfix:lmtp_quit_timeout	Default = "300s"
postfix:lmtp_mail_timeout	Default = "300s"
postfix:fast_flush_purge_time	Default = "7d"
postfix:disable_verp_bounces	Default = no
postfix:lmtp_skip_quit_response	Default = no
postfix:daemon_directory	Default = "/usr/libexec/postfix"
postfix:default_destination_recipient_limit	Default = 50
postfix:smtp_skip_quit_response	Default = yes
postfix:smtpd_recipient_limit	Default = 1000
postfix:virtual_gid_maps	Default = ""
postfix:duplicate_filter_limit	Default = 1000
postfix:rbl_reply_maps	Default = ""
postfix:relay_recipient_maps	Default = 0
postfix:syslog_name	Default = "postfix"
postfix:queue_service_name	Default = "qmgr"
postfix:transport_maps	Default = ""
postfix:smtp_destination_concurrency_limit	Default = "$default_destination_concurrency_limit"
postfix:virtual_mailbox_lock	Default = "fcntl"
postfix:qmgr_fudge_factor	Default = 100
postfix:ipc_timeout	Default = "3600s"
postfix:default_delivery_slot_discount	Default = 50
postfix:relocated_maps	Default = ""
postfix:max_use	Default = 100
postfix:default_delivery_slot_cost	Default = 5
postfix:default_privs	Default = "nobody"
postfix:smtp_bind_address	Default = no
postfix:nested_header_checks	Default = "$header_checks"
postfix:canonical_maps	Default = no

Parameter (mail:)	Description
postfix:debug_peer_level	Default = 2
postfix:in_flow_delay	Default = "1s"
postfix:smtpd_junk_command_limit	Default = 100
postfix:program_directory	Default = "/usr/libexec/postfix"
postfix:smtp_quit_timeout	Default = "300s"
postfix:smtp_mail_timeout	Default = "300s"
postfix:minimal_backoff_time	Default = "1000s"
postfix:queue_file_attribute_count_limit	Default = 100
postfix:body_checks	Default = no
postfix:smtpd_client_restrictions:_array_index:0	Default = ""
postfix:mydestination:_array_index:0	Default = "$myhostname"
postfix:mydestination:_array_index:1	Default = "localhost.$mydomain"
postfix:error_service_name	Default = "error"
postfix:smtpd_sasl_security_options:_array_index:0	Default = "noanonymous"
postfix:smtpd_null_access_lookup_key	Default = "<>"
postfix:virtual_uid_maps	Default = ""
postfix:smtpd_history_flush_threshold	Default = 100
postfix:smtp_pix_workaround_threshold_time	Default = "500s"
postfix:showq_service_name	Default = "showq"
postfix:smtp_pix_workaround_delay_time	Default = "10s"
postfix:lmtp_sasl_security_options	Default = "noplaintext, noanonymous"
postfix:bounce_size_limit	Default = 50000
postfix:qmqpd_timeout	Default = "300s"
postfix:allow_mail_to_files	Default = "alias,forward"
postfix:relay_domains	Default = "$mydestination"
postfix:smtpd_banner	Default = "$myhostname ESMTP $mail_name"
postfix:smtpd_helo_required	Default = no
postfix:berkeley_db_read_buffer_size	Default = 131072
postfix:swap_bangpath	Default = yes
postfix:maximal_queue_lifetime	Default = "5d"
postfix:ignore_mx_lookup_error	Default = no
postfix:mynetworks_style	Default = "host"

Parameter (mail:)	Description
postfix:myhostname	Default = "<hostname>"
postfix:default_minimum_delivery_slots	Default = 3
postfix:recipient_canonical_maps	Default = no
postfix:hash_queue_depth	Default = 1
postfix:hash_queue_names:_array_index:0	Default = "incoming"
postfix:hash_queue_names:_array_index:1	Default = "active"
postfix:hash_queue_names:_array_index:2	Default = "deferred"
postfix:hash_queue_names:_array_index:3	Default = "bounce"
postfix:hash_queue_names:_array_index:4	Default = "defer"
postfix:hash_queue_names:_array_index:5	Default = "flush"
postfix:hash_queue_names:_array_index:6	Default = "hold"
postfix:lmtp_tcp_port	Default = 24
postfix:local_command_shell	Default = 0
postfix:allow_mail_to_commands	Default = "alias,forward"
postfix:non_fqdn_reject_code	Default = 504
postfix:maximal_backoff_time	Default = "4000s"
postfix:smtp_always_send_ehlo	Default = yes
postfix:proxy_read_maps	Default = "$local_recipient_maps $mydestination $virtual_alias_maps $virtual_alias_domains $virtual_mailbox_maps $virtual_mailbox_domains $relay_recipient_maps $relay_domains $canonical_maps $sender_canonical_maps $recipient_canonical_maps $relocated_maps $transport_maps $mynetworks"
postfix:propagate_unmatched_extensions	Default = "canonical, virtual"
postfix:smtp_destination_recipient_limit	Default = "$default_destination_recipient_limit"
postfix:smtpd_restriction_classes	Default = ""
postfix:mime_nesting_limit	Default = 100
postfix:virtual_mailbox_maps	Default = ""
postfix:bounce_service_name	Default = "bounce"
postfix:header_size_limit	Default = 102400

Parameter (mail:)	Description
postfix:strict_8bitmime	Default = no
postfix:virtual_transport	Default = "virtual"
postfix:berkeley_db_create_buffer_size	Default = 16777216
postfix:broken_sasl_auth_clients	Default = no
postfix:home_mailbox	Default = no
postfix:content_filter	Default = ""
postfix:forward_path	Default = "$home/.forward${recipient_delimiter}${extension},$home/.forward"
postfix:qmqpd_error_delay	Default = "1s"
postfix:manpage_directory	Default = "/usr/share/man"
postfix:hopcount_limit	Default = 50
postfix:unknown_virtual_alias_reject_code	Default = 550
postfix:smtpd_sender_login_maps	Default = ""
postfix:rewrite_service_name	Default = "rewrite"
postfix:unknown_address_reject_code	Default = 450
postfix:append_dot_mydomain	Default = yes
postfix:command_expansion_filter	Default = "1234567890!@%-_=+:,./abcdefghijklmnopqrstuvwxyzABCDEFGHIJKLMNOPQRSTUVWXYZ"
postfix:default_extra_recipient_limit	Default = 1000
postfix:lmtp_data_done_timeout	Default = "600s"
postfix:myorigin	Default = "$myhostname"
postfix:lmtp_data_init_timeout	Default = "120s"
postfix:lmtp_data_xfer_timeout	Default = "180s"
postfix:smtp_data_done_timeout	Default = "600s"
postfix:smtp_data_init_timeout	Default = "120s"
postfix:smtp_data_xfer_timeout	Default = "180s"
postfix:default_delivery_slot_loan	Default = 3
postfix:reject_code	Default = 554
postfix:command_directory	Default = "/usr/sbin"
postfix:lmtp_rcpt_timeout	Default = "300s"
postfix:smtp_sasl_security_options	Default = "noplaintext, noanonymous"
postfix:access_map_reject_code	Default = 554
postfix:smtp_helo_timeout	Default = "300s"

Parameter (mail:)	Description
postfix:bounce_notice_recipient	Default = "postmaster"
postfix:smtp_connect_timeout	Default = "30s"
postfix:fault_injection_code	Default = 0
postfix:unknown_client_reject_code	Default = 450
postfix:virtual_minimum_uid	Default = 100
postfix:fast_flush_domains	Default = "$relay_domains"
postfix:default_database_type	Default = "hash"
postfix:dont_remove	Default = 0
postfix:expand_owner_alias	Default = no
postfix:max_idle	Default = "100s"
postfix:defer_transports	Default = ""
postfix:qmgr_message_recipient_minimum	Default = 10
postfix:invalid_hostname_reject_code	Default = 501
postfix:fork_attempts	Default = 5
postfix:allow_untrusted_routing	Default = no
imap:tls_cipher_list:_array_index:0	Default = "DEFAULT"
imap:umask	Default = "077"
imap:tls_ca_path	Default = ""
imap:pop_auth_gssapi	Default = yes
imap:sasl_minimum_layer	Default = 0
imap:tls_cert_file	Default = ""
imap:poptimeout	Default = 10
imap:tls_sieve_require_cert	Default = no
imap:mupdate_server	Default = ""
imap:timeout	Default = 30
imap:quotawarn	Default = 90
imap:enable_pop	Default = no
imap:mupdate_retry_delay	Default = 20
imap:tls_session_timeout	Default = 1440
imap:postmaster	Default = "postmaster"
imap:defaultacl	Default = "anyone lrs"
imap:tls_lmtp_key_file	Default = ""
imap:newsprefix	Default = ""
imap:userprefix	Default = "Other Users"
imap:deleteright	Default = "c"
imap:allowplaintext	Default = yes

Parameter (mail:)	Description
imap:pop_auth_clear	Default = no
imap:imapidresponse	Default = yes
imap:sasl_auto_transition	Default = no
imap:mupdate_port	Default = ""
imap:admins:_array_index:0	Default = "cyrus"
imap:plaintextloginpause	Default = 0
imap:popexpiretime	Default = 0
imap:pop_auth_any	Default = no
imap:sieve_maxscriptsize	Default = 32
imap:hashimapspool	Default = no
imap:tls_lmtp_cert_file	Default = ""
imap:tls_sieve_key_file	Default = ""
imap:sievedir	Default = "/usr/sieve"
imap:debug_command	Default = ""
imap:popminpoll	Default = 0
imap:tls_lmtp_require_cert	Default = no
imap:tls_ca_file	Default = ""
imap:sasl_pwcheck_method	Default = "auxprop"
imap:postuser	Default = ""
imap:sieve_maxscripts	Default = 5
imap:defaultpartition	Default = "default"
imap:altnamespace	Default = yes
imap:max_imap_connections	Default = 100
imap:tls_imap_cert_file	Default = ""
imap:sieveusehomedir	Default = no
imap:reject8bit	Default = no
imap:tls_sieve_cert_file	Default = ""
imap:imapidlepoll	Default = 60
imap:srvtab	Default = "/etc/srvtab"
imap:imap_auth_login	Default = no
imap:tls_pop3_cert_file	Default = ""
imap:tls_pop3_require_cert	Default = no
imap:lmtp_overquota_perm_failure	Default = no
imap:tls_imap_key_file	Default = ""
imap:enable_imap	Default = no
imap:tls_require_cert	Default = no

Parameter (mail:)	Description
imap:autocreatequota	Default = 0
imap:allowanonymouslogin	Default = no
imap:pop_auth_apop	Default = yes
imap:partition-default	Default = "/var/spool/imap"
imap:imap_auth_cram_md5	Default = no
imap:mupdate_password	Default = ""
imap:idlesocket	Default = "/var/imap/socket/idle"
imap:allowallsubscribe	Default = no
imap:singleinstancestore	Default = yes
imap:unixhierarchysep	Default = "yes"
imap:mupdate_realm	Default = ""
imap:sharedprefix	Default = "Shared Folders"
imap:tls_key_file	Default = ""
imap:lmtpsocket	Default = "/var/imap/socket/lmtp"
imap:configdirectory	Default = "/var/imap"
imap:sasl_maximum_layer	Default = 256
imap:sendmail	Default = "/usr/sbin/sendmail"
imap:loginuseacl	Default = no
imap:mupdate_username	Default = ""
imap:imap_auth_plain	Default = no
imap:imap_auth_any	Default = no
imap:duplicatesuppression	Default = yes
imap:notifysocket	Default = "/var/imap/socket/notify"
imap:tls_imap_require_cert	Default = no
imap:imap_auth_clear	Default = yes
imap:tls_pop3_key_file	Default = ""
imap:proxyd_allow_status_referral	Default = no
imap:servername	Default = "<hostname>"
imap:logtimestamps	Default = no
imap:imap_auth_gssapi	Default = no
imap:mupdate_authname	Default = ""
mailman:enable_mailman	Default = no

Mail serveradmin Commands

You can use the following commands with the `serveradmin` tool to manage Mail service.

Command (`mail:command=`)	Description
`getHistory`	View a periodic record of file data throughput or number of user connections. See "Viewing Mail Service Statistics" on this page.
`getLogPaths`	Display the locations of Mail service logs. See "Viewing Mail Service Logs" on page 201.
`writeSettings`	Equivalent to the standard `serveradmin settings` command, but also returns a setting indicating whether the service must be restarted. See "Using the serveradmin Tool" on page 50.

Viewing Mail Service Statistics

You can use the `serveradmin getHistory` command to display a log of periodic samples of the number of user connections and the data throughput. Samples are taken once each minute.

To view samples:

```
$ sudo serveradmin command
mail:command = getHistory
mail:variant = statistic
mail:timeScale = scale
Control-D
```

Parameter	Description
`statistic`	The value you want to display. Valid values: `v1`—Number of connected users (average during sampling period) `v2`—Data throughput (bytes/sec)
`scale`	The length of time in seconds, ending with the current time, for which you want to see samples. For example, to see 24 hours of data, you would specify `mail:timeScale = 86400`.

The computer responds with the following output:

```
mail:nbSamples = <samples>
mail:v2Legend = "throughput"
mail:samplesArray:_array_index:0:vn = <sample>
mail:samplesArray:_array_index:0:t = <time>
mail:samplesArray:_array_index:1:vn = <sample>
mail:samplesArray:_array_index:1:t = <time>
[...]
```

```
mail:samplesArray:_array_index:i:vn = <sample>
mail:samplesArray:_array_index:i:t = <time>
mail:v1Legend = "connections"
afp:currentServerTime = <servertime>
```

Value displayed by `getHistory`	Description
`<samples>`	The total number of samples listed.
`<sample>`	The numerical value of the sample. For connections (`v1`), this is integer average number of users. For throughput, (`v2`), this is integer bytes per second.
`<time>`	The time when the sample was measured. A standard UNIX time (number of seconds since September 1, 1970). Samples are taken every 60 seconds.

Viewing Mail Service Logs

You can use `tail` or another file-listing tool to view the contents of Mail service logs.

To view the latest entries in a log:

```
$ tail log-file
```

You can use the `serveradmin getLogPaths` command to see where Mail service logs are located.

To view the log locations:

```
$ sudo serveradmin command mail:command = getLogPaths
```

The computer responds with the following output:

```
mail:Server Log = <server-log>
mail:Lists smtp = <delivery-log>
mail:Amavisd Log = <amavis-log>
mail:Virus DB Log = <freshclam-log>
mail:Lists subscribe = <subscriptions-log>
mail:Lists smtp-failure = <failures-log>
mail:Lists post = <postings-log>
mail:Lists error = <listerrors-log>
mail:POP Log = <pop-log>
mail:SMTP Log = <smtp-log>
mail:Lists qrunner = <lists-log>
mail:Virus Log = <clamav-log>
mail:IMAP Log = <imap-log>
```

Value	Description
`<server-log>`	The location of the server log. Default = `"/var/log/mailaccess.log"`
`<delivery-log>`	The location of the Mailing Lists Delivery log. Default = `"/var/mailman/logs/smtp"`

Value	Description
<amavis-log>	The location of the mail filtering log. Default = "/var/log/amavis.log"
<freshclam-log>	The location of the virus definition updates log. Default = "/var/log/freshclam.log"
<subscriptions-log>	The location of the Mailing Lists Subscriptions log. Default = "/var/mailman/logs/subscribe"
<failures-log>	The location of the Mailing Lists Delivery Failures log. Default = "/var/mailman/logs/smtp-failure"
<postings-log>	The location of the Mailing Lists Postings log. Default = "/var/mailman/logs/post"
<listerrors-log>	The location of the Mailing Lists Error log. Default = "/var/mailman/logs/error"
<pop-log>	The location of the POP log. Default = "/var/log/mailaccess.log"
<smtp-log>	The location of the server log. Default = "/var/log/mail.log"
<lists-log>	The location of the Mailing Lists log. Default = "/var/mailman/logs/qrunner"
<clamav-log>	The location of the virus scanning log. Default = "/var/log/clamav.log"
<imap-log>	The location of the IMAP log. Default = "/var/log/mailaccess.log"

Backing Up Mail Files

When talking about mail-related backup, IMAP mailboxes are the first thing that come to mind. In addition, you might want to back up configuration files for Cyrus and Postfix. The value of backing up the configuration files is clear: it saves you time when reconfiguring your server if it powers down unexpectedly.

The Server Admin tearoff sheets include configuration information and can be backed up instead of the separate configuration files, unless you manually modified configuration files to include additional configuration not available through Server Admin.

Postfix spool files act as temporary storage and are constantly changing. Backing up and restoring these files can lead to double delivery of mail to users.

To back up the mail database, stop Mail service first. Then copy the following files and folders to a backup destination:

- Cyrus database (/var/imap)
- IMAP folders (/var/spool/imap)
- Cyrus configuration file (/etc/imapd.conf)

- Postfix configuration file (/etc/postfix/main.cf)

The largest database is the mailbox folders database. Each mailbox folder contains the following files:

- Message files—There is one file per message. The file name of each message is the message's UID followed by a period. The UID is a unique ID given to each message.
- cyrus.header—This file contains a magic number and variable-length information about the mailbox.
- cyrus.index—This file contains fixed-length information about the mailbox and each message in the mailbox.
- cyrus.cache—This file contains variable-length information about each message in the mailbox.
- cyrus.seen—This file contains variable-length state information about each reader of the mailbox.

Setting Up SSL for Mail Service

Mail service requires some configuration to provide Secure Sockets Layer (SSL) connections automatically. The basic steps are as follows:

1. Generate a Certificate Signing Request (CSR) and create a keychain.
2. Obtain an SSL certificate from an issuing authority.
3. Import the SSL certificate into the keychain.
4. Create a password file.

These steps are explained in the following sections.

Generating a CSR and Creating a Keychain

To begin configuring Mail service for SSL connections, you generate a CSR and create a keychain by using the `certtool` tool. A CSR is a file that provides information needed to issue an SSL certificate.

To generate a CSR and create a keychain:

1. Log in to the server as root.
2. In the Terminal application, enter the following commands:

```
$ cd /private/var/root/Library/Keychains/
$ /usr/bin/certtool r csr.txt k=certkc c
```

This use of the `certtool` tool begins an interactive process that generates a CSR in the file csr.txt and creates a keychain named certkc.

3. In the New Keychain Passphrase dialog that appears, enter a password for the keychain you're creating, enter the password a second time to verify it, and click OK.

Remember this password, because later you must supply it again.

4 When "Enter key and certificate label:" appears in the Terminal window, enter a one-word key, a blank space, and a one-word certificate label, and then press Return.

For example, you could enter your organization's name as the key and `mailservice` as the certificate label.

The following output appears.

```
Please specify parameters for the key pair you will generate.
  r  RSA
  d  DSA
  f  FEE
Select key algorithm by letter:
```

5 Enter `r`, and then press Return.

The following output appears.

```
Valid key sizes for RSA are 512..2048; default is 512
Enter key size in bits or CR for default:
```

6 Enter a key size, and then press Return.

Larger key sizes are more secure, but require more processing time on your server. Key sizes smaller than 1024 aren't accepted by some certificate-issuing authorities.

The following output appears.

```
You have selected algorithm RSA, key size (size entered above) bits.
OK (y/anything)?
```

7 Enter `y`, and then press Return.

The following output appears.

```
Enter cert/key usage (s=signing, b=signing AND encrypting):
```

8 Enter `b`, and then press Return.

The following output appears.

```
...Generating key pair...
Please specify the algorithm with which your certificate will be signed.
  5  RSA with MD5
  s  RSA with SHA1
Select signature algorithm by letter:
```

9 Enter `s`, and then press Return.

The following output appears.

```
You have selected algorithm RSA with SHA1.
OK (y/anything)?
```

10 Enter `y`, and then press Return.

The following output appears.

```
...creating CSR...
Enter challenge string:
```

11 Enter a phrase or random text, and then press Return.

The following output appears.

```
For Common Name, enter the server's DNS name, such as server.example.com.
For Country, enter the country in which your organization is located.
For Organization, enter the organization to which your domain name is
     registered.
For Organizational Unit, enter something similar to a department name.
For State/Province, enter the full name of your state or province.
```

12 Enter the correct information for each prompt, which requests the components of the certificate's Relative Distinguished Name (RDN), and press Return after each entry.

The following output appears.

```
Is this OK (y/anything)?
```

13 Enter `y`, and then press Return.

The following output appears.

```
Wrote (n) bytes of CSR to csr.txt
```

When you see a message about writing to csr.txt, you have generated a CSR and created the keychain that Mail service needs for SSL connections.

14 Log out from the server.

Note: You can use the `security` command to administer keychains and manipulate keys and certificates. For more information about this command, see its man page.

Obtaining an SSL Certificate

After generating a CSR and a keychain, you continue configuring Mail service for automatic SSL connections by purchasing an SSL certificate from a certificate authority such as Verisign or Thawte. You can do this by completing a form on the certificate authority's website.

When prompted for your CSR, open the csr.txt file using a text editor, such as TextEdit. Then, copy and paste the contents of the file into the appropriate field on the certificate authority's website. The websites for these certificate authorities are at:

- www.verisign.com
- www.thawte.com

When you receive your certificate, save it in a text file named sslcert.txt. You can save this file with the TextEdit application. Make sure that the file is plain text, not rich text, and that it contains only the certificate text.

Importing an SSL Certificate into the Keychain

To import an SSL certificate into a keychain, use the `certtool` tool. This continues the process of configuring Mail service for automatic SSL connections.

To import an SSL certificate into the keychain:

1. Log in to the server as root.
2. Open the Terminal application.
3. Go to the folder where the saved certificate file is located.

   For example, if the certificate file is saved on the desktop of the root user, enter `cd /private/var/root/Desktop` and press Return.
4. Enter the following command, and then press Return:

   ```
   $ certtool i sslcert.txt k=certkc
   ```

   Using `certtool` this way imports a certificate from the file named `sslcert.txt` into the keychain named `certkc`.

   A message on screen confirms that the certificate was successfully imported.

   ```
   ...certificate successfully imported.
   ```
5. Log out from the server.

Accessing Server Certificates

Server Admin keeps a centralized store of your server's certificates for ease of use and management. You can use `certadmin` to access this information from the command line. `certadmin` manipulates the list of certificates stored in the System keychain.

To view the certificates in the System keychain:

```
$ sudo certadmin list
```

By default, `certadmin` prints the Common Name field of each certificate separated by newlines. Adding the option `-x` or `--xml` prints the certificate list to screen as an XML property list (plist).

To export a certificate to OpenSSL:

```
$ sudo certadmin export
```

For more information, see the `certadmin` man page.

Creating a Password File

To create a password file, use TextEdit, and then change the privileges of the file using the Terminal application. This file contains the password you specified when you created the keychain. Mail service uses the password file to unlock the keychain that contains the SSL certificate. Mail service is now configured for automatic SSL connections.

To create a password file:

1. Log in to the server as root.
2. In TextEdit, create a file and enter the password as you entered it when you created the keychain.

   Don't press Return after entering the password.
3. Make the file plain text by choosing Make Plain Text from the Format menu.
4. Save the file, naming it cerkc.pass.
5. Move the file to the root keychain folder.

   The path is /private/var/root/Library/Keychains/.

   To see the root keychain folder in the Finder, choose Go to Folder from the Go menu, enter /private/var/root/Library/Keychains/, and then click Go.
6. In the Terminal application, change the access privileges to the password file so only root can read and write to this file.

   Do this by entering the following commands, pressing Return after each one:

   ```
   cd /private/var/root/Library/Keychains/
   chmod 600 certkc.pass
   ```

   Mail service can now use SSL for secure IMAP connections.
7. Log out from the server.

Note: If Mail service is running, stop it and start it again to make it recognize the new certificate keychain.

Configuring Mailboxes

Mail service keeps track of incoming mail messages with a small database (BerkeleyDB 4.2.52), but the database doesn't contain the messages. Mail service stores each message as a separate file in a mail folder for each user. This is the user's mailbox.

Incoming mail is stored on the startup disk in the /var/spool/imap/user/*username* folder. Cyrus puts a database index file in the folder of user messages. You can change the location of mail folders and database indexes to another folder, disk, or disk partition. Cyrus mail storage can also be split across multiple partitions. This can be done to scale Mail service, or to facilitate data backup.

The `cyradm` tool is included with Mac OS X Server. It is an administration shell for Cyrus, the IMAP Mail service package, and communicates with the `Cyrus::IMAP::Admin` Perl module. You can use `cyradm` to create, delete, or rename mailboxes, as well as set ACLs for mailboxes (for mail clients that support them).

Things to note:

- `cyradm` is a limited shell. It supports shell-style redirection, but does not understand pipes.
- `cyradm` can be used interactively or be scripted, but Perl scripting with `Cyrus::IMAP::Admin` is more flexible.
- You must escape spaces in file or folder names with a backslash (\), just as you would in a shell.

For a complete list of commands, see the `cyradm` man page.

Enabling Sieve Scripting

Mac OS X Server supports Sieve scripting for mail processing. Sieve is an Internet standard mail filtering language for server-side filtering. Sieve scripts interact with incoming mail before final delivery.

Sieve scripting acts much like the rules in various mail programs, to sort or process mail based on user-defined criteria. In fact, some mail clients use Sieve for client-side mail processing. Sieve can provide such functions as vacation notifications, message sorting, and mail forwarding, among other things.

Sieve scripts are kept for each user on the mail server in the /usr/sieve/<first letter of username>/<user> folder.

The folder is owned by the Mail service, so users normally don't have access to it and can't put their scripts there for mail processing.

For security purposes, users and administrators upload their scripts to a Sieve process (timsieved) which transports the scripts to the mail process for use. There are various ways of getting the scripts to timsieved, such as Perl shell scripts ("sieveshell") and even some mail clients.

Enabling Sieve Support

For Sieve to function, you must enable its communications port. Sieve has the vacation extension added by default. All scripts must be placed in the central script repository at /usr/sieve/, and Sieve scripts cannot be used to process mail for mail aliases set up in Workgroup Manager; you must use Postfix-style aliases.

To enable Sieve support:

1. Add the following entry to the services file in /etc/, using a text editor.

```
sieve 2000/tcp #Sieve mail filtering
```

2. Reload the Mail service.

Sample Sieve Scripts

The following scripts are examples of common scripts a user might want to use.

Vacation Notification Script

```
#--------
# This is a sample script for vacation rules.
# Read the comments following the pound/hash to find out
# what the script is doing.
#---------
#
# Make sure the vacation extension is used.
require "vacation";
# Define the script as a vacation script
  vacation
# Send the vacation response to any given sender only once every seven days
     no matter how many messages are sent from him.
  :days 7
#For every message sent to these addresses
  :addresses ["bob@example.com", "robert.fakeuser@server.com"]
# Make a message with the following subject
  :subject "Out of Office Reply"
# And make the body of the message the following
"I'm out of the office and will return on December 31. I won't be able to
     replay until 6 months after that. Love, Bob.";
# End of Script
```

Self-Defined Forwarding Script

```
#--------
# This is a sample script to illustrate how Sieve could be used
# to let users handle their own mail forwarding needs.
# Read the comments following the pound/hash to find out what the
# script is doing.
#---------
#
# No need to add any extension. 'redirect' is built-in.
# Redirect all my incoming mail to the listed address
redirect "my-other-address@example.com";
# But keep a copy of it on the IMAP server keep;
# End of script
```

Basic Sort and Anti-Junk Mail Filter Script

```
#--------
# This is a sample script to show discarding and filing.
# Read the comments following the pound/hash to find out
# what the script is doing
#---------
#
# Make sure filing and rejection are enabled
require "fileinto";
#
```

```
# If it's from my mom...
if header ["From"] :contains ["Mom"]{
# send it to my home email account
redirect "home-address@example.com";
  }
#
# If the subject line has a certain keyword...
else if header "Subject" :contains "daffodil" {
# forward it to the postmaster
     forward "postmaster@server.edu";
  }
#
# If the junk mail filter has marked this as junk...
else if header :contains ["X-Spam-Flag"] ["YES"]{
# throw it out
     discard;
  }
#
# If the junk mail filter thinks this is probably junk
else if header :contains ["X-Spam-Level"] ["****"]{
# put it in my junkmail box for me to check
     fileinto "INBOX.JunkMail";
  }
#
# for all other cases...
else {
# put it in my inbox
     fileinto "INBOX";
  }
# End of script
```

Sieve Scripting Resources

Sieve's complete syntax, commands, and arguments are found in IETF RFC 3028, located at www.ietf.org/rfc/rfc3028.txt?number=3028.

Configuring and Managing Web Technologies

13

Use this chapter to learn the commands to configure and manage Web service and the web components on your server.

Web technologies in Mac OS X Server consist of several components that provide a flexible and scalable server environment. This chapter covers the commands that are used to configure and manage these web technologies, referred to as Web service.

For more information, see *Web Technologies Administration*.

Understanding Web Service

Apple's Web service is based primarily on Apache. Apache is one of the most popular and versatile web servers, and is a community-based, open-source project. Apple has extended Apache in a number of ways to implement Mac OS X–specific features.

Mac OS X Server includes the following versions of the Apache HTTP Server:

- Version 1.3—This is the officially supported version on Mac OS X Server. It is a well-tested, stable, reliable software package used worldwide for many years. In this chapter, references to the Apache server refer to this version.
- Version 2.0—This is an evaluation version that includes several new features, including multithreading and an improved API for plug-in modules. However, the API changes make many third-party modules incompatible with this version.

The locations of Apache 1.3 files on Mac OS X Server are slightly different from a traditional Apache installation. The following table identifies the major folders.

Files	Location
Application binaries	/usr/sbin/
CGI applications	/Library/WebServer/CGI-Executables/
Configuration files	/etc/httpd/
Default documents	/Library/WebServer/Documents/

Files	Location
Log files	/var/log/httpd/
Loadable modules	/usr/libexec/httpd/

Apache 2.0 files are in the /etc/apache2/ folder.

The main configuration file for the Apache web server is /etc/httpd/httpd.conf. The Apache web server (`httpd`) reads this file during startup. In addition, Mac OS X Server maintains a configuration file for each website it hosts. Mac OS X Server stores the website-specific configuration files in the /etc/httpd/sites/ folder.

To change settings that aren't in Server Admin, such as the maximum number of requests that an `httpd` child can process before it dies, edit the httpd.conf file. Each section of the httpd.conf file contains instructions for how to safely edit its options.

Important: To avoid misconfiguring Web service, do not modify the httpd.conf file manually when the Web Settings pane of Server Admin is open. For more information about Apache, see www.apache.org.

Managing Web Service

The following sections describe basic Apache Web service functions.

Starting and Stopping Web Service

To start Web service:

```
$ sudo serveradmin start web
```

To stop Web service:

```
$ sudo serveradmin stop web
```

Checking Web Service Status

To see if Web service is running:

```
$ sudo serveradmin status web
```

To see complete Web service status:

```
$ sudo serveradmin fullstatus web
```

Viewing Web Settings

To view your server's Web service configuration, use `serveradmin`. However, to work with the Web service from the command line, you'll probably find it more straightforward to work directly with the underlying Apache web server.

To view a setting:

```
$ sudo serveradmin settings web:setting
```

To view a group of settings:

You can view a group of settings that have part of their names in common by entering as much of the name as you want, stopping at a colon (:), and entering an asterisk (*) as a wildcard for the remaining parts of the name. For example:

```
$ sudo serveradmin settings web:IFModule:_array_id:mod_alias.c:*
```

To view all Web service settings:

```
$ sudo serveradmin settings web
```

Changing Web Settings

You can use `serveradmin` to modify your server's Web service configuration. However, if you want to work with the Web service from the command line, you'll probably find it more straightforward to work directly with the underlying Apache web server.

Apache Settings and serveradmin

The parameters are written differently in the Apache configuration file than in `serveradmin`. For example, this block of Apache configuration parameters:

```
<IfModule mod_macbinary_apple.c>
  MacBinary On
  MacBinaryBlock html shtml perl pl cgi jsp php phps asp scpt
  MacBinaryBlock htaccess
</IfModule>
```

appears as this block of configuration parameters in `serveradmin`:

```
web:IfModule:_array_id:mod_macbinary_apple.c:MacBinary = yes
web:IfModule:_array_id:mod_macbinary_apple.c:MacBinaryBlock:_array_index:0 =
     "html shtml perl pl cgi jsp php phps asp scpt"
web:IfModule:_array_id:mod_macbinary_apple.c:MacBinaryBlock:_array_index:1 =
     "htaccess".
```

Changing Settings Using serveradmin

Use the `serveradmin` tool to change Web service settings.

To change a setting:

```
$ sudo serveradmin settings web:setting = value
```

Parameter	Description
setting	A Web service setting. To see a list of available settings, enter: `$ sudo serveradmin settings web`
value	An appropriate value for the setting.

To change several settings:

```
$ sudo serveradmin settings
web:setting = value
web:setting = value
web:setting = value
[...]
Control-D
```

Web serveradmin Commands

To manage Web service, use the following commands with the `serveradmin` tool.

Command (`web:command=`)	Description
`getHistory`	View Web service statistics. See "Viewing Service Statistics" on page 214.
`getLogPaths`	Find the access and error logs for each hosted site. See "Viewing Service Logs and Log Paths" on this page.
`getSites`	View existing sites. See "Listing Hosted Sites" on this page.

Listing Hosted Sites

To view a list of sites hosted by the server, along with basic settings and status, use the `serveradmin getSites` command.

To view sites:

```
$ sudo serveradmin command web:command = getSites
```

You can also view the sites using Apache, with the following command:

```
$ httpd -S
```

Viewing Service Logs and Log Paths

To view the contents of Web service access and error logs for each site hosted by the server and to view log paths, use `tail` or another file listing tool.

To view the latest entries in a log:

```
$ tail log-file
```

To see where the current error and activity logs for each site are located, use the `serveradmin getLogPaths` command.

To view log paths:

```
$ sudo serveradmin command web:command = getLogPaths
```

Viewing Service Statistics

To display a log of periodic samples of the number of requests, cache performance, and data throughput, use the `serveradmin getHistory` command. Samples are taken once each minute.

To view samples:

```
$ sudo serveradmin command
web:command = getHistory
web:variant = statistic
web:timeScale = scale
Control-D
```

Parameter	Description
statistic	The value you want to display. Valid values: • v1—Number of requests per second • v2—Throughput (bytes/sec) • v3—Cache requests per second • v4—Cache throughput (bytes/sec)
scale	The length of time in seconds, ending with the current time, that you want to see samples for. For example, to see 30 minutes of data, you would specify `qtss:timeScale = 1800`.

The computer responds with the following output:

```
web:nbSamples = <samples>
web:samplesArray:_array_index:0:vn = <sample>
web:samplesArray:_array_index:0:t = <time>
web:samplesArray:_array_index:1:vn = <sample>
web:samplesArray:_array_index:1:t = <time>
[...]
web:samplesArray:_array_index:i:vn = <sample>
web:samplesArray:_array_index:i:t = <time>
web:vnLegend = "<legend>"
web:currentServerTime = <servertime>
```

Value displayed by `getHistory`	Description
`<samples>`	The total number of samples listed.
`<legend>`	A textual description of the selected statistic: • `"REQUESTS_PER_SECOND"` for `v1` • `"THROUGHPUT"` for `v2` • `"CACHE_REQUESTS_PER_SECOND"` for `v3` • `"CACHE_THROUGHPUT"` for `v4`
`<sample>`	The numerical value of the sample.
`<time>`	The time the sample was measured. A standard UNIX time (number of seconds since September 1, 1970). Samples are taken every 60 seconds.

Example Script for Adding a Website

The following script shows how you can use `serveradmin` to add a website to the server's Web service configuration. The script uses two files:

- `addsite`—The script you run. It accepts values for the site's IP address, port number, server name, and root folder, and uses `sed` to substitute these values in the addsite.in file. This is then sent to `serveradmin`.
- `addsite.in`—Contains the settings (with placeholders for values you provide when you run `addsite`) used to create the website.

The addsite File

```
sed -es#_ipaddr#$1#g -es#_port#$2#g -es#_servername#$3#g
     -es#_docroot#$4#g ./addsite.in | /usr/sbin/serveradmin
```

The addsite.in File

```
web:Sites:_array_id:_ipaddr\:_port__servername = create
web:Sites:_array_id:_ipaddr\:_port__servername:Listen:_array_index:0 =
     "_ipaddr:_port"
web:Sites:_array_id:_ipaddr\:_port__servername:ServerName = _servername
web:Sites:_array_id:_ipaddr\:_port__servername:ServerAdmin =
     admin@_servername
web:Sites:_array_id:_ipaddr\:_port__servername:DirectoryIndex:_array_index:0
     = "index.html"
web:Sites:_array_id:_ipaddr\:_port__servername:DirectoryIndex:_array_index:1
     = "index.php"
web:Sites:_array_id:_ipaddr\:_port__servername:WebMail = yes
web:Sites:_array_id:_ipaddr\:_port__servername:CustomLog:_array_index:0:
     Format = "%{User-agent}i"
web:Sites:_array_id:_ipaddr\:_port__servername:CustomLog:_array_index:0:
     enabled = yes
web:Sites:_array_id:_ipaddr\:_port__servername:CustomLog:_array_index:0:
     ArchiveInterval = 0
web:Sites:_array_id:_ipaddr\:_port__servername:CustomLog:_array_index:0:
     Path = "/private/var/log/httpd/access_log"
web:Sites:_array_id:_ipaddr\:_port__servername:CustomLog:_array_index:0:
     Archive = yes
web:Sites:_array_id:_ipaddr\:_port__servername:Directory:_array_id:
     /Library/WebServer/Documents:Options:Indexes = yes
web:Sites:_array_id:_ipaddr\:_port__servername:Directory:_array_id:
     /Library/WebServer/Documents:Options:ExecCGI = no
web:Sites:_array_id:_ipaddr\:_port__servername:Directory:_array_id:
     /Library/WebServer/Documents:AuthName = "Test Site"
web:Sites:_array_id:_ipaddr\:_port__servername:ErrorLog:ArchiveInterval = 0
web:Sites:_array_id:_ipaddr\:_port__servername:ErrorLog:Path = "/private/
     var/log/httpd/error_log"
web:Sites:_array_id:_ipaddr\:_port__servername:ErrorLog:Archive = no
web:Sites:_array_id:_ipaddr\:_port__servername:Include:_array_index:0 = "/
     etc/httpd/httpd_squirrelmail.conf"
web:Sites:_array_id:_ipaddr\:_port__servername:enabled = yes
```

```
web:Sites:_array_id:_ipaddr\:_port__servername:ErrorDocument:_array_index:0:
     StatusCode = 404
web:Sites:_array_id:_ipaddr\:_port__servername:ErrorDocument:_array_index:0:
     Document = "/nwesite_notfound.html"
web:Sites:_array_id:_ipaddr\:_port__servername:LogLevel = "warn"
web:Sites:_array_id:_ipaddr\:_port__servername:IfModule:_array_id:mod_ssl.c:
     SSLEngine = no
web:Sites:_array_id:_ipaddr\:_port__servername:IfModule:_array_id:mod_ssl.c:
     SSLPassPhrase = ""
web:Sites:_array_id:_ipaddr\:_port__servername:IfModule:_array_id:mod_ssl.c:
     SSLLog = "/private/var/log/httpd/ssl_engine_log"
web:Sites:_array_id:_ipaddr\:_port__servername:DocumentRoot = "_docroot"
web:Sites:_array_id:_ipaddr\:_port__servername
```

To run the script:

```
$ addsite ipaddress port name root
```

Parameter	Description
ipaddress	The IP address for the site
port	The port number to be used for HTTP access to the site
name	The name of the site
root	The root folder for the site's files and subfolders

If you get the message `command not found` when you try to run the script, precede the command with the full path to the script file. For example:

```
/Users/admin/Documents/addsite 10.0.0.2 80 corpsite /Users/webmaster/Sites/
     corpsite
```

Or, use `cd` to change to the folder that contains the file and precede the command with `./`. For example:

```
$ cd /Users/admin/Documents
$ ./addsite 10.0.0.2 80 corpsite /users/webmaster/sites/corpsite
```

Tuning Server Performance

A number of factors can affect server performance: CGI scripts can grow too large, database queries exhaust your computer's resources, there can be too much network traffic, and so on.

Apache provides a basic benchmarking tool, `ab`. You can use `ab` to simulate hits to your web server and get an idea of how long it takes your website to respond, as well as other valuable statistics.

The following command simulates 1,000 requests to the specified URL with the user name and password provided.

```
$ ab -n 1000 -c 1 -A user:password www.studentnumber.example.com/
```

Apache Tomcat

Mac OS X Server comes with Apache Tomcat, the open source servlet container developed by Sun Microsystems. Tomcat runs as part of the Java process.

To start Apache Tomcat:

```
$ su /Library/Tomcat/6.0/bin/startup.sh
```

Note: If you start Tomcat manually, it is not reflected in the Server Admin application. Additionally, it is not monitored by the `launchd` process.

By default, Tomcat uses port 9006. Tomcat comes with several example servlets. You can access these servlets at localhost:9006/examples/servlets/. The example servlets reside in /Library/Tomcat/6.0/webapps/WEB-INF. To deploy your own servlets, place them in /Library/Tomcat/webapps/WEB-INF.

Tomcat's configuration information is in /Library/Tomcat/6.0/conf/. For more information about Tomcat, see jakarta.apache.org/tomcat.

The MySQL Database

Mac OS X Server includes MySQL, a popular open source database you can use with web applications. This database is well-suited for common web-related tasks, such as managing content and implementing discussion boards and guestbooks.

MySQL is one service you can manage using Server Admin. You can use Server Admin to start and stop MySQL (`mysqld`), change the database location, set MySQL's root password, enable or disable network connections, and view MySQL's logs. You can perform these actions from the command line.

When you start MySQL for the first time, or when you change the location of the database using Server Admin (or the serveradmin command line tool), a new MySQL database is initialized for you and MySQL is ready for use.

Mac OS X Server stores the files of the preinstalled MySQL version in the file system, with executables in /usr/bin/ and /usr/libexec/, man pages in /usr/share/man/, and other parts in /usr/share/mysql/. In addition, the MySQL configuration file resides in /etc/my.conf/ and the MySQL database in /var/mysql/.

A default configuration file, appropriate to the memory size of your system, is installed automatically. You can alter this configuration file to customize MySQL to your needs. You can find sample MySQL configuration files (with the .cnf file extension) in the /usr/share/mysql/ folder.

MySQL saves the current state automatically, and launches with each system restart if it is running prior to system shut down or reboot.

To set/change the root password:

```
$ sudo /usr/sbin/serveradmin stop mysql
$ sudo /usr/sbin/serveradmin settings mysql:rootPassword = password
$ sudo /usr/sbin/serveradmin start mysql
```

When you set up MySQL service, set up a password for the MySQL root user to protect your server from unauthorized access.

To change the database location:

```
$ sudo /usr/sbin/serveradmin stop mysql
$ sudo /usr/sbin/serveradmin settings mysql:databaseLocation = /path/to/new/
    database/
$ sudo /usr/sbin/serveradmin start mysql
```

MySQL is preconfigured to use /var/mysql/ as the default database location.

By default, changing the database location creates a database at the chosen path if one does not exist at that location.

To move a database to a new location:

```
$ sudo /usr/sbin/serveradmin stop mysql
$ sudo cp -Rp /oldpath/mysql /newpath/
$ sudo /usr/sbin/serveradmin settings mysql:databaseLocation = /newpath/
    mysql
$ sudo /usr/sbin/serveradmin start mysql
```

To set the network option:

```
$ sudo /usr/sbin/serveradmin stop mysql
$ sudo /usr/sbin/serveradmin settings mysql:allowNetwork = yes
```

or

```
$ sudo /usr/sbin/serveradmin settings mysql:allowNetwork = no
$ sudo /usr/sbin/serveradmin start mysql
```

To start mysqld:

```
$ sudo /usr/sbin/serveradmin start mysql
```

The following are useful tools distributed with MySQL. Each has its own man page:

- `mysql_install_db`—Installs the default MySQL database.
- `mysqladmin`—Administers the MySQL database.
- `mysql`—The MySQL database text-based client.

`mysqld_safe`, the `mysqld` parent (`watchdog`) process used in previous releases of Mac OS X Server, is not used in v10.5. Its function has been replaced with `/usr/bin/launchd`.

Do not use `mysqld_safe` to start `mysqld`. It interferes with the operation of launchd.

For more information about setting up and configuring MySQL, see www.mysql.org.

Configuring and Managing Network Services

14

Use this chapter to learn the commands to configure and manage DHCP, DNS, Firewall, NAT, and VPN services.

Mac OS X Server network services add administrative and managerial capabilities to basic networking protocols. This chapter describes the commands used to configure and manage network services.

For more information, see *Network Services Administration*.

Managing Network Services

Mac OS X Server uses the `xinetd` process to manage many UNIX network services, such as FTP, finger, and so on. `xinetd` listens for requests on specific TCP/IP sockets and is a secure replacement for `inetd`. However, because `xinetd` does not handle RPC services well, `inetd` and `xinetd` are included with Mac OS X.

`xinetd` does the same things as `inetd`, with the added security benefits of access control based on source address, destination address, and time, and provides extensive logging, efficient containment of denial-of-service attacks, and the ability to bind services to specific interfaces.

The configuration files for `xinetd` provide a mapping of services to the executable that should be run to service a request for a given service.

For example, if you enable FTP file sharing, the `ftpd` process is not started immediately. Instead, the configuration file is updated to reflect that `xinetd` should listen for `ftp` requests, and when it receives one, it should launch `ftpd` to service the request.

When the first `ftp` request comes in to the computer, `xinetd` receives the request and then launches `ftpd` to handle it. In this way, `xinetd` can keep the number of services running on a computer lower by launching only those that are requested by a client.

`inetd` and `xinetd` have their own configuration files. `inetd` uses one file, `inetd.conf`, to map a service to its executable. Standard services that `inetd` handles are listed in the file.

xinetd uses a different configuration file for each service it provides. In the /etc/xinetd.d/ folder, there are configuration files for each service that xinetd handles. If you were to enable ftp sharing, Mac OS X will modify the configuration file /etc/xinetd.d/ftp. For more information about xinetd, see www.xinetd.org.

Managing DHCP Service

Dynamic Host Configuration Protocol (DHCP) service lets you administer and distribute IP addresses and other configuration information to client computers from your server. When you configure the DHCP server, you assign a block of IP addresses that can be made available to clients.

Each time a client computer configured to use DHCP starts up, it looks for a DHCP server on your network. If a DHCP server is found, the client computer requests an IP address. The DHCP server checks for an available IP address and sends it to the client computer along with a *lease period* (the length of time the client computer can use the address) and configuration information.

Starting and Stopping DHCP Service

To start the service:

```
$ sudo serveradmin start dhcp
```

To stop the service:

```
$ sudo serveradmin stop dhcp
```

Viewing the Status of DHCP Service

To see summary status of the service:

```
$ sudo serveradmin status dhcp
```

To see detailed status of the service:

```
$ sudo serveradmin fullstatus dhcp
```

Viewing DHCP Service Settings

To view a setting:

```
$ sudo serveradmin settings dhcp:setting
```

To view a group of settings:

You can view a group of settings that have part of their names in common by entering as much of the name as you want, stopping at a colon (:), and entering an asterisk (*) as a wildcard for the remaining parts of the name. For example:

```
$ sudo serveradmin settings dhcp:subnets:*
```

To view DHCP configuration settings:

```
$ sudo serveradmin settings dhcp
```

Changing DHCP Service Settings

To change a DHCP setting:

```
$ sudo serveradmin settings dhcp:setting = value
```

Parameter	Description
setting	A DHCP service setting. See the table below.
value	An appropriate value for the setting.

To change several DHCP settings:

```
$ sudo serveradmin settings
dhcp:setting = value
dhcp:setting = value
dhcp:setting = value
[...]
Control-D
```

To view all service settings:

```
$ sudo serveradmin settings dhcp
```

Also see "DHCP Service Settings" on this page and "DHCP Subnet Settings Array" on page 224.

DHCP Service Settings

To change settings for the DHCP service, use the following parameters with the `serveradmin` tool.

Parameter (`dhcp:`)	Description
`logging_level`	`"LOW"` | `"MEDIUM"` | `"HIGH"` Default = `"MEDIUM"` Corresponds to the Log Detail Level pop-up menu in the Logging pane of DHCP service settings in the Server Admin application.
`subnet_status`	Default = `0`
`subnet_defaults:logVerbosity`	`"LOW"` | `"MEDIUM"` | `"HIGH"` Default = `"MEDIUM"`
`subnet_defaults:logVerbosityList:_array_index:`*n*	Available values for the logVerbosity setting. Default = `"LOW"`, `"MEDIUM"`, and `"HIGH"`
`subnet_defaults:WINS_node_type`	Default = `"NOT_SET"`
`subnet_defaults:routers`	Default = `empty_dictionary`
`subnet_defaults:selected_port_key`	Default = `en0`
`subnet_defaults:selected_port_key_list:_array_index:`*n*	An array of available ports.
`subnet_defaults:dhcp_domain_name`	Default = The last portion of the server's host name, for example, `example.com`.

Parameter (dhcp:)	Description
subnet_defaults:dhcp_domain_name_server:_array_index:*n*	Default = The DNS server addresses provided during server setup, as listed in the Network pane of the server's System Preferences.
subnets:_array_id:<subnetID>...	An array of settings for a subnet. <subnetID> is a unique identifier for each subnet. See "DHCP Subnet Settings Array" on this page.

DHCP Subnet Settings Array

An array of settings listed in the following table is included in the DHCP service settings for each subnet you define. You can add a subnet to the DHCP configuration by using serveradmin to add an array of these settings.

About Subnet IDs

In an actual list of settings, <subnetID> is replaced with a unique ID code for the subnet. The IDs generated by the server are random numbers. The only requirement for the ID is that it be unique among the subnets defined on the server.

Subnet Parameter subnets:_array_id:<subnetID>:	Description
descriptive_name	A textual description of the subnet. Corresponds to the Subnet Name field in the General pane of the subnet settings in Server Admin.
dhcp_domain_name	The default domain for DNS searches, for example, example.com. Corresponds to the Default Domain field in the DNS pane of the subnet settings in Server Admin.
dhcp_domain_name_server:_array_index:*n*	The primary WINS server to be used by clients. Corresponds to the Name Servers field in the DNS pane of the subnet settings in Server Admin.
dhcp_enabled	Whether DHCP is enabled for this subnet. Corresponds to the Enable checkbox in the list of subnets in the Subnets pane of the DHCP settings in Server Admin.
dhcp_ldap_url:_array_index:*n*	The URL of the LDAP folder to be used by clients. Corresponds to the Lease URL field in the LDAP pane of the subnet settings in Server Admin.
dhcp_router	The IPv4 address of the subnet's router. Corresponds to the Router field in the General pane of the subnet settings in Server Admin.
lease_time_secs	Lease time in seconds. Default = "3600" Corresponds to the Lease Time pop-up menu and field in the General pane of the subnet settings in Server Admin.
net_address	The IPv4 network address for the subnet.

Subnet Parameter subnets:_array_id:<subnetID>:	Description
net_mask	The subnet mask for the subnet. Corresponds to the Subnet Mask field in the General pane of the subnet settings in Server Admin.
net_range_end	The highest available IPv4 address for the subnet. Corresponds to the Ending IP Address field in the General pane of the subnet settings in Server Admin.
net_range_start	The lowest available IPv4 address for the subnet. Corresponds to the Starting IP Address field in the General pane of the subnet settings in Server Admin.
selected_port_name	The network port for the subnet. Corresponds to the Network Interface pop-up menu in the General pane of the subnet settings in Server Admin.
WINS_NBDD_server	The NetBIOS Datagram Distribution Server IPv4 address. Corresponds to the NBDD Server field in the WINS pane of the subnet settings in Server Admin.
WINS_node_type	The WINS node type. Can be set to: • "" (not set; default) • BROADCAST_B_NODE • PEER_P_NODE • MIXED_M_NODE • HYBRID-H-NODE Corresponds to the NBT Node Type field in the WINS pane of the subnet settings in Server Admin.
WINS_primary_server	The primary WINS server used by clients. Corresponds to the WINS/NBNS Primary Server field in the WINS pane of the subnet settings in Server Admin.
WINS_scope_id	A domain name such as apple.com. Default = "" Corresponds to the NetBIOS Scope ID field in the WINS pane of the subnet settings in Server Admin.
WINS_secondary_server	The secondary WINS server used by clients. Corresponds to the WINS/NBNS Secondary Server field in the WINS pane of the subnet settings in Server Admin.

Adding a DHCP Subnet

To add other subnets to your DHCP configuration, use the `serveradmin settings` command.

You might already have a subnet for each port you enabled when you installed and set up the server. You can use the `serveradmin settings` command to check for subnets the server set up for you (see "Viewing DHCP Service Settings" on page 222).

Note: Include the special first setting (ending with `= create`). This is how you tell `serveradmin` to create the settings array with the specified subnet ID.

To add a subnet:

```
$ sudo serveradmin settings
dhcp:subnets:_array_id:subnetID = create
dhcp:subnets:_array_id:subnetID:WINS_NBDD_server = nbdd-server
dhcp:subnets:_array_id:subnetID:WINS_node_type = node-type
dhcp:subnets:_array_id:subnetID:net_range_start = start-address
dhcp:subnets:_array_id:subnetID:WINS_scope_id = scope-ID
dhcp:subnets:_array_id:subnetID:dhcp_router = router
dhcp:subnets:_array_id:subnetID:net_address = net-address
dhcp:subnets:_array_id:subnetID:net_range_end = end-address
dhcp:subnets:_array_id:subnetID:lease_time_secs = lease-time
dhcp:subnets:_array_id:subnetID:dhcp_ldap_url:_array_index:0 = ldap-server
dhcp:subnets:_array_id:subnetID:WINS_secondary_server = wins-server-2
dhcp:subnets:_array_id:subnetID:descriptive_name = description
dhcp:subnets:_array_id:subnetID:WINS_primary_server = wins-server-1
dhcp:subnets:_array_id:subnetID:dhcp_domain_name = domain
dhcp:subnets:_array_id:subnetID:dhcp_enabled = (yes|no)
dhcp:subnets:_array_id:subnetID:dhcp_domain_name_server:_array_index:0 =
     dns-server-1
dhcp:subnets:_array_id:subnetID:dhcp_domain_name_server:_array_index:1 =
     dns-server-2
dhcp:subnets:_array_id:subnetID:net_mask = mask
dhcp:subnets:_array_id:subnetID:selected_port_name = port
Control-D
```

Parameter	Description
subnetID	A unique number that identifies the subnet. Can be any number not assigned to another subnet on the server. Can include embedded hyphens (-).
dns-server-n	To specify additional DNS servers, add `dhcp_name_server` settings, incrementing `_array_index:`*n* for each additional value.
Other parameters	The standard subnet settings described in "DHCP Subnet Settings Array" on page 224.

Adding a DHCP Static Map

To add a static map to the DHCP configuration, use the `serveradmin` tool. A static DHCP map allows you to map a specific IP address to a computer based on the Ethernet (MAC) address.

To add a static map:

```
$ sudo serveradmin settings
dhcp:static_maps:_array_id:host name:mapID:static map parameter
```

Static Map Parameter	Description
ip_address	IP address of host
name	Host's DNS name
en_address	Host's Ethernet address

About Static Map IDs

In a list of settings, `<mapID>` is replaced with a unique ID code for the map entry. The IDs generated by the server are random numbers. The only requirement for this ID is that it be unique among the static maps defined on the server.

The *mapID* parameter is used by the administrative software; it is ignored by the `bootpd` process that provides the DHCP service. For more information about `bootpd`, see its man page.

Note: Include the special first setting (ending with `= create`). This is how you instruct `serveradmin` to create the settings array with the specified map ID. The static map for a host is identified with the host name, followed by a slash, followed by a unique ID.

To add maps to your DHCP configuration, use the `serveradmin` settings command.

To create a static map:

```
$ sudo serveradmin settings
dhcp:static_maps:_array_id:examplehost/9681BABD-3329-402E-A7AB-F0C3608E231D
     = create
dhcp:static_maps:_array_id:examplehost/9681BABD-3329-402E-A7AB-
     F0C3608E231D:ip_address = "1.2.3.4"
dhcp:static_maps:_array_id:examplehost/9681BABD-3329-402E-A7AB-
     F0C3608E231D:name = "examplehost"
dhcp:static_maps:_array_id:examplehost/9681BABD-3329-402E-A7AB-
     F0C3608E231D:en_address = "00:30:a1:a2:a1:23"
Control-D
```

Viewing the Location of the DHCP Service Log

To view the location of the DHCP service log, use the following command with the `serveradmin` tool.

Command (`dhcp:command=`)	Description
`getLogPaths`	Display the location of the DHCP service log.

To view the log path:

```
$ sudo serveradmin command dhcp:command = getLogPaths
```

The computer responds with the following output:

```
dhcp:systemLog = <system-log>
```

Value	Description
`<system-log>`	The location of the DHCP service log. Default = `/var/logs/system.log`

Viewing the DHCP Service Log

To view the contents of the DHCP service log, use `tail` or another file listing tool.

To view the latest entries in a log:

```
$ tail log-file
```

Managing DNS Service

Domain Name System (DNS) is a distributed database that maps IP addresses to domain names so users can find the resources by name rather than by numerical address. A DNS server keeps a list of domain names and the IP address associated with each name.

To manage DNS service, use the `serveradmin` tool.

Starting and Stopping DNS Service

To start the service:

```
$ sudo serveradmin start dns
```

To stop the service:

```
$ sudo serveradmin stop dns
```

Checking the Status of DNS Service

To see summary status of the service:

```
$ sudo serveradmin status dns
```

To see detailed status of the service:

```
$ sudo serveradmin fullstatus dns
```

Viewing DNS Service Settings

To view a setting:

```
$ sudo serveradmin settings dns:setting
```

To view a group of settings:

Enter as much of the name as you want, stopping at a colon (`:`), then enter an asterisk (`*`) as a wildcard for the remaining parts of the name. For example:

```
$ sudo serveradmin settings dns:zone:_array_id:localhost:*
```

To view all service configuration settings:

```
$ sudo serveradmin settings dns
```

Changing DNS Service Settings

To modify your server's DNS configuration, use `serveradmin`. However, you'll probably find it more straightforward to work with DNS and BIND using the standard tools and techniques described in the many books on the subject. (For an example, see *DNS and BIND* by Paul Albitz and Cricket Liu.)

DNS Service Settings

To view the settings, see "Viewing DNS Service Settings" on this page.

Available DNS serveradmin Commands

Command (`dns:command=`)	Description
`getLogPaths`	Find the location of the DNS service log. See "Viewing the DNS Service Log and Log Path" on this page.
`getStatistics`	Retrieve DNS service statistics. See "Viewing DNS Service Statistics" on this page.

Viewing the DNS Service Log and Log Path

To view the contents of the DNS service log and the log paths, use `tail` or another file listing tool.

To view the latest entries in a log:

```
$ tail log-file
```

To see where the current DNS log is located, use the `serveradmin getLogPaths` command. The default is /Library/Logs/named.log.

To display the log path:

```
$ sudo serveradmin command dns:command = getLogPaths
```

Viewing DNS Service Statistics

To view a summary of the DNS service workload, use the `serveradmin getStatistics` command.

To view statistics:

```
$ sudo serveradmin command dns:command = getStatistics
```

The computer responds with output similar to the following:

```
dns:queriesArray:_array_index:0:name = "NS_QUERIES"
dns:queriesArray:_array_index:0:value = -1
dns:queriesArray:_array_index:1:name = "A_QUERIES"
dns:queriesArray:_array_index:1:value = -1
dns:queriesArray:_array_index:2:name = "CNAME_QUERIES"
dns:queriesArray:_array_index:2:value = -1
dns:queriesArray:_array_index:3:name = "PTR_QUERIES"
dns:queriesArray:_array_index:3:value = -1
dns:queriesArray:_array_index:4:name = "MX_QUERIES"
dns:queriesArray:_array_index:4:value = -1
dns:queriesArray:_array_index:5:name = "SOA_QUERIES"
dns:queriesArray:_array_index:5:value = -1
dns:queriesArray:_array_index:6:name = "TXT_QUERIES"
dns:queriesArray:_array_index:6:value = -1
dns:nxdomain = 0
dns:nxrrset = 0
dns:reloadedTime = ""
dns:success = 0
dns:failure = 0
dns:recursion = 0
dns:startedTime = "2003-09-10 11:24:03 -0700"
dns:referral = 0
```

Configuring IP Forwarding

You can configure Mac OS X Server to provide routing services by configuring the network interfaces properly and by enabling IP forwarding. A server providing routing services requires at least two interfaces, one to connect to the internal network and one to connect to the public network. Each of these interfaces must be configured correctly to allow it to route network data.

After the interfaces are configured to allow the server to communicate on the two networks, you must enable the computer to forward traffic between the networks. IP forwarding is enabled by using the `sysctl` tool to set the `net.inet.forwarding` kernel variable to `1` as follows:

```
$ sysctl -w net.inet.forwarding=1
```

This change takes place immediately, but is not persistent if you reboot the computer. To enable IP forwarding when Mac OS X Server restarts, set the `IPFORWARDING` flag in the /etc/hostconfig file to `-YES-` to enable IP forwarding during the startup process.

Managing Firewall Service

For its Firewall service, Mac OS X Server uses the reliable open source IPFW2 software. To protect your network applications, Firewall service scans incoming IP packets and rejects or accepts them based on the set of filters you create. You can restrict access to any IP service running on the server, and you can customize filters for all incoming clients or for a range of client IP addresses.

Firewall service relies on the `ipfw` tool included with Mac OS X Server. The `ipfw` tool is a content filter that uses rules to decide which packets to allow and which to deny.

Firewall Startup

Although the firewall is treated as a service by the Server Admin application, it is not implemented by a running process like other services. It is a set of behaviors in the kernel, controlled by the `ipfw` and `sysctl` tools.

To start and stop the firewall, the Server Admin application sets a switch using the `sysctl` tool. When the computer starts, a startup item named IPFilter checks the /etc/hostconfig file for the "IPFILTER" flag. If it is set, the `sysctl` tool is used to enable the firewall:

```
$ sysctl -w net.inet.ip.fw.enable=1
```

Otherwise, it disables the firewall:

```
$ sysctl -w net.inet.ip.fw.enable=0
```

The rules loaded in the firewall remain regardless of this setting. They are ignored when the firewall is disabled.

Starting and Stopping Firewall Service

To start the service:

```
$ sudo serveradmin start ipfilter
```

To stop the service:

```
$ sudo serveradmin stop ipfilter
```

Disabling Firewall Service

To disable the service:

```
$ sudo /usr/sbin/sysctl -w net.inet.ip.fw.enable=0
```

Checking the Status of Firewall Service

To see summary status of the service:

```
$ sudo serveradmin status ipfilter
```

To see detailed status of the service, including rules:

```
$ sudo serveradmin fullstatus ipfilter
```

Viewing Firewall Service Settings

To view a setting:

```
$ sudo serveradmin settings ipfilter:setting
```

To view a group of settings:

Enter as much of the name as you want, stopping at a colon (`:`), then enter an asterisk (`*`) as a wildcard for the remaining parts of the name. For example:

```
$ sudo serveradmin settings ipfilter:ipAddressGroups:*
```

To view all service configuration settings:

```
$ sudo serveradmin settings ipfilter
```

Changing Firewall Service Settings

To change a setting:

```
$ sudo serveradmin settings ipfilter:setting = value
```

Parameter	Description
setting	An `ipfilter` service setting. See "Available Firewall Service Settings" on page 232.
value	A value for the setting.

To change several settings:

```
$ sudo serveradmin settings
ipfilter:setting = value
ipfilter:setting = value
ipfilter:setting = value
[...]
Control-D
```

Available Firewall Service Settings

To change settings for the `ipfilter` service, use the following parameters with the `serveradmin` tool.

Parameter (`ipfilter:`)	Description
`ipAddressGroupsWithRules:_array_id:<group>...`	An array of settings describing the services allowed for specific IP address groups. See "Using ipfilter Groups with the Rules Array" on page 233.
`rules:_array_id:<rule>:...`	Arrays of rule settings, one array per defined rule. See "The ipfilter Rules Array" on page 236.

Parameter (ipfilter:)	Description
logAllDenied	A parameter that specifies whether to log all denials. Default = no
ipAddressGroups:_array_id:n:address	The address of a defined IP address group, the first element of an array that defines an IP address group.
ipAddressGroups:_array_id:n:name	The name of a defined IP address group, the second element of an array that defines an IP address group.
logAllAllowed	Whether to log access allowed by rules. Default = no

Using ipfilter Groups with the Rules Array

An array of the following settings is included in the ipfilter settings for each defined IP address group.

These arrays aren't part of a standard ipfw configuration, but are created by the Server Admin application to implement the IP Address groups in the General pane of the Firewall service settings. In an actual list, <group> is replaced with an IP address group.

Parameter (ipfilter:)	Description
ipAddressGroupsWithRules:_array_id:<group>:rules	An array of rules for the group.
ipAddressGroupsWithRules:_array_id:<group>:addresses	The group's address.
ipAddressGroupsWithRules:_array_id:<group>:name	The group's name.
ipAddressGroupsWithRules:_array_id:<group>:readOnly	Whether the group is set for read-only.

Defining Firewall Rules

To set up firewall rules for your server, use serveradmin. However, a simpler method is to add your rules to a configuration file used by Firewall service.

By modifying the file, you can define your rules using standard rule syntax instead of creating a specialized array to store the rule's components.

Adding Rules by Modifying ipfw.conf

An ipfw configuration, or ruleset, is made of a list of rules numbered from 1 to 65535. The file where you can define your rules is /etc/ipfilter/ipfw.conf. Firewall service reads this file but doesn't modify it. Its contents are annotated and include commented-out rules you can use as models.

Packets are passed to `ipfw` from a number of places in the protocol stack. (Depending on the source and destination of the packet, `ipfw` can be invoked multiple times on the same packet.) The packet passed to the firewall is compared with each rule in the firewall ruleset. When a match is found, the action corresponding to the matching rule is performed.

Important: Misconfiguring the firewall can put your computer in an unusable state, possibly shutting down network services and requiring console access to regain control of it.

You can configure `ipfw` with a variety of commands. For more information, see the `ipfw` man page.

The unmodified ipfw.conf file:

```
# ipfw.conf.default - Installed by Apple, never modified by Server Admin app
#
# ipfw.conf - The servermgrd process (the back end of Server Admin app)
# creates this from ipfw.conf.default if it's absent, but does not modify
# it.
#
# Administrators can place custom ipfw rules in ipfw.conf.
#
# Whenever a change is made to the ipfw rules by the Server Admin
# application and saved:
#    1. All ipfw rules are flushed
#    2. The rules defined by the Server Admin app (stored as plists)
#         are exported to /etc/ipfilter/ipfw.conf.apple and loaded into the
#         firewall via ipfw.
#    3. The rules in /etc/ipfilter/ipfw.conf are loaded into the firewall
#         via ipfw.
# Note that the rules loaded into the firewall are not applied unless the
# firewall is enabled.
#
# The rules resulting from the Server Admin app's IPFirewall and NAT panels
# are numbered:
#    10 - from the NAT Service - this is the NAT divert rule, present only
#         when he NAT service is started via the Server Admin app.
#    1000 - from the "Advanced" panel - the modifiable rules, ordered by
#         their relative position in the drag-sortable rule list
#    12300 - from the "General" panel - "allow"" rules that punch specific
#         holes in the firewall for specific services
#    63200 - from the "Advanced" panel - the non-modifiable rules at the
#         bottom of the panel's rule list
#
# Refer to the man page for ipfw(8) for more information.
#
# The following default rules are already added by default:
#
#add 01000 allow all from any to any via lo0
```

```
#add 01010 deny all from any to 127.0.0.0/8
#add 01020 deny ip from 224.0.0.0/4 to any in
#add 01030 deny tcp from any to 224.0.0.0/4 in
#add 12300 ("allow" rules from the "General" panel)
#...
#add 65534 deny ip from any to any
```

To add an entry that denies all TCP packets from cracker.evil.org to the Telnet port of my.host.org from being forwarded by the host:

```
$ ipfw add deny tcp from cracker.evil.org to my.host.org telnet
```

To disallow any connection from the cracker.evil.org network to my.host.org:

1 Ping cracker.evil.org to determine its IP address.

```
$ ping cracker.evil.org
PING cracker.evil.org (123.45.67.10): 56 data types
64 bytes from 123.45.67.10: icmp_seq=0 ttl=52 time=24.953 ms
64 bytes from 123.45.67.10: icmp_seq=1 ttl=52 time=19.406 ms
64 bytes from 123.45.67.10: icmp_seq=2 ttl=52 time=18.871 ms
64 bytes from 123.45.67.10: icmp_seq=3 ttl=52 time=29.776 ms
64 bytes from 123.45.67.10: icmp_seq=4 ttl=52 time=26.209 ms
```

2 Deny access to a range of IP addresses associated with cracker.evil.org.

```
$ ipfw add deny ip from 123.45.67.0/24 to my.host.org
```

Adding Rules Using serveradmin

If you prefer not to work with the ipfw.conf file, you can use the `serveradmin settings` command to add firewall rules to your configuration.

Note: Be sure to include the special first setting (ending with `= create`). This is how you instruct `serveradmin` to create the necessary rule array with the specified rule number.

To add a rule:

```
$ sudo serveradmin settings
ipfilter:rules:_array_id:rule = create
ipfilter:rules:_array_id:rule:source = source
ipfilter:rules:_array_id:rule:protocol = protocol
ipfilter:rules:_array_id:rule:destination = destination
ipfilter:rules:_array_id:rule:action = action
ipfilter:rules:_array_id:rule:enableLocked = (yes|no)
ipfilter:rules:_array_id:rule:enabled = (yes|no)
ipfilter:rules:_array_id:rule:log = (yes|no)
ipfilter:rules:_array_id:rule:readOnly = (yes|no)
ipfilter:rules:_array_id:rule:source-port = port
Control-D
```

Parameter	Description
rule	A unique rule number.
Other parameters	The standard rule settings described under "The ipfilter Rules Array" on page 236.

An example of this is the following:

```
$ sudo serveradmin settings
ipfilter:rules:_array_id:1111 = create
ipfilter:rules:_array_id:1111:source = "10.10.41.60"
ipfilter:rules:_array_id:1111:protocol = "udp"
ipfilter:rules:_array_id:1111:destination = "any via en0"
ipfilter:rules:_array_id:1111:action = "allow"
ipfilter:rules:_array_id:1111:enableLocked = yes
ipfilter:rules:_array_id:1111:enabled = yes
ipfilter:rules:_array_id:1111:log = no
ipfilter:rules:_array_id:1111:readOnly = yes
ipfilter:rules:_array_id:1111:source-port = ""
Control-D
```

The ipfilter Rules Array

An array of the following settings is included in the `ipfilter` settings for each defined firewall rule. In an actual list, `<rule>` is replaced with a rule number. You can add a rule by using `serveradmin` to create an array in the firewall settings (see "Adding Rules Using serveradmin" on page 235).

Parameter (`ipfilter:`)	Description
`rules:_array_id:<rule>:source`	The source of traffic governed by the rule.
`rules:_array_id:<rule>:protocol`	The protocol for traffic governed by the rule.
`rules:_array_id:<rule>:destination`	The destination of traffic governed by the rule.
`rules:_array_id:<rule>:action`	The action to be taken.
`rules:_array_id:<rule>:enabled`	Whether the rule is enabled.
`rules:_array_id:<rule>:log`	Whether activation of the rule is logged.
`rules:_array_id:<rule>:readOnly`	Whether read-only is set.
`rules:_array_id:<rule>:source-port`	The source port of traffic governed by the rule.

Firewall serveradmin Commands

To manage Firewall service, use the following commands with the `serveradmin` tool.

Command (`ipfilter:command=`)	Description
`getLogPaths`	Find the current location of the log used by the service. Default = `/var/log/system.log`
`getStandardServices`	Retrieve a list of standard services as they appear on the General pane of the Firewall service settings in Server Admin.
`writeSettings`	Equivalent to the standard `serveradmin settings` command, but also returns a setting indicating whether the service must be restarted. See "Using the serveradmin Tool" on page 50.

Viewing the Firewall Service Log and Log Path

To view the contents of the `ipfilter` service log to view log paths, use `tail` or another file listing tool.

To view the latest entries in the log:

```
$ tail log-file
```

To see where the `ipfilter` service log is located, use the `serveradmin getLogPaths` command.

To view the log path:

```
$ sudo serveradmin command ipfilter:command = getLogPaths
```

The computer responds with output similar to the following:

```
ipfilter:systemLog = <system-log>
```

Value	Description
`<system-log>`	The location of the ipfilter service log. Default = `/var/log/ipfw.log`

Using Firewall Service to Simulate Network Activity

You can use Firewall service in Mac OS X with Dummynet, a general-purpose network load simulator. For more information about Dummynet, see ai3.asti.dost.gov.ph/sat/dummynet.html. Also see the `ipfw` man page.

Managing NAT Service

Network Address Translation (NAT) is sometimes referred to as IP masquerading. NAT is used to allow multiple computers to access the Internet with only one assigned public or external IP address. NAT allows you to create a private network that accesses the Internet through a NAT router or gateway.

The NAT router takes traffic from your private network and remembers which internal address made the request. When the NAT router receives the response to the request, it forwards it to the originating computer. Traffic that originates from the Internet does not reach computers behind the NAT router unless port forwarding is enabled.

Note: Firewall service must be configured and running NAT service. The NAT service divert rule is run through `ipfw`.

Starting and Stopping NAT Service

To start the service:

```
$ sudo serveradmin start nat
```

To stop the service:

```
$ sudo serveradmin stop nat
```

Viewing the Status of NAT Service

To see a summary status of the service:

```
$ sudo serveradmin status nat
```

To see detailed status of the service:

```
$ sudo serveradmin fullstatus nat
```

Viewing NAT Service Settings

To view a setting:

```
$ sudo serveradmin settings nat:setting
```

To view all settings:

```
$ sudo serveradmin settings nat
```

Changing NAT Service Settings

To change a setting:

```
$ sudo serveradmin settings nat:setting = value
```

Parameter	Description
setting	A NAT service setting. To see a list of available settings, enter `$ sudo serveradmin settings nat` or see "NAT Service Settings" on page 238.
value	An appropriate value for the setting.

To change several settings:

```
$ sudo serveradmin settings
nat:setting = value
nat:setting = value
nat:setting = value
[...]
Control-D
```

NAT Service Settings

To change settings for NAT service, use the following parameters with the `serveradmin` tool.

Parameter (`nat:`)	Description
`deny_incoming`	`yes` | `no` Default = `no`
`log_denied`	`yes` | `no` Default = `no`
`clamp_mss`	`yes` | `no` Default = `yes`
`reverse`	`yes` | `no` Default = `no`

Parameter (nat:)	Description
log	yes|no Default = yes
proxy_only	yes|no Default = no
dynamic	yes|no Default = yes
use_sockets	yes|no Default = yes
interface	The network port. Default = "en0"
unregistered_only	yes|no Default = no
same_ports	yes|no Default = yes

NAT serveradmin Commands

To manage NAT service, use the following commands with the serveradmin tool.

Command (nat:command=)	Description
getLogPaths	Find the location of the log used by the NAT service. See "Viewing the NAT Service Log and Log Path" on this page.
updateNATRuleInIpfw	Update the firewall rules defined in the ipfilter service to reflect changes in NAT settings.
writeSettings	Equivalent to the standard serveradmin settings command, but also returns a setting indicating whether the service must be restarted. See "Using the serveradmin Tool" on page 50.

Port Mapping

You can configure port mapping by adding a redirect_port directive to the configuration file passed to the natd process. You can accomplish this by editing the plist version of the configuration file /etc/nat/natd.plist. This file is then processed by the serveradmin tool, and is used to create the configuration file /etc/nat/natd.conf.apple, which is passed to the natd process. For details about configuring natd, see the natd man page.

Note: Don't edit the /etc/nat/natd.conf.apple file directly, because it is regenerated every time the serveradmin start nat command is executed.

To configure NAT to use the port mapping rule redirect_port tcp 1.2.3.4:80 80, add the following lines to /etc/nat/natd.plist, inside the configuration dictionary:

```
<key>redirect_port</key>
<array>
```

```
<dict>
<key>proto</key>
<string>tcp</string>
<key>targetIP</key>
<string>1.2.3.4</string>
<key>targetPortRange</key>
<string>80</string>
<key>aliasPortRange</key>
<string>80</string>
</dict>
</array>
```

Confirm those settings using the `serveradmin` tool:

```
$ sudo serveradmin settings nat
...
nat:redirect_port:_array_index:0:proto = "tcp"
nat:redirect_port:_array_index:0:targetPortRange = "80"
nat:redirect_port:_array_index:0:aliasPortRange = "80"
nat:redirect_port:_array_index:0:targetIP = "1.2.3.4"
Control-D
```

Viewing the NAT Service Log and Log Path

To view the contents of the NAT service log or to view log paths, use `tail` or another file listing tool.

To view the latest entries in the log:

```
$ tail log-file
```

To see where the NAT service log is located, use the `serveradmin getLogPaths` command.

To view the log path:

```
$ sudo serveradmin command nat:command = getLogPaths
```

The computer responds with the following output:

```
nat:natLog = <nat-log>
```

Value	Description
`<nat-log>`	The location of the NAT service log. Default = `/var/log/alias.log`

Managing VPN Service

A Virtual Private Network (VPN) is two or more computers or networks (nodes) connected by a private link of encrypted data. This link simulates a local connection, as if the remote computer were attached to the local area network (LAN).

VPNs allow users at home or away from the LAN to securely connect to it using any network connection, such as the Internet. From the user's perspective, the VPN connection appears as a dedicated private link.

Starting and Stopping VPN Service

To start the service:

```
$ sudo serveradmin start vpn
```

To stop the service:

```
$ sudo serveradmin stop vpn
```

Checking the Status of VPN Service

To see a summary status of service:

```
$ sudo serveradmin status vpn
```

To see a detailed status of service:

```
$ sudo serveradmin fullstatus vpn
```

Viewing VPN Service Settings

To view a setting:

```
$ sudo serveradmin settings vpn:setting
```

To view all settings:

```
$ sudo serveradmin settings vpn
```

Changing VPN Service Settings

To change a setting:

```
$ sudo serveradmin settings vpn:setting = value
```

Parameter	Description
setting	A VPN service setting. To see a list of available settings, enter `$ sudo serveradmin settings vpn` or see "Available VPN Service Settings" on page 242.
value	An appropriate value for the setting.

To change several settings:

```
$ sudo serveradmin settings
vpn:setting = value
vpn:setting = value
vpn:setting = value
[...]
Control-D
```

Available VPN Service Settings

To change settings for VPN service, use the following parameters with the `serveradmin` tool.

Parameter (`vpn:Servers:`)	Description
`com.<name>.ppp.l2tp:Server:VerboseLogging`	Default = `1`
`com.<name>.ppp.l2tp:Server:MaximumSessions`	Default = `128`
`com.<name>.ppp.l2tp:Server:LogFile`	Default = `"/var/log/ppp/vpnd.log"`
`com.<name>.ppp.l2tp:IPSec:IPSecSharedSecretEncryption`	Default = `"Keychain"`
`com.<name>.ppp.l2tp:IPSec:SharedSecret`	Default = `"com.apple.ppp.l2tp"`
`com.<name>.ppp.l2tp:IPSec:LocalIdentifier`	Default = `""`
`com.<name>.ppp.l2tp:IPSec:LocalCertificate`	Default = `""`
`com.<name>.ppp.l2tp:IPSec:AuthenticationMethod`	Default = `"SharedSecret"`
`com.<name>.ppp.l2tp:IPSec:IdentifierVerification`	Default = `"None"`
`com.<name>.ppp.l2tp:IPSec:RemoteIdentifier`	Default = `""`
`com.<name>.ppp.l2tp:L2TP:Transport`	Default = `"IPSec"`
`com.<name>.ppp.l2tp:IPv4:DestAddressRanges`	Default = `_empty_array`
`com.<name>.ppp.l2tp:IPv4:OfferedRouteMasks`	Default = `_empty_array`
`com.<name>.ppp.l2tp:IPv4:OfferedRouteAddresses`	Default = `_empty_array`
`com.<name>.ppp.l2tp:IPv4:OfferedRouteTypes`	Default = `_empty_array`
`com.<name>.ppp.l2tp:IPv4:ConfigMethod`	Default = `"Manual"`
`com.<name>.ppp.l2tp:DNS:OfferedSearchDomains`	Default = `_empty_array`
`com.<name>.ppp.l2tp:DNS:OfferedServerAddresses`	Default = `_empty_array`
`com.<name>.ppp.l2tp:Interface:SubType`	Default = `"L2TP"`

Parameter (vpn:Servers:)	Description
com.<name>.ppp.l2tp: Interface:Type	Default = "PPP"
com.<name>.ppp.l2tp: PPP:LCPEchoFailure	Default = 5
com.<name>.ppp.l2tp: PPP:ACSPEnabled	Default = 1
com.<name>.ppp.l2tp: PPP:VerboseLogging	Default = 1
com.<name>.ppp.l2tp: PPP:AuthenticatorACLPlugins	Default = DSACL
com.<name>.ppp.l2tp: PPP:AuthenticatorEAPPlugins	Default = EAP-KRB
com.<name>.ppp.l2tp: PPP:AuthenticatorPlugins: _array_index:*n*	Default = "DSAuth"
com.<name>.ppp.l2tp: PPP:LCPEchoInterval	Default = 60
com.<name>.ppp.l2tp: PPP:LCPEchoEnabled	Default = 1
com.<name>.ppp.l2tp: PPP:IPCPCompressionVJ	Default = 0
com.<name>.ppp.l2tp: PPP:AuthenticatorProtocol: _array_index:*n*	Default = "MSCHAP2"
com.<name>.ppp.l2tp: PPP:LogFile	Default = "/var/log/ppp/vpnd.log"
com.<name>.ppp.pptp: Server:VerboseLogging	Default = 1
com.<name>.ppp.pptp: Server:MaximumSessions	Default = 128
com.<name>.ppp.pptp: Server:LogFile	Default = "/var/log/ppp/vpnd.log"
com.<name>.ppp.pptp: IPv4:DestAddressRanges	Default = _empty_array
com.<name>.ppp.pptp: IPv4:OfferedRouteMasks	Default = _empty_array
com.<name>.ppp.pptp: IPv4:OfferedRouteAddresses	Default = _empty_array
com.<name>.ppp.pptp: IPv4:OfferedRouteTypes	Default = _empty_array
com.<name>.ppp.pptp: IPv4:ConfigMethod	Default = "Manual"

Parameter (vpn:Servers:)	Description
com.<name>.ppp.pptp: DNS:OfferedSearchDomains	Default = _empty_array
com.<name>.ppp.pptp: DNS:OfferedServerAddresses	Default = _empty_array
com.<name>.ppp.pptp: Interface:SubType	Default = "PPTP"
com.<name>.ppp.pptp: Interface:Type	Default = "PPP"
com.<name>.ppp.pptp: PPP:CCPProtocols:_array_index:*n*	Default = "MPPE"
com.<name>.ppp.pptp: PPP:LCPEchoFailure	Default = 5
com.<name>.ppp.pptp: PPP:MPPEKeySize128	Default = 1
com.<name>.ppp.pptp: PPP:ACSPEnabled	Default = 1
com.<name>.ppp.pptp: PPP:AuthenticatorACLPlugins	Default = DSACL
com.<name>.ppp.pptp: PPP:AuthenticatorEAPPlugins	Default = EAP-RSA
com.<name>.ppp.pptp: PPP:VerboseLogging	Default = 1
com.<name>.ppp.pptp: PPP:AuthenticatorPlugins: _array_index:*n*	Default = "DSAuth"
com.<name>.ppp.pptp: PPP:MPPEKeySize40	Default = 0
com.<name>.ppp.pptp: PPP:LCPEchoInterval	Default = 60
com.<name>.ppp.pptp: PPP:LCPEchoEnabled	Default = 1
com.<name>.ppp.pptp: PPP:CCPEnabled	Default = 1
com.<name>.ppp.pptp: PPP:IPCPCompressionVJ	Default = 0
com.<name>.ppp.pptp: PPP:AuthenticatorProtocol: _array_index:*n*	Default = "MSCHAP2"
com.<name>.ppp.pptp: PPP:LogFile	Default = "/var/log/ppp/vpnd.log"

Available VPN serveradmin Commands

To manage VPN service, use the following commands with the `serveradmin` tool.

Command (`vpn:command=`)	Description
`getLogPaths`	Find the location of the VPN service log. See "Viewing the VPN Service Log and Log Path" on this page.
`writeSettings`	Equivalent to the standard `serveradmin` settings command but also returns a setting indicating whether the service must be restarted. See "Using the serveradmin Tool" on page 50.

Viewing the VPN Service Log and Log Path

To view the contents of the VPN service log or to view log paths, use `tail` or another file listing tool.

To view the latest entries in the log:

```
$ tail log-file
```

To see where the current VPN service log is located, use the `serveradmin getLogPaths` command.

To view the log path:

```
$ sudo serveradmin command vpn:command = getLogPaths
```

The computer responds with the following output:

```
vpn:vpnLog = <vpn-log>
```

Value	Description
`<vpn-log>`	The location of the VPN service log. Default = `/var/log/vpnd.log`

Site-to-Site VPN

Site-to-site VPN is implemented by the daemon `vpnd`, which is a wrapper around the `racoon` daemon and the `setkey` tool. The `racoon` daemon negotiates and configures a set of parameters of IPsec. `setkey` manipulates Security Association Database (SAD) entries as well as Security Policy Database (SPD) entries in the kernel.

For more information, see the `racoon` and `setkey` man pages. `racoon` also has a webpage at www.kames.com/racoon. You might also find the `ipsec` man page helpful.

Apple provides an interactive `s2svpnadmin` tool, in /usr/sbin/, that enables you to configure and set up site-to-site VPN. The `s2svpnadmin` tool accesses configuration information for the Client Server VPN application in Server Admin.

The `s2svpnadmin` tool does not start the VPN service. You must start the VPN service separately from Server Admin.

The `s2svpnadmin` tool can:

- List configured site-to-site VPN servers
- Display their configuration details
- Add a configuration
- Delete a configuration

You can use this tool to configure a local VPN server, not a remote one. To set up a site-to-site server, you must configure the two VPN gateway servers at the two sites independently.

You must run `s2svpnadmin` with root privileges.

Configuring Site-to-Site VPN

To configure a site-to-site VPN, run `s2svpnadmin` with root privileges, choose the "Configure a new site-to-site server" option, and provide the following information:

- A configuration name used to identify the server. Do not include spaces in it.
- The external gateway address of the local site.
- The external gateway address of the remote site.
- The form of IPSec security to use (certificate or shared-secret). Before choosing certificate-based authentication, be sure that at least one certificate is installed on the server.

  `s2svpnadmin` displays a list of installed certificates and prompts the user to choose one.

  Certificates can be created, self-signed, and installed using Server Admin. To use a shared secret, be sure the same shared secret is configured on the VPN server at the other site.
- Policies consisting of local and remote subnet addresses. A policy includes a local network and a remote network. A network is specified by a network address and the number of prefix bits that must be masked in an IPv4 address to determine the network address it corresponds to. Be sure that a compatible policy is configured on both VPN servers.

If you make an invalid entry, `s2svpnadmin` forces you to start over again.

Note: `s2svpnadmin` prompts if the server must be enabled. By default, it is enabled. `s2svpnadmin` does not support editing a configuration, so if the server is not enabled, you must delete the configuration and then recreate it and enable it later. Alternatively, you can edit the configuration file. The configuration file is a plist file in /Library/Preferences/SystemConfiguration/com.apple.RemoteAccessServers.plist.

Adding a VPN Keyagent User

To enable PPTP in your VPN server, add a keyagent user in the LDAP folder that hosts your users. If you have more than one folder with VPN users, add a keyagent in each folder.

Use the `vpnaddkeyagentuser` tool to add the required VPN PPTP keyagent user to a folder. The tool prompts you for the administrator user name and password of the folder. It then sets up the keyagent user. This step is necessary to proceed with the configuration of the VPN PPTP server.

Note: You must run the `vpnaddkeyagentuser` command on the computer running the VPN service.

To add the keyagent user to the OpenLDAP master on your local computer:

```
$ sudo vpnaddkeyagentuser /LDAPv3/127.0.0.1
```

If your OpenLDAP master is not running on the local computer, replace `127.0.0.1` with the IP address of the OpenLDAP master.

You must run `vpnaddkeyagentuser` with root privileges. If no argument is specified, the keyagent user is added to the local directory domain.

Setting Up IP Failover

IP failover allows a secondary server to acquire the IP address of a primary server if the primary server ceases to function. After the primary server returns to normal operation, the secondary server relinquishes the IP address. This allows your website to remain available on the network even if the primary server temporarily goes offline.

IP failover isn't a complete solution; it is one tool you can use to increase your server's availability to your clients.

Note: IP failover only allows a secondary server to acquire a primary server's IP address. You need additional software tools, such as `rsync`, to provide capabilities such as mirroring the primary server's data on the secondary server. For more information, see the `rsync` man page.

IP Failover Prerequisites

To use IP failover, set up the following hardware and software.

Hardware Requirements

IP failover requires the following hardware setup:

- Primary server
- Secondary server
- Public network (the servers must be on same subnet)
- Private network between servers (requires an additional network interface card)

Note: Because IP failover uses broadcast messages, both servers must have IP addresses on the same subnet of the public network. Both servers must also have IP addresses on the same subnet of the private network.

Software Requirements

IP failover requires the following software setup:

- Unique IP addresses for each network interface (public and private)
- Software to mirror primary server data to the secondary server
- Scripts to control failover behavior on the secondary server

IP Failover Operation

When IP failover is active, the primary server periodically broadcasts a brief message confirming normal operation on the public and private networks. This message is monitored by the secondary server.

- If the broadcast is interrupted on both public and private networks, the secondary server initiates the failover process.
- If status messages are interrupted on only one network, the secondary server sends a mail notification of a network anomaly, but doesn't acquire the primary server's IP address.

Mail notification is sent when the secondary server detects a failover condition or a network anomaly and when the IP address is relinquished back to the primary server.

Enabling IP Failover

You enable IP failover by adding command lines to the file /etc/hostconfig on the primary and the secondary server. Enter these lines exactly as shown, with regard to spaces and punctuation marks.

To enable IP failover:

1 On the primary server, add the following line to /etc/hostconfig:

```
FAILOVER_BCAST_IPS="10.0.0.255 100.0.255.255"
```

Substitute the broadcast addresses used on your server for the public and private networks. This instructs the server to send broadcast messages over relevant network interfaces, indicating that the server at those IP addresses is functioning.

2 Restart the primary server so your changes can take effect.

3 Disconnect the primary server from the public and private networks.

4 On the secondary server, add the following lines to /etc/hostconfig:

```
FAILOVER_PEER_IP="10.0.0.1"
FAILOVER_PEER_IP_PAIRS="en0:100.0.0.10"
FAILOVER_EMAIL_RECIPIENT="admin@example.com"
```

In the first line, substitute the IP address of the primary server on the private network.

In the second line, enter the local network interface that should adopt the primary server's public IP address, then a colon, and then the primary server's public IP address.

In the third line, enter the mail address for notification messages regarding the primary server status. If this line is omitted, mail notifications are sent to the root account on the local computer.

5 Restart the secondary server so your changes can take effect and allow the secondary server to acquire the primary's public IP address.

Important: Before you enable IP failover, verify on both servers that the port used for the public network is at the top of the Network Port Configurations list in the Network pane of System Preferences. Also verify that the port used for the private network contains no DNS configuration information.

6 Reconnect the primary server to the private network, wait 15 seconds, and then reconnect the primary server to the public network.

7 Verify that the secondary server relinquishes the primary server's public IP address.

Configuring IP Failover

You configure failover behavior using scripts. The scripts must be executable (for example, shell scripts, Perl, compiled C code, or executable AppleScripts). You place these scripts in /Library/IPFailover/<IP_address> on the secondary server.

You must create a folder named with the public IP address of the primary server to contain the failover scripts for that server (for example, /Library/IPFailover/100.0.0.10).

Notification Only

You can use a script named `Test` located in the failover scripts folder to control whether, in the event of a failover condition, the secondary server acquires the primary server's IP address, or only sends a mail notification.

If no script exists, or if the script returns a zero result, the secondary server acquires the primary's IP address.

If the script returns a nonzero result, the secondary server skips IP address acquisition and only sends a mail notification of the failover condition.

You run the `Test` script to determine whether the IP address should be acquired and to determine if the IP address should be relinquished when the primary server returns to service.

A simple way to set up this notification-only mode is to copy the script at /usr/bin/false to the folder named with your primary server IP address, and then change the name of the script to `Test`. This script always returns a nonzero result.

Using the `Test` script, you can configure the primary server to monitor the secondary server and send mail notification if the secondary server becomes unavailable.

Pre- and Post- Scripts

You can configure the failover process with scripts that can run before acquiring the primary IP address (before acquisition), after acquiring the IP address (post acquisition), before relinquishing the primary IP address (before relinquish), and after relinquishing the IP address back to the primary server (after relinquish).

These scripts reside in the /Library/IPFailover/<IP_address> folder on the secondary server. The scripts use the following prefixes:

- `PreAcq`—Run before acquiring the IP address from the primary server.
- `PostAcq`—Run after acquiring the IP address from the primary server.
- `PreRel`—Run before relinquishing the IP address back to the primary server.
- `PostRel`—Run after relinquishing the IP address back to the primary server.

Important: Before you activate IP failover on the secondary server, be sure the primary server is up and functioning normally. If the primary server isn't sending broadcast messages, the secondary server initiates the failover process and acquires the primary's public IP address.

You might have more than one script at each stage. The scripts in each prefix group are run in the order in which their file names appear in a folder listing using the `ls` tool.

For example, your secondary server might perform other services on the network, such as running a statistical analysis application and distributed image processing software. A preacquisition script quits the running applications to free the CPU for the web server. A postacquisition script starts the web server. After the primary server is up and running again, a prerelinquish script quits the web server, and a postrelinquish script starts the image processing and statistical analysis applications.

The sequence of scripted events might look like this:

```
<Failover condition detected>
Test (if present)
PreAcq10.StopDIP
PreAcq20.StopSA
PreAcq30.CleanupTmp
<Acquire IP address>
```

```
PostAcq10.StartTimer
PostAcq20.StartApache
<Primary server returns to service>
PreRel10.StopApache
PreRel20.StopTimer
<Relinquish IP address>
PostRel10.StartSA
PostRel20.StartDIP
PostRel30.MailTimerResultsToAdmin
```

Enabling PPP Dial-In

To set up Point-to-Point Protocol (PPP) dial-in service, use the `pppd` daemon. For more information, see the `pppd` man page.

The "Examples" section of the man page shows an example of setting up dial-in service.

Restoring the Default Configuration for Server Services

When you use applications such as Server Admin to configure a Mac OS X Server service, your settings are stored in places such as a configuration file (.conf), a preference list (.plist), an XML file, or the local directory database.

In some cases, you might want to reset a service to its default settings, which you can do by renaming or deleting a service's configuration file. Mac OS X Server then creates a default copy of the file.

To restore NAT service to its default:
Rename or delete the natd.plist file in the /etc/nat/ folder.

To restore Firewall service to its default:
Rename or delete the ip_address_groups.plist, standard_services.plist, and ipfw.conf files in the /etc/ipfilter/ folder.

To restore DHCP service to its default:

1. Remove the subnet configuration from the /config/dhcp/ folder in the local directory database by using the `dscl` tool:

   ```
   $ sudo dscl . -delete /config/dhcp
   ```

2. Remove the static Ethernet / IP Address static maps from the /machines/ folder in the local directory database.

   The easiest way to do this is to delete the folder:

   ```
   $ sudo dscl . -delete /machines
   ```

3 Recreate the two default records:

```
$ sudo dscl . -create /machines/localhost
$ sudo dscl . -append /machines/localhost ip_address 127.0.0.1
$ sudo dscl . -append /machines/localhost serves ./local
$ sudo dscl . -create /machines/broadcasthost
$ sudo dscl . -append /machines/broadcasthost ip_address 255.255.255.255
$ sudo dscl . -append /machines/broadcasthost serves ../network
```

To restore QTSS Publisher service to its default:
Rename or delete these files:

- /Library/Application Support/Apple/QTSS Publisher/Links.plist
- /Library/Application Support/Apple/QTSS Publisher/Poster Images.plist
- /Library/Caches/com.apple.qtsspublisher.plist

The libraries and templates are in the/Library/Application Support/Apple/QTSS Publisher/* folder. The content varies, based on what's been uploaded:

To restore QTSS service to its default:
Rename or delete these files:

- /Library/QuickTimeStreaming/Config/streamingserver.xml
- /Library/QuickTimeStreaming/Config/relayconfig.xml

You can also rename or delete the qtusers and qtgroups files, which should then be recreated using `qtpasswd`.

- /Library/QuickTimeStreaming/Config/qtusers
- /Library/QuickTimeStreaming/Config/qtgroups

To restore DNS service to its default:

1 From the /etc/named.conf/var/named/* folder, remove the files for each forward zone, named similar to my.domain.com.zone.

2 From the /etc/named.conf/var/named/* folder, remove the separate files for each reverse zone, named similar to db.10.1.0.

3 Do not remove the localhost.zone, named.ca, or named.local files.

To restore VPN service to its default:
Rename the com.apple.RemoteAccessServers.plist file in the /Library/Preferences/SystemConfiguration/ folder.

To restore SERVERMGR_MAIL service to its default:
Rename these files:

- /etc/MailServicesOther.plist
- /var/mailman/data/listinfo.plist

Configuring and Managing Open Directory

15

Use this chapter to learn the commands to configure and manage the Open Directory service.

This chapter discusses the tools and commands used when working with Open Directory.

Open Directory is the standards-based directory and network authentication services architecture used by Mac OS X and Mac OS X Server. In Mac OS X Server, Open Directory relies on open source technologies such as OpenLDAP and Kerberos to provide directory and authentication services, but Open Directory does much more.

Open Directory supports conventional authentication methods in addition to Kerberos. Open Directory also integrates with other directory services including Microsoft Active Directory, Novell eDirectory, and other standards-based LDAP directory services.

For more information, see *Open Directory Administration*.

Understanding Open Directory

To provide access to directory service data, Mac OS X Server relies on Lightweight Directory Access Protocol (LDAP). LDAP is provided on Mac OS X Server by OpenLDAP, a best-of-breed open source LDAP service.

Apple has made very few changes to the stock distribution of OpenLDAP. For most functions, you should be able to treat LDAP on Mac OS X Server as a standard OpenLDAP distribution.

In addition to Open Directory, a number of third-party directory services use LDAP for identification. This allows Mac OS X to interoperate easily with these systems.

This chapter includes descriptions of tools for working with LDAP and the Open Directory Password Server.

Using General Directory Tools

This section describes how to test Open Directory configurations, modify Open Directory directory domains, and test Open Directory plug-ins.

Testing Your Open Directory Configuration

To test your directory services configuration, use the `dscl` tool. For more information, see the `dscl` man page.

Modifying a Directory Domain

To create, modify, or delete directory information in a directory domain, use the `dscl` tool.

Testing Open Directory Plug-ins

To check the performance of protocol-specific plug-ins used by Open Directory, use the `dsperfmonitor` tool. It can list the API calls being made to plug-ins, how long the plug-ins take to reply, and recent API call errors. For more information, see the `dsperfmonitor` man page.

Directory services API support is provided by the `DirectoryService` daemon. For more information, see the `DirectoryService` man page.

For information about the data types used by directory services, see the `DirectoryServiceAttributes` man page.

Finally, for information about the internals of Open Directory and its plug-ins, including source code you can examine or adopt, click the Open Directory link at www.apple.com/darwin.

Changing Open Directory Service Settings

To change settings for the Open Directory service, use the following parameters with the `serveradmin` tool. Be sure to add `dirserv:` to the beginning of any parameter you use.

To see the role the server is playing in the directory hierarchy:

```
$ sudo serveradmin settings dirserv:<parameter>
```

Parameter	Description
replicationUnits	Default = `"days"`
replicaLastUpdate	Default = `""`
LDAPSettings:LDAPDataBasePath	Default = `""`
replicationPeriod	Default = `4`
LDAPSettings:LDAPSearchBase	Default = `""`

Parameter	Description
passwordOptionsString	Default = `"usingHistory=0 usingExpirationDate=0 usingHardExpirationDate=0 requiresAlpha=0 requiresNumeric=0 expirationDateGMT=12/31/69 hardExpireDateGMT=12/31/69 maxMinutesUntilChangePassword=0 maxMinutesUntilDisabled=0 maxMinutesOfNonUse=0 maxFailedLoginAttempts=0 minChars=0 maxChars=0 passwordCannotBeName=0"`
LDAPSettings:LDAPSSLCertificatePath	Default = `""`
masterServer	Default = `""`
LDAPServerType	Default = `"standalone"`
replicationWhen	Default = `"periodic"`
LDAPSettings:useSSL	Default = `"YES"`
LDAPDefaultPrefix	Default = `"dc=<domain>,dc=com"`
LDAPSettings:LDAPTimeoutUnits	Default = `"minutes"`
LDAPSettings:LDAPServerBackend	Default = `"BerkeleyDB"`

Managing OpenLDAP

To provide directory services for mixed-platform environments, Open Directory uses OpenLDAP, the open source implementation of LDAP. A common language for directory access lets you consolidate information from different platforms and define a single name space for network resources.

Whether you have Mac, Windows, or Linux computers on your network, you can set up and manage a single directory, eliminating the need to maintain a separate directory or separate user records for each platform.

Configuring LDAP

The OpenLDAP server daemon is `slapd`, in /usr/libexec/. `slapd` is launched by the LDAP startup item. The primary configuration files for OpenLDAP are in /etc/openldap/. There you will find the slapd.conf file, which contains basic configuration information.

Most configuration for Open Directory is stored in the slapd_macosxserver.conf file. An include statement in the slapd.conf file includes slapd_macosxserver.conf.

Although the directives in these files can be modified using the administration applications, avoid modifying these directives. Instead, use your own configuration file by adding an include directive for it in the slapd.conf file.

The slapd_macosxserver.conf file contains an entry for the root user of the LDAP database, the directive `rootdn`. This root user is a user who has control over all data inside the LDAP database. Access controls do not apply to the root user.

An example value for `rootdn` is `uid=root,cn=users,dc=example,dc=com`.

An administrator can edit the /etc/openldap/slapd_macosxserver.conf file to add a password hash, or plain-text password, to the file, at which point that administrator user could administer the LDAP database. This is especially useful when your LDAP database is damaged or the passwords are lost or forgotten.

Configuring slapd and slurpd Daemons

To configure the `slapd` and `slurpd` LDAP daemons and related search policies, use the `slapconfig` tool. For more information, see the `slapconfig` man page.

Standard Distribution Tools

Two types of tools come with OpenLDAP:

- Tools that operate directly on the LDAP databases—These tools begin with `slap`.
- Tools that go through the LDAP protocol—These tools begin with `ldap`.

You must run the `slap` tools on the computer hosting the LDAP database. When using the `slap` tools, shut down the LDAP service. If you don't, your database can get out of sync.

These tools are included in the standard OpenLDAP distribution.

Tool	Used to
`/usr/bin/ldapadd`	Add entries to the LDAP directory.
`/usr/bin/ldapcompare`	Compare a directory entry's actual attributes with known attributes.
`/usr/bin/ldapdelete`	Delete entries from the LDAP directory.
`/usr/bin/ldapmodify`	Change an entry's attributes.
`/usr/bin/ldapmodrdn`	Change an entry's relative distinguished name (RDN).
`/usr/bin/ldappasswd`	Set the password for an LDAP user. Apple recommends using `passwd` instead of `ldappasswd`. For more information, see the `passwd` man page.
`/usr/bin/ldapsearch`	Search the LDAP directory. See the usage note under "Searching the LDAP Server" on page 257.
`/usr/bin/ldapwhoami`	Obtain the primary authorization identity associated with a user.
`/usr/sbin/slapadd`	Add entries to the LDAP directory.
`/usr/sbin/slapcat`	Export LDAP Directory Interchange Format files.
`/usr/sbin/slapindex`	Regenerate directory indexes.
`/usr/sbin/slappasswd`	Generate user password hashes.

Idle Rebinding Options

The following LDAPv3 plug-in parameters are documented in *Open Directory Administration*. The parameters are used in the file /Library/Preferences/DirectoryService/DSLDAPv3PlugInConfig.plist.

Delay Rebind

This parameter specifies how long the LDAP plug-in waits before attempting to reconnect to a server that fails to respond. You can increase this value to prevent continuous reconnection attempts.

```
<key>Delay Rebind Try in seconds<\key>
<integer>n<\integer>
```

You can find this parameter in the DSLDAPv3PlugInConfig.plist file near `<key>OpenClose Timeout in seconds<\key>`. If not, add it there.

Idle Timeout

This parameter specifies how long the LDAP plug-in sits idle before disconnecting from the server. You can adjust this value to reduce overloading the server's connections from remote clients.

```
<key>Idle Timeout in minutes<\key>
<integer>n<\integer>
```

If this parameter doesn't exist in the DSLDAPv3PlugInConfig.plist file, add it near `<key>OpenClose Timeout in seconds<\key>`.

Searching the LDAP Server

The `ldapsearch` tool connects to an LDAP server, authenticates, finds entries, and returns attributes of the entries found.

To query the LDAP server for a user's information:
Enter the following command, replacing the example search base (cn=users, dc=example, dc=com) with an actual search base:

```
$ ldapsearch -H ldap://127.0.0.1 -b cn=users,dc=example,dc=com
```

By default, `ldapsearch` tries to connect to the LDAP server using the Simple Authentication and Security Layer (SASL) method. If the server doesn't support this method, you see this error message:

```
ldap_sasl_interactive_bind_s: No such attribute (16)
```

To avoid this error, include the `-x` option when you enter the command. For example:

```
$ ldapsearch -h 192.168.100.1 -b "dc=example,dc=com" -x
```

The `-x` option forces `ldapsearch` to use simple authentication instead of SASL. The `-x` option also works on other LDAP tools.

`ldapsearch` can also be used for debugging issues with LDAP, independent of the directory services LDAPv3 plug-in.

For example, you can read the root directory server entry (DSE) like this (`-LLL` omits some output, `-x` means no SASL, `-h` specifies the hostname, `-b` specifies the search base and `-s` specifies the type of search):

```
$ ldapsearch -LLL -x -h ldap.psu.edu -b "" -s base
dn:
namingcontexts: CN=SCHEMA
namingcontexts: CN=LOCALHOST
namingcontexts: CN=PWDPOLICY
namingcontexts: CN=IBMPOLICIES
namingcontexts: DC=PSU,DC=EDU
subschemasubentry: cn=schema
supportedextension: 1.3.18.0.2.12.1
supportedextension: 1.3.18.0.2.12.3
supportedextension: 1.3.18.0.2.12.5
supportedextension: 1.3.18.0.2.12.6
supportedextension: 1.3.18.0.2.12.15
supportedextension: 1.3.18.0.2.12.16
supportedextension: 1.3.18.0.2.12.17
supportedextension: 1.3.18.0.2.12.19
supportedextension: 1.3.18.0.2.12.44
supportedextension: 1.3.18.0.2.12.24
supportedextension: 1.3.18.0.2.12.22
supportedextension: 1.3.18.0.2.12.20
supportedextension: 1.3.18.0.2.12.28
supportedextension: 1.3.18.0.2.12.30
supportedextension: 1.3.18.0.2.12.26
supportedextension: 1.3.6.1.4.1.1466.20037
supportedextension: 1.3.18.0.2.12.35
supportedextension: 1.3.18.0.2.12.40
supportedextension: 1.3.18.0.2.12.46
supportedextension: 1.3.18.0.2.12.37
supportedcontrol: 2.16.840.1.113730.3.4.2
supportedcontrol: 1.3.18.0.2.10.5
supportedcontrol: 1.2.840.113556.1.4.473
supportedcontrol: 1.2.840.113556.1.4.319
supportedcontrol: 1.3.6.1.4.1.42.2.27.8.5.1
supportedcontrol: 1.2.840.113556.1.4.805
supportedcontrol: 2.16.840.1.113730.3.4.18
supportedcontrol: 1.3.18.0.2.10.15
supportedcontrol: 1.3.18.0.2.10.18
security: none
port: 389
supportedsaslmechanisms: CRAM-MD5
supportedsaslmechanisms: DIGEST-MD5
supportedldapversion: 2
supportedldapversion: 3
ibmdirectoryversion: 5.2
```

```
ibm-ldapservicename: tr17n01.aset.psu.edu
ibm-serverId: 0f876740-64d2-102b-8f0b-8ab9d7eaa702
ibm-supportedacimechanisms: 1.3.18.0.2.26.3
ibm-supportedacimechanisms: 1.3.18.0.2.26.4
ibm-supportedacimechanisms: 1.3.18.0.2.26.2
vendorname: International Business Machines (IBM)
vendorversion: 5.2
ibm-sslciphers: N/A
ibm-slapdisconfigurationmode: FALSE
ibm-slapdSizeLimit: 200
ibm-slapdTimeLimit: 900
ibm-slapdDerefAliases: always
ibm-supportedAuditVersion: 2
ibm-sasldigestrealmname: tr17n01.aset.psu.edu
```

If the server is an OpenLDAP server, specify `+` for operational attributes or specify the attributes of interest:

```
$ ldapsearch -LLL -x -h xtra.apple.com -b "" -s base +
dn:
structuralObjectClass: OpenLDAProotDSE
namingContexts: dc=apple,dc=com
supportedControl: 2.16.840.1.113730.3.4.18
supportedControl: 2.16.840.1.113730.3.4.2
supportedControl: 1.3.6.1.4.1.4203.1.10.1
supportedControl: 1.2.840.113556.1.4.1413
supportedControl: 1.2.840.113556.1.4.1339
supportedControl: 1.2.840.113556.1.4.319
supportedControl: 1.2.826.0.1.334810.2.3
supportedExtension: 1.3.6.1.4.1.1466.20037
supportedExtension: 1.3.6.1.4.1.4203.1.11.1
supportedExtension: 1.3.6.1.4.1.4203.1.11.3
supportedFeatures: 1.3.6.1.4.1.4203.1.5.1
supportedFeatures: 1.3.6.1.4.1.4203.1.5.2
supportedFeatures: 1.3.6.1.4.1.4203.1.5.3
supportedFeatures: 1.3.6.1.4.1.4203.1.5.4
supportedFeatures: 1.3.6.1.4.1.4203.1.5.5
supportedLDAPVersion: 3
supportedSASLMechanisms: CRAM-MD5
supportedSASLMechanisms: GSSAPI
subschemaSubentry: cn=Subschema
```

Usually the `namingContexts` value is the first thing you want to determine:

```
$ ldapsearch -LLL -x -h xtra.apple.com -b "" -s base namingContexts
dn:
namingContexts: dc=apple,dc=com
```

After you determine the value, search for a record with a command, like this:

```
$ ldapsearch -LLL -x -h xtra.apple.com -b "dc=apple,dc=com"
uid=ajohnson uid cn
dn: uid=ajohnson,cn=users,dc=apple,dc=com
```

```
uid: ajohnson
cn: Anne Johnson
```

Using LDIF Files

Lightweight Directory Interchange Format (LDIF) is a file format used to represent LDAP entries in text form. LDAP tools such as `ldappadd`, `ldapmodify`, and `ldapsearch` read and write LDIF files.

Here is an example of an LDIF file containing three entries. Multiple entries in an LDIF file are separated by blank lines.

```
dn: cn=Mei Chen,dc=example,dc=com
cn: Mei Chen
cn: M Chen
objectclass: person
description:< file:///tmp/babs
sn: Chen

dn: cn=Anne Johnson,dc=example,dc=com
cn: Anne Johnsone
cn: A Johnson
objectclass: person
sn: Johnson

dn: cn=Tom Clark,dc=example,dc=com
cn: Tom Clark
cn: T Clark
objectclass: person
sn: Clark
```

WARNING: LDAP tools can modify or add entries to the LDAP directory. Changing raw data in a directory can have unexpected and undesirable consequences. You could inadvertently incapacitate users or computers, or you could unintentionally authorize users to access more resources.

To load an LDIF file into the LDAP directory:

Replace the *`appleserver.example.com`* with the location of the LDAP directory and *`myusers.ldif`* with the name of your LDIF file:

```
$ ldapadd -H ldap://appleserver.example.com -f myusers.ldif
```

Additional Information About LDAP

The LDAP server in Mac OS X Server is based on OpenLDAP. Additional information about OpenLDAP, including an administrator's guide, is available at www.openldap.org.

WARNING: Apple doesn't support the OpenLDAP administrator's guide, so carefully test procedures documented in it before using them on an Open Directory server that's in service.

Managing Open Directory Passwords

When a user's account has a password type of Open Directory, the user can be authenticated by Kerberos or the Open Directory Password Server.

Kerberos is a network authentication system that uses credentials issued by a trusted server.

The Open Directory Password Server supports traditional password authentication methods that some network services or users' client applications require.

Services can be configured to not allow Kerberos. In that case they use Password Server for user accounts with Open Directory passwords.

Neither Kerberos nor the Open Directory Password Server stores the password in the user's account. Both Kerberos and the Open Directory Password Server store passwords in secure databases apart from the directory domain and they never allow passwords to be read. Passwords can only be set and verified.

Open Directory Password Server

Password Server uses standard Simple Authentication and Security Layer (SASL) technology to negotiate an authentication method between a client and a service.

Password Server supports multiple authentication methods, including APOP, CRAM-MD5, DHX, Digest-MD5, MS-CHAPv2, NTLMv1 and NTLMv2, LAN Manager, and WebDAV-Digest.

Open Directory also provides authentication services using shadow passwords, which support the same authentication methods as Password Server.

To back up and restore the Password Server and Kerberos databases, use the `slapconfig` tool with the `-backupdb` and `-restoredb options`, respectively. You can also use this tool with the `-mergedb` option to merge a backup archive into an existing directory system. For more information, see the `slapconfig` man page.

To create or modify the password database used by Password Server, use the `mkpassdb` tool. For more information, see the `mkpassdb` man page.

Viewing or Changing Password Policies

To view or change the authentication policies used by Password Server, use the `pwpolicy` tool. For more information, see the `pwpolicy` man page.

Kerberos and Apple Single Sign-On

A robust authentication server that uses MIT's Kerberos Key Distribution Center (KDC) is built into Open Directory—providing strong authentication with support for secure single sign-on. That means users authenticate once, with a single user name and password pair, to access a broad range of Kerberized network services.

The following tools are available for setting up your Kerberos and Apple single sign-on environment. For more information about a tool, see the related man page.

Tool (in `usr/sbin/`)	Description
`kdcsetup`	Creates necessary setup files and adds `krb5kdc` and `kadmind` servers for the Apple Open Directory KDC.
`sso_util`	Sets up, interrogates, and tears down the Kerberos configuration in the Apple single sign-on environment.
`kerberosautoconfig`	Creates the `edu.mit.Kerberos` file based on the Open Directory `KerberosClient` record.

Backing Up the Kerberos Database

`kdb5_util` is a tool for maintaining the Kerberos database. The `kdb5_util` tool is useful for dumping the principal database to text to get a reliable backup.

The data is extremely sensitive. By definition, creating a copy of it decreases your overall security. These backups should be subject to the same security precautions as other KDC files.

Note: Do not back up the KDC while the `krb5kdc` process is running.

To dump the KDC's database:

```
$ sudo kdb5_util dump > /path/to/secure/backup
```

Replace *`/path/to/secure/backup`* with the path to the location you are backing up the database to.

To load KDC data from a dumped file:

```
$ sudo kdb5_util load /path/to/secure/backup
```

Replace *`/path/to/secure/backup`* with the path to the location of your backup database.

You can also use `kdb5_util` to create and delete Kerberos databases and to manage the location of the stash file used to encrypt the database.

Principal Management

Mac OS X Server uses MIT's Kerberos administration architecture for principal management. The Kerberos `kadmind` administration daemon is responsible for making changes to the Kerberos database. Aside from Open Directory, `kadmind` is largely manipulated by `kadmin` and `kadmin.local`.

Generally in Mac OS X, Apple applications are responsible for telling `kadmin` what to do, so manual modifications are rarely needed.

The configuration files for `kadmin` and `krb5kdc` are in /var/db/krb5kdc/. The kadm5.acl file is a list of Kerberos principals that have various administrative privileges.

The principal.kadm5 database is the `kadmind` process' policy database. It is located in /var/db/krb5kdc/. Although principals and their keys are stored in /var/db/krb5kdc/principal, policies, which can be applied to principals, are stored in principal.kadm5.

Principal.kadm5.lock is a lock file used by `kadmind`. However, it is unlike most lock files because `kadmind` does not write to the policy or principal database unless it exists.

The `kadmin` tool, in /usr/sbin/, is the native MIT administrative client to `kadmind`. `kadmin` reads the Kerberos configuration file, edu.mit.kerberos, to discover the network location of the `kadmind` server.

Unlike `kadmin`, `kadmin.local` cannot be run remotely, nor is it bound by the access controls of `kadmind`. Instead, it is a brute-force tool that you must always run with root privileges, with full administrative privileges over the `kadmind` and KDC databases. Both `kadmin` and `kadmin.local` can be run interactively or in query mode (using the `-q` flag).

The following examples show basic `kadmin` tool uses.

To add a principal:

```
$ sudo kadmin.local -q "add_principal student1"
```

Replace *`student1`* with the principal you are adding to the database.

To add a service principal:

```
$ sudo kadmin.local -q "add_principal afpserver/server.example.com"
```

Replace *`afpserver/server.example.com`* with the service principal you are adding to the database.

To delete a principal:

```
$ sudo kadmin.local -q "delete_principal student1"
```

Replace *`student1`* with the principal you are deleting from the database.

To view all principals:

```
$ sudo kadmin.local -q list_principals
```

Using kadmin to Kerberize a Service

You can use `kadmin` to kerberize additional services, depending on your specific configuration requirements. Although Mac OS X Server kerberizes many services for you, you can use Kerberos command-line tools to kerberize additional services with Open Directory Kerberos.

A kerberized service must know its principal name. The service type for most services is compiled into the binary.

Often the server administrator can assume that its server's principal name is `serviceType/fqdn@REALM`. For example, the service principal for the AFP server on the host "server.example.com" in the realm "EXAMPLE.COM" is afpserver/server.example.com@EXAMPLE. However, the service type is service-specific and the primary place to get the information is from the service documentation.

To kerberize a service (from a terminal running on that host):

1. To create the service principal, use `kadmin`.

```
$ sudo kadmin -p admin_principal -q "addprinc -randkey service-principal"
```

2. Import the principal key into the `keytab` file.

```
$ sudo kadmin -p admin_principal -q "ktadd service-principal"
```

3. Configure the service to use the new principal.

   This step is service-specific. For information about how to perform this step, see the service documentation.

Using Directory Service Tools

The following are miscellaneous directory service tools that you can use to configure directory services and to troubleshoot problems.

Operating on Directory Service Domains

Use `dscl`, a general-purpose tool, for operating on directory domains. You can create, read, and manage directory data. If invoked without commands, `dscl` runs in an interactive mode, reading commands from standard input.

The following example shows basic `dscl` tool uses:

To verify that you can access an LDAPv3 directory:

```
$ dscl localhost
> cd /LDAPv3/directory.example.com/Users
> ls
```

You should see a list of the server's network user accounts

For more information, see the `dscl` man page.

Manipulating a Single Named Group Record

Use `dseditgroup` to manipulate a single named group record on the default local directory domain or on the specified directory domain. The following examples show uses for `dseditgroup`.

To view the attributes of a group in the local directory domain:

```
$ dseditgroup -o read groupname
```

To create a group in a domain:

```
$ dseditgroup -o create -n /LDAPv3/ldap.example.com -u diradmin_name -P
     diradmin_password -r "Group Name" -c "comment" -s 1234 -k "some
     keyword" groupname
```

To create a Windows group in a domain:

1 Create the group.

```
$ dseditgroup -o create -n /LDAPv3/ldap.example.com -u diradmin_name
     -P diradmin_password -r "Group Name" groupname
```

2 Set the domain group relative identifier (RID).

```
$ dscl -u diradmin_name -P diradmin_password /LDAPv3/ldap.example.com
     -create /Groups/groupname SMBRID RID
```

To delete a group from a domain:

```
$ dseditgroup -o delete -n /LDAPv3/ldap.example.com -u diradmin_name -P
     diradmin_password groupname
```

Parameter	Description
diradmin_name	Name of the directory administrator
diradmin_password	Password of the directory administrator
Group Name	Real name to add or replace
comment	Comment or add or replace
1234	Time-to-live, in seconds, to add or replace
some keyword	Keyword to add
groupname	Group name

For more information, see the `dseditgroup` man page.

Adding or Removing LDAP Server Configurations

Use `dsconfigldap` to add or remove LDAP server configurations in directory services.

To add an LDAP server:

```
$ dsconfigldap -v -a myldap.example.com
```

To remove an LDAP server:

```
$ dsconfigldap -v -r myldap.example.com
```

Configuring the Active Directory Plug-In

Use `dsconfigad` to configure the Active Directory plug-in from the command-line. `dsconfigad` has the same functionality for configuring the Active Directory plug-in as the Directory Access application.

To add a computer to a directory:

```
$ dsconfigad -a computerid -u "administrator" -ou
    "CN=Computers,OU=Engineering,DC=ads,DC=demo,DC=com" -domain
    domain.ads.apple.com
```

Parameter	Description
computerid	The computer ID to add to the domain.
administrator	The user name of a network account that has administrator privileges.
CN=Computers,OU=Engineering,DC=ads,DC=demo,DC=com	The LDAP domain name of the container used for adding the computer. If this is not specified, it defaults to the container.
domain	The fully-qualified domain name of the domain used when adding the computer to the directory.

For more information, see the `dsconfigad` man page.

Configuring the RADIUS Server

To view and configure most RADIUS server settings, use the `radiusconfig` tool.

To view RADIUS server settings:

```
$ sudo radiusconfig -appleversion -getconfig -getconfigxml -nascount
    -naslist -naslistxml -ver -help -q
```

Command Option	Description
`-appleversion`	Displays the version of the toll, including the build version.
`-getconfig`	Displays configuration data stored in the radiusd.conf and eap.conf files in an abbreviated, user-friendly format.
`-getconfigxml`	Displays configuration data stored in the radiusd.conf and eap.conf files in xml plist format.
`-nascount`	Displays the number of RADIUS clients.
`-naslist`	Displays the list of RADIUS clients formatted for the clients.conf file.
`-naslistxml`	Displays the list of RADIUS clients in xml plist format.
`-ver`	Displays a specific build version.
`-help`	Displays usage information.
`-q`	Suppresses prompts.

To start the RADIUS server:

```
$ sudo radiusconfig -start
```

To stop the RADIUS server:

```
$ sudo radiusconfig -stop
```

To disable Transport Level Security (TLS):

```
$ sudo radiusconfig -disable-tls
```

This command disables TLS by commenting-out the TLS section in the eap.conf file.

To enable TLS:

```
$ sudo radiusconfig -enable-tls
```

This command enables TLS by activating the TLS section in the eap.conf file.

To add a Radius client:

```
$ sudo radiusconfig -addclient nas-name shortname [type]
```

Parameter	Description
nas-name	The name of the client.
shortname	The shortname of the client.
type	(Optional) The type of the client.

To import Radius clients:

```
$ sudo radiusconfig -importclients xml-plist-file
```

Parameter	Description
xml-plist-file	The name of the file, including the path, to import clients from.

To remove Radius clients:

```
$ sudo radiusconfig -removeclient nas-name [nas-name …]
```

Parameter	Description
nas-name	The name of the client to remove from the server.

To configure RADIUS service parameters:

```
$ sudo radiusconfig -setconfig key value [key value …]
```

Parameter	Description
key	The name of the key to configure in the radiusd.conf or eap.conf files.
value	The value of the key.

To assign an access control group to a client of the RADIUS service:

```
$ sudo radiusconfig -setgroup nas-name group-name
```

Parameter	Description
nas-name	The name of the client.
group-name	The name of the access control group.

To configure the rotation of RADIUS service logs:

```
$ sudo radiusconfig -rotatelog [-n file-count] base-file
```

Parameter	Description
file-count	The number of log files to preserve.
base-file	The name of the log file.

To configure the automatic rotation of RADIUS service logs:

```
$ sudo radiusconfig -autorotatelog [on | off] [-n file-count]
```

Parameter	Description
on	Enables automatic log rotation.
off	Disables automatic log rotation.
file-count	Specifies the number of log files to preserve.

To configure RADIUS service certificates:

```
$ sudo radiusconfig -installcerts private-key certificate
     [trusted-ca-list [yes | no [common-name]]]
```

Parameter	Description
private-key	The file path to the client's private key to use in the certificate.
certificate	The file path to the certificate.
trusted-ca-list	The file path to the trusted CA list.
yes	A request to check a certificate revocation list.
no	A request to not check a certificate revocation list.
common-name	The common name.

This command changes eap.conf to contain an active TLS section and configures the certificates. This command also replaces the random file and creates the dh file if absent.

Configuring and Managing QuickTime Streaming Server

16

Use this chapter to learn the commands to configure and manage the QuickTime Streaming Server service.

This chapter describes the commands used to configure and manage the QuickTime Streaming Server (QTSS).

Streaming is the delivery of media, such as movies and live presentations, over a network in real time. A streaming server sends the media to a client computer, which plays the media as it is delivered. With streaming, no files are downloaded to the viewer's hard disk.

For more information, see *QuickTime Streaming and Broadcasting Administration*.

Understanding QTSS

Mac OS X Server v10.5 includes the latest version of QTSS, providing a complete solution for streaming live and on-demand media to audiences everywhere. Mac OS X Server makes it easy and affordable to enhance and extend the reach of your communications with rich video and audio content.

QuickTime is one of the most versatile, cost-effective platforms for creating, playing, and streaming digital media over the Internet. It supports all the latest digital media standards, including H.264, AAC, MP3, MPEG-4, and 3GPP, so your content can be played anywhere using standards-compliant media players.

Performing QTSS Tasks

To start QTSS, use the `serveradmin` or `quicktimestreamingserver` tool to specify additional service parameters when you start the service.

Starting and Stopping QTSS

To start QTSS:

```
$ sudo serveradmin start qtss
```

or

```
$ sudo quicktimestreamingserver
```

To see a list of quicktimestreamingserver tool options:

```
$ sudo quicktimestreamingserver -h
```

To stop QTSS:

```
$ sudo serveradmin stop qtss
```

Viewing QTSS Status

To see if the service is running:

```
$ sudo serveradmin status qtss
```

To see complete service status:

```
$ sudo serveradmin fullstatus qtss
```

Viewing QTSS Settings

To view a setting:

```
$ sudo serveradmin settings qtss:setting
```

To view a group of settings:

You can view a group of settings that have part of their names in common by entering as much of the name as you want, stopping at a colon (`:`), and entering an asterisk (`*`) as a wildcard for the remaining parts of the name. For example:

```
$ sudo serveradmin settings qtss:modules:_array_id:QTSSAdminModule:*
```

To view all service settings:

```
$ sudo serveradmin settings qtss
```

Changing QTSS Settings

You can change QTSS settings by using the `serveradmin` tool or by editing the QTSS parameter list file.

To change a setting:

```
$ sudo serveradmin settings qtss:setting = value
```

Parameter	Description
setting	A QTSS service setting. To see a list of available settings, enter: `$ sudo serveradmin settings qtss` or see "Available QTSS Parameters" on page 271.
value	An appropriate value for the setting.

To change several settings:

```
$ sudo serveradmin settings
qtss:setting = value
qtss:setting = value
qtss:setting = value
[...]
Control-D
```

Available QTSS Parameters

To change the QTSS service settings, use the following parameters with the `serveradmin` tool.

Parameters

To see descriptions of most QTSS parameters, see the streamingserver.xml-sample file in /Library/QuickTimeStreaming/Config/.

Look for XML module and pref names that match the last two segments of the parameter name.

For example, to see a description of:

```
modules:_array_id:QTSSFileModule:record_movie_file_sdp
```

look in the sample file for:

```
<MODULE NAME="QTSSFileModule">...
     <PREF NAME="record_movie_file_sdp".
```

Parameter (`qtss:`)	Description
`broadcaster:password`	Default = `""`
`broadcaster:username`	Default = `""`
`modules:_array_id:QTSSAccessLogModule:request_logfile_dir`	Default = `"/Library/QuickTime Streaming/Logs/"`
`modules:_array_id:QTSSAccessLogModule:request_logfile_interval`	Default = `7`

Parameter (qtss:)	Description
modules:_array_id:QTSSAccessLogModule: request_logfile_name	Default = "StreamingServer"
modules:_array_id:QTSSAccessLogModule: request_logfile_size	Default = 10240000
modules:_array_id:QTSSAccessLogModule: request_logging	Default = yes
modules:_array_id:QTSSAccessLogModule: request_logtime_in_gmt	Default = yes
modules:_array_id:QTSSAccessModule: modAccess_groupsfilepath	Default = "/Library/Quick TimeStreaming/Config/ qtgroups"
modules:_array_id:QTSSAccessModule: modAccess_qtaccessfilename	Default = "qtaccess"
modules:_array_id:QTSSAccessModule: modAccess_usersfilepath	Default = "/Library/Quick TimeStreaming/Config/ qtusers"
modules:_array_id:QTSSAdminModule: AdministratorGroup	Default = "admin"
modules:_array_id:QTSSAdminModule: Authenticate	Default = yes
modules:_array_id:QTSSAdminModule: enable_remote_admin	Default = yes
modules:_array_id:QTSSAdminModule: IPAccessList	Default = "127.0.0.*"
modules:_array_id:QTSSAdminModule: LocalAccessOnly	Default = yes
modules:_array_id:QTSSFileModule: add_seconds_to_client_buffer_delay	Default = 0
modules:_array_id:QTSSFileModule: admin_email	Default = ""
modules:_array_id:QTSSFileModule: record_movie_file_sdp	Default = no
modules:_array_id:QTSSHomeDirectoryModule: enabled	Default = no
modules:_array_id:QTSSHomeDirectoryModule: movies_directory	Default = "/Sites/Streaming"
modules:_array_id:QTSSMP3StreamingModule: mp3_broadcast_buffer_size	Default = 8192
modules:_array_id:QTSSMP3StreamingModule: mp3_broadcast_password	Default = ""
modules:_array_id:QTSSMP3StreamingModule: mp3_max_flow_control_time	Default = 10000

Parameter (qtss:)	Description
modules:_array_id:QTSSMP3StreamingModule: mp3_request_logfile_dir	Default = "/Library/QuickTime Streaming/Logs/"
modules:_array_id:QTSSMP3StreamingModule: mp3_request_logfile_interval	Default = 7
modules:_array_id:QTSSMP3StreamingModule: mp3_request_logfile_name	Default = "mp3_access"
modules:_array_id:QTSSMP3StreamingModule: mp3_request_logfile_size	Default = 10240000
modules:_array_id:QTSSMP3StreamingModule: mp3_request_logging	Default = yes
modules:_array_id:QTSSMP3StreamingModule: mp3_request_logtime_in_gmt	Default = yes
modules:_array_id:QTSSMP3StreamingModule: mp3_streaming_enabled	Default = yes
modules:_array_id:QTSSReflectorModule: allow_broadcasts	Default = yes
modules:_array_id:QTSSReflectorModule: allow_non_sdp_urls	Default = yes
modules:_array_id:QTSSReflectorModule: BroadcasterGroup	Default = "broadcaster"
modules:_array_id:QTSSReflectorModule: broadcast_dir_list	Default = ""
modules:_array_id:QTSSReflectorModule: disable_overbuffering	Default = no
modules:_array_id:QTSSReflectorModule: enable_broadcast_announce	Default = yes
modules:_array_id:QTSSReflectorModule: enable_broadcast_push	Default = yes
modules:_array_id:QTSSReflectorModule: ip_allow_list	Default = "127.0.0.*"
modules:_array_id:QTSSReflectorModule: kill_clients_when_broadcast_stops	Default = no
modules:_array_id:QTSSReflectorModule: minimum_static_sdp_port	Default = 20000
modules:_array_id:QTSSReflectorModule: timeout_broadcaster_session_secs	Default = 20
modules:_array_id:QTSSRelayModule: relay_prefs_file	Default = "/Library/Quick TimeStreaming/Config/ relayconfig.xml"
server:authentication_scheme	Default = "digest"
server:auto_restart	Default = yes
server:default_authorization_realm	Default = "Streaming Server"

Parameter (qtss:)	Description
server:do_report_http_connection_ip_address	Default = no
server:error_logfile_dir	Default = "/Library/QuickTimeStreaming/Logs/"
server:error_logfile_name	Default = "Error"
server:error_logfile_size	Default = 256000
server:error_logfile_verbosity	Default = 2
server:error_logging	Default = yes
server:force_logs_close_on_write	Default = no
server:maximum_bandwidth	Default = 102400
server:maximum_connections	Default = 1000
server:module_folder	Default = "/Library/QuickTimeStreaming/Modules/"
server:movie_folder	Default = "/Library/QuickTimeStreaming/Movies/"
server:pid_file	Default = "/var/run/QuickTimeStreamingServer.pid"
server:reliable_udp	Default = yes
server:reliable_udp_dirs	Default = "/"
server:run_group_name	Default = "qtss"
server:run_num_threads	Default = 0
server:run_user_name	Default = "qtss"
web_admin:enabled	Default = no
web_admin:password	Default = ""
web_admin:username	Default = ""

Managing QTSS

To manage QTSS, use the following commands with the serveradmin tool.

Command (qtss:command=)	Description
getConnections	View QTSS connections. See "Viewing QTSS Connections" on this page.
getHistory	View QTSS statistics. See "Viewing QTSS Statistics" on page 275.
getLogPaths	Find the current location of the service logs. See "Viewing Service Logs and Log Paths" on page 276.

Viewing QTSS Connections

To retrieve information about QTSS connections, use the `serveradmin getConnections` command.

To view a list of connected users:

```
$ sudo serveradmin command qtss:command = getConnections
```

Viewing QTSS Statistics

To display a log of periodic samples of the number of connections and the data throughput, use the `serveradmin getHistory` command. Samples are taken once each minute.

To view samples:

```
$ sudo serveradmin command
qtss:command = getHistory
qtss:variant = statistic
qtss:timeScale = scale
Control-D
```

Parameter	Description
statistic	The value you want to display. Values: • `v1`—Number of connected users (average during sampling period) • `v2`—Throughput (bytes/sec)
scale	The length of time in seconds, ending with the current time, that you want to see samples for. For example, to see 30 minutes of data, specify `qtss:timeScale = 1800`.

The computer responds with the following output:

```
qtss:nbSamples = <samples>
qtss:samplesArray:_array_index:0:vn = <sample>
qtss:samplesArray:_array_index:0:t = <time>
qtss:samplesArray:_array_index:1:vn = <sample>
qtss:samplesArray:_array_index:1:t = <time>
[...]
qtss:samplesArray:_array_index:i:vn = <sample>
qtss:samplesArray:_array_index:i:t = <time>
qtss:vnLegend = "<legend>"
qtss:currentServerTime = <servertime>
```

Value displayed by `getHistory`	Description
`<samples>`	The total number of samples listed.

`<legend>`	A textual description of the selected statistic.
	`"CONNECTIONS"` for `v1`
	`"THROUGHPUT"` for `v2`

Value displayed by `getHistory`	Description
`<sample>`	The numerical value of the sample. For connections (`v1`), this is integer average number of connections. For throughput, (`v2`), this is integer bytes per second.
`<time>`	The time when the sample was measured. A standard UNIX time (number of seconds since September 1, 1970). Samples are taken every 60 seconds.

Viewing Service Logs and Log Paths

To view the contents of the QTSS logs, use `tail` or another file listing tool.

To view the latest entries in a log:

```
$ tail log-file
```

To see where the current QTSS error and activity logs are located, use the `serveradmin getLogPaths` command.

To view log paths:

```
$ sudo serveradmin command qtss:command = getLogPaths
```

The computer responds with the following output:

```
qtss:accessLog = <access-log>
qtss:errorLog = <error-log>
```

Value	Description
`<access-log>`	The location of the QTSS service access log. Default = `/Library/QuickTimeStreaming/Logs/StreamingServer.log`
`<error-log>`	The location of the QTSS service error log. Default = `/Library/QuickTimeStreaming/Logs/Error.log`

Forcing QTSS to Reread Preferences

You can force QTSS to reread preferences without restarting the server.

To force QTSS to reread preferences:

1. Log in as root.
2. List the QTSS processes:

```
$ ps -ax | grep QuickTimeStreamingServer
```

You should see a list similar to the following:

```
949 ??          0:00.00 QuickTimeStreamingServer
950 ??          0:00.13 QuickTimeStreamingServer
965 ttys000     0:00.00 grep QuickTimeStreamingServer
```

3 Send a HUP signal to one of the two process IDs (PIDs) for QuickTimeStreamingServer (949 or 950). For example:

```
$ kill -HUP 950
```

Preparing Older Home Folders for User Streaming

To enable QTSS home folder stream for home folders created using an earlier version of Mac OS X Server (before v10.3), use the `createuserstreamingdir` tool to set up the streaming media folder in each user's Home folder.

To set up Sites/Streaming/ in older Home folders:

```
$ createuserstreamingdir user
```

Parameter	Description
user	The user in whose Home folder the Sites/Streaming/ folder is created.

Configuring Streaming Security

Some security is inherent in real-time streaming, because content is delivered only as the client needs it and no files remain afterward. However, other security issues usually need to be addressed. Aspects of streaming security covered in this section include:

- Setting up password protection for content
- Configuring `qtaccess` to limit access to the media folder

Resetting the Streaming Server Admin User Name and Password

If you forget the Streaming Server Web Admin or Broadcast user name or password, you can reset them from the command line.

To add a user or reset an existing user's password:

1 Log in to the server computer, open a Terminal window, and enter:

```
$ sudo qtpasswd someUserName
```

Replace *someUserName* with a name of your choice.

2 Follow the prompts to enter and confirm the password.

To reset the web-based administrator user name:

1 Log in to the server computer and open a Terminal window.

2 Remove the old admin username by entering:

```
$ sudo qtpasswd -R admin
```

3 Add a new admin username by entering:

```
$ sudo qtpasswd -A admin someUserName
```

4 If the new admin user doesn't exist, follow the prompts to enter and confirm the password.

To reset the broadcaster user name and password:

1. Log in to the server computer and open a Terminal window.
2. Remove the old broadcaster username by entering:

```
$ sudo qtpasswd -R broadcaster
```

3. Add a new broadcaster username by entering:

```
$ sudo qtpasswd -A broadcaster someUserName
```

4. If the new broadcaster user doesn't exist, follow the prompts to enter and confirm the password.

Controlling Access to Streamed Media

You can set up authentication to control client access to streamed media files. Two schemes of authentication are supported: basic and digest. By default, the server uses the more secure digest authentication.

You can also control playlist access and administrator access to your streaming server. Authentication does not control access to media streamed from a relay server. The administrator of the relay server must set up authentication for relayed media. The ability to manage user access is built into the streaming server, so it is always enabled.

For folder-level access control to work, an access file must be present in the streaming media folder of the file being accessed. If an access file is not present in the folder of the requested file, access is controlled by the Server Admin QTSS guest access setting (the default is guest access enabled) in combination with the Server Admin Access settings for nonguest users.

When an access file is present, it functions as a fine-grain folder level control and overrides other access settings. That is, even when guest access is enabled, a streaming media folder with an access file might still require authenticated access for specified users and groups or allow guest access to a specific folder when guest access is disabled by Server Admin.

To set up qtpasswd-based user access control:

1. Create user accounts using the `qtpasswd` tool.
2. Create an access file containing the users and groups created with `qtpasswd` and place it in the media folder you want to protect.
3. If you want to disable authentication for a media folder, remove the access file (called `qtaccess`) or rename it (for example, `qtaccess.disabled`).

To set up Open-Directory-based user access control:

1. Create user accounts and passwords using Open-Directory-based user account services (for example, users created with System Settings, System Preferences, or Workgroup Manager).
2. Create an access file containing the Open Directory users and groups and place it in the media folder you want to protect.

Note: You can designate `qtpasswd`-based and Open-Directory-based users and groups in the same access file.

Note: QTSS supports Open-Directory-based user authentication and group membership checking that is compatible with `qtaccess` file authorization, Server Admin SACL support, and Server Admin to easily enable and disable authenticated access.

Creating an Access File

An access file is a text file called `qtaccess` that contains information about users and groups authorized to view media in the folder where the access file is stored.

The folder you use to store streamed media can contain other folders, and each folder can have its own access file.

When a user tries to view a media file, the server looks for an access file to see whether the user is authorized to view the media. The server looks first in the folder where the media file is located. If an access file is not found, it looks in the enclosing folder.

The first access file that's found is used to determine whether the user is authorized to view the media file. The access file for the streaming server works like the Apache web server access file.

You can create an access file with any text editor. The filename must be qtaccess and the file can contain some or all of the following information:

```
AuthName message
AuthUserFile user filename
AuthGroupFile group filename
require user username1 username2
require group groupname1 groupname2
require valid-user
require any-user
```

Terms not in angle brackets are keywords. Anything in angle brackets is information you supply. Save the access file as plain text (not as .rtf or another file format).

The following is a description of the parameters in the qtaccess file:

Parameter	Description
message	(Optional) Text your users see when the login window appears. If your message contains white space (such as a space character between terms), enclose the text in quotation marks.
user filename	The path and filename of the user file: • For Mac OS X, the default is /Library/QuickTimeStreaming/Config/qtusers. • For other supported platforms, it is /etc/streaming/qtusers.
group filename	(Optional) The path and filename of the group file: • For Mac OS X, the default is /Library/QuickTimeStreaming/Config/qtgroups. • For other supported platforms, it is /etc/streaming/qtgroups. If you have many users, it may be easier to set up one or more groups, and then enter the group names, than to list each user.
username	A user authorized to log in and view the media file. The user's name must be in the user file you specified. You can also specify the following: • `valid-user` is any user defined in the `qtusers` file or any bound directory server. The statement `require valid-user` specifies that any authenticated user can access the media files. If this tag is used, the server prompts users for an appropriate user name and password. • `any-user` allows any user to view media without providing a name or password.
groupname	A group whose members are authorized to log in and view the media file. The group and its members must be listed in the group file you specified.

You can also add the keyword `AuthScheme` with the values `basic` or `digest` to a `qtaccess` file. This overrides the global authentication setting on a folder-by-folder basis.

Accessing Protected Media

To access a media file that digest authentication is enabled for, users must have QuickTime 5 or later. If your streaming server is set up to use basic authentication, users need QuickTime 4.1 or later.

Users must enter their user names and passwords to view the media file.

Users who try to access a media file with an earlier version of QuickTime will see the error message `401: Unauthorized`.

Adding User Accounts and Passwords

You can add a user account and password if you log in to the server computer.

To add a user account:

1. Enter the following:

```
$ sudo qtpasswd -f user filename user-name
```

2. Enter a password for the user and reenter it when prompted.

Adding or Deleting Groups

You can edit the /Library/QuickTimeStreaming/Config/qtgroups file with any text editor as long as the file follows this format:

```
groupname: user-name1 user-name2 user-name3
```

To add or delete a group, edit the group file you set up.

Making Changes to the User or Group File

You can make changes to the user or group file if you log in to the server computer.

To delete a user from a user or group file:

- Log in to the server computer as administrator, open Terminal window, and enter:

```
$ sudo qtpasswd -d user
```

To change a user password:

1. Enter the following:

```
$ sudo qtpasswd user-name
```

2. Enter a new password for the user.

The password you enter replaces the password in the file.

Manipulating QuickTime and MP4 Movies

To manipulate QuickTime and MP4 movies, use the `qtmedia` tool. You can add hint tracks, prepare for fast-start, and edit annotations.

For more information, run the `qtmedia` tool to display the command-line options.

Creating Reference Movies

To create reference movies that can be used to embed QuickTime content in Web pages, use the `qtref` tool. You can use the following options:

Parameter	Description
`-r`	Create QuickTime Atom ref movie with extension `.qtl`
`-t`	Create XML text ref movie with extension `.qtl`
`-a`	Create alternate data rate movie with extension `.qtl`

For more information about using `qtref`, enter the command without arguments to display usage information.

Configuring the Podcast Producer Service

17

Use this chapter to learn how to control and manage Podcast Capture and the Podcast Producer service.

Mac OS X Server v10.5 provides command-line tools for controlling a Podcast Producer solution. These commands provide the same functionality available in Podcast Capture and the Podcast Producer pane of Server Admin, and more.

For more information about Podcast Capture, see its online help. For more information about Podcast Producer, see *Podcast Producer Administration*.

Controlling Podcast Capture

You can use the `/usr/bin/podcast` tool to control the client-side functionality of Podcast Producer, which is represented by Podcast Capture.

For more information about `podcast`, see its man page.

Connecting to a Podcast Producer Server

If you run `podcast` on a client system remotely connected to a Podcast Producer server, you must specify the server, username, and password when you use the `podcast` command to send or receive information from the Podcast Producer server.

For example, to submit a job, you enter:

```
$  podcast --server pcast.example.com --user annej --pass password
     --submit --file file_path --workflow workflow_name
```

Note: In the following sections, the server, username, and password information is omitted. However, when using `podcast` to interact with the Podcast Producer server, provide this information.

Submitting QuickTime Movies for Processing

To submit a QuickTime movie to Podcast Producer for processing:

```
$ podcast --submit --file file_path --workflow workflow_name
     [--upload_buffer_size size]
```

You can submit multiple files and specify metadata (submission description) and upload buffer size (optional).

For example, to specify a .plist file containing a description of the submission and to specify a smaller upload size than the default 128 KB, enter:

```
$ podcast --submit --file file_path --workflow workflow_name
     --upload_buffer_size 64
```

The following is an example of a .plist file containing metadata describing a job submission:

```
<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE plist PUBLIC "-//Apple//DTD PLIST 1.0//EN" "http://
     www.example.com/DTDs/PropertyList-1.0.dtd">
<plist version="1.0">
             <dict>
                     <key>Author</key>
                     <string>Professor Rick Fernwood</string>
                     <key>Comment</key>
                     <string>Strangely, things fall together.</string>
                     <key>Copyright</key>
                     <string>Copyright (c) 2007 Rick Fernwood</string>
                     <key>Description</key>
                     <string>Gravitation is a phenomenon through which all
     objects attract each other.</string>
                     <key>Keywords</key>
                     <string>gravitation phys13</string>
                     <key>Title</key>
                     <string>Lecture 4 -- The theory of gravitation</string>
             </dict>
</plist>
```

Viewing Cameras and Workflows

To view available cameras for capturing video:

```
$ podcast --listcameras
```

To view available workflows for using when submitting content:

```
$ podcast --listworkflows
```

Viewing and Clearing Uploads

To view upload status and a list of uploaded movies:

```
$ podcast --list_uploads
```

To clear workflows from the list of uploaded workflows:

```
$ podcast --clear_completed_workflows
```

Binding and Unbinding Cameras

To bind a camera to the Podcast Producer server:

```
$ sudo podcast --bind camera_name
```

To unbind a camera from the Podcast Producer server:

```
$ sudo podcast --unbind camera_name
```

To find out whether a camera is bound to the Podcast Producer server:

```
$ sudo podcast --isbound
```

This command returns `1` if the local camera agent is bound to the Podcast Producer server; otherwise, it returns `0`.

Configuring Podcast Producer Agent

To view available capture presets for camera agent configuration:

```
$ podcast --presets
```

To view available audio and video devices for camera agent configuration:

```
$ podcast --devices
```

To view agent configuration settings:

```
$ sudo podcast --presets
```

To configure camera agent settings:

```
$ sudo podcast --setconfig key=value[;key=value]
```

For example, to set the capture setting for the camera agent to the best video resolution (Better):

```
$ sudo podcast --setconfig Capture=Video:Better
```

Controlling Cameras

To get the status of a camera:

```
$ podcast --status camera_name
```

To start video capture on a camera:

```
$ podcast --start camera_name
```

To start audio-only capture on a camera:

```
$ podcast --start camera_name --audio_only
```

To stop video capture on a camera:

```
$ podcast --stop camera_name --metadata metadata_file_path
     --workflow workflow_name
```

To pause video capture on a camera:

```
$ podcast --pause camera_name
```

To resume video capture on a camera:

```
$ podcast --resume camera_name
```

Configuring Podcast Producer Service

Use the `/usr/bin/pcastconfig` tool to configure the Podcast Producer service.

For more information about `pcastconfig`, see its man page.

Configuring Workflows

To enable a workflow that has been added to the Podcast Producer database:

```
$ sudo pcastconfig --enable_workflow workflow_name
```

To disable a workflow that has been added to the Podcast Producer database:

```
$ sudo pcastconfig --disable_workflow workflow_name
```

To validate the contents of a workflow in the Podcast Producer database:

```
$ sudo pcastconfig --validate_workflow workflow_name
```

To validate the contents of all workflows in the Podcast Producer database:

```
$ sudo pcastconfig --validate_all_workflows
```

To validate the contents of a workflow not in the Podcast Producer database:

```
$ sudo pcastconfig --validate_workflow_at_path path_to_workflow
```

To update the Podcast Producer database:

```
$ sudo pcastconfig --update_workflows_in_db
```

This command updates the database to include all workflows stored in:

- /System/Library/PodcastProducer/Workflows/
- /Library/PodcastProducer/Workflows/

To cache a workflow in the Podcast Producer shared file system:

```
$ sudo pcastconfig --cache_workflow workflow_name
```

Configuring Cameras

To enable a camera:

```
$ sudo pcastconfig --enable_camera camera_name
```

To disable a camera:

```
$ sudo pcastconfig --disable_camera camera_name
```

Configuring Properties

To add a custom property:

```
$ sudo pcastconfig --add_property p_name --value p_value [--protect]
```

To remove a custom property:

```
$ sudo pcastconfig --remove_property p_name
```

Controlling Access to Properties

To control access to a list of properties:

```
$ sudo pcastconfig --add_access access_group --properties property_list
```

This command lets you create a one-time access key that allows the specified group to access a list of colon-separated properties (for example, "p1:p2:p3").

To remove access to a list of properties:

```
$ sudo pcastconfig --remove_access access_group
```

Setting Up Podcast Producer as an Upload-Only Node

To set up Podcast Producer to be an upload-only node:

```
$ sudo pcastconfig --create_upload_node shared_file_system_path
```

In this mode, the server runs only Apache and HTTPS upload CGI.

Controlling Podcast Producer Service

Use the `/usr/sbin/pcastctl` tool to start, stop, and restart the Podcast Producer server or agent and to view the status of running daemons.

For more information about `pcastctl`, see its man page.

Starting and Stopping the Podcast Producer Service

To start the Podcast Producer service:

```
$ sudo pcastctl server start
```

To stop the Podcast Producer service:

```
$ sudo pcastctl server start
```

To restart the Podcast Producer service:

```
$ sudo pcastctl server restart
```

Viewing Status Information

To view the status of the Podcast Producer service:

```
$ sudo pcastctl server status
```

To view the status of the Podcast Producer agent daemon:

```
$ sudo pcastctl agent status
```

Launching Podcast Producer Server Upon System Startup

To edit the launchd configuration to launch the Podcast Producer server upon system startup:

```
$ sudo pcastctl server on
```

To edit the launchd configuration to not launch the Podcast Producer server upon system startup:

```
$ sudo pcastctl server off
```

Processing Submitted Content

Use the `/usr/bin/pcastaction` tool in workflows. It provides a rich set of commands for processing and producing audio and video podcasts.

The following is a description of the `pcastaction` commands you can use in workflows.

Command	Description
`pcastaction annotate`	Adds annotations to the input movie.
`pcastaction approval`	Submits content for approval.
`pcastaction archive`	Archives the input movie at the specified location.
`pcastaction encode`	Encodes the input movie using the specified codec.
`pcastaction iTunes`	Instructs the iTunes Store to check the specified RSS feed for new episodes.
`pcastaction iTunesU`	Posts the input video at the specified iTunes U tab.
`pcastaction groupblog`	Posts to a group's wiki blog.
`pcastaction mail`	Sends a notification mail to the specified user using the mail template in the workflow's Resources/Templates folder.
`pcastaction merge`	Merges two movies with a fade transition between them.
`pcastaction preflight`	Runs the preflight script (System/Library/PodcastProducer/Resources/Tools/preflight_script) with the specified arguments.
`pcastaction postflight`	Runs the postflight script (System/Library/PodcastProducer/Resources/Tools/postflight_script) with the specified arguments.
`pcastaction publish`	Publishes the input file to a web or QTSS server.
`pcastaction qceffect`	Applies a Quartz Composer effect (composition) to a movie.
`pcastaction shell`	Runs the specified shell script with the specified arguments.
`pcastaction template`	Processes a web or mail template into a localized content block to be used in mail or web postings.
`pcastaction title`	Adds the supplied title to the input video.
`pcastaction unpack`	Unpacks folder archives before running the main part of a workflow.
`pcastaction watermark`	Superimposes the specified image as a watermark over the input video.

For more information about `pcastaction` and its commands, see its man page. You can also view help information about the commands of `pcastaction` by entering:

```
$ pcastaction help command
```

Applying Quartz Composer Compositions to Movies

Quartz Composer supports the notion of composition protocols and repositories where compositions are stored.

Quartz Composer supports these types of composition protocols:

- Animation
- Transition
- Effect

Podcast Producer leverages compositions that abide by the Transition and Effect protocols. These compositions must reside in one of the following repositories:

Repository	Description
/System/Library/Compositions	Contains Apple-supplied compositions.
/Library/Compositions	Contains user-created or third party compositions.

When specifying a Quartz Composer composition that abides by either the Effect or Transition protocol, use the composition's repository identifier. Otherwise, you can pass the composition's full file path.

Applying a Quartz Composer Transition

To apply a Quartz Composer transition to a QuickTime movie:

```
$ pcastaction merge --basedir=basedir --input1=first_input_movie
     --input2=second_input_movie --output=output_movie
     --duration=transition_duration
     --transition=composition_repository_identifier
```

For example, to apply the Copy Machine transition between movies, enter:

```
$ pcastaction merge --basedir=basedir --input1=first_input_movie
     --input2=second_input_movie --output=output_movie
     --duration=transition_duration
     --transition="/copy machine"
```

or

```
$ pcastaction merge --basedir=basedir --input1=first_input_movie
     --input2=second_input_movie --output=output_movie
     --duration=transition_duration
     --transition=/copy\ machine
```

Note: Enclose the repository identifier in double quotes (`"/copy machine"`) or escape the spaces by adding a backslash character before a space (`/copy\ machine`).

Podcast Producer provides the following Quartz Composer transitions:

Transition	Composition Repository Identifiers
Copy Machine	"/copy machine"
Cube	"/cube"
Dissolve	"/dissolve"
Mask	"/mask"
Mod	"/mod"
Mosaic Flip	"/mosaic flip"
Push	"/flip"
Swing	"/swing"
Zoom Dissolve	"/zoom dissolve"

Applying a Quartz Composer Effect

To apply a Quartz Composer effect to a QuickTime movie:

```
$ pcastaction qceffect --basedir=basedir --input=input_movie
     --output=output_movie --composition=composition_repository_identifier
```

For example, to apply the Blue Print effect to a movie, enter:

```
$ pcastaction qceffect --basedir=basedir --input=input_movie
     --output=output_movie --composition="/blue print"
```

or

```
$ pcastaction qceffect --basedir=basedir --input=input_movie
     --output=output_movie --composition=/blue\ print
```

Podcast Producer provides the following Quartz Composer effects:

Transition	Composition Repository Identifiers
ASCII Art	"/ascii art"
Black and White	"/black and white"
Blue Print	"/blue print"
Blur	"/blur"
Bulge	"/bulge"
City Lights	"/city lights"
Color Controls	"/color controls"
Color Invert	"/color invert"
Color Pencil	"/color pencil"
Comic Book	"/comic book"

Transition	Composition Repository Identifiers
Compound Eye	"/compound eye"
Concert	"/concert"
Crystallize	"/crystallize"
Dent	"/dent"
Dot Screen	"/dot screen"
Exposure Adjust	"/exposure adjust"
False Color	"/false color"
Film Stock	"/film stock"
Fish Eye	"/fish eye"
Flip Flop	"/flip flop"
Gamma Adjust	"/gamma adjust"
Glow	"/glow"
Image Resizer	"/image resizer"
Kaleidescope	"/kaleidescope"
Light Tunnel	"/light tunnel"
Line Overlay	"/light overlay"
Line Screen	"/line screen"
Mirror	"/mirror"
Monochrome	"/monochrome"
Motion Blur	"/motion blur"
Neon	"/neon"
Pinch	"/pinch"
Pixellate	"/pixellate"
Pointillize	"/pointillize"
Pop Art	"/pop art"
Posterize	"/posterize"
Sepia	"/sepia"
Sharpen	"/sharpen"
Squeeze	"/squeeze"
Stretch	"/stretch"
Thermal Camera	"/thermal camera"
Tracer	"/tracer"
Twirl	"/twirl"
X-Ray	"/x-ray"

Shared File System Uploading Mechanisms

Podcast Producer provides the following mechanisms for uploading content to the shared file system:

- Copy upload (file_upload_url)
- FTP upload (ftp_upload_url)
- HTTPS CGI POST upload (https_upload_url)

Podcast Producer stores the configuration information for these mechanisms in its server preferences file (/Library/Preferences/com.apple.pcastserverd.plist), as shown in this example:

```
…
<key>file_upload_url</key>
<string>file:///Xgrid/PodcastProducer/Submissions</string>
<key>ftp_upload_url</key>
<string>ftp://my_username:my_password@example.com</string>
<key>https_upload_url</key>
<string>https://nbrosnahan1.apple.com:8170/cgi-bin/upload.cgi</string>
…
```

To upload content to the shared file system, Podcast Producer tries to use the copy upload method first. If it can't use this method, it tries to use the FTP upload method. If Podcast Producer can't use this method, it uses the third method (HTTPS CGI POST upload).

In the /Library/Preferences/com.apple.pcastserverd.plist file, if an upload method key does not have a value, Podcast Producer skips this upload method. Also, if the value of the ftp_upload_url key is `username:password`, Podcast Producer assumes that this key hasn't been configured and won't use it.

Note: The value of the ftp_upload_url key in the /Library/Preferences/com.apple.pcastserverd.plist file is provided as an example.

Copy Upload

Podcast Producer uses the file_upload_url mechanism only if the file system specified in the URL is available on the system executing the `podcast` CLI tool.

The `podcast` tool attempts to copy the file into the appropriate folder, which requires that any user executing `podcast` be a member of the submissions_groupname group.

FTP Upload

Podcast Producer uses the ftp_upload_url mechanism as a last resort because it is not a secure mechanism for uploading.

If the Podcast Producer system is deployed in a secure setting, there can be a significant speed advantage to using FTP for uploading content because there is no encryption overhead.

Important: The FTP user you define for the ftp_upload_url key must be a member of the submissions_groupname group.

HTTPS CGI POST Upload

The https_upload_url mechanism performs a file-upload POST command to a CGI script running on the Podcast Producer server (or on another server where the CGI is configured).

Because the Apache instance that hosts the CGI scripts on Mac OS X Server v10.5 runs as the _pcastserver user, the CGI script requires that _pcastserver be a member of the submissions_groupname group.

Configuring and Managing iCal Service and iChat Service

18

Use this chapter to learn the commands to configure and administer iCal and iChat services.

This chapter describes the commands for configuring and managing iCal and iChat services.

For more information, see *iCal Service Administration* and *iChat Service Administration*.

Configuring iCal Service

To start and stop the iCal service and to configure its settings, use the `caldavd` command-line tool. For more information about this tool, see its man page.

To view help information:

```
$ caldavd -h
```

To start the service:

```
$ sudo caldavd start
```

To stop the service:

```
$ sudo caldavd stop
```

To restart the service:

```
$ sudo caldavd restart
```

To specify the path to the configuration file:

```
$ sudo caldavd -f path
```

Replace *path* with the name of the configuration file, including the path (for example, /etc/caldavd/caldavd.plist).

To modify service settings:

1. Open the iCal service configuration file (caldavd.plist), which is stored in the /etc/caldavd/ folder by default.
2. Modify the following settings:
   - To specify the document root for iCal service, modify the Document Root key.
   - To specify the port number the service uses, modify the Port key.
   - To enable or disable SS, modify the SSEnable key.

Configuring iChat Service

To start and stop the iChat service and to configure its settings, use the `serveradmin` command-line tool.

To start the service:

```
$ sudo serveradmin start jabber
```

To stop the service:

```
$ sudo serveradmin stop jabber
```

To view service settings:

```
$ sudo serveradmin settings jabber
```

The following is an example of the output:

```
jabber:savedChatsArchiveInterval = 7
jabber:enableAutoBuddy = yes
jabber:s2sAllowedDomains = _empty_array
jabber:requireSecureS2S = no
jabber:sslKeyFile = "/etc/certificates/Default.crtkey"
jabber:hosts:_array_index:0 = "pb4server"
jabber:s2sRestrictDomains = no
jabber:eventLogArchiveInterval = 7
jabber:savedChatsLocation = "/var/jabberd/message_archives"
jabber:enableXMPP = yes
jabber:enableSavedChats = no
jabber:welcomeMessage = "Welcome to the iChat Server at pb4server!"
jabber:logLevel = "ALL"
```

To set service settings:

```
$ sudo serveradmin settings jabber:setting = value
```

Replace *`setting`* with the name of the setting and *`value`* with the value of the setting.

For more information about iChat service, see the man page for `serveradmin`.

Configuring and Managing System Logging

19

Use this chapter to learn the commands to configure and manage system logging.

Logging System Events

Logs are text files that form a record of what has occurred on the system, much like a journal.

Configuring the Log File

Log files are maintained in the /Library/Logs/ and /var/log/ folders. Some commonly monitored log files include console.log and system.log. Applications can have their own log files located in different folders.

Console.log is located in /Library/Logs/Console/*uid*, where *uid* is the user ID. Console.log contains recent console activity.

System.log is located in /var/log/ and contains a log of all system activity, including console log information.

Configuring System Logging

The configuration file for the system logging daemon, `syslogd`, is /etc/syslog.conf. Each line in /etc/syslog.conf consists of text containing the following types of data:

- Facility: Specifies categories of log messages. Standard facilities include `mail`, `news`, `user`, and `kern` (kernel).
- Priority: Specifies the urgency of the message. The order from least to most critical is `debug`, `info`, `notice`, `warning`, `err`, `crit`, `alert`, and `emerg`. The priority of the log message is set by the application sending it, not by `syslogd`.
- Action: Specifies what to do with a log message of a specific facility and priority. Messages can be sent to files, named pipes, devices, or to a remote host.

The following example specifies that for any log messages in the category `mail`, with a priority of `emerg` or higher, the message is written to the /var/log/mail.log file:

```
mail.emerg /var/log/mail.log
```

The facility and priority are separated by a single period, and these are separated from the action by tabs. Wildcards ("*") can also be used. The following example line logs all messages of any facility or priority to the file /var/log/all.log:

```
*.* /var/log/all.log
```

For information about the configuration of this file, see the `syslog.conf` man page.

Local Logging

The default configuration in /etc/syslog.conf is appropriate for a Mac OS X Server system if a remote log server is not available. The computer is set to rotate log files using a cron job at the time intervals specified in the file /etc/crontab.

Rotation entails compressing the current log file, incrementing the integer in the filename of compressed log files, and creating a log file for new messages. For example, the following files were created in the /var/log/ folder:

```
system.log
system.log.0.gz
system.log.1.gz
system.log.2.gz
system.log.3.gz
system.log.4.gz
```

The log files are rotated by a cron job, and the rotation occurs if the computer is on when the job is scheduled. By default, log rotation tasks are scheduled for early in the morning (for example, 4:30 a.m. on Saturday) to be as unobtrusive as possible. If the computer will not be on at this time, adjust the settings in /etc/crontab.

The following example shows the default for running the weekly log rotation script, which is configured for 4:15 a.m. on the last day of the week, Saturday (Sunday is 0). An asterisk denotes "any," so a line of all asterisks would execute every minute.

```
DayOf DayOf
#Minute Hour Month Month Week User Command
15 4 * * 6 root periodic weekly
```

The following line would change the time to 12:15 p.m. on Tuesday, when the computer is more likely to be on:

```
DayOf DayOf
#Minute Hour Month Month Week User Command
15 12 * * 2 root periodic weekly
```

For more information about editing the /etc/crontab file, see the `crontab` man page.

Remote Logging

Using remote logging in addition to local logging is strongly recommended for any server system, because local logs can easily be altered if the system is compromised.

Several security issues must also be considered when making the decision to use remote logging:

- The `syslog` process sends log messages as clear text, which could expose sensitive information.
- Too many log messages can fill storage space on the logging system, making further logging impossible.
- Log files can indicate suspicious activity only if a baseline of normal activity has been established, and if the files are regularly monitored for such activity.

If these security issues outweigh the security benefit of remote logging, do not use remote logging.

Configuring Remote Logging on a Client Computer

To configure a client computer for remote logging, alter the syslog.conf configuration file. The following instructions assume that a remote log server has been configured on the network.

To enable remote logging:

1. On the client computer, open the /etc/syslog.conf file with root privileges.
2. Add the following line to the top of the file, replacing *your.log.server* with the name or IP address of the log server and keeping all other lines intact:

   ```
   *.* @your.log.server
   ```

3. Exit, saving changes.
4. Send a hangup signal to `syslogd` to make it reload the configuration file:

   ```
   $ sudo killall - HUP syslogd
   ```

Configuring Remote Logging on a Server

The remote logging software included with Mac OS X Server is the syslog daemon `syslogd`. This service accepts and stores log messages from other systems on the network. If another system is compromised, its local logs can be altered, so the log server might contain the only accurate system records.

Only enable remote logging across a trusted internal network or VPN.

By default, Mac OS X Server performs only local logging and does not act as a log server.

Configuring Mac OS X Server to act as a remote log server involves changing `syslogd` command-line arguments. Enabling remote logging services requires removal of the `-s` tag from the `syslogd` tool, which allows any host to send traffic via UDP to the logging computer, which can present security risks.

To better control the hosts that are allowed to send logging message traffic, use the `-a` option to ensure that log messages from only specific IP addresses are accepted. You can use the `-a` option multiple times to specify additional hosts. Follow the `-a` option with an address in this format:

```
-a ipaddress/masklen[:service]
```

This format is the IPv4 address with a mask bit length. Optionally, the service can be a name or number of the UDP port the source packet must belong to.

When using the `-a` option, do not omit the `masklen` portion, because the default `masklen` might be very small and the corresponding matching addresses could be almost anything. The default [:`service`] is `syslog`, which should not need to be changed.

For example, match a subnet of 255 hosts as follows:

```
-a 192.168.1.0/24
```

or match a single host like this:

```
-a 192.168.1.23/32
```

You can specify host names or domain names instead of IP addresses, but this is not recommended.

To configure Mac OS X Server as a log server that accepts log messages from other systems on the network:

1. Open /etc/rc and locate the following line:

   ```
   /usr/sbin/syslogd -s -m 0
   ```

2. Replace the IP address after `-a` with your network information and change the line to:

   ```
   /usr/sbin/syslogd -n -a 192.168.1.0/24
   ```

   The `-n` option disables DNS lookups.

3. Insert this command as the next to last line of the file, before the `exit 0` line:

   ```
   killall -HUP syslogd #re-load configuration
   exit 0
   ```

`syslogd` contains features not documented on its man page. A more recent man page that fully describes its features is available at www.freebsd.org/cgi/man.cgi?query=syslogd.

PCI RAID Card Command Reference

Use this appendix to learn the megaraid commands to manage a PCI RAID card.

The `megaraid` tool uses are described in the following table, along with parameter explanations.

Parameter	Description
`megaraid -alarm -on | -off | -silence`	Turns the alarm on, off, or to silence. When the alarm is set to silence, it turns off for the current failure, but turns on again for the next failure.
`megaraid -changepolicy` *`ld`* `[-writecache` *`enable`* `|` *`disable`*`] [-readahead` *`on`* `|` *`off`* `|` *`adaptive`*`] [-iopolicy` *`direct`* `|` *`cached`*`] [-log` *`file`*`]`	Changes the policy of an existing logical drive. The parameter *`ld`* is the logical drive ID. This option applies to all RAID levels; however, the policies apply only to individual logical drives.
`megaraid -changestate` *`pd`* `-online | -fail [-log` *`file`*`]`	Changes the state of an existing physical drive to `online` or `fail`.
`megaraid -chkcon` *`ld`* `-start | -stop | -status [-log` *`file`*`]`	Starts, stops, or checks the status (percentage of progress) of a consistency check for a specific logical drive. The parameter *`ld`* is the logical drive ID.
`megaraid -create auto [-numld` *`n`*`] [-log` *`file`*`]`	Destroys configured logical drives and creates a RAID level based on the physical drive or drives present. It can create from 1 to 40 logical drives, depending on the number of logical drives (`numld` *`n`*) parameter. By default `numld` is `1`.

Parameter	Description
`megaraid -create R0 | R1 | R5 -drive {0 1 2 3} [-stripesize n] [-size x] [-writecache enable | disable] [-readahead on | off | adaptive] [-iopolicy direct | cached] [-log file]`	Creates a logical drive and adds it to the configuration. The RAID level and participating physical drives' parameters are required. Other parameters are optional. If `size` is not specified, the remaining size of the array is used. If `the stripesize` and `iopolicy` parameters are not specified, the default values are used. The `stripesize` parameter is in kilobytes, and valid stripe sizes are *16*, *32*, *64*, and *128* kilobytes. The `size` parameter is in megabytes. You cannot create a logical drive smaller than100 MB. After you create a logical drive, you can change the cache policy using the `changepolicy` command. Default values are as follows: • `stripesize: 64K` • `writecache: disabled` • `readcache: off` • `iopolicy: direct`
`megaraid -destroyconfig [-yes] [-log file]`	Clears the configuration. If you don't specify the `yes` parameter, the computer prompts for confirmation before clearing the configuration.
`megaraid -flash flashFileName [-log file]`	Flashes new firmware from the flash file to the adapter. The firmware becomes operational only after the computer restarts.
`megaraid -initialize ld -start | -stop | -status [-log file]`	Initializes, starts, stops, or displays the status (percentage of progress) of a specific logical drive. The parameter *ld* is the logical drive ID.
`megaraid -rebuild pd -start | -stop | -status [-log file]`	Rebuilds, starts, stops, or displays the status of a specific physical drive. The parameter *pd* is the physical drive ID.
`megaraid -showadapter [-log file]`	Displays information about the adapter, including product identification, battery status, number of logical drives created, cache size, and more.

Parameter	Description
`megaraid -showconfig [ld] [-log file]`	Displays the RAID configuration of the computer, including logical drive ID, RAID level, size, status, and participating physical drives. The logical drive status can be `failed`, `degraded`, or `optimal`. You cannot access a failed logical drive or recover data from it. You can access all data on a degraded logical drive (without a failure) even if attached physical drives are not in good condition. A degraded logical drive state does not apply to RAID 0, because RAID 0 is not a redundant array. A logical drive reported to be in the optimal state is in perfect condition.
`megaraid -showdevices [-log file]`	Displays drives connected to the PCI RAID card. The command displays drive ID, identification, size, status, and SMART alerts. The status of a drive is reported as `online`, `failed`, `ready`, `hotspare`, or `not responding`.
`megaraid -spare pd -create | -delete [-log file]`	Creates or deletes a global hot spare. You can create hot spares from a pool of ready drives. After deletion, a hot spare drive becomes a ready drive. The parameter *pd* is the physical drive ID.

Note: For more information, see the `megaraid` man page. You can also use `megaraid` commands with a `[-log file]` parameter, which logs displayed information with the date and time in the file you specify.

Glossary

This glossary defines terms and spells out abbreviations you may encounter while working with online help or the various reference manuals for Mac OS X Server. References to terms defined elsewhere in the glossary appear in italics.

ACE Access Control Entry. An entry within the ACL that controls access rights. See **ACL**.

ACL Access Control List. A list, maintained by a system, that defines the rights of users and groups to access resources on the system.

Active Directory The directory and authentication service of Microsoft Windows 2000 Server, Windows Server 2003, and Windows Server 2003 R2.

address A number or other identifier that uniquely identifies a computer on a network, a block of data stored on a disk, or a location in a computer's memory. See also **IP address**, **MAC address**.

administrator A user with server or directory domain administration privileges. Administrators are always members of the predefined "admin" group.

administrator computer A Mac OS X computer onto which you've installed the server administration applications from the Mac OS X Server Admin CD.

AFP Apple Filing Protocol. A client/server protocol used by Apple file service on Macintosh-compatible computers to share files and network services. AFP uses TCP/IP and other protocols to communicate between computers on a network.

Berkeley Internet Name Domain See **BIND**.

Berkeley Software Distribution See **BSD**.

BIND Berkeley Internet Name Domain. The program included with Mac OS X Server that implements DNS. The program is also called the name daemon, or named, when the program is running.

Bonjour A protocol developed by Apple for automatic discovery of computers, devices, and services on IP networks. Formerly called Rendezvous, this proposed Internet standard protocol is sometimes referred to as ZeroConf or multicast DNS.

boot ROM Low-level instructions used by a computer in the first stages of starting up.

BootP An older method of allocating IP addresses to clients on a network See also **DHCP**.

BSD Berkeley Software Distribution. A version of UNIX on which Mac OS X software is based.

canonical name The "real" name of a server when you've given it a "nickname" or alias. For example, mail.apple.com might have a canonical name of MailSrv473.apple.com.

certificate Sometimes called an "identity certificate" or "public key certificate." A file in a specific format (Mac OS X Server uses the X.509 format) that contains the public key half of a public-private keypair, the user's identity information such as name and contact information, and the digital signature of either a **Certificate Authority** (CA) or the key user.

Certificate Authority An authority that issues and manages digital certificates in order to ensure secure transmission of data on a public network. See also **certificate.**

CGI Common Gateway Interface. A script or program that adds dynamic functions to a website. A CGI sends information back and forth between a website and an application that provides a service for the site.

child A computer that gets configuration information from the shared directory domain of a parent.

client A computer (or a user of the computer) that requests data or services from another computer, or server.

Common UNIX Printing System See **CUPS**.

computer account A computer account stores data that allows Mac OS X Server to identify and manage an individual computer. You create a computer account for each computer that you intend to add to a computer group. See also **computer group**.

computer group A set of computers and computer groups, which all receive the managed preference settings defined for the group. New in Mac OS X Server version 10.5. See also **computer list**.

computer list A set of computers that all receive the managed preference settings defined for the list, and that are all available to a particular set of users and groups. A computer can be a member of only one computer list. Computer lists are created in Mac OS X Server version 10.4 or earlier. See also **computer group**.

computer name The default name used for SLP and SMB service registrations. The Network Browser in the Finder uses SLP to find computers advertising Personal File Sharing and Windows File Sharing. It can be set to bridge subnets depending on the network router settings. When you turn on Personal File Sharing, users see the computer name in the Connect to Server dialog in the Finder. Initially it is "<first created user>'s Computer" (for example, "John's Computer") but can be changed to anything. The computer name is used for browsing for network file servers, print queues, Bluetooth® discovery, Apple Remote Desktop clients, and any other network resource that identifies computers by computer name rather than network address. The computer name is also the basis for the default local host name.

CUPS Common UNIX Printing System. A cross-platform printing facility based on the Internet Printing Protocol (IPP). The Mac OS X Print Center, its underlying print system, and the Mac OS X Server print service are based on CUPS. For more information, visit www.cups.org.

DHCP Dynamic Host Configuration Protocol. A protocol used to dynamically distribute IP addresses to client computers. Each time a client computer starts up, the protocol looks for a DHCP server and then requests an IP address from the DHCP server it finds. The DHCP server checks for an available IP address and sends it to the client computer along with a lease period—the length of time the client computer may use the address.

directory domain A specialized database that stores authoritative information about users and network resources; the information is needed by system software and applications. The database is optimized to handle many requests for information and to find and retrieve information quickly. Also called a directory node or simply a directory.

directory domain hierarchy A way of organizing local and shared directory domains. A hierarchy has an inverted tree structure, with a root domain at the top and local domains at the bottom.

directory node See **directory domain**.

directory services Services that provide system software and applications with uniform access to directory domains and other sources of information about users and resources.

disk image A file that, when opened, creates an icon on a Mac OS X desktop that looks and acts like an actual disk or volume. Using NetBoot, client computers can start up over the network from a server-based disk image that contains system software. Disk image files have a filename extension of either .img or .dmg. The two image formats are similar and are represented with the same icon in the Finder. The .dmg format cannot be used on computers running Mac OS 9.

DNS Domain Name System. A distributed database that maps IP addresses to domain names. A DNS server, also known as a name server, keeps a list of names and the IP addresses associated with each name.

DNS domain A unique name of a computer used in the Domain Name System to translate IP addresses and names. Also called a **domain name**.

DNS name A unique name of a computer used in the Domain Name System to translate IP addresses and names. Also called a **domain name**.

domain Part of the domain name of a computer on the Internet. It does not include the top-level domain designator (for example, .com, .net, .us, .uk). Domain name "www.example.com" consists of the subdomain or host name "www," the domain "example," and the top-level domain "com."

domain name See **DNS name**.

Domain Name System See **DNS**.

dynamic IP address An IP address that's assigned for a limited period of time or until the client computer no longer needs it.

Ethernet A common local area networking technology in which data is transmitted in units called packets using protocols such as TCP/IP.

everyone Any user who can log in to a file server: a registered user or guest, an anonymous FTP user, or a website visitor.

filter A "screening" method used to control access to a server. A filter is made up of an IP address and a subnet mask, and sometimes a port number and access type. The IP address and the subnet mask together determine the range of IP addresses to which the filter applies.

file system A scheme for storing data on storage devices that allows applications to read and write files without having to deal with lower-level details.

firewall Software that protects the network applications running on your server. IP firewall service, which is part of Mac OS X Server software, scans incoming IP packets and rejects or accepts these packets based on a set of filters you create.

FireWire A hardware technology for exchanging data with peripheral devices, defined by IEEE Standard 1394.

folder Also known as a directory. A hierarchically organized list of files and/or other folders.

FTP File Transfer Protocol. A protocol that allows computers to transfer files over a network. FTP clients using any operating system that supports FTP can connect to a file server and download files, depending on their access privileges. Most Internet browsers and a number of freeware applications can be used to access an FTP server.

group A collection of users who have similar needs. Groups simplify the administration of shared resources.

group folder A folder that organizes documents and applications of special interest to group members and allows group members to pass information among themselves.

guest computer A computer that doesn't have a computer account.

guest user A user who can log in to your server without a user name or password.

GUID Globally unique identifier. A hexadecimal string that uniquely identifies a user account, group account, or computer list. Also used to provide user and group identity for access control list (ACL) permissions, and to associate particular users with group and nested group memberships. GUIDs are 128-bit values, which makes the generation of duplicate GUIDs extremely unlikely.

home directory See **home folder**.

home folder A folder for a user's personal use. Mac OS X also uses the home folder to store system preferences and managed user settings for Mac OS X users. Also known as a home directory.

HTML Hypertext Markup Language. The set of symbols or codes inserted in a file to be displayed on a web browser page. The markup tells the web browser how to display a webpage's words and images for the user.

HTTP Hypertext Transfer Protocol. The client/server protocol for the World Wide Web. HTTP provides a way for a web browser to access a web server and request hypermedia documents created using HTML.

Hypertext Markup Language See **HTML**.

Hypertext Transfer Protocol See **HTTP**.

iChat The Mac OS X instant messaging application.

iChat service The Mac OS X Server service that hosts secure chats. iChat service uses Open Directory authentication to verify the identity of chatters and SSL to protect the privacy of users while they chat.

ICMP Internet Control Message Protocol. A message control and error-reporting protocol used between host servers and gateways. For example, some Internet software applications use ICMP to send a packet on a round trip between two hosts to determine round-trip times and discover problems on the network.

image See **disk image**.

IMAP Internet Message Access Protocol. A client-server mail protocol that allows users to store their mail on the mail server rather than downloading it to the local computer. Mail remains on the server until the user deletes it.

IP Internet Protocol. Also known as IPv4. A method used with Transmission Control Protocol (TCP) to send data between computers over a local network or the Internet. IP delivers data packets and TCP keeps track of data packets.

IP address A unique numeric address that identifies a computer on the Internet.

IP subnet A portion of an IP network, which may be a physically independent network segment, that shares a network address with other portions of the network and is identified by a subnet number.

IPP Internet Printing Protocol. A client-server protocol for printing over the Internet. The Mac OS X printing infrastructure and the Mac OS X Server print service that's built on it support IPP.

IPSec A security addition to IP. A protocol that provides data transmission security for L2TP VPN connections. IPSec acts at the network layer, protecting and authenticating IP packets between participating IPSec nodes.

IPv4 See **IP**.

IPv6 Internet Protocol version 6. The next-generation communication protocol to replace IP (also known as IPv4). IPv6 allows a greater number of network addresses and can reduce routing loads across the Internet.

ISO International Standards Organization. The international standards body. ISO-published standards have the status of international treaties.

ISP Internet service provider. A business that sells Internet access and often provides web hosting for e-commerce applications as well as mail services.

KB Kilobyte. 1,024 (2^{10}) bytes.

KDC Kerberos Key Distribution Center. A trusted server that issues Kerberos tickets.

Kerberos A secure network authentication system. Kerberos uses tickets, which are issued for a specific user, service, and period of time. After a user is authenticated, it's possible to access additional services without retyping a password (called single sign-on) for services that have been configured to take Kerberos tickets. Mac OS X Server uses Kerberos v5.

Kerberos Key Distribution Center See **KDC**.

Kerberos realm The authentication domain comprising the users and services that are registered with the same Kerberos server. The registered users and services trust the Kerberos server to verify each other's identities.

kernel The part of an operating system that handles memory management, resource allocation, and other low-level services essential to the system.

LDAP Lightweight Directory Access Protocol. A standard client-server protocol for accessing a directory domain.

lease period A limited period of time during which IP addresses are assigned. By using short leases, DHCP can reassign IP addresses on networks that have more computers than available IP addresses.

Lightweight Directory Access Protocol See **LDAP**.

Line Printer Remote See **LPR**.

local directory domain A directory of identification, authentication, authorization, and other administrative data that's accessible only on the computer where it resides. The local directory domain isn't accessible from other computers on the network.

local domain A directory domain that can be accessed only by the computer it resides on.

local home directory See **local home folder**.

local home folder A home folder that resides on disk on the computer a user is logged in to. It's accessible only by logging directly in to the computer where it resides, unless you log in to the computer using SSH.

local hostname A name that designates a computer on a local subnet. It can be used without a global DNS system to resolve names to IP addresses. It consists of lowercase letters, numbers, or hyphens (except as the last characters), and ends with ".local" (For example, bills-computer.local). Although the default name is derived from the computer name, a user can specify this name in the Sharing pane of System Preferences. It can be changed easily, and can be used anywhere a DNS name or fully qualified domain name is used. It can only resolve on the same subnet as the computer using it.

log in (verb) To start a session with a computer (often by authenticating as a user with an account on the computer) in order to obtain services or access files. Note that logging in is separate from connecting, which merely entails establishing a physical link with the computer.

logical disk A storage device that appears to a user as a single disk for storing files, even though it might actually consist of more than one physical disk drive. An Xsan volume, for example, is a logical disk that behaves like a single disk even though it consists of multiple storage pools that are, in turn, made up of multiple LUNs, each of which contains multiple disk drives. See also **physical disk**.

LPR Line Printer Remote. A standard protocol for printing over TCP/IP.

MAC address Media access control address. A hardware address that uniquely identifies each node on a network. For AirPort devices, the MAC address is called the AirPort ID.

mail host The computer that provides your mail service.

managed client A user, group, or computer whose access privileges and/or preferences are under administrative control.

managed network The items managed clients are allowed to see when they click the Network icon in a Finder window. Administrators control this setting using Workgroup Manager. Also called a network view.

managed preferences System or application preferences that are under administrative control. Workgroup Manager allows administrators to control settings for certain system preferences for Mac OS X managed clients.

MIME Multipurpose Internet Mail Extensions. An Internet standard for specifying how a web browser handles a file with certain characteristics. A file's suffix describes its type. You determine how the server responds when it receives files with certain suffixes. Each suffix and its associated response make up a MIME type mapping.

MX record Mail exchange record. An entry in a DNS table that specifies which computer manages mail for an Internet domain. When a mail server has mail to deliver to an Internet domain, the mail server requests the MX record for the domain. The server sends the mail to the computer specified in the MX record.

MySQL An open source relational database management tool frequently used by web servers.

name server A server on a network that keeps a list of names and the IP addresses associated with each name. See also **DNS**, **WINS**.

NetBIOS Network Basic Input/Output System. A program that allows applications on different computers to communicate within a local area network.

NetBoot server A Mac OS X server on which you've installed NetBoot software and have configured to allow clients to start up from disk images on the server.

Network File System See **NFS**.

NFS Network File System. A client/server protocol that uses Internet Protocol (IP) to allow remote users to access files as though they were local. NFS exports shared volumes to computers based on IP address, rather than user name and password.

nfsd daemon An NFS server process that runs continuously behind the scenes and processes read and write requests from clients. The more daemons that are available, the more concurrent clients can be served.

Open Directory master A server that provides LDAP directory service, Kerberos authentication service, and Open Directory Password Server.

Open Directory password A password that's stored in secure databases on the server and can be authenticated using Open Directory Password Server or Kerberos (if Kerberos is available).

Open Directory Password Server An authentication service that validates passwords using a variety of conventional authentication methods required by the different services of Mac OS X Server. The authentication methods include APOP, CRAM-MD5, DHX, LAN Manager, NTLMv1, NTLMv2, and WebDAV-Digest.

open relay A server that receives and automatically forwards mail to another server. Junk mail senders exploit open relay servers to avoid having their own mail servers blacklisted as sources of junk mail.

Open Relay Behavior-modification System See **ORBS**.

ORBS Open Relay Behavior-modification System. An Internet service that blacklists mail servers known to be or suspected of being open relays for senders of junk mail. ORBS servers are also known as black hole servers.

owner The owner of an item can change access permissions to the item. The owner may also change the group entry to any group the owner is a member of. By default, the owner has Read & Write permissions.

parent A computer whose shared directory domain provides configuration information to another computer.

password An alphanumeric string used to authenticate the identity of a user or to authorize access to files or services.

password policy A set of rules that regulate the composition and validity of a user's password.

Password Server See **Open Directory Password Server**.

pathname The location of an item within a file system, represented as a series of names separated by slashes (/).

PDC Primary domain controller. In Windows networking, a domain controller that has been designated as the primary authentication server for its domain.

permissions Settings that define the kind of access users have to shared items in a file system. You can assign four types of permissions to a share point, folder, or file: Read & Write, Read Only, Write Only, and No Access. See also **privileges**.

PHP PHP Hypertext Preprocessor (originally Personal Home Page). A scripting language embedded in HTML that's used to create dynamic webpages.

physical disk An actual, mechanical disk. Compare with **logical disk**.

PID Process ID. A number assigned to a UNIX process when it starts. The PID allows you to refer to the process at a later time.

POP Post Office Protocol. A protocol for retrieving incoming mail. After a user retrieves POP mail, it's stored on the user's computer and is usually deleted automatically from the mail server.

port A sort of virtual mail slot. A server uses port numbers to determine which application should receive data packets. Firewalls use port numbers to determine whether data packets are allowed to traverse a local network. "Port" usually refers to either a TCP or UDP port.

port name A unique identifier assigned to a Fibre Channel port.

portable home directory A portable home directory provides a user with both a local and network home folder. The contents of these two home folders, as well as the user's directory and authentication information, can be automatically kept in sync.

POSIX Portable Operating System Interface for UNIX. A family of open system standards based on UNIX, which allows applications to be written to a single target environment in which they can run unchanged on a variety of systems.

Post Office Protocol See **POP**.

Postscript Printer Description file See **PPD file**.

PPD file Postscript Printer Description file. A file that contains information about the capabilities of a particular printer model. The PPD file provides the controls you need to take advantage of special features such as multiple paper trays, special paper sizes, or duplex printing. The printer model you choose when you add a printer specifies the PPD file used with the printer.

predefined accounts User accounts that are created automatically when you install Mac OS X. Some group accounts are also predefined.

preferences cache A storage place for computer preferences and preferences for groups associated with that computer. Cached preferences help you manage local user accounts on portable computers.

presets Default attributes you specify for accounts you create using Workgroup Manager. You can use presets only during account creation.

primary domain controller See **PDC**.

primary group A user's default group. The file system uses the ID of the primary group when a user accesses a file he or she doesn't own.

primary group ID A unique number that identifies a primary group.

print queue An orderly waiting area where print jobs wait until a printer is available. The print service in Mac OS X Server uses print queues on the server to facilitate management.

privileges The right to access restricted areas of a system or perform certain tasks (such as management tasks) in the system.

process A program that has started executing and has a portion of memory allocated to it.

process ID See **PID**.

protocol A set of rules that determines how data is sent back and forth between two applications.

proxy server A server that sits between a client application, such as a web browser, and a real server. The proxy server intercepts all requests to the real server to see if it can fulfill the requests itself. If not, it forwards the request to the real server.

public key One of two asymmetric keys used in a PKI security system. The public key is distributed to other communicating parties. It can encrypt messages that can be decrypted only by the holder of the corresponding private key, and it can verify the signature on a message originating from a corresponding private key.

public key certificate See **certificate**.

QTSS QuickTime Streaming Server. A technology that lets you deliver media over the Internet in real time.

queue An orderly waiting area where items wait for some type of attention from the system. See also **print queue**.

QuickTime Streaming Server See **QTSS**.

RADIUS Remote Authentication Dial-In User Service.

RADIUS server A computer on the network that provides a centralized database of authentication information for computers on the network.

RAID Redundant Array of Independent (or Inexpensive) Disks. A grouping of multiple physical hard disks into a disk array, which either provides high-speed access to stored data, mirrors the data so that it can be rebuilt in case of disk failure, or both. The RAID array is presented to the storage system as a single logical storage unit. See also **RAID array**, **RAID level**.

RAID 0 A RAID scheme in which data is distributed evenly in stripes across an array of drives. RAID 0 increases the speed of data transfer, but provides no data protection.

RAID 0+1 A combination of RAID 0 and RAID 1. This RAID scheme is created by striping data across multiple pairs of mirrored drives.

RAID 1 A RAID scheme that creates a pair of mirrored drives with identical copies of the same data. It provides a high level of data availability.

RAID 10 A hybrid RAID level that uses software RAID striping to stripe data across RAID 1 (or mirrored) arrays.

RAID 3 A RAID scheme that stripes data across two or more drives and stores parity data on a dedicated drive. In the event of a disk failure, the redundant parity bits can be used to reconstruct data on any drive.

RAID 30 A hybrid RAID level that uses software RAID striping to stripe data across RAID 3 arrays.

RAID 5 A RAID scheme that distributes both data and parity information across an array of drives one block at a time, with each drive operating independently. This enables maximum read performance when accessing large files.

RAID 50 A hybrid RAID level that uses software RAID striping across RAID 5 arrays.

RAID array A group of physical disks organized and protected by a RAID scheme and presented by RAID hardware or software as a single logical disk. In Xsan, RAID arrays appear as LUNs, which are combined to form storage pools.

RAID level A storage allocation scheme used for storing data on a RAID array. Specified by a number, as in RAID 3 or RAID 0+1.

RAID set See **RAID array**.

Real Time Streaming Protocol See **RTSP**.

Real-Time Transport Protocol See **RTP**.

realm General term with multiple applications. See **WebDAV realm**, **Kerberos realm**.

relay In QuickTime Streaming Server, a relay receives an incoming stream and then forwards that stream to one or more streaming servers. Relays can reduce Internet bandwidth consumption and are useful for broadcasts with numerous viewers in different locations. In Internet mail terms, a relay is a mail SMTP server that sends incoming mail to another SMTP server, but not to its final destination.

relay point See **open relay**.

root An account on a system that has no protections or restrictions. System administrators use this account to make changes to the system's configuration.

RTP Real-Time Transport Protocol. An end-to-end network-transport protocol suitable for applications transmitting real-time data (such as audio, video, or simulation data) over multicast or unicast network services.

RTSP Real Time Streaming Protocol. An application-level protocol for controlling the delivery of data with real-time properties. RTSP provides an extensible framework to enable controlled, on-demand delivery of real-time data, such as audio and video. Sources of data can include both live data feeds and stored clips.

SACL Service Access Control List. Lets you specify which users and groups have access to specific services. See **ACL**.

scope A group of services. A scope can be a logical grouping of computers, such as all computers used by the production department, or a physical grouping, such as all computers located on the first floor. You can define a scope as part or all of your network.

SDP Session Description Protocol. A text file used with QuickTime Streaming Server that provides information about the format, timing, and authorship of a live streaming broadcast and gives the user's computer instructions for tuning in.

search path See **search policy**.

search policy A list of directory domains searched by a Mac OS X computer when it needs configuration information; also, the order in which domains are searched. Sometimes called a search path.

session The period of time during which two programs, or two users running programs, communicate across a network. For example, when a user logs in to a file server, a session is initiated that continues until the user logs out or the session is terminated by the file service.

Session Description Protocol See **SDP**.

shadow image A file created by the NetBoot daemon process for each NetBoot client where applications running on the client can write temporary data.

shadow password A password that's stored in a secure file on the server and can be authenticated using a variety of conventional authentication methods required by the different services of Mac OS X Server. The authentication methods include APOP, CRAM-MD5, DHX, LAN Manager, NTLMv1, NTLMv2, and WebDAV-Digest.

share See **share point**.

share point A folder, hard disk (or hard disk partition), or optical disc that's accessible over the network. A share point is the point of access at the top level of a group of shared items. Share points can be shared using AFP, SMB, NFS (an export), or FTP.

shared secret A value defined at each node of an L2TP VPN connection that serves as the encryption key seed to negotiate authentication and data transport connections.

short name An abbreviated name for a user. The short name is used by Mac OS X for home folders, authentication, and email addresses.

Simple Mail Transfer Protocol See **SMTP**.

Simple Network Management Protocol See **SNMP**.

SLP DA Service Location Protocol Directory Agent. A protocol that registers services available on a network and gives users easy access to them. When a service is added to the network, the service uses SLP to register itself on the network. SLP DA uses a centralized repository for registered network services.

SMB Server Message Block. A protocol that allows client computers to access files and network services. It can be used over TCP/IP, the Internet, and other network protocols. SMB services use SMB to provide access to servers, printers, and other network resources.

SMTP Simple Mail Transfer Protocol. A protocol used to send and transfer mail. Its ability to queue incoming messages is limited, so SMTP is usually used only to send mail, and POP or IMAP is used to receive mail.

SNMP Simple Network Management Protocol. A set of standard protocols used to manage and monitor multiplatform computer network devices.

spam Unsolicited email; junk mail.

SSL Secure Sockets Layer. An Internet protocol that allows you to send encrypted, authenticated information across the Internet. More recent versions of SSL are known as TLS (Transport Level Security).

static IP address An IP address that's assigned to a computer or device once and is never changed.

subnet A grouping on the same network of client computers that are organized by location (for example, different floors of a building) or by usage (for example, all eighth-grade students). The use of subnets simplifies administration. See also **IP subnet**.

subnet mask A number used in IP networking to specify which portion of an IP address is the network number.

systemless client A computer that doesn't have an operating system installed on its local hard disk. Systemless computers can start up from a disk image on a NetBoot server.

TCP Transmission Control Protocol. A method used with the Internet Protocol (IP) to send data in the form of message units between computers over the Internet. IP handles the actual delivery of the data, and TCP keeps track of the units of data (called packets) into which a message is divided for efficient routing through the Internet.

Transmission Control Protocol See **TCP**.

TTL Time-to-live. The specified length of time that DNS information is stored in a cache. When a domain name–IP address pair has been cached longer than the TTL value, the entry is deleted from the name server's cache (but not from the primary DNS server).

UDP User Datagram Protocol. A communications method that uses the Internet Protocol (IP) to send a data unit (called a datagram) from one computer to another on a network. Network applications that have very small data units to exchange may use UDP rather than TCP.

UID User ID. A number that uniquely identifies a user within a file system. Mac OS X computers use the UID to keep track of a user's folder and file ownership.

Unicode A standard that assigns a unique number to every character, regardless of language or the operating system used to display the language.

Uniform Resource Locator See **URL**.

URL Uniform Resource Locator. The address of a computer, file, or resource that can be accessed on a local network or the Internet. The URL is made up of the name of the protocol needed to access the resource, a domain name that identifies a specific computer on the Internet, and a hierarchical description of a file location on the computer.

user ID See **UID**.

user name The long name for a user, sometimes referred to as the user's real name. See also **short name**.

user profile The set of personal desktop and preference settings that Windows saves for a user and applies each time the user logs in.

virtual user An alternate email address (short name) for a user. Similar to an alias, but it involves creating another user account.

volume A mountable allocation of storage that behaves, from the client's perspective, like a local hard disk, hard disk partition, or network volume. In Xsan, a volume consists of one or more storage pools. See also **logical disk**.

VPN Virtual Private Network. A network that uses encryption and other technologies to provide secure communications over a public network, typically the Internet. VPNs are generally cheaper than real private networks using private lines, but they rely on having the same encryption system at both ends. The encryption may be performed by firewall software or by routers.

WebDAV Web-based Distributed Authoring and Versioning. A live authoring environment that allows client users to check out webpages, make changes, and then check the pages back in to the site while the site is running.

WebDAV realm A region of a website, usually a folder or directory, that's defined to provide access for WebDAV users and groups.

wildcard A range of possible values for any segment of an IP address.

Windows domain The Windows computers on a network that share a common directory of user, group, and computer accounts for authentication and authorization. An Open Directory master can provide directory services for a Windows domain.

Windows Internet Naming Service See **WINS**.

WINS Windows Internet Naming Service. A name resolution service used by Windows computers to match client names with IP addresses. A WINS server can be located on the local network or externally on the Internet.

workgroup A set of users for whom you define preferences and privileges as a group. Any preferences you define for a group are stored in the group account.

Index

D

N

O

P

Q

R

S

T

U

V

W

X