Mac OS X Server

Command-Line Administration
For Version 10.4 or Later
Second Edition

Every effort has been made to ensure that the information in this manual is accurate. Apple Computer, Inc., is not responsible for printing or clerical errors.

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

019-0635/2-15-2006

Contents

About This Guide

This guide describes Mac OS X Servers command-line interface tools and commands, including the syntax, purpose, and parameters, as well as examples of usage and any output that they generate.

This guide 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 commonly 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 4.4BSD commands. While some commands are modified to function differently, most of the commands are either 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 funtions 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 frequently 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 (enter one or the other)
`italicized`	A parameter you must replace with a value
`[...]`	A parameter that may be repeated
`<angle brackets>`	A displayed value that depends on your server configuration

Commands and Other Terminal Text

Commands or command parameters that you might enter, along with other text that normally 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 the Return key. (Terminal is found in /Applications/Utilities).

Command Parameters and Options

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

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 either `doit` or `doit -w later`. The result might vary, but the command will be performed 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 either `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.

Getting Documentation Updates

Periodically, Apple posts revised guides and solution papers. To download the latest guides and solution papers in PDF format, go to the Mac OS X Server documentation webpage: 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.

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.

Mac OS X Server website (www.apple.com/macosx/server/)—Gateway to extensive product and technology information.

AppleCare Service & Support website (www.apple.com/support/)—Access to hundreds of articles from Apple's support organization.

Apple customer training (train.apple.com)—Instructor-led and self-paced courses for honing your server administration skills.

Apple discussion groups (discussions.info.apple.com)—A way to share questions, knowledge, and advice with other administrators.

Apple mailing list folder (www.lists.apple.com)—Subscribe to mailing lists so you can communicate with other administrators using email.

The public source website (developer.apple.com/darwin/)—Access to Darwin source code, developer information, and FAQs.

Mac OS X Server suite documentation (www.apple.com/server/documentation/)—The Mac OS X Server documentation includes a suite of guides that explain the available services and provide instructions for configuring, managing, and troubleshooting those services.

This guide ...	tells you how to:
Mac OS X Server Getting Started for Version 10.4 or Later	Install Mac OS X Server and set it up for the first time.
Mac OS X Server Upgrading and Migrating to Version 10.4 or Later	Use data and service settings that are currently being used on earlier versions of the server.
Mac OS X Server User Management for Version 10.4 or Later	Create and manage users, groups, and computer lists. Set up managed preferences for Mac OS X clients.

This guide …	tells you how to:
Mac OS X Server File Services Administration for Version 10.4 or Later	Share selected server volumes or folders among server clients using these protocols: AFP, NFS, FTP, and SMB/CIFS.
Mac OS X Server Print Service Administration for Version 10.4 or Later	Host shared printers and manage their associated queues and print jobs.
Mac OS X Server System Imaging and Software Update Administration for Version 10.4 or Later	Use NetBoot and Network Install to create disk images from which Macintosh computers can start up over the network. Set up a software update server for updating client computers over the network.
Mac OS X Server Mail Service Administration for Version 10.4 or Later	Set up, configure, and administer mail services on the server.
Mac OS X Server Web Technologies Administration for Version 10.4 or Later	Set up and manage a web server, including WebDAV, WebMail, and web modules.
Mac OS X Server Network Services Administration for Version 10.4 or Later	Set up, configure, and administer DHCP, DNS, VPN, NTP, IP firewall, and NAT services on the server.
Mac OS X Server Open Directory Administration for Version 10.4 or Later	Manage directory and authentication services.
Mac OS X Server QuickTime Streaming Server Administration for Version 10.4 or Later	Set up and manage QuickTime streaming services.
Mac OS X Server Windows Services Administration for Version 10.4 or Later	Set up and manage services including PDC, BDC, file, and print for Windows computer users.
Mac OS X Server Migrating from Windows NT for Version 10.4 or Later	Move accounts, shared folders, and services from Windows NT servers to Mac OS X Server.
Mac OS X Server Java Application Server Administration For Version 10.4 or Later	Configure and administer a JBoss application server on Mac OS X Server.
Mac OS X Server Command-Line Administration for Version 10.4 or Later	Use commands and configuration files to perform server administration tasks in a UNIX command shell.
Mac OS X Server Collaboration Services Administration for Version 10.4 or Later	Set up and manage weblog, chat, and other services that facilitate interactions among users.
Mac OS X Server High Availability Administration for Version 10.4 or Later	Manage IP failover, link aggregation, load balancing, and other hardware and software configurations to ensure high availability of Mac OS X Server services.

This guide …	tells you how to:
Mac OS X Server Xgrid Administration for Version 10.4 or Later	Manage computational Xserve clusters using the Xgrid application.
Mac OS X Server Glossary: Includes Terminology for Mac OS X Server, Xserve, Xserve RAID, and Xsan	Interpret terms used for server and storage products.

Executing Commands

1

In this chapter you will find out how to execute commands and 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 a complete execution context, called a shell, that is separate from all other execution contexts. The shell itself 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 of your choice, the examples in this book assume that you are using `bash`, the standard Mac OS X shell.

Opening Terminal

To enter shell commands or run server command-line tools, you need access to a UNIX shell prompt. Both Mac OS X and Mac OS X Server include Terminal, an application you can use to start a UNIX shell command-line session 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 (located 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 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 and the prompt displays 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, wherever a command is shown as you might enter it, 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.

Some of the standard shortcuts used to represent specific folders in the computer 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 represents 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 currently logged in. In Mac OS X, this folder resides either 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 only letters, numbers, a period, or the underscore character. Most other characters, including space characters, should be avoided. Although some Mac OS X file systems permit the use of these other characters, including spaces, you may have to add single or double quotation marks around any 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 would become either "My Disk" or My\ Disk.

Modifying Flow Control

Many commands are capable of receiving text input from the user and printing text out to the console. They do so using *standard pipes*, which are created by the shell and passed to the command automatically.

The standard pipes include:

- `stdin`—The standard input pipe is the means through which data enters a command. By default, this is data entered by the user 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 back to the command line. You can also redirect the output from the command 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 may 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 having to enter that data.

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 any lines containing the word "commands." The result is a listing of only those lines with the specified text, instead of the entire man page.

See the `bash` man page for more information about redirection.

Using Environment Variables

Some commands require the use of environment variables for their execution. Environment variables are variables inherited by all commands executed in the shell's context. The shell itself 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 also create environment variables and use them to control the behavior of your command without modifying the command itself. For example, you might use an environment variable to tell your command to print debug information to the console.

To set the value of an environment variable, you 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 will modify the environment variable `PATH` with the value assigned. To view all of the environment variables, enter the following:

```
$ env
```

When you launch an application from a shell, the application inherits much of the shell's environment, including any exported environment variables. This form of inheritance can be a useful way to configure the application dynamically. For example, your application can check 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. Once you close a Terminal window, any 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, you must enter the complete pathname of the tool's executable file, followed by any arguments, and then press the Return key. If a command is located in one of the shell's known folders, you can omit any path information and just enter the command name. The list of known folders is stored in the shell's PATH environment variable and includes the folders containing most of the command-line tools.

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

```
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 either use the open command (open MyProg.app) or launch the tool by typing 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.

```
server:/ anne$ serversetup -getAllPort
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$ /System/Library/ServerSetup/serversetup -getAllPort
1
Built-in Ethernet
```

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$ ./serversetup -getAllPort
1
Built-in Ethernet
```

or

```
server:/System/Library/ServerSetup anne$ cd /
server:/ anne$ PATH="$PATH:/System/Library/ServerSetup"
server:/ anne$ serversetup -getAllPort
1
Built-in Ethernet
```

Correcting Typing Errors

To correct a typing error before you press Return to execute the command, press Left Arrow or Right Arrow to skip over parts of the command you don't want to change, press the Delete key to remove characters, enter regular characters to insert them, and finally press Return to execute the command.

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

Repeating Commands

To repeat a command, press Up Arrow until you see the command, make any modifications, and then 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 Within a File

To locate a unique 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 sunshine filename
```

where *filename* is the name of the file you wish to search through and *sunshine* is the unique string.

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.

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

```
$ sudo serveradmin list
```

You're prompted for the root password if you haven't used `sudo` recently. The root user password is set to the administrator user password when you install Mac OS X Server.

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

```
$su root
```

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

Important: As the root user, you have sufficient privileges to do things that can cause your server to stop working properly. Don't execute commands as the root user unless you know what you're doing. Logging in as an administrator user and using `sudo` selectively might prevent you from making unintended changes.

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 may install signal handlers to trap this signal and respond differently.

Scheduling Tasks

You can create scheduled tasks using the `cron` tool. `cron` is a daemon that executes scheduled commands from a crontab file. 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. `cron` also searches for crontab files in the /etc/crontab folder, which are in a different format. `cron` then cycles every minute, examining all stored crontab files and checking each command to see if it should be run in the current minute.

When commands execute, any 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 such exists. When a crontab file has been modified, cron needs to be restarted. crontab is the program used to install, deinstall, or list the tables used to drive the cron daemon. Each user 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      /usr/local/vscanx folder-name
50   23   *    *     0        /usr/local/vscanx --summary folder-name
15   10   *    *     6        /usr/local/vscanx --load /usr/local/conf1 /uz
45   8    *    *     1        /usr/local/vscanx --f /usr/local/biglist
```

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

The following crontab entry schedules a scan operation to run and produce a summary at 18:30 every day, Monday through Friday:

```
30 18 * * 1-5 /usr/local/vscanx folder-name
```

The following crontab entry schedules a scan operation to run and produce a summary at 23:50 every Sunday:

```
50 23 * * 0 /usr/local/vscanx --summary folder-name
```

The following crontab entry schedules a scan operation to run on the uz folder at 10:15 a.m. every Saturday in accordance with options specified in a configuration file conf1:

```
15 10 * * 6 /usr/local/vscanx --load /usr/local/conf1 /uz
```

The following crontab entry schedules a scan operation to run at 8:45 a.m. every Monday on the files specified in the file biglist:

```
45 8 * * 1 /usr/local/vscanx --f /usr/local/biglist
```

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.

See Chapter 2, "Connecting to Remote Computers," on page 31 for information about sending commands to remote computers.

Viewing Command Information

Most command-line documentation comes in the form of man pages. These are formatted pages that 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 any 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 the man pages are so long that they do not fit on your screen, you can use the `more` or `less` command to automatically paginate the file. This allows you to view the file faster by loading full 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` will return you to the prompt and `less` will wait 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 seraching 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 pop-up list of options and parameters you can use with the command, enter the command without any 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

In this chapter you will find commands you can use to connect to remote computers.

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

Understanding Secure Shell

Secure Shell (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:

- The local and remote computers exchange their public keys. If the local computer has never encountered a given public key before, both SSH and a web browser will prompt you whether to accept the unknown key.
- The two computers use the public keys to negotiate a session key that is used to encrypt all subsequent session data.
- 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 username/password combination. See "Password-Less Logins Using SSH Keys" on page 32 for information about setting up certificate authentication.
- After successful authentication, the session begins. Either a remote shell, a secure file transfer, a remote command, or so on, is begun through the encrypted tunnel.

You should be aware of the following SSH tools:

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

Password-Less Logins Using SSH Keys

The standard method of SSH authentication is supplying login credentials in the form of a user name and password. Identity key pair authentication enables you to log in to the server without having to supply a password. This process works by:

- Generating a private and public key associated with a user name to establish that user's authenticity. When you attempt to log in as that user, the user name is sent to the remote computer.
- 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.
- A challenge is then sent to the user based on his or her public key.
- The user verifies his or her identity by using the private portion of the key pair to decode the challenge.
- Once decoded, the user is logged in without the need for a password. This is especially useful when automating remote scripts.

To generate the identity key pair, use the following command on the local computer:

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

When prompted, enter a filename in which to save the keys in the user's folder. 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, is also displayed. This secures the public key, making it computationally infeasible for duplication.

Copy the resultant public file, which contains the local computer's public key to the user's home folder in .ssh/ on the remote computer. 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 already logged in using the account, you do not have to supply a pasword 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). See the Open Directory administration guide for more information.

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 compared against in later sessions. Be sure this is the correct key before accepting it. If at all possible, provide your users with the encryption key either through FTP, email, 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 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 either 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.
- Are attempting to SSH in to a computer that has the same IP address as a computer that you previously used SSH with on another network.

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

What is an SSH Man-in-the-Middle Attack?

An attacker may be able to get access to your network and compromise proper routing information, such that packets intended for a remote computer are instead routed to the attacker who 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 the 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 the information that goes back and forth, and can modify it.

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

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

Protect against 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. Be sure you understand why the key on the remote computer has changed before you delete its entry from the known_hosts file.

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 are always allowed to open a connection using SSH. The `ssh` tool uses the SSH service. For information about controlling access to the SSH service, see the Open Directory administration guide.

Connecting to a Remote Computer

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

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 Enter the following command to log in to the remote computer, and then press Return:

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

where *username* is the name of an administrator user on the remote computer and *server* is the name or IP address of the remote computer. For example:

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

3 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 is displayed. Enter `yes` and press Return.

4 When prompted, enter the user's password (the user's password on the remote computer) and press Return.

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

```
10.0.1.2:~ anne$
```

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

To close a remote connection, enter `logout` and press Return.

To authenticate and send a command using a single line, append the command you want to execute to the basic `ssh` tool. For example, to delete a file:

```
$ 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 it isn't as secure as SSH, Telnet access is disabled by default.

To enable Telnet access:

```
$ service telnet start
```

To disable Telnet access:

```
$ service telnet stop
```

You are strongly advised not to enable Telnet. When you log in using Telnet, your login information, user name, and password are passed along the Internet in clear text. In fact, your entire Telnet session is also passed along the Internet in clear text. Any person on the network running tcpdump, ethereal, or similar applications can effortlessly 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 will be compromised as well.

To access a remote computer using `telnet`:

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

where *username* is the name of an administrator user on the remote computer and *server* is the name or IP address of the remote computer. For example:

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

Once connected, the remote computer will prompt for a login name, and then the 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 completely logged in. When you are finished working, log out from the remote computer by typing `logout` or `exit` at the remote computer's prompt. The telnet client will automatically exit when you log out from the remote computer.

See the `telnet` man page for more information.

Installing Server Software and Finishing Basic Setup

3

In this chapter you will find commands you can use to install, set up, and update Mac OS X Server software on local or remote computers.

Some computers come with Mac OS X Server software already installed. However, you might want to upgrade from a previous version, change a computer configuration, automate software installation, or completely refresh your server environment. This chapter covers the commands needed to perform a variety of software setup and installation tasks.

Installing Server Software

You can use the `/usr/sbin/installer` tool to install Mac OS X Server or other software on a computer. 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 out the details of the installation. It's useful for monitoring progress.

See the `installer` man page for detailed information.

To use 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 the target computer's actual IP address for <ip address>:

```
$ ssh root@<ip address>
```

If you don't know the IP address, you can use the `sa_srchr` tool to identify computers on the local subnet on which 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 10.4#RDY4PkgInstall#2.0#512
```

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

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 had been 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 will first want to 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. You can use the `sa_srchr` tool to identify the IP address of each computer that's ready for installation on your subnet.

Note: To locate computers, you must have booted the computer from the installation CD.

To list 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 would come from `sa_rspndr` running on a computer started up from the Mac OS X Server installation CD. The computer will respond with output similar to the following:

```
localhost#unknown#<ip address>#<mac address>#Mac OS X Server
     10.3#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

Use the `installer` tool to specify the target computer volume onto which you want to install the server software.

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/ServerNetworkImage10.4/System/
     Installation/Packages/OSInstall.mpkg
```

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

```
/Volumes/Mount 01
/Volumes/Mount1
/Volumes/Mount02
```

Preparing the Target Volume for a Clean Installation

If the target volume has Mac OS X Server version 10.3 or version 10.2.8 installed, when you run `installer`, it will upgrade the server to version 10.4 and preserve user files.

If you're not upgrading but 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 `diskutil` to partition the volume and to set up mirroring. For more information, see the `diskutil` man page or Chapter 7, "Working with Disks and Volumes," on page 83.

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 need to reinstall or upgrade system software.

Installing from Multiple CDs

If you're using CDs for server installation, use the `sa_srchr` tool to install the remaining software from the remaining installation CDs. Server Assistant opens automatically when installation is complete.

1 To use the next installation disc, use the `sa_srchr` command to locate the computer that's waiting. For <ip address>, specify the address you used in step 2:

```
$ /System/Library/Serversetup/sa_srchr <ip address>
```

2 When the `sa_srchr` response includes the string "#InstallInProgress", insert the next installation disc:

```
$ mycomputer.example.com#PowerMac4,4#<ip address>#<mac address> #Mac OS X
     Server 10.4#InstallInProgress#2.0#2080
```

Restarting After Installation

When installation from the disc is complete, restart the computer. Enter:

```
$ /sbin/reboot
```

or

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

Automating Server Setup

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 up and 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. You can automate this initial setup task by providing a configuration file that contains these settings.

Servers that have previously had Mac OS X Server version 10.4 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 might want to define generic setup data for a computer that's on order, or to configure 50 Xserve computers you want to be identically configured. You can also save setup data that's specifically tailored for a particular computer.

Important: When you perform an upgrade installation, 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 saved 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 any number of 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 more than one computer, don't specify network names (computer name and local hostname), and make sure that 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 then 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; when searching for setup files, target computers search for names in the order listed:
   - *MAC-address-of-server*.plist (include any 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—A 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 particular computer, it will use the file named generic.plist.

To create a configuration file at any time after initial setup:

1. Open Server Assistant (located in /Applications/Server/).
2. In the Welcome pane, select "Save setup information in a file or folder 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" 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; when searching for setup files, target computers search for names in the order listed here:
   - *MAC-address-of-server*.plist (include any 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—A 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 particular computer, it will use 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 the target computer or 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 new 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 any 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—A 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 the passphrase. Click Send.

Customizing a Configuration File

After you create a configuration file, you can modify it directly using a text editor, or write a script to automatically 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 example shows the basic structure and contents of a configuration file for a computer with the following 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 Computer//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>Administrator</string>
     <key>uid</key>
     <string>501</string>
  </dict>
  <key>ComputerName</key>
  <string>server1.example.com</string>
  <key>DS</key>
  <dict>
     <key>DSClientInfo</key>
     <string>2 - NetInfo client - broadcast dhcp static -192.168.42.250
     network</string>
     <key>DSClientType</key>
     <string>2</string>
     <key>DSType</key>
     <string>2 - directory client</string>
  </dict>
  <key>HostName</key>
  <string>server1.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>ResID</key>
   <integer>0</integer>
   <key>ResName</key>
   <string>U.S.</string>
   <key>ScriptID</key>
   <integer>0</integer>
</dict>
<key>NetworkInterfaces</key>
<array>
   <dict>
        <key>ActiveAT</key>
        <true/>
        <key>ActiveTCPIP</key>
        <true/>
        <key>DNSDomains</key>
        <array>
             <string>example.com</string>
        </array>
        <key>DNSServers</key>
        <array>
             <string>192.168.100.10</string>
        </array>
        <key>DeviceName</key>
        <string>en0</string>
        <key>EthernetAddress</key>
        <string>00:0a:93:bc:6d:1a</string>
        <key>PortName</key>
        <string>Built-in Ethernet</string>
        <key>Settings</key>
        <dict>
             <key>DHCPClientID</key>
             <string></string>
             <key>Type</key>
             <string>DHCP Configuration</string>
        </dict>
   </dict>
</array>
<key>PrimaryLanguage</key>
<string>English</string>
<key>Bonjour</key>
<dict>
   <key>BonjourEnabled</key>
   <true/>
   <key>BonjourName</key>
   <string>beasbe3</string>
</dict>
<key>SerialNumber</key>
<string>XSVR-123-456-A-BCD-7EF-GHI-89J-1KL-MNO-2</string>
```

```
  <key>ServiceNTP</key>
  <dict>
     <key>HostNTP</key>
     <false/>
     <key>HostNTPServer</key>
     <string>Local</string>
     <key>UseNTP</key>
     <false/>
  </dict>
  <key>ServicesAutoStart</key>
  <dict>
     <key>ARD</key>
     <false/>
     <key>Apache</key>
     <false/>
     <key>FTP</key>
     <false/>
     <key>File</key>
     <false/>
     <key>IChat</key>
     <false/>
     <key>Mail</key>
     <false/>
     <key>NetBoot</key>
     <false/>
     <key>QTSS</key>
     <false/>
     <key>SMB</key>
     <false/>
     <key>SWUPD</key>
     <false/>
     <key>WebDAV</key>
     <false/>
     <key>Weblog</key>
     <false/>
     <key>XgridA</key>
     <false/>
     <key>XgridC</key>
     <false/>
  </dict>
  <key>TimeZone</key>
  <string>US/Pacific</string>
  <key>VersionNumber</key>
  <integer>2</integer>
</dict>
</plist>
```

Note: The actual contents of a configuration file depend on the hardware configuration of the computer on which it's created, 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 any device volume mounted in /Volumes.

Devices you can use to provide configuration files include:

- A partition on one of the computer's hard disks
- 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`—Directory service command line is a general purpose tool that allows you to create, read, and manage directory service data. If invoked without any commands, `dscl` runs interactively, reading commands from standard input. See Chapter 8, "Working with Users and Groups," for more information about the usage of this command.
- `systemsetup`—Use `systemsetup` to set a number of system-wide preferences. If you were going through Server Assistant, you would have to select the proper keyboard and time zone. The `systemsetup` tool can configure both these preferences, and more. See Chapter 5, "Setting General System Preferences," for mor information on the usage of this command.
- `networksetup`—Anything that you can configure in the Network pane of System Preferences can also be configured using `networksetup`. See Chapter 6, "Setting Network Preferences," for more information about the usage of this command.

See the man pages related to these tools for more information. The man pages for `systemsetup` and `networksetup` are only available 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 -getAllPort
```

If you want to use the tool to perform several commands, you can change your working folder and enter a shorter command:

```
$ cd /System/Library/ServerSetup
$ ./serversetup -getAllPort
$ ./serversetup -getDefaultInfo
```

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

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

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

Using the serveradmin Tool

The `serveradmin` tool is used for administering service-related tasks. Some services need to be restarted after you change certain 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 is displayed 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 have information about using the `serveradmin` tool 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.

- `servermgrd` uses SSL for encryption and client authentication, 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. You can use the Certificate Manager in Server Admin to create and manage certificates. See the mail service administration guide for more information.

- 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 only 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 the Mail Service administration guide 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.
- The `servermgrd` SSL encryption options can be changed at any time 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

See the following for information about changing general system preferences and network settings:

- Chapter 5, "Setting General System Preferences," on page 57
- Chapter 6, "Setting Network Preferences," on page 63

Viewing, Validating, and Setting the Software Serial Number

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

To display 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
```

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 that they can be tracked to a specific company, group, or individual. If a serial number has watermarking strings associated with it, then it is necessary to supply the watermark information when setting or validating the serial number.

To check whether 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 will be 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 [recommmended] [restart]
  * J2SE50Release3-3.0
     **PRERELEASE** J2SE 5.0 Release 3 (8M318) (3.0), 44020K [recommmended]
  - 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 the update responses. See the `softwareupdate` man page for more information about the command.

Moving a Server

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

If you must move a server after initial setup, you need to 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 need to 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 on which other computers depend.
- 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 will need to change 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, NetInfo, and LDAP folders on the server. See "Changing a Server's IP Address" on page 66. You may need to manually adjust some network configurations, such as the local DNS database, after using the tool.
- 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 the Open Directory administration guide.

Restarting or Shutting Down a Computer

4

In this chapter you will find commands you can use to shut down or restart a local or remote computer.

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

Restarting a Computer

You can use the `reboot` or `shutdown -r` command to restart a computer at a specific time. See the relevant man pages for more information.

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 automatically after a power failure or system freeze. See "Viewing or Changing Automatic Restart Settings" on page 59.

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 desired 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

You can use the `shutdown` tool to shut down a computer at a specific time. See the `shutdown` man page for more information.

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.

Manipulating Open Firmware NVRAM Variables

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

To view the different NVRAM variables:

```
$ nvram -p
```

Monitoring and Restarting Critical Services

In earloier 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 version 10.4, `watchdog` has been replaced by `launchd`. The `launchd` daemon manages other daemons, both for the computer as a whole and for individual 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 rest of the computer. In Mac OS X Server, it is preferable to have your daemon started by `launchd`.

Note: Some system administrators need to 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 whatever changes they require, and avoid modifying `rc` or creating a SystemStarter Startup Item. The `rc` command script may be phased out in the future.

The configuration files are located 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

In this chapter you will find commands you can use to set system preferences, usually set using the System Preferences graphical application.

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 or time
- A computer's time zone
- Whether a server uses a network time server

These settings can also be changed using the Date & Time pane of System Preferences.

Viewing or Changing the System Date

To view the current system date:

```
$ sudo systemsetup -getdate
```

or

```
$ serversetup -getDate
```

To set the current system date:

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

or

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

Viewing or Changing the System Time

To view the current system time:

```
$ sudo systemsetup -gettime
```

or

```
$ serversetup -getTime
```

To change the current 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 current time zone:

```
$ sudo systemsetup -gettimezone
```

or

```
$ serversetup -getTimeZone
```

To view the 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 use of a network time server:

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

To view the current network time server:

```
$ sudo systemsetup -getnetworktimeserver
```

To specify a network time server:

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

Viewing or Changing the Energy Saver Settings

You can use the `systemsetup` tool to view or change a server's energy saver settings. These can also be changed using 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 the Power Management Settings

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

- Display dim timer
- Disk spindown 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 can configure different settings for the different power modes using `pmset`. There are four flags you can use: `-a`, `-b`, `-c`, and `-u`. `-b` applies the settings to battery operation, `-c` to charger (wall power), `-u` to UPS, and `-a` to all.

To set disk spindown timer for all modes of operation:

```
$ sudo pmset -u spindown minutes
```

Parameter	Description
minutes	Must be a multiple of 30 seconds.

To display the current settings:

```
$ sudo pmset -g command
```

See the `pmset` man page for more information.

Viewing or Changing the Startup Disk Settings

You can use the `systemsetup` tool to view or change a computer's startup disk. This can also be set using the Startup Disk pane of System Preferences.

To view the current startup disk:

```
$ sudo systemsetup -getstartupdisk
```

To view the available startup disks:

```
$ sudo systemsetup -liststartupdisks
```

To change the current startup disk:

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

Viewing or Changing the Sharing Settings

You can use the `systemsetup` tool to view or change Sharing settings. These can also be set using 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
```

Telnet access is disabled by default because it isn't as secure as SSH. You can, however, 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)
```

Viewing or Changing the International Settings

You can use the `serversetup` tool to view or change language settings. These can also be set using the International pane of System Preferences.

To view the current primary language:

```
$ serversetup -getPrimaryLanguage
```

To view the installed primary language:

```
$ serversetup -getInstallLanguage
```

To change the installation language:

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

To view the script setting:

```
$ serversetup -getPrimaryScriptCode
```

Viewing and Changing the 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

In this chapter you will find commands you can use to change the 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

Mac OS X Server includes `ifconfig`, the standard UNIX tool for configuring networks. Both `ifconfig` and `networksetup` make system calls to change the interface configuration. However, `ifconfig` and `networksetup` do not communicate with each other. `ifconfig` changes the network interface settings.

Warning: If you use `ifconfig`, your computer will be out of sync and will revert back to the contents of preferences.plist after a restart.

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

It's best to rely on `networksetup` and `serversetup` for your manual configuration. You are encouraged to view the man pages of both commands to see all the available configuration options.

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 65.

Viewing Port Names and Hardware Addresses

To list all port names:

```
$ serversetup -getAllPort
```

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 particular 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 that is transmitted over a network travels in data packets. The size of the data packets is called maximum transmission units (MTU), which if too large or too small will affect performance. You can use the `networksetup` tool to change the MTU size for a port.

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 the 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 the 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 particular 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 an existing 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 the port configurations:

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

Managing TCP/IP Settings

TCP/IP is a set of layered protocols that allow shared applications 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

Changing a server's IP address isn't as simple as changing the TCP/IP settings. Address information is set throughout the system when you set up the server. To make sure that all the necessary changes are made, use the `changeip` tool.

`changeip` is a python script that runs tools out of the /usr/libexec/changeip folder. There are currently three tools 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 NetInfo 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:

```
$ 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, you must 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 DNS host name of the server.
new-hostname	(optional) The new DNS host name of the server.

See the `changeip` man page for more information and examples.

2 Use the `networksetup` or `serversetup` tool (or the Network pane of System Preferences) to change the server's IP address in its network settings.

3 Restart the server.

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

```
$ changeip /LDAPv3/127.0.0.1 192.0.0.12 192.0.1.10 oldhost newhost
```

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

To change the IP address of a standalone server:

```
$ changeip - 192.0.0.12 192.0.1.10 oldhost newhost
```

To change the IP address of a server bound to a parent NetInfo directory domain:

```
$ changeip /NetInfo/root/netinfonode 192.0.0.12 192.0.1.10 oldhost newhost
```

To change the IP address of a server bound to a parent NetInfo directory domain, where the old and new IP addresses map to the same name:

```
$ changeip /NetInfo/root/netinfonode 192.0.0.12 192.0.1.10
```

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

You can use the `serversetup` and `networksetup` tools to change a computer's TCP/IP settings.

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 66.

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 port en0:

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

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

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

To change TCP/IP settings for a particular 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

You can use the `serversetup` tool to view and modify the Domain Name Server (DNS) settings.

To view the DNS servers for port en0:

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

To change the DNS servers for port en0:

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

To view the DNS servers for a particular port or device:

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

To change the DNS servers for a particular port or device:

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

To list the DNS servers for a configuration:

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

To view the DNS search domains for port en0:

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

To change the DNS search domains for port en0:

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

To view the DNS search domains for a particular port or device:

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

To change the DNS search domains for a particular port or device:

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

To list the DNS search domains for a configuration:

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

To set the DNS servers for a configuration:

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

To set the 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

Use the `serversetup` tool to enable or disable TCP/IP on a computer.

To enable TCP/IP on a particular port:

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

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

To disable TCP/IP on a particular port:

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

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

Working with 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 the devices that support VLANs:

```
$ networksetup -listdevicesthatsupportVLAN
```

IEEE 802.3ad Ethernet Link Aggregation

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 any modifications. The advantage of aggregation is that the virtual interface provides increased bandwidth by merging the bandwidth of the 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 any port or cable fails. All traffic that was being routed over the failed port is automatically rerouted to use one of the remaining ports. This failover is completely transparent to the software using the connection. This feature provides increased bandwidth and automatic failover for the server environment.

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* is the name of the pseudo device and the *physical_interface* is the actual 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 have their Ethernet address reprogrammed so that all 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 all 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 will remain inactive. To monitor the IEEE 802.3ad Link Aggregation state, use the `-b` option.

See the `ifconfig` man page for more information.

Configuring Ethernet Link Aggregation

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

To display if the 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. Use the `serversetup` tool to enable or disable AppleTalk.

To enable AppleTalk on a particular port:

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

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

To disable AppleTalk on a particular 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 check 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 uses a manager/agent design.

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 the manager and the agent.

Installing SNMP

To use SNMP for monitoring or data collection, an SNMP agent (`snmpd`) must be running on the monitored Mac OS X Server host computer. Mac OS X Server version 10.1.5 or later includes a version of SNMP (UCD-SNMP v. 4.2.3 or later).

If you do not have the file /usr/sbin/snmpd, then SNMP is not installed. Mac OS X Server version 10.1.4 or earlier require that SNMP be built and installed. Mac OS X Server v10.1.5 or later Admin CDs include the SNMP package on the CD used to install UCD-SNMP 4.2.3 on these older systems. If you do not have access to the CD, you may download current SNMP source from the NET-SNMP Project Home Page (www.net-snmp.org/).

Warning: Once SNMP is active, anyone with a route to the SNMP host will be able to collect SNMP data from it. To learn more, consult the various SNMP information sources listed below.

The default configuration of `snmpd` uses privileged port 161. For this reason and others, it must be executed by root or by using `setuid`. You should only use `setuid` as root if you understand the ramifications. If you do not, seek assistance or additional information. There are flags available for `snmpd` that will change the UID and GID of the process after it starts. See the `snmpd` man page for more information.

Starting SNMP

To start SNMP you have three options:

- Click the checkbox to enable SNMP in the Server Admin application. This modifies the hostconfig file for you.
- Modify the hostconfig file to start SNMP automatically at system startup.
- Start the SNMP agent manually.

To start SNMP on Mac OS X Server version 10.4 or later by modifying the hostconfig file:

1. Open the /etc/hostconfig file.
2. Locate the line:

```
SPOTLIGHT=-YES-
```

3. Immediately above it, add this line:

```
SNMPSERVER=-YES-
```

4. Save the file.

To start SNMP on Mac OS X 10.4 client computers by modifying the hostconfig file:

Mac OS X 10.4 client systems already have the `SNMPSERVER:=-NO-` line in their hostconfig file by default.

1 Open the /etc/hostconfig file.

2 Locate the line:

```
SNMPSERVER=-NO-
```

3 Change `NO` to `YES`.

4 Save the file.

Note: Systems running Mac OS X Server version 10.3 or earlier will need to have the line added.

Changing the `SNMPSERVER` line in the hostconfig file, causes `snmpd` to be executed during system startup, with no options, as dictated by the /System/Library/StartupItems/SNMP/SNMP file. For further instruction on editing configuration files, including important precautionary statements, see technical document 106619, "Mac OS X Server: How to Edit Configuration Files".

To start the snmp agent manually:

```
$ /usr/sbin/snmpd
```

Configuring SNMP

The configuration (conf) file for `snmpd` is typically in the /usr/share/snmp/ folder, and is named snmpd.conf or snmpd.local.conf. If you have an environment variable SNMPCONF, `snmpd` will read any files named snmpd.conf and snmpd.local.conf in these folders. The SNMP agent can be started with a `-c` flag to indicate other conf files. See the `snmpd` man page for more information about which conf files can be used.

Configuration files can be created and installed more easily using the included script /usr/bin/snmpconf. As root, use this script with the `-i` flag to install the file in the /usr/share/snmp/ folder. Otherwise, the default location for the file to be written is the user's home directory (~/). Only root has write permission for /usr/share/snmp/.

Because `snmpd` reads its conf files at startup, changes to the conf files require that the process be stopped and restarted. To do this, you must identify the process id.

To identify the process id:

```
$ ps aux |grep snmpd
```

To stop `snmpd` :

```
$ kill <pid>
```

Once `snmpd` is stopped, you can customize the snmpd.conf file as needed.

To customize the data provided by snmpd, you may add an snmpd.conf file using /usr/bin/snmpconf:

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

You will then see a series of text menus. Make these choices in this order:

1. Select File: 1 (snmpd.conf)
2. Select section: 5 (System Information Setup)
3. Select section: 1 (The [typically physical] location of the system)
4. The location of the system: type text string here—such as `server_room`
5. Select section: f (finish)
6. Select section: f (finish)
7. Select File: q (quit)

This creates an snmpd.conf file with a creation date of today.

To view the snmp.conf file:

```
$ ls -l /usr/share/snmpd.conf
```

Once the configuration file is created, restart the `snmpd` process.

To start snmpd, execute this as root:

```
$ sudo /usr/sbin/snmpd
```

Collecting SNMP Information from the Host

To get the SNMP information you just added, execute this command from a host that has the SNMP tools installed, where *`hostname`* is replaced with the actual name of the target host:

```
$ snmpget -v 1 -c public hostname system.sysLocation.0
```

You should see the location you provided. In this example, you would see:

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

The other options in the menu you were working in are:

```
$ snmpget -v 1 -c hostname public system.sysContact.0
$ snmpget -v 1 -c hostname public system.sysServices.0
```

The final `.0` indicates you are looking for the index object. The word `public` is the name of the SNMP community, which you did not alter. If you need information about either of these, or explanations of SNMP syntax, there are tutorials available at www.netsnmp.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:

```
$ sudo snmpwalk -v 1 -c public localhost
```

This will list 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 retrieve specific SNMP management values, use the `snmpget` tool as shown in the following examples.

To view the system name:

```
$ snmpget -v 1 -c public localhost system.sysName.0
SNMPv2-MIB::sysName.0 = STRING: xlabxs06.apple.com
```

To view the system location:

```
$ snmpget -v 1 -c public localhost system.sysLocation.0
SNMPv2-MIB::sysLocation.0 = STRING: "server_room"
```

To view the system uptime:

```
$ snmpget -v 1 -c public localhost system.sysUptime.0
SNMPv2-MIB::sysUpTime.0 = Timeticks: (72239) 0:12:02.39
```

For a list of `snmp` man pages, enter the following:

```
$ 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 allows for a more secure environment. Use the `networksetup` tool to view or change the proxy settings.

Viewing or Changing FTP Proxy Settings

To view the FTP proxy information for a configuration:

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

To set the 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 the web proxy information for a configuration:

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

To set the 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 the secure web proxy information for a configuration:

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

To set the 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 the streaming proxy information for a configuration:

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

To set the 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 Settings

To view the gopher proxy information for a configuration:

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

To set the 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 the SOCKS firewall proxy information for a configuration:

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

To set the 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 the proxy bypass domains for a configuration:

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

To set the 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. Use the `networksetup` tool to view or change the AirPort settings.

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 current AirPort network:

```
$ sudo networksetup -getairportnetwork
```

To join an AirPort network:

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

Managing the Computer, Host, and Bonjour Names

These names are used by networking applications to identify a computer.

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. Use the `serversetup` tool to view or modify the computer name.

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 that the network uses to identify a device attached to the network. Use the `serversetup` tool to view or modify the host name.

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` tools.

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 automatically 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. Use the `serversetup` tool to view or change the Bonjour name.

To display the server's Bonjour name:

```
$ serversetup -getBonjourname
```

To change the server's Bonjour name:

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

The command displays `0` if the name was changed.

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

Managing Preference Files and the Configuration Daemon

The various sets of configuration information that 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. This storage is in the form of key-value pairs. The key is a description of what is being stored, and the value is the actual 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 trying to debug 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), perform the following:

```
$ scutil
> open
```

This opens a session with `configd.` Once the session is open, you can list all of the keys in 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 actual state of the computer. File indicates stored information as of the last time the configuration file was updated.

Using `scutil`, you can view data in the keys. First you must get the data, and then you can 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, such that if its state changes, `scutil` will alert you. To quit the `scutil` session, enter `quit` at the prompt.

```
> quit
```

You can also manage system configuration parameters from within `scutil` using the `--get` and `--set` options. These provide a means of reporting and updating a select 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	This is the new hostname value you wish to set.

To get the hostname of a system:

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

See the `scutil` man page for more information or enter `help` at the `scutil` prompt.

Changing Network Locations

A network location contains all of the 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 may need to switch between locations quickly. `scselect` allows you to access these configuration sets or locations.

To view the current locations:

```
$ scselect
```

The computer will respond 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 listed that you want to switch to:

```
$ scselect 1
```

In this example, the network location will switch to AirPort.

Working with Disks and Volumes 7

In this chapter you will find commands that are used to initialize and test disks and volumes.

Computers use disks and partitions to store and organize data. This chapter covers the commands that are 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 would be called /dev/disk0s1, the second partition would be /dev/disk0s2, and so on.

Although Mac OS X Server assigns a device name to each device, the files on a particular device are not accessed in this way. A virtual file system is created where all files on all devices appear to exist under a single hierarchy. This sets one root folder and every file exisiting 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 would like those files to appear. The folder given to the operating system is the mount point. Different volumes on a computer may 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:

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

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

To view a list of currently mounted file systems:

```
$ sudo mount
```

To mount a network folder:

```
$ mount /dev/
```

mount returns the value 0 if the mount succeeded.

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
```

umount returns the value 0 if the umount succeeded. See the umount man page for more information.

Displaying Disk Information

The df tool located in /bin is designed to display free disk space. In addition, df is a useful way to find out what your current disk partitions are, how much space each one takes up, which block each partition starts on, which device file is associated with each partition, and where each partition is mounted.

To display disk information:

```
$ df
```

The computer will respond 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. diskspacemonitor is disabled by default.

To enable diskspacemonitor:

```
$ sudo diskspacemonitor on.
```

You may be prompted for your password. See the diskspacemonitor man page for more information.

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. It lets you specify how often you want to monitor disk space, and specify 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. See the comments in the file for additional information.
- 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 email 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.

  See the comments in the script and configuration files for more information about these files.
- If you want 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 will be 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

Three predefined scripts are executed automatically, in order to reclaim space used on your server for log files generated by:

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

The scripts use values in the following configuration files to determine whether and how to reclaim space:

- 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 is intended to run weekly, but is currently empty. Its configuration file is /etc/diskspacemonitor/weekly.server.conf.
- The script /etc/periodic/monthly/600.monthly.server is intended to run monthly, but is currently empty. Its configuration file is /etc/diskspacemonitor/monthly.server.conf.

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

- 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.

There are several additional parameters you can specify. See comments in the configuration files for information about all the parameters and how to set them. The scripts ignore all 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.

Erasing, Modifying, Verifying, and Repairing Disks

You can use `diskutil` to erase, modify, verify, and repair disks. This command provides functionality that overlaps with the functionality of `pdisk`, `newfs_hfs`, and `disktool`. For example, you can use both `diskutil` and `pdisk` to partition a disk. However, unlike `pdisk`, which lets you partition tables at their most basic level by setting the exact 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 currently 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 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 mount a drive:

```
$ diskutil mountDisk diskvol
```

Parameter	Description
diskvol	Device name.

To erase and repartition a disk:

```
$ diskutil partitionDisk disk numberOfPartitions part1Format part1Name
      part1Size
```

Parameter	Description
disk	Device name (such as `disk0`).
numberOfPartitions	
part1Format	HFS+ or UFS.
part1Name	
part1Size	Can be either 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 may be times when you would want to set the file system to be case sensitive. You can use the `diskutil` tool to format a drive for case-sensitive HFS+.

Note: Volumes you format as case-sensitive HFS+ are also journaled.

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`

See the `diskutil` man page for more options and information about repairing and modifying disks.

Partitioning and Formatting Disks

Disk partitions are subsdivisions of a disk to which you apply operating-system–specific formatting.

Partitioning a Disk

You can use `pdisk`, located in /usr/sbin, to edit the disk partition table. You can initialize the disk, create partitions, and delete partitions. The `pdisk` tool is menu-driven, which means that once it is launched, you are prompted to enter a `pdisk` command. You can find the commands by typing `?` at the pdisk prompt. The following are some of the more useful commands:

Command	Description
`L`	Lists the partition maps of all the drives. `pdisk` lists all the 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, you have to use the raw device file as the argument.

Once 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 new partition. There are two partition types, Apple_HFS and Apple_UFS.
`w`	Writes the modifications to the partition map on-disk. Before that, all edits and modifications are only in memory and not yet implemented.

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

After a partition has been created on a device, the partition needs to be formatted before the computer will be able to store data on the device. Formatting a disk partition creates the volume and sets the file system.

Labeling a Disk

Once a disk is formatted, it needs to 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 be writable by anne:

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

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

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

See the `disklabel` man page for more information about creating disk labels.

Formatting a Disk

You can use `newfs`, located in /sbin, to create a new volume. `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. Extreme care should be taken to ensure a successful format when modifying the settings beyond the default. Before running `newfs`, the disk must be labeled using the `disklabel` tool.

To fomat a disk:

```
$ newfs
```

See the `newfs` man page for options in detail.

To format a disk to HFS+, you would need to use the newfs_hfs tool located in /sbin:

```
$ newfs_hfs
```

See the `newfs_hfs` man page for more information.

Checking for Disk Problems

You can use the `diskutil` or `fsck` tool (`fsck_hfs` for HFS volumes) to check the physical condition and file system integrity of a volume. See the related man pages for more information.

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.

Checking to See If Journaling is Enabled

You can use the `mount` tool to see if journaling is enabled on a volume.

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 an Existing Volume

You can use the `diskutil` tool to enable journaling on a volume without affecting existing files on the volume.

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 the exisiting 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

You can use the `newfs_hfs` tool to set up and enable journaling when you erase a disk.

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. Whenever 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 all of the applications using Spotlight, and changes are reflected dynamically to the user.

The Spotlight store retains information that is extracted into two seperate 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 as root using your favorite editor. For example:

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

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

You can also 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 as root.

2 Change the value of the *spotlight* parameter to `-YES-`.

You can also 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, which is similar to the `ls` tool, lists all of the metadata attributes for a specific file.

To view the metadata of a file:

```
$ mdls filename
```

The computer will respond 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, use 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 any volume.

To enable indexing on a volume:

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

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

To disable indexing on a volume:

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

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

See the `mdutil` man page for more information.

Managing RAID Volumes

In addition to standard drive management options, `diskutil` has the ability 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 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 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	Specifies the slice number to replace.
fromDisk	Specifies the mirror source.
toDisk	Specifies the repaired mirror destination.

Note: Xsan RAID volumes have their own set of commands, which are described in an appendix of the Xsan administrators guide. See the appendix for informatian about the `megaraid` tool, used for managing a PCI RAID card.

Imaging and Cloning Volumes Using ASR

You can use Apple Software Restore (ASR) to copy a disk image onto a volume or to prepare existing 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 is already there, and block copies, which restore entire disk images. The `asr` tool doesn't create the disk images. You can use `hdiutil` to create disk images from volumes or folders.

You must run ASR as root. 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 "To create an image from a folder:" on page 177.

5 Mount the disk image.

6 Remove cache files, host-specific preferences, and virtual memory files. See the `asr` man page for examples of what files to remove.

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
```

See the `asr` man page for command syntax, limitations, and image preparation instructions.

Working with Users and Groups

8

In this chapter you will find commands you can use 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. There are several command-line tools that facilitate working with the directory domains that hold these accounts.

Understanding Accounts

There are three kinds of accounts you can set up with Workgroup Manager: user accounts, group accounts, and computer lists. 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 perhaps to personalize the user's environment. Along with accounts you create, Mac OS X Server has some 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 Accounts

A user account stores data that Mac OS X Server needs to validate the user's identity and provide services for the user. This section provides an overview of user accounts.

User accounts, as well as group accounts and computer lists, 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, a NetInfo domain, or other read/write directory domain) or it can reside on a non-Apple server (for example, a non-Apple LDAP or Active Directory server). This section describes how to administer user accounts stored in various kinds of directory domains.

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 info about or change the settings of a particular server. Domain administrator privileges determine the extent to which the user can view or change the account settings for users, groups, and computer lists in the directory domain.

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

To create nonadministrator users, see "Creating a Nonadministrator User Account" on page 100. To create administrator users in a network directory domain, see "Creating a Domain Administrator User Account" on page 99.

To create a local administrator user account:

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

The name, short name, and password must be entered 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 an local administrator user with a specific UID:

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

The name, short name, password, and UID must be entered 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 an local administrator user with a specific UID and home folder:

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

The name, short name, password, and UID must be entered 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

In order to create a domain administrator user account for a networked directory, you need to already have a domain administrator user account.

Before starting, you should already 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 100.

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 Once connected to the directory, choose the directory domain. Change the current folder to LDAPv3/*ipaddress*/Groups.

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

Replace *`ipaddress`* with the IP address of your directory server. If using a NetInfo directory domain, enter `cd /NetInfo/root/Groups` at the prompt.

3 Create an administrator user.

```
>append admin Member adminusername
```

This command creates an administrator user, but it doesn't add the GUID (globally unique identifier) of the administrator user to the group account.

4 Add the administrator user to the group.

```
> append admin GroupMembers guid
```

Replace *`guid`* with the globally unique identifier.

5 Quit the `dscl` tool.

```
>quit
```

To find the GUID of the administrator user:

```
> cd /Users/
> read adminusername GeneratedUID
```

Checking a User's Administrator Privileges

Use the `serversetup` tool to verify the administrator privileges of a specific user.

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 new 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. Each user on a server must have a unique user ID. Use 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 is displayed in the search path in Directory Access). If you connect to a NetInfo domain, replace `UniqueID` with `uid`.

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: Pick a user ID that isn't on either list and that is greater than 501. 501 is the user ID of the local administrator user that gets 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. 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. If using a NetInfo directory domain, enter `cd /NetInfo/root/Users` at the prompt.

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 new 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:

```
apple-generateduid:1B2A3456-E7C8-9EC1-2345-678D912E3456
cn: anne johnson
gidNumber: 99
HomeDirectory: /LDAPv3/ipaddress/Users/ajohnson
loginShell: /bin/bash
objectClass: inetOrgPerson posixAccount shadowAccount apple-user extensible
     object organizationalPerson top person
sn: ajohnson
uid: ajohnson
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
```

You will be prompted to enter a password.

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 all of the home folders:

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

where *username* is the name of an administrator user on the remote server and *server* is 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.

```
$ sudo createhomedir -s -u ajohnson
```

To create a group account for the new user, see "Creating a Group Account" on page 111 before doing this step.

The new user account is now complete and can be used for login. See the dscl man page for more information.

Retreiving a User's GUID

When a user account is created, the computer generates a 128-bit integer called a globally unique identifier (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. If using a NetInfo directory domain, enter `cd /NetInfo/root/Users` at the prompt.

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 particular 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 delete 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. If using a NetInfo directory domain, enter `cd /NetInfo/ipaddress/Users` at the prompt.

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. See "Removing a Group Account" on page 112, for information about deleting this group.

Revoking a User's Right to Access His or Her Account

There are times when it is necessary to revoke a user's ability to access the computer. This involves preventing the user from logging in and then terminating all of the user's processes. This can be done by forcing the user to log out and then killing any remaining processes, or by just killing all of the user's processes.

To prevent a user from logging in:

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. If using a NetInfo directory domain, enter `cd /NetInfo/root/Users` at the prompt.

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 Quit `dscl` by entering:

```
> quit
```

5 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.

To terminate all of a user's processes:

After disabling the user account, you need to kill all of the user's active processes that are currently running on the directory server.

> *Warning:* Unconditionally killing all of a user's processes will cause the user to lose any 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. To terminate all user processes unconditionally, enter the following command, replacing *ajohnson* with the user name:

```
$ sudo killall -9 -u ajohnson
```

Refer to the `killall` man page for more information about terminating processes.

To reenable a user account that is disabled:

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. If using a NetInfo directory domain, enter `cd /NetInfo/root/Users` at the prompt.

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 Quit `dscl` by entering:

```
> quit
```

5 Enable the user account by entering the following command. Replace *ajohnson* with the short name of the user account and replace *diradmin* with the short name of your domain administrator account.

```
$ pwpolicy -a diradmin -u ajohnson -setpolicy "isDisabled=0"
```

Checking a Server User's Name, UID, or Password

You can use the following commands to check the name, UID, or password of a user in the server's local directory domain.

Note: These tasks apply only to the local directory domain on the server.

To see if a full name is already in use:

```
$ sudo /System/Library/ServerSetup/serversetup -verifyRealName "longname"
```

The command displays a `1` if the name is already in use, or a `0` if it isn't.

To see if a short name is already in use:

```
$ sudo /System/Library/ServerSetup/serversetup -verifyName shortname
```

The command displays a `1` if the name is already in use, or a `0` if it isn't.

To see if a UID is already in use:

```
$ sudo /System/Library/ServerSetup/serversetup -verifyUID uid
```

The command displays a `1` if the UID is already 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 the 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 PDC and mapping domain privileges to UNIX groups. See the `net` man page for more information.

Modifying a User Account

You can change the value of an attribute in a user account by using `dscl`.

There are many attributes that can be set for users. The following table describes some of the user account attributes you can modify 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'sTerminal shell.
sn	User's sir 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'sTerminal 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. If using a NetInfo directory domain, enter `cd /NetInfo/root/Users` at the prompt.

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, replacing *ajohnson* with the user account's short name, *attribute* with the name of the attribute whose value you wish to change, and *newvalue* with the value:

```
> create ajohnson attribute newvalue
```

5 Quit `dscl` by entering:

```
> quit
```

Creating a Mobile User Account

Mobile accounts are network accounts that have been set up to be accessible even when the user is not connected to the server where the account resides. The mobile account user is provided with a local home folder on the computer the user is logged in to. This functionality reduces network traffic and improves overall performance.

You can use the MCXCacher tool to create a mobile account from the command line. `MCXCacher` performs the pre-login checks and refreshes cache if required. This tool will only work if the client is bound to a network directory system containing the target user record.

Important: Creating a mobile user account is a client-only operation. These commands must be either performed on the client computer or while connected through SSH to a client computer.

To create a mobile account:

1 Use the `MCXCacher` to create a mobile account on the current computer.

```
$ sudo /System/Library/CoreServices/mcxd.app/Contents/Resources/
     MCXCacher -U ajohnson
```

Where *ajohnson* is the short name of a user in the parent folder and */Users/ajohnson* is the Home Folder.

2 Run the `passwd` command to change passwords.

```
$ passwd ajohnson
```

Then enter verify passwords. You can also set the password by logging in while connected to the network.

3 Create a standard home folder for a user with a mobile account.

```
$ sudo createhomedir -u ajohnson -c -l
```

When a mobile account is enabled, it appears in the login window and in the Accounts pane of System Preferences with the label Mobile. You can alsol select the user in Workgroup Manager and click Preferences > Mobility. If "synchronize account for offline use" is checked, the account is mobile.

The `MCXCacher` tool does not have a man page. This tool, located in the /System/Library/CoreServices/mcxd.app/Contents/Resources/ folder, performs the pre-login checks and refreshes cache if necessary. The following examples describe other options for `MCXCacher` tool.

To create (or overwrites an existing) mobile account on the current machine:

Enter the following, replacing *usershortname* with the user's short name and *homepath* with the location of the user's Home Folder.

```
$ sudo /System/Library/CoreServices/mcxd.app/Contents/Resources/
     MCXCacher -U usershortname [-h homepath]
```

To perform the post–login checks and refreshes caches and caches the current user's mcx_settings:

Enter the following, replacing *usershortname* with the user's short name.

```
$ sudo /System/Library/CoreServices/mcxd.app/Contents/Resources/
     MCXCacher -U usershortname
```

To flush the cache:

```
$ sudo /System/Library/CoreServices/mcxd.app/Contents/Resources/
     MCXCacher -f
```

To dirty the cache so that it will be refreshed at the next login:

```
$ sudo /System/Library/CoreServices/mcxd.app/Contents/Resources/
     MCXCacher -d
```

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 particular 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)

See the `createhomedir` man page for more information.

In all cases, the home folders are created on the server where you run the tool.

To create a home folder for a particular 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 particular 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

You can use `mnthome` to mount a user's home folder. 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 directory on an AFP server:

```
$ mnthome -p password
```

See the `mnthome` man page for more information.

Administering Group Accounts

A group is simply a collection of users who have similar needs. For example, you can add all users with a particular 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 various 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 is used to help control user access to folders and files. Individual users may 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 new 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 is displayed in the search path in Directory Access). If you connect to a NetInfo domain, use:

```
$ dscl /NetInfo/root -list /Groups gid | awk ‘{print $2}’ | sort -n.
```

After you enter the command, the `dscl` tool displays a list of assigned IDs similar to the following output:

```
-2
0
1
99
25
26
27
70
71
76
77
78
79
501
```

Important: Pick an ID that isn't on either 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. If using a NetInfo directory domain, enter `cd /NetInfo/root/Groups` at the prompt.

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 new 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 new group by entering the following command, replacing *officegroup* with the new group account's short name.

```
> read officegroup
```

`dscl` displays the settings for your new group account, similar to the following output:

```
apple-generateduid:4B3A5678-E9C1-2EC3-4567-891D234E5678
cn: officegroup
gidNumber: 600
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
```

See the `dscl` man page for more information about using the `dscl` command-line tool.

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. If using a NetInfo directory domain, enter `cd /NetInfo/root/Groups` at the prompt.

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. If using a NetInfo directory domain, enter `cd /NetInfo/root/Users` at the prompt.

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 *ajohnson* with the short name of the user account and *officegroup* with the short name of the group account:

```
> append admin Member adminusername
```

This creates an administrator user, but it does not add the GUID (globally unique identifier) of the administrator user to the group account. This may cause security and compatibility issues.

5 Add the administrator user to the admin group.

```
> append admin GroupMembers guid
```

6 Review the new settings of the group by entering the following command, replacing *officegroup* with the group account's short name:

```
> read officegroup
```

`dscl` displays the settings for the group account, similar to the following output:

```
apple-generateduid:4B3A5678-E9C1-2EC3-4567-891D234E5678
cn: officegroup
gidNumber: 600
MemberUid: mchen ajohnson bmiller
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
```

7 Quit `dscl` by entering:

```
> quit
```

To find the guid of the administrator user:

```
> cd /Users/
> read adminusername 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. If using a NetInfo directory domain, enter `cd /NetInfo/root/Groups` at the prompt.

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:

```
apple-generateduid:4B3A5678-E9C1-2EC3-4567-891D234E5678
cn: officegroup
gidNumber: 600
MemberUid: mchen ajohnson bmiller
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:

```
apple-generateduid:4B3A5678-E9C1-2EC3-4567-891D234E5678
cn: officegroup
gidNumber: 600
MemberUid: mchen bmiller
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 Nested Group

Nested groups allow for one group (child) to be a member of a second group (parent), thus inheriting the permissions and attributes of the parent group. All members of a nested group will become child members of the parent group as well.

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/ipaddess] 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. If using a NetInfo directory domain, enter `cd /NetInfo/root/Groups` at the prompt.

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 (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:

```
apple-generateduid:4B3A5678-E9C1-2EC3-4567-891D234E5678
apple-group-nestedgroup:1A2B3456-C7D8-9EF1-2345-678G912H3456
cn: parentgroup
gidNumber: 700
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
```

Once a nested group is established, it can be split apart or 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/ipaddess] 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

You can use dsEditGroup to add, remove, or edit group records in the local directory service.

To display the information about a particular group:

```
$ dseditgroup officegroup
```

To delete a group:

```
$ dseditgroup -o delete -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.

This will prompt you for your diradmin password, which is much more secure than putting the password in the command you are sending.

See the `dseditgroup` man page for more information.

Creating a Group Folder

A group folder facilitates the sharing of files between members of a group. Once you set up a group folder in Workgroup Manager you need to use the `CreateGroupFolder` tool to create the actual group folder. Group folders should be created on the server that hosts the group folders.

To create a group folder:

```
$ sudo /usr/bin/CreateGroupFolder
```

See the `CreateGroupFolder` man page for more information.

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. Any 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 selected workgroup in its "workgroup" key.

Important: Viewing the workgroup a user selects at login must be performed on the client computer.

To view the workgroup a user selects at login, from the client computer:

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 key "workgroup" 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.

Importing Users and Groups

You can use `dsimport` to import user and group accounts. into a folder. The `dsimport` tool permits logging at three levels with the `-l` switch. You can use the `dsimport` tool to import any number of records from a flexible text–delimited file. See the `dsimport` man page for more information.

See the Open Directory administration guide for a list of record types and attributes. 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/.

See "Creating a Character-Delimited User Import File" on page 120 for information about the formats of the files you can import.

```
$ dsimport (-g|-s|-p) file path (O|M|I|A) -u user -p password [options]
```

Parameter	Description
`-g|-s|-p`	You must 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 version 10.1.x `-p` for an XML file exported from AppleShare IP version 6.x
`file`	The path of the file to import.
`path`	The path to the Open Directory directory domain where the records will be added.
`O|M|I|A`	Specifies how user data is handled if a record for an imported user already 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 existing record unchanged. `A`: Append data from import record to existing record.
`user`	The name of the folder administrator.
`password`	The password of the folder 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 120.

Open Directory supports up to 200,000 records. For a local NetInfo directory, make sure the file contains no more than 10,000 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 will overwrite 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 or a NetInfo domain into a file. 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. Any line of 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 the Open Directory administration guide

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
```

The method for setting an imported user's password type to Open Directory requires that the imported data actually have a password value. If the password value is missing for a user, then the corresponding user record will be created with a password type of Crypt or Shadow Password.

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.

Using the 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:Adl47E$:408:20:A. Johnson, M.D.:/Network/Servers/somemac/Homes/anne:/
     bin/csh
```

Using the 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
```

Setting Permissions

To control access to your information, Mac OS X automatically 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 the 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 will be allowed into the folder.

Mac OS X provides distinct permissions for three 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

There are four levels of permission:

- Read & Write allows a user to open the item to see its contents and change it.
- Read Only allows a user to open the item to see its contents, but not change or copy the contents.
- Write Only 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 blocks all access to the item so that 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.

Some 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--
```

See the `ls` man page for more information about viewing permissions.

Setting the umask for Individual Users

The global umask setting determines the permissions of new 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 newly created files.
23 (octal equivalent 027)	User and members of the user's default group can read newly created 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 will not be readable by every other user on the computer, but will still be readable by members of his assigned group. The owner of the file or folder can still make it accessible to others by changing the permissions in the Finder's Get Info window or by using the `chmod` tool.

To set the NSUmask settings for all 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 NSUmask, but there is no guarantee that they will. Also, users can override their own NSUmask setting at any time. The changes to the umask settings take effect at next login.

> *Warning:* Setting permissions to group, or all, will allow any private, or confidential information in these folders to be visible to others. To prevent private files being accessed, the user should create a folder and restrict the permissions.

Changing Permissions

Use the `chmod` tool to change permissions for an item.

```
$ chmod securitygroup changetype permission fileorfolder
```

Parameter	Description
securitygroup	The person or group whose permission you are changing. Can be any of the following: • u - user • g - group • o - other • all - all
changetype	Type of change. Whether you are adding or subtracting the permission: • "+" - 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 write access permission for group and others from the file myfile:

```
$ chmod go-w myfile
```

To add read and write access permission for group and others to files myfile1 and myfile2:

```
$ chmod go+rw myfile1 myfile2
```

To add read, write, and execute permission for everyone to myfile1:

```
$ chmod ugo+rwx myfile1
```

See the `chmod` man page for more information.

Changing the Owner

Use the `chown` tool to change the owner of a file or folder.

```
$ 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
```

See the `chown` man page for more information.

Changing the Group

Use the `chgrp` tool to change the group of a file or folder.

```
$ 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
```

See the `chgrp` man page for more information.

Securing System Accounts

Security is very important when setting up and administering system accounts. The following sections cover security settings for user accounts.

Securing Initial System Accounts

Two accounts on the computer require attention before any further configuration is done. First, the permissions on the home folder of the initial administrator account should be changed. Second, any 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
```

where *adminname* is the name of the account. The 700 permission setting allows only the administrator to read and browse files in his 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 login, and then the `sudo` tool used to perform actions as root.

The computer uses a file called /etc/sudoers to determine which users have the authority to use the `sudo` program, and this file initially specifies that all accounts with administrator privileges may 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 /NetInfo/root/Users by entering the path at the prompt:

```
> cd /NetInfo/root/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 The following commands 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

The list of administrators allowed to use the `sudo` tool should be limited to only those administrators who require the ability to run commands as root.

To change the /etc/sudoers file:

1 Edit the /etc/sudoers file using the `visudo` tool, which allows for safe editing of the file. The command must be run as root:

```
$ sudo visudo
```

2 Enter the root password when prompted.

Note: 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. See the `sudo` and `sudoers` man pages for more information.

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 will mean that any time a new administrator is added to a system, that administrator must be added to the /etc/sudoers file as described above if that administrator requires the ability to use the `sudo` tool.

5 Save and quit `visudo`.

See the `vi` and `visudo` man pages for more information.

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, the Open Firmware settings should be altered. 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, enter the following command:

```
$ nvram security-mode="command"
```

In command mode, the computer will boot from the boot device specified in the computer's boot device variable and disallow users from providing any boot arguments.

To test that the computer has been put into command mode as recommended:

1 Close all applications and choose Restart from the Apple menu.

2 A confirmation window will pop up. Restart the computer by clicking the Restart button.

3 Hold down the key combination Command-S while the computer boots.

4 If the command mode has been set correctly, the computer will display the Mac OS X login window. Normally, holding down the Command-S key combination while starting up would cause the computer to start up in single-user mode.

5 If the computer did start 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; 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), the Open Firmware password will be disabled.

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. A strong password should be chosen; in this instance, a computer-generated random password would be a good choice. This password should be written down, and secured in the same location as the Master FileVault password. This password will not be needed except for situations where the computer must be booted from an alternate disk, such as if the startup disk fails or its file system is in need of repair.

4 To restart the computer and enable the settings, enter the command:

```
> reset-all
```

5 The computer should restart and display the login window.

Note: An Open Firmware password provides some protection, although it can be reset if a user has physical access to the computer and can change the physical memory configuration of the computer.

Setting Password Policy

Us the `pwpolicy` tool to adjust the password policies of your users. This tool can be used to view or set global password policies that force users to change passwords, limit the number and type of characters in a password, the length of time before passwords can be reused, and when passwords must be changed.

For secure passwords, you should require every password to have a minimum of 5 characters. You may use a higher number of characters if a more secure password is desired. It is also a good idea to have users change passwords frequently.

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	User name of the user whose password is changing.
newpassword	The 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 all users requiring:

- the password to 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 thirty days

Parameter	Description
ipaddress	Location of the LDAP directory.
adminusername	User name of an administrator.
minChars	Minimum number of characters in the password.
usingHistory	Sets the number of previous passwords that the user is not allowed to 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 an individual user to 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	The administrator password (omit this to prompt for the password)
usertochange	User name of the user whose password is changing.
newPasswordRequired	Set to 1 to prompt the user to enter a new password.

See the `pwpolicy` man page for more information.

Finding User Account Information

The `lookupd` daemon acts as an information broker and cache. It is called by various routines in the system framework to find information about user accounts, groups, printers, email aliases and distribution lists, computer names, Internet addresses, and several other kinds of information. You can use it interactively to find out user account information.

To query for a user by name:

```
$ lookupd -d
> userWithName: admin
```

To see a list of all the different commands that run with lookupd:

```
$ lookupd -d
>?
```

To get a description of a specific command that you can run with lookupd:

Access the help prompt and enter the command name.

```
$ lookupd -d
>help
help> [command]
```

See the `lookupd` man page for more information.

Working with File Services

9

In this chapter you will find commands you can use to create share points and manage 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 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/Common Internet File System (SMB/CIFS) for Windows, as well as WebDAV and FTP for Internet clients. This chapter covers the commands that are used to configure and manage these file services.

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.

You can use the `sharing` tool to list, create, and modify share points. See the `sharing` man page for more information.

Listing Share Points

To list existing share points:

```
$ sharing -l
```

In the resulting list, there's 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:

```
$ 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 this 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 separate from 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/CIFS clients.
shareflags	A three-digit binary number indicating which protocols are used to share the folder. The digits represent, from left to right, AFP, FTP, and SMB/CIFS. 1=shared, 0=not shared.

Parameter	Description
guestflags	A group of three 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/CIFS. `1`=guests allowed, `0`=guests not allowed.
inheritflags	A group of two flags indicating whether new items in AFP or SMB/CIFS 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/CIFS. `1`=inherit, `0`=don't inherit.
creationmask	The SMB/CIFS creation mask. Default=`0644`.
directorymask	The SMB/CIFS folder mask. Default=`0755`.
oplockflag	Specifies whether opportunistic locking is allowed for an SMB/CIFS share point. 1=enable oplocks, 0=disable oplocks. For more information about oplocks, see the file services administration guide.
strictlockingflag	Specifies whether strict locking is used on an SMB/CIFS share point. `1`=enable strict locking, `0`=disable. For more information about strict locking, see the file services administration guide.

To create a share point that uses AFP, FTP, and SMB/CIFS protocols:

Enter the following command, replacing *100GB* with the name of the volume containing the share point and *Archive* with the actual share point name:

```
$ 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 actual share point name so that it appears as WinDocs for server management purposes, and Documents for SMB/CIFS file service users:

```
$ sharing -a /Volumes/100GB/Windows\ Docs -n WinDocs -S Documents -s 001
     -o 1
```

This share point is shared using only the SMB/CIFS protocol with oplocks enabled.

Modifying a Share Point

To change share point settings:

```
$ 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 under "Creating a Share Point" on page 134.

Disabling a Share Point

To disable a share point:

```
$ sharing -r sharepointname
```

Parameter	Description
sharepointname	The current name of the share point.

Managing the AFP Service

Apple Filing Protocol (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 shared disks don't unmount 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
```

Checking AFP Service Status

To see if AFP service is running:

```
$ sudo serveradmin status afp
```

To see complete AFP status:

```
$ sudo serveradmin fullstatus afp
```

Viewing AFP Settings

To list all AFP service settings:

```
$ sudo serveradmin settings afp
```

To list a particular setting:

```
$ sudo serveradmin settings afp setting
```

Parameter	Description
setting	Any of the AFP service settings. For a complete list of settings, enter `$ sudo serveradmin settings afp` or see "List of AFP Settings" on page 137.

To list a group of settings:

You can list a group of settings that have part of their names in common by typing only as much of the name as you want, stopping at a colon (`:`), and typing an asterisk (`*`) as a wildcard for the remaining parts of the name. For example:

```
$ sudo serveradmin settings afp:loggingAttributes:*
```

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 "List of AFP Settings" on page 137.
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
```

List of 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 all volumes. Default = `yes`
`adminGetsSp`	Set to yes to force administrator users on Mac OS 9 to see share points instead of all 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 an administrator user to masquerade as another user. Default = yes
authenticationMode	Authentication mode. Can be: standard kerberos standard_and_kerberos Default = "standard_and_kerberos"
autoRestart	Whether the AFP service should 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	The location of the error log. Default = /Library/Logs/AppleFileService/AppleFileServiceError.log
errorLogSize	Rollover size (in kilobytes) for the error log. Used 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	The 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	The login greeting message. Default = ""
loginGreetingTime	The last time the login greeting was set or updated.
maxConnections	Maximum number of simultaneous user sessions allowed by the server. Default = -1 (unlimited)
maxGuests	Maximum number of simultaneous guest users allowed. Default = -1 (unlimited)
maxThreads	Maximum number of 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.

List of 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. See the examples in the following sections for details on how to use these commands.

Command (afp:command=)	Description
cancelDisconnect	Cancel a pending user disconnect. See "Canceling a User Disconnect" on page 143.
disconnectUsers	Disconnect AFP users. See "Disconnecting AFP Users" on page 142.
getConnectedUsers	List settings for connected users. See "Listing Connected Users" on this page.
getHistory	View a periodic record of file data throughput or number of user connections. See "Listing AFP Service Statistics" on page 144.
getLogPaths	Display the locations of the AFP service activity and error logs.

Command (afp:command=)	Description
sendMessage	Send a text message to connected AFP users. See "Sending a Message to AFP Users" on page 142.
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 48.

Listing Connected Users

You can use the getConnectedUsers command with the serveradmin tool to retrieve information about connected AFP users. In particular, you can use this command to retrieve the session IDs you need to disconnect or send messages to users.

To list connected users:

```
$ sudo serveradmin command afp:command = getConnectedUsers
```

The computer will respond with the following array of 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 particular disconnect. This will appear once a disconnect has been issued.
<flags>	Indicates the type of user. 1-session belongs to the administrator 2-session belongs to a guest 4-session is sleeping
<ipAddress>	The user's IP address.
<lastUseElapsed>	Time since the command was last run.
<login-elapsed-time>	The elapsed time since the user connected.
<minsToDisconnect>	The number of minutes between the time the command is issued and the user is disconnected
<name>	The user's name.

Value returned by getConnectedUsers (afp:usersArray:_array_index:*n*:)	Description
<serviceType>	The share point the user is accessing.
<sessionID>	An integer that identifies the user session.
<state>	State of the service.

Sending a Message to AFP Users

You can use the sendMessage command with the serveradmin tool to send a text message to connected AFP users. 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	The message that appears on client computers.
sessionidn	The session ID of a user you want to receive the message. To list the session IDs of connected users, use the getConnectedUsers command. See "Listing Connected Users" on page 141.

Disconnecting AFP Users

You can use the disconnectUsers command with the serveradmin tool to disconnect AFP users. Users are specified by session ID. You can specify a delay time before disconnect and 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 text of a 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 "Listing Connected Users" on page 141.

The computer will repond 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 particular disconnect. You can use this ID with the `cancelDisconnect` command to cancel the disconnect.
`<user listing>`	A standard array of user settings for each user scheduled for disconnect. For a description of these settings, see "Listing Connected Users" on page 141.
`<status>`	A command status code. `0` = command successful.

Canceling a User Disconnect

You can use the `cancelDisconnect` command with the `serveradmin` tool to cancel a `disconnectUsers` command. 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 any user scheduled to be disconnected and looking at the value of the `disconnectID` setting for the user.

The computer will respond with the following output:

```
afp:command = "cancelDisconnect"
afp:timeStamp = "<time>"
afp:status = <status>
```

Value	Description
`<time>`	The time at which the command was executed.
`<status>`	A command status code: `0` = command successful

Listing AFP Service Statistics

You can use the `serveradmin getHistory` command to display a log of periodic samples of the number of connections and the data throughput. Samples are taken once each minute.

To list 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, for which you want to see samples. For example, to see 30 minutes of data, you would specify `afp:timeScale = 1800`.

The computer will respond 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 integer average number of users. For throughput, (v2), this is integer bytes per second.
`<time>`	The time at which the sample was measured. A standard UNIX time (number of seconds since Sep 1, 1970). Samples are taken every 60 seconds.

Viewing AFP Log Files

You can use `tail` or any other file listing tool to view the contents of the AFP service logs.

To view the latest entries in a log:

```
$ tail log-file
```

You can use the `getLogPaths` command with the `serveradmin` tool to see where the current AFP error and activity logs are located.

To display the log paths:

```
$ sudo serveradmin command afp:command = getLogPaths
```

The computer will respond 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`

Managing the NFS Service

Network File System (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 is started automatically 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.

Checking NFS Service Status

To see if NFS service and related processes are running:

```
$ sudo serveradmin status nfs
```

To see complete NFS status:

```
$ sudo serveradmin fullstatus nfs
```

Viewing NFS Service Settings

To list all NFS service settings:

```
$ sudo serveradmin settings nfs
```

To list a particular setting:

```
$ sudo serveradmin settings nfs:setting
```

Changing NFS Service Settings

Use the following parameters with the `serveradmin` tool to change settings for the NFS service.

Parameter (`nfs:`)	Description
`nbDaemons`	To reduce the number of daemons, you must restart the server after changing this value. Default = `6`
`useTCP`	You must restart the server after changing this value. Default = `yes`
`useUDP`	You must restart the server after changing this value. Default = `yes`

Managing the FTP Service

Mac OS X Server features a robust File Transfer Protocol (FTP) file service for Internet file sharing from any platform. The FTP protocol provides the broadest compatibility across platforms, making it ideal for anonymous downloads or sharing files that are too large to be sent over email. 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 FTP service:

```
$ sudo serveradmin start ftp
```

Stopping FTP Service

To stop FTP service:

```
$ sudo serveradmin stop ftp
```

Checking FTP Service Status

To see if FTP service is running:

```
$ sudo serveradmin status ftp
```

To see complete FTP status:

```
$ sudo serveradmin fullstatus ftp
```

Viewing FTP Service Settings

To list all FTP service settings:

```
$ sudo serveradmin settings ftp
```

To list a particular setting:

```
$ sudo serveradmin settings ftp:setting
```

To list a group of settings:

You can list a group of settings that have part of their names in common by typing only as much of the name as you want, stopping at a colon (`:`), and typing an asterisk (`*`) as a wildcard for the remaining parts of the name. For example:

```
$ sudo serveradmin settings ftp:logCommands:*
```

Changing FTP Service Settings

You can change FTP service settings using 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 "List of FTP Service Settings" on this page.
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
```

List of FTP Service Settings

Use the following parameters with the `serveradmin` tool to change settings for the FTP service.

Parameter (`ftp:`)	Description
`administratorEmailAddress`	Sets the administrator email address. Default = `"user@hostname"`
`anonymous-root`	Sets the anonymous root directory. Default = `"/Library/FTPServer/FTPRoot"`
`anonymousAccessPermitted`	To allow anonymous access to the FTP 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 prompted to log in to the 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 in which 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 the 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.`
	`-------------------------------------"`

List of FTP serveradmin Commands

You can use the following commands with the `serveradmin` tool to manage FTP service. See the examples in the following sections for details on how to use these commands.

Command (`ftp:command=`)	Description
`getConnectedUsers`	List connected users. See "Checking for Connected FTP Users" on page 150.
`getLogPaths`	Show location of the FTP transfer log file. See "Viewing the FTP Transfer Log" on page 150.
`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 48.

Viewing the FTP Transfer Log

You can use `tail` or any other 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. You can use the `serveradmin getLogPaths` command to see where the current transfer log is located.

To display the log path:

```
$ sudo serveradmin command ftp:command = getLogPaths
```

Checking for Connected FTP Users

To see how many FTP users are connected:

```
$ ftpcount
```

or

```
$ sudo serveradmin command ftp:command = getConnectedUsers
```

Managing the SMB/CIFS Service

Mac OS X Server offers integration of Samba 3, a popular open-source project that delivers high-performance SMB/CIFS 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) just like a Windows server. This enables Windows clients to browse folders and share files without having to install additional software.

Starting and Stopping SMB/CIFS Service

To start SMB/CIFS service:

```
$ sudo serveradmin start smb
```

To stop SMB/CIFS service:

```
$ sudo serveradmin stop smb
```

Checking SMB/CIFS Service Status

To see if SMB/CIFS service is running:

```
$ sudo serveradmin status smb
```

To see complete SMB/CIFS status:

```
$ sudo serveradmin fullstatus smb
```

Viewing SMB/CIFS Service Settings

To list all SMB/CIFS service settings:

```
$ sudo serveradmin settings smb
```

To list a particular setting:

```
$ sudo serveradmin settings smb:setting
```

Parameter	Description
setting	An SMB/CIFS service setting. To see a list of available settings, enter `$ sudo serveradmin settings smb` or see "List of SMB/CIFS Service Settings" on page 152.

To list a group of settings:

You can list a group of settings that have part of their names in common by typing only as much of the name as you want, stopping at a colon (`:`), and typing an asterisk (`*`) as a wildcard for the remaining parts of the name. For example:

```
$ sudo serveradmin settings smb:adminCommands:*
```

Changing SMB/CIFS Service Settings

You can change SMB/CIFS service settings using the `serveradmin` tool.

To change a setting:

```
$ sudo serveradmin settings smb:setting = value
```

Parameter	Description
setting	An SMB/CIFS service setting. To see a list of available settings, enter `$ sudo serveradmin settings smb` or see "List of SMB/CIFS Service Settings" on page 152.
value	An appropriate value for the setting. For a list of values that correspond to GUI controls in the Server Admin application, see "List of SMB/CIFS Service Settings" on page 152.

To change several settings:

```
$ sudo serveradmin settings
smb:setting = value
smb:setting = value
smb:setting = value
[...]
Control-D
```

List of SMB/CIFS Service Settings

Use the following parameters with the `serveradmin` tool to change settings for the SMB/CIFS service.

Parameter (`smb:`)	Description
`adminCommands:homes`	Whether home folders are mounted automatically when Windows users log in so you don't have to set up individual 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: `CP437` (Latin US) `CP737` (Greek) `CP775` (Baltic) `CP850` (Latin1) `CP852` (Latin2) `CP861` (Icelandic) `CP866` (Cyrillic) `CP932` (Japanese SJIS) `CP936` (Simplified Chinese) `CP949` (Korean Hangul) `CP950` (Traditional Chinese) `CP1251` (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/CIFS 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.

List of SMB/CIFS serveradmin Commands

You can use these commands with the `serveradmin` tool to manage SMB/CIFS service. See the examples in the following sections for details on how to use these commands.

Command (`smb:command=`)	Description
`disconnectUsers`	Disconnect SMB/CIFS users. See "Disconnecting SMB/CIFS Users" on page 156.
`getConnectedUsers`	List users currently connected to an SMB/CIFS service. See "Listing SMB/CIFS Users" on page 155.
`getHistory`	List connection statistics. See "Listing SMB/CIFS Service Statistics" on page 156.
`getLogPaths`	Show location of service log files. See "Viewing SMB/CIFS Service Logs" on page 157.
`syncPrefs`	Update the service to recognize changes in share points. See "Updating Share Point Information" on page 157.
`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 48.

Listing SMB/CIFS Users

You can use the `serveradmin getConnectedUsers` command to retrieve information about connected SMB/CIFS users. For example, you can use this command to retrieve the session IDs you need in order to disconnect users.

To list connected users:

```
$ sudo serveradmin command smb:command = getConnectedUsers
```

The computer will respond with the folowing array of settings displayed 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/CIFS Users

You can use the `serveradmin disconnectUsers` command to disconnect SMB/CIFS users. 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 "Listing SMB/CIFS Users" on page 155.

The computer will respond with the following output:

```
smb:command = "disconnectUsers"
smb:status = <status>
```

Value	Description
`<status>`	A command status code. 0 = command successful

Listing SMB/CIFS Service Statistics

You can use the `smbstatus` command to display a list of the number of SMB/CIFS connections.

To list SMB/CIFS connections:

```
$ sudo 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 2006
No Locked Files
```

Updating Share Point Information

After you make a change to an SMB/CIFS share point using the `sharing` tool, you need to update the SMB/CIFS service information.

To update SMB/CIFS share point information:

```
$ sudo serveradmin command smb:command = syncPrefs
```

Viewing SMB/CIFS Service Logs

You can use `tail` or any other file-listing tool to view the contents of the SMB/CIFS service logs.

To view the latest entries in a log:

```
$ tail log-file
```

You can use the `serveradmin getLogPaths` command to see where the current SMB/CIFS logs are located.

To display the log paths:

```
$ sudo serveradmin command smb:command = getLogPaths
```

The computer will respond 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 (ACL). 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 throughout 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 the 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, review the Overview chapter of the file services administration guide.

Using chmod to Modify ACLs

Using `chmod`, you can add and delete ACEs for a file or a folder. Here are a few of the parameters to 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 some of the common permissions you can assign to files:

Permission	Description
`delete`	Grants permission to delete the item
`readattr`	Read an object's basic attributes
`read`	Read the object
`write`	Write to the object
`writeattr`	Write an object's basic attributes
`readextattr`	Read extended attributes
`writeextattr`	Write extended attributes
`readsecurity`	Read an object's extended security information (ACL)
`writesecurity`	Write an object's security information (ACL)
`chown`	Change an object's ownership

The following are the permissions applicable to folders:

Permission	Description
`list`	List entries
`add_file`	Add a file
`add_sudirectory`	Add a subfolder
`delete_child`	Delete 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
1: guest deny read
2: user1 allow write
```

See the `chmod` man page for more information.

Working with the Print Service

10

In this chapter you will find commands you can use to configure and manage the print service.

The print service in Mac OS X Server lets you share network and direct-connect printers among clients on your network. The print service also includes support for managing print queues, monitoring print jobs, extensive logging, and using print quotas. This chapter covers the commands needed to view, modify, or change the print service settings.

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 either a Mac OS X application or the command line. CUPS allows Mac OS X to support all the 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 both the 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 directly 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

You can use the lpadmin tool, or the CUPS web interface, to add a print queue. 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.

Once the file is copied to /var/spool/cups, cupsd begins the process of preparing the file for printing.

For more information about CUPS and tools specific to CUPS, review the documentation available at: www.cups.org/documentation.php. You can also see the man pages for the following CUPS commands: accept, backend, cancel, disable, enable, filter, lp, lpadmin, lpinfo, lpoptions, lpq, lpr, lpstat, and reject.

Performing Print Service Tasks

Use the serveradmin tool in conjunction with commands that interact with CUPS to perform print service tasks.

Starting and Stopping Print Service

To start print service:

```
$ sudo serveradmin start print
```

To stop print service:

```
$ sudo serveradmin stop print
```

Checking 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 list print service configuration settings:

```
$ sudo serveradmin settings print
```

To list a particular setting:

```
$ sudo serveradmin settings print:setting
```

To list a group of settings:

You can list a group of settings that have part of their names in common by typing only as much of the name as you want, stopping at a colon (:), and typing 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`).

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 "Print Service Settings" on page 164.
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
```

Print Service Settings

Use the following parameters with the serveradmin tool to change settings for the print service.

Parameter (print:)	Description
serverLogArchiveEnable	Default = no; yes enforces log size limits
<queue arrays>	See "Queue Data Array" on page 165
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 = supress 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)

As well as 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 existing print queue. The array is a set of parameters that define values for each queue. The array of sharing services has been expanded to include IPP. This is the same service as Mac OS X version 10.3 printer sharing, now integrated with Mac OS X Server version 10.4.

Many of the following parameters are CUPS parameters. You can get more details about the CUPS parameters in 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 Internet Printing Protocol (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/CIFS.
`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 the Print Service

Use the `serveradmin` tool in conjunction with the following commands that interact with CUPS to modify and manage the print service.

Command (`print:command=`)	Description
`getJobs`	List information about the jobs waiting 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 167.
`getLogPaths`	Finding the locations of the print service and job logs. See "Viewing Print Service Log Files" on page 169.
`getQueues`	List print service queues. See "Listing Queues" on page 167.
`setJobState`	Hold or release a job. The name required for this command is the *sharing* name given to the queue by the administrator as previously described. See "Holding a Job" on page 168.
`setQueueState`	Pauses or release 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 a Queue" on this page.
`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 48.

Listing Queues

You can use the `serveradmin getQueues` command to list print service queues.

```
$ sudo serveradmin command print:command = getQueues
```

Pausing 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 167.

To release the queue:

```
$ sudo serveradmin command
print:command = setQueueState
print:state = RESUMED
print:namesArray:_array_index:0 = queue
Control-D
```

Listing Jobs and Job Information

You can use the `serveradmin getJobs` command to list information about print jobs.

```
$ 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 167.

For each job, the command lists:

- Document name
- Document size
- Job ID
- Submitting user
- Submitting host
- Job name
- Job state
- Job priority

Holding a Job

You can use the serveradmin setJobState command to hold or release a job.

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 167.
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 167.

To release the job for printing, change its state to PENDING.

To release the 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

You can use `tail` or any other file-listing tool to view the contents of the print service logs.

To view the latest entries in a log:

```
$ tail log-file
```

The following are the log files for the 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)

You can use the `serveradmin getLogPaths` command to see where the current logs are located.

To display the 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

In this chapter you will find commands you can use to configure and manage the NetBoot Service and system images.

NetBoot is used to host a standard operating system and application configuration on all of the clients in a network from the server.This chapter describes the commands used to configure and manage the NetBoot service.

Understanding the NetBoot Service

The 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 drive. This allows you to create a standard configuration and use it on all of the desktop computers on a network—or host multiple images customized for different workgroups.

You can also create server configurations and run all of your servers from one image. Updating the disk image on the NetBoot server updates all of these computers automatically the next time 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 NetBoot service:

```
$ sudo serveradmin start netboot
```

If you get the following response:

```
$ netboot:state = "STOPPED"
$ netboot:status = 5000
```

you have not yet enabled NetBoot on any network port.

To stop NetBoot service:

```
$ sudo serveradmin stop netboot
```

Checking NetBoot Service Status

To see if NetBoot service is running:

```
$ sudo serveradmin status netboot
```

To see complete NetBoot status:

```
$ sudo serveradmin fullstatus netboot
```

Viewing NetBoot Settings

To list all NetBoot service settings:

```
$ sudo serveradmin settings netboot
```

Changing NetBoot Settings

You can change NetBoot service settings using the `serveradmin` tool.

To change a NetBoot 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. Use the following parameters with the `serveradmin` tool to change settings for the NetBoot service.

Parameter (`netboot:`)	Description
`filterEnabled`	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 "Storage Record Array" on page 173.
`netBootFiltersRecordsArray...`	An array of values for each computer explicitly allowed or disallowed access to images. For a description, see "Filters Record Array" on page 174.
`netBootImagesRecordsArray...`	An array of values for each boot or installation image stored on the server. For a description, see "Image Record Array" on page 174.
`netBootPortsRecordsArray...`	An array of values for each server network port used to deliver boot or installation images. For a description, see "Port Record Array" on page 175.

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"`

Filters Record Array

An array of the following values appears in the 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.

Image Record Array

An array of the following values appears in the 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	
netBootImagesRecordsArray: _array_index:<n>:BootFile	Name of 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 the 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.

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), to use NetBoot, you need to enable NetBoot 1.0. You may do so by using the `nicl` tool.

To enable NetBoot:

```
$ sudo nicl . 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`.

Note: NetBoot 1.0 and 2.0 can run on the same network interface simultaneously.

Working with System Images

A boot image is a file that looks and acts like a mountable disk or volume. NetBoot boot images contain the system software needed to act as a startup disk for client computers across 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 just 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. Using NetBoot with Mac OS X client management services, you can provide a personalized work environment for each client computer user.

Updating an Image

You can use the `installer` tool to update a package from the command line, the same way you would install new packages on your default installation volume.

To update an image:

```
$ installer -pkg pkg.mpkg -target image_path
```

Booting from an Image

You can set the `nvram` environment variables to boot from an image. You can do so using the `nvram` tool, or by booting into open firmware mode.

To boot from an image:

1. Boot into open firmware by clicking (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 to Work with System Images

You can use the `hdiutil` tool to manipulate disk images. This tool is used to perform many functions, such as creating, compressing, mounting, unmouting, and resizing images. You can also display image information and burn images onto CDs. See the `hdiutil` man page for information about how to manipulate disk images.

The following examples provide some basic `hdiutil` tool functions:

To verify an image against 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-R export image with a .toast extention:

```
$ hdiutil convert master.dmg -format UDTO -o master
```

To burn an image onto the CD drive:

```
$ hdiutil burn myImage.dmg
```

To create an image from a folder:

```
$ hdiutil create -srcfolder mydir mydir.dmg
```

Using asr to Restore System Images

The `asr` tool can efficiently copy disk images onto volumes. `asr` can also accurately clone volumes.

To clone a volume:

```
$ sudo asr -source /Volumes/Classic -target /Volumes/install
```

To restore an system image onto a volume:

```
$ sudo asr -source compressedimage -target <targetvol> -erase
```

Note: The target drive will be erased.

Imaging Multiple Clients Using Multicast asr

You can enable a multicast image server using Mac OS X Server. Multicast `asr` can restore multiple clients simultaneously from one looping multicast of an asr disk image. Each client can start receiving 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 has received 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. Use the `asr` tool with the `-server` flag and a correctly built image and plist to enable the image server.

To start up a multicast server for a specified image:

```
$ asr -source <compressedimage> -server <configuration.plist>
```

where the specified image used the parameters in the configuration.plist file. The image will not start multicasting on the network until a client attempts to start a restore. The server will continue 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 will receive the multicast stream from `<hostname>` and save it to a client. Add `-erase` to overwrite any existing image. Passing `-erase` with `-target` indicates any existing image should be overwritten when doing a multicast.

Choosing a Boot Device Using systemsetup

You can use the `systemsetup` tool to choose your boot device. When setting the startup disk, you simply have to know the full path to core services. For example, to boot from "Disk 2," which is now mounted in /Volumes, you would enter:

```
$ systemsetup -setstartupdisk /Volumes/Disk\ 2/System/Library/CoreServices
```

Working with the Mail Service

12

In this chapter you will find commands you can use to manage the mail service.

Mac OS X Server provides a full complement of tools for setting up and managing email service for your users. You can use the commands described in this chapter to control the individual components that make up the mail service.

Understanding the Mail Service

The Mail service in Mac OS X Server consists of three 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

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 against 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 as root, or run as any other user.

Postfix uses the configuration file main.cf in /etc/postfix. Whenever Server Admin modifies Postfix settings, it overwrites the main.cf file. If you want to make a manual change to the configuration file of Postfix, be aware that Server Admin will overwrite 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. See www.postfix.org for more information about postfix.

Cyrus

Cyrus was developed at Carnegie Mellon University with the purpose of creating a highly scalable enterprise mail system for use in small- to large-enterprise environments. The 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 the user folders are located in /var/spool/imap/.

In brief, Cyrus works as follows: The Cyrus deliver application will receive mail from the Postfix delivery agent, update the mailboxes database located at /var/imap/mailboxes.db, and store the mail in the users spool files located at /var/spool/imap/*username/folder*. The user will then be able to use the IMAP or POP protocol to retrieve messages.

See asg.web.cmu.edu/cyrus/ for more information about 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 it 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 the Mail Service

Mac OS X Server ships with some powerful tools to help administer you 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 list mail service configuration settings:

```
$ sudo serveradmin settings mail
```

To list a particular setting:

```
$ sudo serveradmin settings mail:setting
```

To list a group of settings:

You can list a group of settings that have part of their names in common by typing only as much of the name as you want, stopping at a colon (`:`), and typing 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, if you want to work with the mail service from the command line, you'll probably find it more straightforward to work directly with the underlying Postfix and Cyrus agents.

For more information about these agents:

- See www.postfix.org for information about Postfix.
- See asg.web.cmu.edu/cyrus for information about Cyrus IMAP/POP.

Mail Service Settings

Use the following parameters with the serveradmin tool to change settings for the 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!@%-_=+:,./abcdefghijklmnopqrstuvwxyzABCDEFGHIJKLMNOPQRSTUVWXYZ"
postfix:smtpd_expansion_filter	Default = "\t\40!"#$%&'()*+,-./0123456789:;<=>?@ABCDEFGHIJKLMNOPQRSTUVWXYZ[\\]^_`abcdefghijklmnopqrstuvwxyz{|}~"
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 "Listing Mail Service Statistics" on this page.
`getLogPaths`	Display the locations of the Mail service logs. See "Viewing the Mail Service Logs" on page 195.
`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 48.

Listing 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 list 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 at which the sample was measured. A standard UNIX time (number of seconds since Sep 1, 1970). Samples are taken every 60 seconds.

Viewing the Mail Service Logs

You can use `tail` or any other file-listing tool to view the contents of the mail service logs.

To view the latest entries in a log:

```
$ tail log-file
```

You can use the `serveradmin getLogPaths` command to see where the mail service logs are located.

To display the log locations:

```
$ sudo serveradmin command mail:command = getLogPaths
```

The computer responds with the following output:

```
mail:Server Log = <server-log>
mail:Lists qrunner = <lists-log>
mail:Lists post = <postings-log>
mail:Lists smtp = <delivery-log>
mail:Lists subscribe = <subscriptions-log>
mail:SMTP Log = <smtp-log>
mail:POP Log = <pop-log>
mail:Lists error = <listerrors-log>
mail:IMAP Log = <imap-log>
mail:Lists smtp-failure = <failures-log>
```

Value	Description
`<server-log>`	The location of the server log. Default = `srvr.log`
`<lists-log>`	The location of the Mailing Lists log. Default = `/private/var/mailman/logs/qrunner`

Value	Description
`<postings-log>`	The location of the Mailing Lists Postings log. Default = `/private/var/mailman/logs/post`
`<delivery-log>`	The location of the Mailing Lists Delivery log. Default = `/private/var/mailman/logs/smtp`
`<subscriptions-log>`	The location of the Mailing Lists Subscriptions log. Default = `/private/var/mailman/logs/subscribe`
`<smtp-log>`	The location of the server log. Default = `smtp.log`
`<pop-log>`	The location of the server log. Default = `pop3.log`
`<listerrors-log>`	The location of the Mailing Lists Error log. Default = `/private/var/mailman/logs/error`
`<imap-log>`	The location of the server log. Default = `imap.log`
`<failures-log>`	The location of the Mailing Lists Delivery Failures log. Default = `/private/var/mailman/logs/smtp-failure`

Backing Up the Mail Files

When talking about mail-related backup, IMAP mailboxes are the first thing that come to mind. Aside from the IMAP folders, you might want to back up the configuration files for both Cyrus and Postfix. The value of backing up the configuration files is clear: it will save you time should you have to reconfigure your server after it powers down unexpectedly. The Server Admin tearoff sheets include configuration information and can thus be backed up instead of the separate configuration files, unless you have manually modified the 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 may lead to double delivery of emails to the users.

To back up the mail database, you need to stop the mail service first. You then copy the following files and folders onto 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. 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 that is 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.

Reconstructing the Mail Database

The `reconstruct` tool can be used to recover from corruption in mailbox folders. If `reconstruct` can find existing header and index files, it attempts to preserve any data in them that can't be derived from the message files. `reconstruct` attempts to preserve a mail database state that includes the flag names, flag state, and internal date. All other information is derived from the message files. An administrator may recover from a damaged disk by restoring message files from a backup and then running reconstruct to regenerate what it can of the other files.

The mailboxes file, /var/imap/mailboxes.db, is the most critical file in the entire Cyrus IMAP system. It contains a sorted list of each mailbox on the server, along with the mailboxes quota root and Access Control List (ACL). To reconstruct a corrupted mailbox file, run the `reconstruct -m` command. The `reconstruct` tool, when invoked with the `-m` switch, scavenges and corrects whatever data it can find in the existing mailboxes file. It then scans all partitions listed in the imapd.conf file for additional mailbox folders to put in the mailboxes file.

The cyrus.header file in each mailbox folder stores a redundant copy of the mailbox ACL, to be used as a backup when rebuilding the mailboxes file.

Read the documentation pages at asg.web.cmu.edu/cyrus/download/imapd/overview.html for more information about the Cyrus backup.

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:

- Generate a Certificate Signing Request (CSR) and create a keychain.
- Obtain an SSL certificate from an issuing authority.
- Import the SSL certificate into the keychain.
- Create a password file.

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.

1 Log in to the server as root.

2 In the Terminal application, enter the following two 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.

5 Enter `r` when prompted to select a key algorithm, and then press Return.

```
Please specify parameters for the key pair you will generate.
  r   RSA
  d   DSA
  f   FEE
Select key algorithm by letter:
```

6 Enter a key size at the next prompt, and then press Return.

```
Valid key sizes for RSA are 512..2048; default is 512
Enter key size in bits or CR for default:
```

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.

7 Enter `y` when prompted to confirm the algorithm and key size, and then press Return.

```
You have selected algorithm RSA, key size (size entered above) bits.
OK (y/anything)?
```

8 Enter `b` when prompted to specify how this certificate will be used, and then press Return.

```
Enter cert/key usage (s=signing, b=signing AND encrypting):
```

9 Enter `s` when prompted to select a signature algorithm, and then press Return.

```
...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:
```

10 Enter `y` when asked to confirm the selected algorithm, and then press Return.

```
You have selected algorithm RSA with SHA1.
OK (y/anything)?
```

11 Enter a phrase or some random text when prompted to enter a challenge string, and then press Return.

```
...creating CSR...
Enter challenge string:
```

12 Enter the correct information at the next five prompts, which request the various components of the certificate's Relative Distinguished Name (RDN). Press Return after each entry.

```
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.
```

13 Enter `y` when asked to confirm the information you entered, and then press Return.

```
Is this OK (y/anything)?
```

When you see a message about writing to csr.txt, you have successfully generated a CSR and created the keychain that mail service needs for SSL connections.

```
Wrote (n) bytes of CSR to csr.txt
```

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 configuration of 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: Enter `cd /private/var/root/Desktop` and press Return if the certificate file is saved on the desktop of the root user.

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.
```

Accessing the 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 list the certificates stored in the System keychain:

```
$ certadmin list
```

By default, `certadmin` will print the "Common Name" field of each certificate separated by newlines. Adding the option -x or --xml will print the certificate list to screen as an xml property list (plist).

To export the given certificate to OpenSSL:

```
$ certadmin export
```

See the `certadmin` man page for more information.

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 will automatically use the password file to unlock the keychain that contains the SSL certificate. The 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 new file and enter the password exactly as you entered it when you created the keychain.

   Don't press Return after typing 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, then 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 typing the following two commands, pressing Return after each one:

   ```
   cd /private/var/root/Library/Keychains/
   chmod 600 certkc.pass
   ```

   Mac OS X Server mail service can now use SSL for secure IMAP connections.
7. Log out as root.

Note: If the mail service is running, you need to stop it and start it again to make it recognize the new certificate keychain.

Configuring Mailboxes

The mail service keeps track of incoming email messages with a small database (BerkeleyDB 4.2.52), but the database doesn't contain the messages themselves. The 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 any or all of the 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 services, 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. `cyradm` can be used to create, delete, or rename mailboxes, as well as set ACLs for mailboxes (for email 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.
- All spaces in file or folder names must be escaped with a backslash (\), just as you would in a shell.

See the `cyradm` man page for a complete list of commands.

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.

The Sieve acts much like the rules in various email programs, to sort or process mail based on user-defined criteria. In fact, some email clients use Sieve for client-side email 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"), web mail plug-ins ("avelsieve"), and even some email clients.

Enabling Sieve Support

In order 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 email 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 some common scripts that 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 on the Web at www.ietf.org/rfc/rfc3028.txt?number=3028. Other information about Sieve and a sample script archive can be found at www.cyrusoft.com/sieve.

Working with Web Technologies

13

In this chapter you will find commands you can use to configure and manage web services and web components of 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.

Understanding Web Technology

Apple's web services are 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 two 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, and reliable software package that has been used worldwide for many years. In this chapter, references to the Apache server refer to this version.
- Version 2.0—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 the default 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/conf`
Default documents	`/Library/WebServer/Documents`
Log files	`/var/log/httpd`
Loadable modules	`/usr/libexec/httpd`

Apache web server version 2.0 files are in the /opt/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 directly. Each section of the httpd.conf file contains detailed instructions for how to safely edit its options.

Important: Do not modify the httpd.conf file manually when the Web Settings pane of Server Admin is open, to avoid misconfiguring your web services. For more information about apache, see www.apache.org.

Managing the Web Service

Web service in Mac OS X Server is based on Apache, an open-source HTTP web server. A web server responds to requests for HTML web pages stored on your site. The following sections describe some basic 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

You can use `serveradmin` to view 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.

To list all Web service settings:

```
$ sudo serveradmin settings web
```

To list a particular setting:

```
$ sudo serveradmin settings web:setting
```

To list a group of settings:

You can list a group of settings that have part of their names in common by typing only as much of the name as you want, stopping at a colon (:), and typing an asterisk (*) as a wildcard for the remaining parts of the name. For example:

```
$ sudo serveradmin settings web:IFModule:_array_id:mod_alias.c:*
```

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.

serveradmin and Apache Settings

The parameters are written differently in the Apache configuration file than they are 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 paramters 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

You can change web service settings using the `serveradmin` tool.

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

You can use the following commands with the `serveradmin` tool to manage web service.

Command (`web:command=`)	Description
`getHistory`	View Web service statistics. See "Viewing Service Statistics" on page 210.
`getLogPaths`	Finding the access and error logs for each hosted site. See "Viewing Service Logs" on this page.
`getSites`	Listing existing sites. See "Listing Hosted Sites" on this page.

Listing Hosted Sites

You can use the `serveradmin getSites` command to display a list of the sites hosted by the server, along with basic settings and status.

To list sites:

```
$ sudo serveradmin command web:command = getSites
```

You can also list the sites using Apache, with the following command:

```
$ httpd -S
```

Viewing Service Logs

You can use `tail` or any other file listing tool to view the contents of web service access and error logs for each site hosted by the server.

To view the latest entries in a log:

```
$ tail log-file
```

You can use the `serveradmin getLogPaths` command to see where the current error and activity logs for each site are located.

To display the log paths:

```
$ sudo serveradmin command web:command = getLogPaths
```

Viewing Service Statistics

You can use the `serveradmin getHistory` command to display a log of periodic samples of the number of requests, cache performance, and data throughput. Samples are taken once each minute.

To list 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, for which you want to see samples. 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 at which the sample was measured. A standard UNIX time (number of seconds since Sep 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 --set -i
```

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 to 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 the Server Performance

When trying to analyze the server's performance, keep in mind that a lot of factors can affect performance: CGI scripts growing too large, database queries exhausting your computer's resources, 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 thus get an idea of how long it takes your website to respond, as well as other valuable statistics. The following command will simulate 1000 requests to the specified URL with the user name and password provided.

```
$ ab -n 1000 -c 1 -A user:password www.student number.example.com/
```

Working with Application Servers and Java

With the built-in JBoss application server and full support for JSPs, Java Servlets and SOAP, Mac OS X Server provides a complete solution for hosting Java 2 Platform Enterprise Edition (J2EE) applications. It also features powerful deployment tools that simplify configuration of application resources and EJB components. Mac OS X Server includes several Jave application server components, including:

- Apache Tomcat
- Java virtual machine (J2SE)
- JBoss Server (EJB)
- MySQL
- WebObjects
- Apache Axis

For more information about Java and J2EE, visit java.sun.com/j2ee/overview.html.

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:

```
$ /Library/Tomcat/bin./startup.sh start
```

Note: If you start Tomcat manually, it will not be reflected in the Server Admin application. Additionally, it will not be monitored by the `launchd` process.

Tomcat uses port 9006 by default. Tomcat comes with several example servlets. You can access these servlets at localhost:9006/examples/servlets/. The example servlets reside in /Library/Tomcat/webapps/examples/servlets/WEB-INF. To deploy your own servlets, place them in /Library/Tomcat/webapps/WEB-INF.

Tomcat's configuration information is located in /Library/Tomcat/conf/. For more information about Tomcat, see jakarta.apache.org/tomcat.

JBoss Server

Mac OS X Server includes JBoss, an open source application server and Enterprise JavaBeans (EJB) container. JBoss runs as part of the Java process.

Server Admin stores configuration information inside the conf folder that corresponds to the selected configuration option. For example, if you choose the default option (deploy-standalone), JBoss uses the configuration information located in /Library/JBoss/3.2/server/deploy-standalone/.

To start JBoss, enter the following:

```
/Library/JBoss/3.2/bin/run.sh -c deploy-standalone
```

When you use this command, the system updates the Application Server pane of Server Admin to reflect the status of JBoss. Sometimes, however, you might need to click Refresh to show the configuration changes.

You can monitor the JBoss logs by reading the logs in /Library/Logs/JBoss/.

To stop JBoss, enter the following:

```
/Library/JBoss/3.2/bin/shutdown.sh
```

You can also stop JBoss by terminating the running `run.sh` command. JBoss uses Tomcat as its default web server and servlet container.

MySQL Database

Mac OS X Server includes MySQL, a popular open source database that you can use with web applications. This database is well suited for common web-related tasks, such as content management and implementing web features, such as discussion boards and guestbooks.

Before you can start MySQL for the first time, you need to install default files needed for MySQL to run. For instructions, refer to the web technologies administration guide.

Mac OS X Server stores the files of the preinstalled MySQL version in the file system, with executables in /usr/sbin and /usr/bin, 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. By default the configuration file doesn't exist, so the default configuration is applied. You can find sample MySQL configuration files in /usr/share/mysql/.

To have MySQL run every time the computer restarts, add the following line to the /etc/hostconfig file:

```
MYSQL=-YES
```

There is no Server Admin support for the MySQL database management system, but there is an application called MySQL Manager which provides a graphical way to install the default database, set a root password, set the network option, and start and stop the `mysqld` daemon. All these actions can also be performed from the command line.

To install the default database:

```
$ sudo /usr/bin/mysql_install_db --user=mysql -u mysql
```

To set the root password:

```
$ sudo /usr/bin/mysqladmin shutdown
$ sudo /usr/bin/mysqld_safe --skip-grant-tables --skip-networking &
$ sudo /usr/bin/mysqladmin -u root flush-privileges password new-password
```

When you set up MySQL service for the first time, make sure to set up a password for the MySQL root user to protect your server from unauthorized access.

To create a database:

```
$ mysqladmin -u root password "password"
> create database mydatabase
```

To set the network option:

Edit /etc/mysqlManager.plist and set the string value of the `allowNetwork` key to either `"yes"` or `"no"`.

To start mysqld:

1. Edit /etc/hostconfig and set `MySQL` to `-YES-`.
2. Start `mysqld`.

```
$ SystemStarter start MySQL
```

To stop mysqld (and clear flag in /etc/hostconfig so it does not start upon reboot):

1. Edit `/etc/hostconfig` and set `MySQL` to `-NO-`.
2. Stop `mysqld`.

```
$ sudo SystemStarter stop MySQL
```

The MySQL startup item launches the `mysqld` daemon with arguments extracted from the configuration file /etc/mysqlManager.plist. It uses the Apple-provided `mysqld_manager_options` tool to do this.

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
- `mysqld_safe`—The `mysqld` parent (watchdog) process
- `mysql`—The MySQL database text-based client

For more information about setting up and configuring MySQL, see www.mysql.org.

Working with Network Services 14

In this chapter you will find commands you can use to configure and manage DHCP, DNS, Firewall, NAT, and VPN services in Mac OS X Server.

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.

Managing Network Services

Mac OS X Server uses the `xinetd` process to manage many of its UNIX network services, such as FTP, finger, and so on. `xinetd` listens for requests on certain TCP/IP sockets. `xinetd` is a secure replacement for `inetd`. However, because `xinetd` does not handle RPC services very well, both `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, 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 particular computer lower by launching only those that are requested by a client.

`inetd` and `xinetd` each have their own configuration files. `inetd` uses one file, `inetd.conf`, to map a given service to its executable. All standard services that `inetd` handles are already listed in the file. `xinetd`, on the other hand, uses a different configuration file for each service it provides. In the `/etc/xinetd.d` folder, there are configuration files for each of the services 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 the 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 DHCP service:

```
$ sudo serveradmin start dhcp
```

To stop DHCP service:

```
$ sudo serveradmin stop dhcp
```

Checking the Status of DHCP Service

To see summary status of DHCP service:

```
$ sudo serveradmin status dhcp
```

To see detailed status of DHCP service:

```
$ sudo serveradmin fullstatus dhcp
```

Viewing DHCP Service Settings

To list DHCP service configuration settings:

```
$ sudo serveradmin settings dhcp
```

To list a particular setting:

```
$ sudo serveradmin settings dhcp:setting
```

To list a group of settings:

You can list a group of settings that have part of their names in common by typing only as much of the name as you want, stopping at a colon (`:`), and typing an asterisk (`*`) as a wildcard for the remaining parts of the name. For example:

```
$ sudo serveradmin settings dhcp:subnets:*
```

Changing DHCP Service Settings

To see a list of available service settings:

```
$ sudo serveradmin settings dhcp
```

Also see "DHCP Service Settings" on this page and "DHCP Subnet Settings Array" on page 220.

To change a single DHCP setting:

```
$ sudo serveradmin settings dhcp:setting = value
```

Parameter	Description
setting	A DHCP service setting. See table below.
value	An appropriate value for the setting.

To change several DHCP settings at once:

```
$ sudo serveradmin settings
dhcp:setting = value
dhcp:setting = value
dhcp:setting = value
[...]
Control-D
```

DHCP Service Settings

Use the following parameters with the `serveradmin` tool to change settings for the DHCP service.

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 particular subnet. <subnetID> is a unique identifier for each subnet. See "DHCP Subnet Settings Array" on this page.

DHCP Subnet Settings Array

An array of the 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 just 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 the Server Admin application.
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 the Server Admin application.
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 the Server Admin application.
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 the Server Admin application.
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 the Server Admin application.
dhcp_router	The IPv4 address of the subnet's router. Corresponds to the Router field in the General pane of the subnet settings in the Server Admin application.

Subnet Parameter subnets:_array_id:<subnetID>:	Description
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 the Server Admin application.
net_address	The IPv4 network address for the subnet.
net_mask	The subnet mask for the subnet. Corresponds to the Subnet Mask field in the General pane of the subnet settings in the Server Admin application.
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 the Server Admin application.
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 the Server Admin application.
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 the Server Admin application.
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 the Server Admin application.
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 the Server Admin application.
WINS_primary_server	The primary WINS server to be used by clients. Corresponds to the WINS/NBNS Primary Server field in the WINS pane of the subnet settings in the Server Admin application.

Subnet Parameter	
subnets:_array_id:<subnetID>:	Description
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 the Server Admin application.
WINS_secondary_server	The secondary WINS server to be used by clients. Corresponds to the WINS/NBNS Secondary Server field in the WINS pane of the subnet settings in the Server Admin application.

Adding a DHCP Subnet

You may 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 that the server set up for you (see "Viewing DHCP Service Settings" on page 218).

You can use the `serveradmin settings` command to add other subnets to your DHCP configuration.

Note: Be sure to include the special first setting (ending with `= create`). This is how you tell `serveradmin` to create the necessary 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 already assigned to another subnet defined on the server. Can include embedded hyphens (-).
dns-server-n	To specify additional DNS servers, add additional `dhcp_name_server` settings, incrementing `_array_index:`*n* for each additional value.
Other parameters	The standard subnet settings described under "DHCP Subnet Settings Array" on page 220.

Adding a DHCP Static Map

A static DHCP map allows you to map a specific IP address to a computer based on the Ethernet (MAC) address. You can use the `serveradmin` tool to add a static map to the DHCP configuration.

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 an actual list of settings, `<mapID>` is replaced with a unique ID code for the map entry. The IDs generated by the server are just 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 actually provides the DHCP service.

Note: Be sure to include the special first setting (ending with `= create`). This is how you tell `serveradmin` to create the necessary settings array with the specified map ID. Also note that the static map for a host is identified with the host name, followed by a slash, followed by a unique ID.

You can use the `serveradmin` settings command to add maps to your DHCP configuration.

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
```

The static map entries are stored in the local NetInfo database in the computers record, so they can also be manipulated with NetInfo tools, such as `nidump`. (See the `nidump` man page for details.) For example, the static map created by the `servreadmin` tool above could be viewed as follows:

```
$ nidump -r /machines .
{
  "name" = ( "machines" );
  CHILDREN = (
...
    {
      "name" = ( "examplehost" );
      "en_address" = ( "00:30:a1:a2:a1:23" );
      "ip_address" = ( "1.2.3.4" );
      "uuid" = ( "9681BABD-3329-402E-A7AB-F0C3608E231D" );
    }
...
  )
}
```

List of DHCP serveradmin Commands

You can use the following command with the `serveradmin` tool to manage DHCP service.

Command (`dhcp:command=`)	Description
`getLogPaths`	Determine the location of the DHCP service logs.

Viewing the DHCP Service Log

You can use `tail` or any other file listing tool to view the contents of the DHCP service log.

To view the latest entries in a log:

```
$ tail log-file
```

You can use the `serveradmin getLogPaths` command to see where the current DHCP log is located.

To display the log path:

```
$ sudo serveradmin command dhcp:command = getLogPaths
```

The computer will respond with the following output:

```
dhcp:systemLog = <system-log>
```

Value	Description
`<system-log>`	The location of the DNS service log. Default = `/var/logs/system.log`

Managing the DNS Service

The Domain Name System (DNS) is a distributed database that maps IP addresses to domain names so your clients can find the resources by name rather than by numerical address. A DNS server keeps a list of domain names and the IP addresses associated with each name.

Starting and Stopping the DNS Service

To start DNS service:

```
$ sudo serveradmin start dns
```

To stop DNS service:

```
$ sudo serveradmin stop dns
```

Checking the Status of DNS Service

To see summary status of DNS service:

```
$ sudo serveradmin status dns
```

To see detailed status of DNS service:

```
$ sudo serveradmin fullstatus dns
```

Viewing DNS Service Settings

To list DNS service configuration settings:

```
$ sudo serveradmin settings dns
```

To list a particular setting:

```
$ sudo serveradmin settings dns:setting
```

To list a group of settings:

Enter only 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:*
```

Changing DNS Service Settings

You can use `serveradmin` to modify your server's DNS configuration. However, you'll probably find it more straightforward to work directly with DNS and BIND using the standard tools and techniques described in the many books on the subject. (See, for example, *DNS and BIND* by Paul Albitz and Cricket Liu.)

DNS Service Settings

To list the settings, see "Viewing DNS Service Settings" on this page.

List of DNS serveradmin Commands

Command (`dns:command=`)	Description
`getLogPaths`	Find the location of the DNS service log. See "Viewing the DNS Service Log" on this page.
`getStatistics`	Retrieve DNS service statistics. See "Listing DNS Service Statistics" on this page.

Viewing the DNS Service Log

You can use `tail` or any other file listing tool to view the contents of the DNS service log.

To view the latest entries in a log:

```
$ tail log-file
```

You can use the `serveradmin getLogPaths` command to see where the current DNS log is located. The default is /Library/Logs/named.log.

To display the log path:

```
$ sudo serveradmin command dns:command = getLogPaths
```

Listing DNS Service Statistics

You can use the `serveradmin getStatistics` command to display a summary of current DNS service workload.

To list statistics:

```
$ sudo serveradmin command dns:command = getStatistics
```

The computer will respond 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 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 needs to be configured correctly to allow it to route network data.

After the interfaces are configured to allow the server computer to communicate on the two networks, you need to enable the computer to forward traffic between the two 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 once you reboot the computer. To enable IP forwarding once Mac OS X Server restarts, you must set the `IPFORWARDING` flag in the /etc/hostconfig file to `-YES-` to enable IP forwarding during the startup process.

Managing the Firewall Service

Mac OS X Server uses the reliable open source IPFW2 software for its firewall service. To protect your network applications, the 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 .

The 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 simply 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
```

Note that the rules loaded in the firewall remain there regardless of this setting. It's just that they are ignored when the firewall is disabled.

Starting and Stopping Firewall Service

To start Firewall service:

```
$ sudo serveradmin start ipfilter
```

To stop Firewall service:

```
$ sudo serveradmin stop ipfilter
```

Checking the Status of Firewall Service

To see summary status of Firewall service:

```
$ sudo serveradmin status ipfilter
```

To see detailed status of Firewall service, including rules:

```
$ sudo serveradmin fullstatus ipfilter
```

Viewing Firewall Service Settings

To list Firewall service configuration settings:

```
$ sudo serveradmin settings ipfilter
```

To list a particular setting:

```
$ sudo serveradmin settings ipfilter:setting
```

To list a group of settings:

Enter only 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:*
```

Changing Firewall Service Settings

To change a setting:

```
$ sudo serveradmin settings ipfilter:setting = value
```

Parameter	Description
setting	An `ipfilter` service setting. See "Firewall Service Settings" on page 229.
value	An appropriate value for the setting.

To change several settings:

```
$ sudo serveradmin settings
ipfilter:setting = value
ipfilter:setting = value
ipfilter:setting = value
[...]
Control-D
```

Firewall Service Settings

Use the following parameters with the `serveradmin` tool to change settings for the `ipfilter` service.

Parameter (`ipfilter:`)	Description
`ipAddressGroupsWithRules:_array_id:<group>...`	An array of settings describing the services allowed for specific IP address groups. See "ipfilter Groups with Rules Array" on page 230.
`rules:_array_id:<rule>:...`	Arrays of rule settings, one array per defined rule. See "ipfilter Rules Array" on page 233.
`logAllDenied`	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`

ipfilter Groups with 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 of settings, <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

You can use serveradmin to set up firewall rules for your server. However, a simpler method is to add your rules to a configuration file used by the firewall service. By modifying the file, you'll be able to 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 in which you can define your rules is /etc/ipfilter/ipfw.conf. The firewall service reads this file, but doesn't modify it. Its contents are annotated and include commented-out rules you can use as models. Its default contents are listed below.

Packets are passed to ipfw from a number of different places in the protocol stack (depending on the source and destination of the packet, it is possible that ipfw is invoked multiple times on the same packet). The packet passed to the firewall is compared against each of the rules 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.

ipfw can be configured with a variety of commands. See the ipfw man page for more information.

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 which 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 entire cracker.evil.org network to my host:

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 tell `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 “ipfilter Rules Array” on page 233.

An example of this would be similar to 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
```

ipfilter Rules Array

An array of the following settings is included in the ipfilter settings for each defined firewall rule. In an actual list of settings, <rule> is replaced with a rule number. You can add a rule by using serveradmin to create such an array in the firewall settings (see "Adding Rules Using serveradmin" on page 232).

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

You can use the following commands with the serveradmin tool to manage the firewall service.

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 the standard services as they appear on the General pane of the Firewall service settings in the Server Admin application.
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 48.

Viewing Firewall Service Log

You can use tail or any other file listing tool to view the contents of the ipfilter service log.

To view the latest entries in the log:

```
$ tail log-file
```

You can use the serveradmin getLogPaths command to see where the current ipfilter service log is located.

To display the log path:

```
$ sudo serveradmin command ipfilter:command = getLogPaths
```

The computer will respond 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 the Firewall service in Mac OS X service in conjunction 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 the NAT Service

Network Address Translation (NAT) is sometimes referred to as IP masquerading. NAT is used to allow multiple computers access to 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 all the 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 any of the computers behind the NAT router unless Port forwarding is enabled.

Note: The Firewall service must be configured and running to have NAT service. The NAT service divert rule is run through ipfw.

Starting and Stopping NAT Service

To start NAT service:

```
$ sudo serveradmin start nat
```

To stop NAT service:

```
$ sudo serveradmin stop nat
```

Checking the Status of NAT Service

To see summary status of NAT service:

```
$ sudo serveradmin status nat
```

To see detailed status of NAT service:

```
$ sudo serveradmin fullstatus nat
```

Viewing NAT Service Settings

To list NAT service configuration settings:

```
$ sudo serveradmin settings nat
```

To list a particular setting:

```
$ sudo serveradmin settings nat:setting
```

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 236.
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

Use the following parameters with the serveradmin tool to change settings for NAT service.

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
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

You can use the following commands with the serveradmin tool to manage NAT service.

Command (nat:command=)	Description
getLogPaths	Find the current location of the log used by the NAT service. See "Viewing the NAT Service Log" on this page.
updateNATRuleInIpfw	Update the firewall rules defined in the ipfilter service to reflect changes in the NAT 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 48.

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 in turn processed by the `serveradmin` tool, and used to create the configuration file /etc/nat/natd.conf.apple, which is passed to the `natd` process. See the `natd` man page for details about configuring `natd`.

Note: Don't edit the /etc/nat/natd.conf.apple file directly, since 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`, you would 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>
```

You can then 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

You can use `tail` or any other file listing tool to view the contents of the NAT service log.

To view the latest entries in the log:

```
$ tail log-file
```

You can use the `serveradmin getLogPaths` command to see where the current NAT service log is located.

To display the log path:

```
$ sudo serveradmin command nat:command = getLogPaths
```

The computer will respond 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 the VPN Service

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 VPN service:

```
$ sudo serveradmin start vpn
```

To stop VPN service:

```
$ sudo serveradmin stop vpn
```

Checking the Status of VPN Service

To see summary status of VPN service:

```
$ sudo serveradmin status vpn
```

To see detailed status of VPN service:

```
$ sudo serveradmin fullstatus vpn
```

Viewing VPN Service Settings

To list VPN service configuration settings:

```
$ sudo serveradmin settings vpn
```

To list a particular setting:

```
$ sudo serveradmin settings vpn:setting
```

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 "List of VPN Service Settings" on page 239.
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
```

List of VPN Service Settings

Use the following parameters with the `serveradmin` tool to change settings for VPN service.

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"`

Parameter (vpn:Servers:)	Description
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"
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"

Parameter (vpn:Servers:)	Description
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"
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"

Parameter (vpn:Servers:)	Description
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"

List of VPN serveradmin Commands

You can use the following commands with the `serveradmin` tool to manage VPN service.

Command (vpn:command=)	Description
getLogPaths	Find the current location of the VPN service log. See "Viewing the VPN Service Log" on this page.
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 48.

Viewing the VPN Service Log

You can use `tail` or any other file listing tool to view the contents of the VPN service log.

To view the latest entries in the log:

```
$ tail log-file
```

You can use the `serveradmin getLogPaths` command to see where the current VPN service log is located.

To display the log path:

```
$ sudo serveradmin command vpn:command = getLogPaths
```

The computer will respond 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 in turn 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. See the racoon and setkey man pages for more information. racoon also has a webpage: www.kames.com/racoon. You might also find the ipsec man page helpful in getting more information.

Apple provides an interactive s2svpnadmin tool, located 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. Note that s2svpnadmin does not start the VPN service. You have to start the VPN service separately from Server Admin.

The s2svpnadmin tool can list currently configured site-to-site VPN servers, display their configuration details, add a new configuration, and delete an existing configuration. This tool can be used to configure only a local VPN server, not a remote one. To set up a site-to-site server successfully, you need to configure the two VPN gateway servers at the two sites independently.

s2svpnadmin must be run as root.

Configuring Site-to-Site VPN

To configure a site-to-site VPN, run s2svpnadmin as root and choose the "Configure a new site-to-site server" option. You will need to provide the following information:

- A configuration name used to identify the server. This string should not have any 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, ensure that at least one certificate is currently installed on the server. `s2svpnadmin` will display a list of currently installed certificates and prompt the user to choose one of these. Certificates can be created, self-signed, and installed using the Server Admin application. If a shared secret is desired, ensure that the same shared secret is configured on the VPN server at the other site.
- One or more policies consisting of local and remote subnet addresses. A policy is made of 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. Ensure that a compatible policy is configured on both VPN servers.

If an invalid entry is made, `s2svpnadmin` will force you to start all over again.

Note: `s2svpnadmin` will ask if the server needs to be enabled. By default, it is enabled. Currently, `s2svpnadmin` does not support editing a configuration, so if the server is not enabled, the configuration will need to be deleted and recreated and enabled at a later time; alternatively, you can edit the configuration file directly. The configuration file is a plist file located in /Library/Preferences/SystemConfiguration/com.apple.RemoteAccessServers.plist.

Adding a VPN Keyagent User

To enable the PPTP protocol in your VPN server, you must add a keyagent user in the LDAP folder that hosts your users. If you have more than one folder with VPN users, you must add a keyagent in each of the folders.

The `vpnaddkeyagentuser` tool lets you add the required VPN PPTP keyagent user to a folder. The tool will prompt you for the administrator user name and password of the folder. It will then set up the keyagent user. This step is necessary to be able 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. `vpnaddkeyagentuser` must be run as root. If no argument is specified, the keyagent user is added to the local netinfo 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. Once 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.

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. See the `rsync` man page for more information.

IP Failover Prerequisites

IP failover isn't a complete solution; it is one tool you can use to increase your server's availability to your clients. To use IP failover, you need to 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 the 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 both 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 email notification of a network anomaly, but doesn't acquire the primary server's IP address.

Email 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. Be sure to 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 tells 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 that your changes can take effect.

3 Disconnect the primary server from both 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 email address for notification messages regarding the primary server status. If this line is omitted, email 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 need to 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 simply sends an email notification. If no script exists, or if the script returns a zero result, then the secondary server acquires the primary's IP address. If the script returns a nonzero result, then the secondary server skips IP address acquisition and only sends email notification of the failover condition. The `Test` script is run 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 located 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 email 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 (pre acquisition), after acquiring the IP address (post acquisition), before relinquishing the primary IP address (pre relinquish), and after relinquishing the IP address back to the primary server (post relinquish). These scripts reside in the /Library/IPFailover/*IP_address* folder on the secondary server. The scripts use these four 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: Always be sure that the primary server is up and functioning normally before you activate IP failover on the secondary server. If the primary server isn't sending broadcast messages, the secondary server will initiate the failover process and acquire the primary's public IP address.

You may 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 may perform other services on the network, such as running a statistical analysis application and distributed image processing software. A pre acquisition script quits the running applications to free up the CPU for the Web server. A post acquisition script starts the Web server. Once the primary server is up and running again, a pre relinquish script quits the Web server, and a post relinquish 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

You can use the `pppd` daemon to set up Point-to-Point Protocol (PPP) dial-in service. See the `pppd` man page for more information.

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 NetInfo database. In certain cases, you might want to reset a service back to its default settings, which can be done by simply renaming or deleting a service's configuration file. Mac OS X Server will then create a fresh default copy of the file.

To restore the NAT service to its default configuration:
Rename or delete the natd.plist file located in the /etc/nat/ folder.

To restore the Firewall service to its default configuration:
Rename or delete the ip_address_groups.plist, standard_services.plist, and ipfw.conf files located in the /etc/ipfilter/ folder.

To restore the DHCP service to its default configuration:

1 Remove the subnet configuration from the /config/dhcp folder in the local NetInfo database by using the `nicl` tool:

```
$ sudo nicl . -delete /config/dhcp
```

2 Remove the static Ethernet / IP Address static maps from the /machines folder in the local NetInfo database. The easiest way to do this is to delete the folder:

```
$ sudo nicl . -delete /machines
```

3 Re-create the two default records:

```
$ sudo nicl . -create /machines/localhost
$ sudo nicl . -append /machines/localhost ip_address 127.0.0.1
$ sudo nicl . -append /machines/localhost serves ./local
$ sudo nicl . -create /machines/broadcasthost
$ sudo nicl . -append /machines/broadcasthost ip_address 255.255.255.255
$ sudo nicl . -append /machines/broadcasthost serves ../network
```

To restore the QTSS Publisher service to its default configuration:
Rename or delete these three 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 reside in the/Library/Application Support/Apple/QTSS Publisher/* folder. The content varies, based on what's been uploaded:

To restore the QTSS service to its default configuration:
Rename or delete these two files:

- /Library/QuickTimeStreaming/Config/streamingserver.xml
- /Library/QuickTimeStreaming/Config/relayconfig.xml

You may 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 the DNS service to its default configuration:

1 Remove the files for each forward zone, named similar to my.domain.com.zone from the /etc/named.conf /var/named/* folder.

2 Remove the separate files for each reverse zone, named similar to db.10.1.0 from the /etc/named.conf/var/named/* folder.

3 Do not remove the localhost.zone, named.ca, or named.local files.

To restore the VPN service to its default configuration:

Rename the com.apple.RemoteAccessServers.plist file located in the /Library/Preferences/SystemConfiguration/ folder.

To restore the SERVERMGR_MAIL service to it's default configuration:

Rename these two files:

- /etc/MailServicesOther.plist
- /var/mailman/data/listinfo.plist

Working with Open Directory 15

In this chapter you will find commands used to configure and manage the Open Directory service.

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. It 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. This chapter discusses the tools and commands used when working with Open Directory.

Understanding Open Directory

Mac OS X Server relies on the Lightweight Directory Access Protocol (LDAP) to provide access to directory service data. 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 wide variety 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, NetInfo, 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

You can use the `dscl` tool to test your directory services configuration. See the `dscl` man page for more information.

Modifying a Directory Domain

You can use the dscl tool to create, modify, or delete directory information in a directory domain.

Testing Open Directory Plug-ins

You can use the dsperfmonitor tool to check the performance of the protocol-specific plug-ins used by Open Directory. It can list the API calls being made to plug-ins, how long the plug-ins take to reply, and recent API call errors. See the dsperfmonitor man page for more information.

Directory services API support is provided by the DirectoryService daemon. See the DirectoryService man page for more information.

See the DirectoryServiceAttributes man page for information about the data types used by directory services.

Finally, for information about the internals of Open Directory and its plug-ins, including source code you can examine or adopt, follow the Open Directory link at www.apple.com/darwin/.

Registering URLs with SLP

You can use the slp_reg tool to register service URLs using the Service Location Protocol (SLP). See the slp_reg man page for more information.

SLP registration is handled by the SLP daemon slpd. See the slpd man page for more information.

Changing Open Directory Service Settings

Use the following parameters with the serveradmin tool to change settings for the Open Directory service. Be sure to add dirserv: to the beginning of any parameter you use.

To see the role that the server is playing in the directory hierarchy:

```
$ sudo serveradmin settings dirserv:LDAPServerType
```

Parameter (dirserv:)	Description
replicationUnits	Default = "days"
replicaLastUpdate	Default = " "
LDAPDataBasePath	Default = " "
replicationPeriod	Default = 4
LDAPSearchBase	Default = " "

Parameter (dirserv:)	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"
NetInfoRunStatus	Default = " "
LDAPSSLCertificatePath	Default = " "
masterServer	Default = " "
LDAPServerType	Default = "standalone"
NetInfoDomain	Default = " "
replicationWhen	Default = "periodic"
useSSL	Default = "YES"
LDAPDefaultPrefix	Default = "dc=<domain>,dc=com"
LDAPTimeoutUnits	Default = "minutes"
LDAPServerBackend	Default = "BerkeleyDB"

Managing OpenLDAP

Open Directory uses OpenLDAP, the open source implementation of LDAP, to provide directory services for mixed-platform environments. A common language for directory access lets you consolidate information from different platforms and define a single name space for all 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, located in /usr/libexec/. slapd is launched automatically by the LDAP startup item. The primary configuration files for OpenLDAP are kept in /etc/openldap/. There you will find the slapd.conf file, which contain basic configuration information. Most of the 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, it's advisable that you not modify these directives. Instead, use your own configuration file by adding an include directive for it in the slapd.conf file.

The slapd_macosx.conf file contains an entry for the root user of the LDAP database, the directive `rootdn`. This root user is not the same as the root user in the local NetInfo database, but rather it is a user who has total 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 user on the computer can edit the slapd_macosxserver.conf file to add a new password hash, or plain-text password, to the file, at which point that administrator user would be able to administrator the LDAP database. This is especially useful when your LDAP database has become damaged or the passwords have been lost or forgotten.

Configuring slapd and slurpd Daemons

You can use the `slapconfig` tool to configure the `slapd` and `slurpd` LDAP daemons and related search policies. See the `slapconfig` man page for more information.

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`.

The slap tools must be run directly on the computer hosting the LDAP database. You should shut down the LDAP service when using the slap tools, or else your database may become 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`. See the `passwd` man page for more information.
`/usr/bin/ldapsearch`	Search the LDAP directory. See the usage note under "Searching the LDAP Server" on page 255.
`/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.

Tool	Used to
/usr/sbin/slapindex	Regenerate directory indexes.
/usr/sbin/slappasswd	Generate user password. hashes.

Idle Rebinding Options

The following two LDAPv3 plug-in parameters are documented in the Open Directory administration guide. 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, you can add it there.

Idle Timeout

This parameter specifies how long the LDAP plug-in will sit idle before disconnecting from the server. You can adjust this value to reduce overloading of the server's connections from remote clients.

```
<key>Idle Timeout in minutes<\key>
<integer>n<\integer>
```

If this parameter doesn't already exist in the DSLDAPv3PlugInConfig.plist file, you can 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 all the 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 the 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: 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: 1.3.18.0.2.12.24
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
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: 1.3.18.0.2.10.15
supportedcontrol: 1.3.18.0.2.10.18
security: none
port: 389
supportedsaslmechanisms: CRAM-MD5
supportedldapversion: 2
supportedldapversion: 3
ibmdirectoryversion: 5.1
ibm-ldapservicename: tr17n01.aset.psu.edu
ibm-adminid: CN=MANAGER,DC=PSU,DC=EDU
```

```
ibm-serverId: 71d3fb40-c90a-1028-9ef7-8e62f6ed25ed
ibm-supportedacimechanisms: 1.3.18.0.2.26.3
ibm-supportedacimechanisms: 1.3.18.0.2.26.2
vendorname: International Business Machines (IBM)
vendorversion: 5.1
ibm-sslciphers: N/A
ibm-supportedcapabilities: 1.3.18.0.2.32.1
ibm-supportedcapabilities: 1.3.18.0.2.32.2
ibm-supportedcapabilities: 1.3.18.0.2.32.3
ibm-supportedcapabilities: 1.3.18.0.2.32.4
ibm-supportedcapabilities: 1.3.18.0.2.32.5
ibm-supportedcapabilities: 1.3.18.0.2.32.6
ibm-enabledcapabilities: 1.3.18.0.2.32.1
ibm-enabledcapabilities: 1.3.18.0.2.32.2
ibm-enabledcapabilities: 1.3.18.0.2.32.3
ibm-enabledcapabilities: 1.3.18.0.2.32.4
ibm-enabledcapabilities: 1.3.18.0.2.32.5
ibm-enabledcapabilities: 1.3.18.0.2.32.6
ibm-slapdisconfigurationmode: FALSE
```

If the server is an OpenLDAP server, you will need to either specify `+` for all operational attributes or specify the particular 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 get that, you can 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 within the same 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:* Many of the LDAP tools will 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, use the ldapadd tool as follows:

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 you should carefully test all procedures documented in it before using them on an Open Directory server that's in service.

Managing NetInfo

NetInfo is the built-in Mac OS X directory service used for the local directory domain on every Mac OS X Server and Mac OS X computer. NetInfo stores information about users and resources and makes it available to Mac OS X processes that want to use it.

Note: NetInfo may not be supported in future releases. Administrators should use `dscl` and other tools that work on LDAP or NetInfo whenever possible.

Configuring NetInfo

You can use the following tools to manage the NetInfo directory. For more information about a particular tool, see the related man page.

Tool	Used to
`NeST`	Configure a NetInfo directory domain. There can actually be more than one NetInfo directory domain on an upgraded server, and NeST can also be used on a client computer's NetInfo directory domain.
`nicl`	Create, view, and modify entries in the NetInfo directory.
`nidomain`	Creates and destroys NetInfo directories. Tells you which domains are served from which directories by servers running on a particular computer.
`nifind`	Search the NetInfo directory for a particular entry.
`nigrep`	Search the NetInfo directory for all instances of a string you specify.
`nidump`	Export NetInfo data to text or flat files.
`niload`	Import flat files into the NetInfo directory.
`nireport`	Print tables of NetInfo directory entries.
`niutil`	Reads from a NetInfo directory and writes to one.

In addition, you can use the NeST tool to get and set authentication methods used by Open Directory Password Server, as described in "Enabling or Disabling Authentication Methods" on page 260.

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 the traditional password authentication methods that some network services or users' client applications require. Services can be configured to not allow Kerberos, in which 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 never allow passwords to be read. Passwords can only be set and verified.

Open Directory Password Server

Password Server uses the standard Simple Authentication and Security Layer (SASL) technology to negotiate an authentication method between a client and a service. It 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.

You can use the `mkpassdb` tool to create, modify, or back up the password database used by the Server Password Server. See the `mkpassdb` man page for more information.

Viewing or Changing Password Policies

You can use the `pwpolicy` tool to view or change the authentication policies used by the Mac OS X Server Password Server. See the `pwpolicy` man page for more information.

Enabling or Disabling Authentication Methods

All password authentication methods supported by the Open Directory Password Server are initially enabled. You can disable and enable the Open Directory Password Server authentication methods by using the `NeST` tool.

To see a list of available methods:

```
$ NeST -getprotocols
```

To disable or enable a method:

```
$ NeST -setprotocols protocol (on|off)
```

Replace `protocol` with any of the protocol names listed by `NeST -getprotocols` (for example, `SMB-LAN-MANAGER`). For information about the available methods, see the Open Directory administration guide.

Kerberos and Apple Single Sign-On

Built into Open Directory is a robust authentication server that uses MIT's Kerberos Key Distribution Center (KDC)—providing strong authentication with support for secure single sign-on. That means users need authenticate only once, with a single user name and password pair, for access to 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 within 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. Keep in mind that the data in question is extremely sensitive—creating a copy of it, by definition, decreases your overall security. These backups should be subject to the same security precautions as the other KDC files.

Note: Do not back up the KDC while the `krb5kdc` process is running.

To dump the KDC's database:

Replace *`/path/to/secure/backup`* with the path to the location you are backing up the database to.

```
$ sudo kdb5_util dump > /path/to/secure/backup
```

To load KDC data from a dumped file:

Replace *`/path/to/secure/backup`* with the path to the location of your backup database.

```
$ sudo kdb5_util load /path/to/secure/backup
```

`kdb5_util` can be used to create and delete Kerberos databases and to manage the location of the stash file used to encrypt the database as well.

Principal Management

Mac OS X Server uses MIT's Kerberos administration architecture for principal management. The Kerberos administration daemon `kadmind` 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, and hence, manual modifications are rarely needed.

The configuration files for `kadmin` and `krb5kdc` are located in /var/db/krb5kdc. The kadm5.acl file is a list of Kerberos principals that have various administrative privileges.

The database named principal.kadm5 is the `kadmind` process' policy database. It is located in /var/db/krb5kdc. While 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`. It is unlike most lock files though, as `kadmind` will not write to either the policy or principal database unless it exists.

The `kadmin` tool, located 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 is always run as root, 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 some basic `kadmin` tool uses.

To add a principal:

Replace *student1* with the new principal that you are adding to the database.

```
$ sudo kadmin.local -q "add_principal student1"
```

To add a service principal:

Replace *afpserver/server.example.com* with the new service principal that you are adding to the database.

```
$ sudo kadmin.local -q "add_principal afpserver/server.example.com"
```

To delete a principal:

Replace *student1* with the principal that you are deleting from the database.

```
$ sudo kadmin.local -q "delete_principal student1"
```

To list all principals:

```
$ sudo kadmin.local -q list_principals
```

Using kadmin to kerberize a service

kadmin can be used to kerberize additional services, depending on your specific configuration requirements. While 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 needs to 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 info is from the service documentation.

To kerberize a service (from a terminal running on that host):

1 Use kadmin to create the service principal.

```
$ 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. Make sure to check the service documentation for how to perform this step.

Using Directory Service Tools

The following are miscellaneous directory service tools that you can use to configure directory services and to troubleshoot any problems.

Operating on Directory Service Directory Domains

dscl is a general-purpose tool for operating on directory domains. Its commands allow one to create, read, and manage directory data. If invoked without any commands, dscl runs in an interactive mode, reading commands from standard input.

The following examples show some basic dscl tool uses:

To verify that you are able to 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

See the dscl man page for more information.

Finding Network Information

The lookupd daemon acts as an information broker and cache. It is called by various routines in the System framework to find information about user accounts, groups, printers, email aliases and distribution lists, computer names, Internet addresses, and several other kinds of information. lookupd also has a channel to query Open Directory, allowing access to data from LDAP and other directory services.

To look up a user by name:

```
$ lookupd -q user -a name anne
```

This returns the user records that have a short name of "anne."

To run lookupd in interactive mode:

```
$ lookupd -d
>?
```

Typing ? at the lookupd interactive promt (>) displays all the possible commands for lookupd.

To list the attributes of a user:

```
> userWithName: anne
```

See the lookupd man page for more information.

Manipulating a Single Named Group Record

dseditgroup allows manipulation of a single named group record on either the default local directory domain or the specified directory domain. The following examples show some uses for dseditgroup.

To display the attributes of a group in the local directory domain:

```
$ dseditgroup -o read groupname
```

To create a group in a specified domain:

```
$ dseditgroup -o create -n /LDAPv3/ldap.example.com -u myusername -P
     mypassword -r "Group Name" -c "comment" -s 1234 -k "some keyword"
     groupname
```

To delete a group from a specified domain:

```
$ dseditgroup -o delete -n /LDAPv3/ldap.example.com -u myusername -P
     mypassword groupname
```

Parameter	Description
myuser	User name authenticated with administrator user
mypassword	User password
Group Name	Real name to add or replace
comment	Comment or add or replace
1234	Time to livein seconds to add or replace

Parameter	Description
some keyword	Keyword to add
groupname	Group name

See the `dseditgroup` man page for more information.

Adding or Removing LDAP Server Configurations

`dsconfigldap` allows you 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

`dsconfigad` allows you 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	Add the computer ID to the specified domain.
administrator	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 will default to the container.
domain	Fully-qualified domain name of the domain to be used when adding the computer to the directory.

See the `dsconfigad` man page for more information.

Working with QuickTime Streaming Server

16

In this chapter you will find commands you can use to configure and manage the QuickTime Streaming Server service.

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. This chapter describes the commands used to configure and manage the QuickTime Streaming Server.

Understanding QuickTime Streaming Server

Mac OS X Server version 10.4 includes the latest version of the popular QuickTime Streaming Server, 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 Service Tasks

You can use the `serveradmin` tool to start QTSS service, or you can use the `quicktimestreamingserver` tool to specify additional service parameters when you start the service.

Starting and Stopping the QTSS Service

To start QTSS service:

```
$ sudo serveradmin start qtss
```

or

```
$ sudo quicktimestreamingserver
```

To see a list of quicktimestreamingserver tool options:

```
$ sudo quicktimestreamingserver -h
```

To stop QTSS service:

```
$ sudo serveradmin stop qtss
```

Checking QTSS Service Status

To see if QTSS service is running:

```
$ sudo serveradmin status qtss
```

To see complete QTSS status:

```
$ sudo serveradmin fullstatus qtss
```

Viewing QTSS Settings

To list all QTSS service settings:

```
$ sudo serveradmin settings qtss
```

To list a particular setting:

```
$ sudo serveradmin settings qtss:setting
```

To list a group of settings:

You can list a group of settings that have part of their names in common by typing only as much of the name as you want, stopping at a colon (`:`), and typing an asterisk (`*`) as a wildcard for the remaining parts of the name. For example:

```
$ sudo serveradmin settings qtss:modules:_array_id:QTSSAdminModule:*
```

Changing QTSS Settings

You can change QTSS service settings by using the `serveradmin` tool or by editing the QTSS parameter list file directly.

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 "QTSS Settings" on page 269.
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
```

QTSS Settings

Use the following parameters with the `serveradmin` tool to change settings for the QTSS service.

Descriptions of Settings

To see descriptions of most QTSS settings, you can look in the streamingserver.xml-sample file located 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/QuickTimeStreaming/Logs/"`
`modules:_array_id:QTSSAccessLogModule:request_logfile_interval`	Default = `7`
`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/QuickTimeStreaming/Config/qtgroups"`
`modules:_array_id:QTSSAccessModule:modAccess_qtaccessfilename`	Default = `"qtaccess"`

Parameter (qtss:)	Description
modules:_array_id:QTSSAccessModule:modAccess_usersfilepath	Default = "/Library/QuickTimeStreaming/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
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

Parameter (qtss:)	Description
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/QuickTimeStreaming/Config/relayconfig.xml"
server:authentication_scheme	Default = "digest"
server:auto_restart	Default = yes
server:default_authorization_realm	Default = "Streaming Server"
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

Parameter (qtss:)	Description
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

You can use the following commands with the serveradmin tool to manage the QTSS service.

Command (qtss:command=)	Description
getConnections	List current QTSS connections. See "Listing Current Connections" on this page.
getHistory	View service statistics. See "Viewing QTSS Service Statistics" on page 273.
getLogPaths	Find the current location of the service logs. See "Viewing Service Logs" on page 274.

Listing Current Connections

You can use the serveradmin getConnectedUsers command to retrieve information about QTSS connections.

To list connected users:

```
$ sudo serveradmin command qtss:command = getConnectedUsers
```

Viewing QTSS Service Statistics

You can use the `serveradmin getHistory` command to display a log of periodic samples of the number of connections and the data throughput. Samples are taken once each minute.

To list samples:

```
$ sudo serveradmin command
qtss:command = getHistory
qtss:variant = statistic
qtss: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, for which you want to see samples. For example, to see 30 minutes of data, you would specify `qtss:timeScale = 1800`.

The computer will respond 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
<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 at which the sample was measured. A standard UNIX time (number of seconds since Sep 1, 1970). Samples are taken every 60 seconds.

Viewing Service Logs

You can use `tail` or any other file listing tool to view the contents of the QTSS service logs.

To view the latest entries in a log:

```
$ tail log-file
```

You can use the `serveradmin getLogPaths` command to see where the current QTSS error and activity logs are located.

To display the log paths:

```
$ sudo serveradmin command qtss:command = getLogPaths
```

The computer will respond 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 its Preferences

You can force QTSS to reread its preferences without restarting the server. You must log in as root to perform this task.

To force QTSS to reread its preferences:

1 List the QTSS processes:

```
$ ps -ax | grep QuickTimeStreamingServer
```

You should see a list similar to the following:

```
949  ??  Ss     0:00.00 /usr/sbin/QuickTimeStreamingServer
950  ??  S      0:00.13 /usr/sbin/QuickTimeStreamingServer
965 std  S+     0:00.00 grep QuickTimeStreamingServer
```

2 Find the larger of the two process IDs (PIDs) for the `QuickTimeStreamingServer` processes (in this case, `950`).

3 Send a HUP signal to this process:

```
$ kill -HUP 950
```

Preparing Older Home Folders for User Streaming

If you want to enable QTSS home folder streaming for home folders created using an earlier version of Mac OS X Server (before version 10.3), you need to set up the necessary streaming media folder in each user's home folder. You can use the `createuserstreamingdir` tool to set up the needed Sites/Streaming/ 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

A certain level of security is inherent in real-time streaming, since content is delivered only as the client needs it and no files remain afterward. But 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 Admin user name and password, you can reset them.

To reset the user name and password:

1 Log in to the server computer as root, open a Terminal window, and enter the following:

```
$ qtpasswd someUserName
```

where *someUserName* is a name of your choice.

2 Follow the prompts by entering the administrator user name and a password you want to assign to the user *someUserName*.

3 Using a text editor, modify the /Library/QuickTimeStreaming/Config/qtgroups file. For Windows, modify the c:\Program Files\Darwin Streaming Server\qtgroups file. For other supported platforms, modify the /etc/streaming/qtgroups file. Modify the file so that the user name you just created or modified is included in the group Admin, as follows:

```
admin: someUserName
```

4 Save the file as ordinary text (not as .rtf or any other file format).

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 access control to work, an access file must be present in the folder you selected as your media folder. If an access file is not present in the streaming server media folder, all clients are allowed access to the media in the folder.

To set up access control:

1. Use the `qtpasswd` tool to create new user accounts with passwords.
2. Create an access file and place it in the media folder that 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`).

Creating an Access File

An access file is a text file called `qtaccess` that contains information about users and groups that are authorized to view media in the folder in which 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 checks 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 any other file format).

Parameter	Description
message	Text your users see when the login window appears. It's optional. If your message contains any white space (such as a space character between terms), make sure you enclose the entire message 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 Windows, it is c:\Program Files\Darwin Streaming Server\qtusers. • For other supported platforms, it is: /etc/streaming/qtusers.
group filename	The path and filename of the group file. • For Mac OS X, the default is /Library/QuickTimeStreaming/Config/qtgroups. • For Windows, it is c:\Program Files\Darwin Streaming Server\qtgroups. • For other supported platforms, it is /etc/streaming/qtgroups. A group file is optional. 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 who is 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 `valid-user`, which designates any valid user.
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 use these additional user tags:

- `valid-user` is any user defined in the `qtusers` file. The statement `require valid-user` specifies that any authenticated user in the `qtusers` file can have access to the media files. If this tag is used, the server will prompt users for an appropriate user name and password.
- `any-user` allows any user to view media without providing a name or password.

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 folderfolder-by-folder basis.

If you make changes to the default qtaccess access file, be aware that making any changes to broadcast user settings in Streaming Server Admin will modify the default qtaccess file at the root level of the Movies folder. Any modifications you made prior to this will not be preserved.

Accessing Protected Media

Users must have QuickTime 5 or later to access a media file for which digest authentication is enabled. 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 it follows this format:

```
groupname: user-name1 user-name2 user-name3
```

For Windows, the path is c:\Program Files\Darwin Streaming Server\qtgroups. For other supported platforms, it is /etc/streaming/qtgroups.

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:

1 Log in to the server computer as administrator, and use a text editor to open the user or group file.

2 Delete the user name and encrypted passwords line from the user file.

3 Delete the user name from the group file.

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

You can use the `qtmedia` tool to manipulate QuickTime and MP4 movies. 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

You can use the `qtref` tool to create reference movies that can be used to embed QuickTime content in Web pages. You can use the following options with `qtref`.

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 any arguments to display usage information.

Configuring System Logging 17

In this chapter you will find commands you can use 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 may have their own log files located in different folders. Console.log is located in /Library/Logs/Console/*uid*, where *uid* is the user ID. The console.log file contains recent console activity. System.log is located in /var/log/ and contains all system activity, including console log information.

Configuring Your System Logging

The configuration file for the system logging daemon, `syslogd`, is /etc/syslog.conf. Each line within /etc/syslog.conf consists of text containing three types of data:

- Facility: categories of log messages. The standard facilities include `mail`, `news`, `user`, and `kern` (kernel).
- Priority: urgency of the message. In order from least to most critical, they are: `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 line specifies that for any log messages in the category `mail`, with a priority of `emerg` or higher, the message will be 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 one or more tabs. Wildcards ("*") may also be used in the configuration file. The following example line logs all messages of any facility or priority to the file /var/log/all.log:

```
*.* /var/log/all.log
```

See the `syslog.conf` man page for information about the configuration of this file.

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 new 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 will only occur if the computer is on when the job is scheduled. By default, the log rotation tasks are scheduled for very early in the morning (for example, 4:30 a.m. on Saturday) in order to be as unobtrusive as possible. If the computer will not be on at this time, adjust the settings in /etc/crontab.

For example, the following line 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 much more likely to be on:

```
DayOf DayOf
#Minute Hour Month Month Week User Command
15 12 * * 2 root periodic weekly
```

See the `crontab` man page for more information about editing the /etc/crontab file.

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. First, the `syslog` process sends log messages as clear text, which could expose sensitive information. Second, too many log messages may fill storage space on the logging system, making further logging impossible. Third, log files can indicate suspicious activity only if a baseline of normal activity has been established, and if they are regularly monitored for such activity. If these security issues outweigh the security benefit of remote logging for the network being configured, then remote logging should not be used.

Configuring Remote Logging on a Client Computer

To configure a client computer for remote logging, you must 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 on a client computer:

1 Open the /etc/syslog.conf file as root.

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. Make sure to keep 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. In the event that another system is compromised, its local logs can be altered, so the log server may contain the only accurate system records. Remote logging should only be enabled across a trusted internal network or VPN. By default, Mac OS X Server performs only local logging and will not act as a log server.

Configuring Mac OS X Server to act as a remote log server involves changing the `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. In order to better control what hosts are allowed to send logging message traffic, the `-a` option should be used to ensure that log messages from only certain IP addresses are accepted. The `-a` option may be used multiple times to specify additional hosts. The `-a` option should be followed with an address in the following 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, as the default `masklen` may be very small and the corresponding matching addresses could, therefore, 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
```

It is also possible to 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 Replacing the IP address after `-a` with your network information, 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, right before the "`exit 0`" line:

```
killall -HUP syslogd #re-load configuration
exit 0
```

`syslogd` contains features not documented in 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

In this appendix you will find information about the megaraid command, used for managing a PCI RAID Card.

The `megaraid` tool uses are described in the following table, along with parameter explanations.

```
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 will turn 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 particular logical drive. The parameter *ld* is the logical drive ID.

```
megaraid -create auto [-numld n] [-log file]
```

Automatically destroys all current 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`.

```
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 existing configuration. The RAID level and participating physical drives' parameters are required. All other parameters are optional. If `size` is not specified, the remaining size of the array will automatically be 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 new firmware becomes operational only after the computer is restarted.

```
megaraid -initialize ld -start | -stop | -status [-log file]
```

Initializes, starts, stops, or displays the status (percentage of progress) of a particular 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 particular 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.

```
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 all the 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 all drives connected to the PCI RAID Card. The command displays drive ID, identification, size, status, and any 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: See the `megaraid` man page for more information. You can also use all `megaraid` commands with a `[-log` *`file`*`]` parameter, which logs all the displayed information with 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.

administrator A user with server or directory domain administration privileges. Administrators are always members of the predefined "admin" group.

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.

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 application is running.

boot ROM Low-level instructions used by a computer in the first stages of starting up.

BSD Berkeley System 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.

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.

computer account See **computer list**.

computer list A list of computers that have the same preference settings and are available to the same users and groups.

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 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.

dynamic IP address An IP address that's assigned for a limited period of time or until the client computer no longer needs it.

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.

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.

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.

full name See **long name**.

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 back and forth among themselves.

guest computer An unknown computer that isn't included in a computer list on your server.

guest user A user who can log in to your server without a user name or password.

home folder A folder for a user's personal use. Mac OS X also uses the home folder, for example, to store system preferences and managed user settings for Mac OS X users.

HTML Hypertext Markup Language. The set of symbols or codes inserted in a file to be displayed on a World Wide 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. The HTTP protocol provides a way for a web browser to access a web server and request hypermedia documents created using HTML.

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.

idle user A user who is connected to the server but hasn't used the server volume for a period of time.

IMAP Internet Message Access Protocol. A client-server mail protocol that allows users to store their mail on the mail server rather than download 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 packets of data, while TCP keeps track of data packets.

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.

ISP Internet service provider. A business that sells Internet access and often provides web hosting for ecommerce applications as well as mail services.

Kerberos A secure network authentication system. Kerberos uses tickets, which are issued for a specific user, service, and period of time. Once a user is authenticated, it's possible to access additional services without retyping a password (this is called single sign-on) for services that have been configured to take Kerberos tickets. Mac OS X Server uses Kerberos v5.

Kerberos realm The authentication domain comprising the users and services that are registered with the same Kerberos server. The registered services and users trust the Kerberos server to verify each other's identities.

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.

load balancing The process of distributing client computers' requests for network services across multiple servers to optimize performance.

local domain A directory domain that can be accessed only by the computer on which it resides.

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 into 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 name is derived by default from the computer name, a user can specify this name in the Network 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.

long name The long form of a user or group name. See also **user name**.

LPR Line Printer Remote. A standard protocol for printing over TCP/IP.

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.

MTA Mail Transfer Agent. A mail service that sends outgoing mail, receives incoming mail for local recipients, and forwards incoming mail of nonlocal recipients to other MTAs.

multicast DNS A protocol developed by Apple for automatic discovery of computers, devices, and services on IP networks. Called "Bonjour" (previously "Rendezvous") by Apple, this proposed Internet standard protocol is sometimes referred to as "ZeroConf" or "multicast DNS." For more information, visit www.apple.com or www.zeroconf.org. To see how this protocol is used in Mac OS X Server, see **local hostname**.

multihoming The ability to support multiple network connections. When more than one connection is available, Mac OS X selects the best connection according to the order specified in Network preferences.

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.

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. An application 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.

NetInfo One of the Apple protocols for accessing a directory domain.

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 according to 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 The Apple directory services architecture, which can access authoritative information about users and network resources from directory domains that use LDAP, NetInfo, or Active Directory protocols; BSD configuration files; and network services.

Open Directory master A server that provides LDAP directory service, Kerberos authentication service, and Open Directory Password Server.

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.

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 set Read & Write, Read only, or No Access permissions for Owner; Group; and Others. The owner also can assign ownership of an item to another user, and Group privileges to another group. By default the owner has Read & Write permissions.

parent A computer whose shared directory domain provides configuration information to another computer.

PHP PHP Hypertext Preprocessor (originally Personal Home Page). A scripting language embedded in HTML that's used to create dynamic webpages.

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.

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 Initial default attributes you specify for new accounts you create using Workgroup Manager. You can use presets only during account creation.

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.

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.

QTSS QuickTime Streaming Server. A technology that lets you deliver media over the Internet in real time.

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**.

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.

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.

shadow image A file created by the NetBoot daemon process for each NetBooted client where applications running on the client can write temporary data.

share point A folder, hard disk (or hard disk partition), or CD 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, Windows SMB, NFS (an "export"), or FTP protocols.

short name An abbreviated name for a user. The short name is used by Mac OS X for home folders, authentication, and email addresses.

Simplified Finder A user environment featuring panels and large icons that provide novice users with an easy-to-navigate interface. Mounted volumes or media to which users are allowed access appear on panels instead of on the standard desktop.

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/CIFS Server Message Block/Common Internet File System. 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. Windows services use SMB/CIFS 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 usually is 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 Transport Level Security (TLS).

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 (different floors of a building, for example) or by usage (all eighth-grade students, for example). The use of subnets simplifies administration. See also **IP subnet**.

system-less client A computer that doesn't have an operating system installed on its local hard disk. System-less computers can start up from a disk image on a NetBoot server.

TCP Transmission Control Protocol. A method used along with the Internet Protocol (IP) to send data in the form of message units between computers over the Internet. IP takes care of handling the actual delivery of the data, and TCP takes care of keeping track of the individual units of data (called packets) into which a message is divided for efficient routing through the Internet.

Tomcat The official reference implementation for Java Servlet 2.2 and JavaServer Pages 1.1, two complementary technologies developed under the Java Community Process.

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 in 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.

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 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.

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 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 while a 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.

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