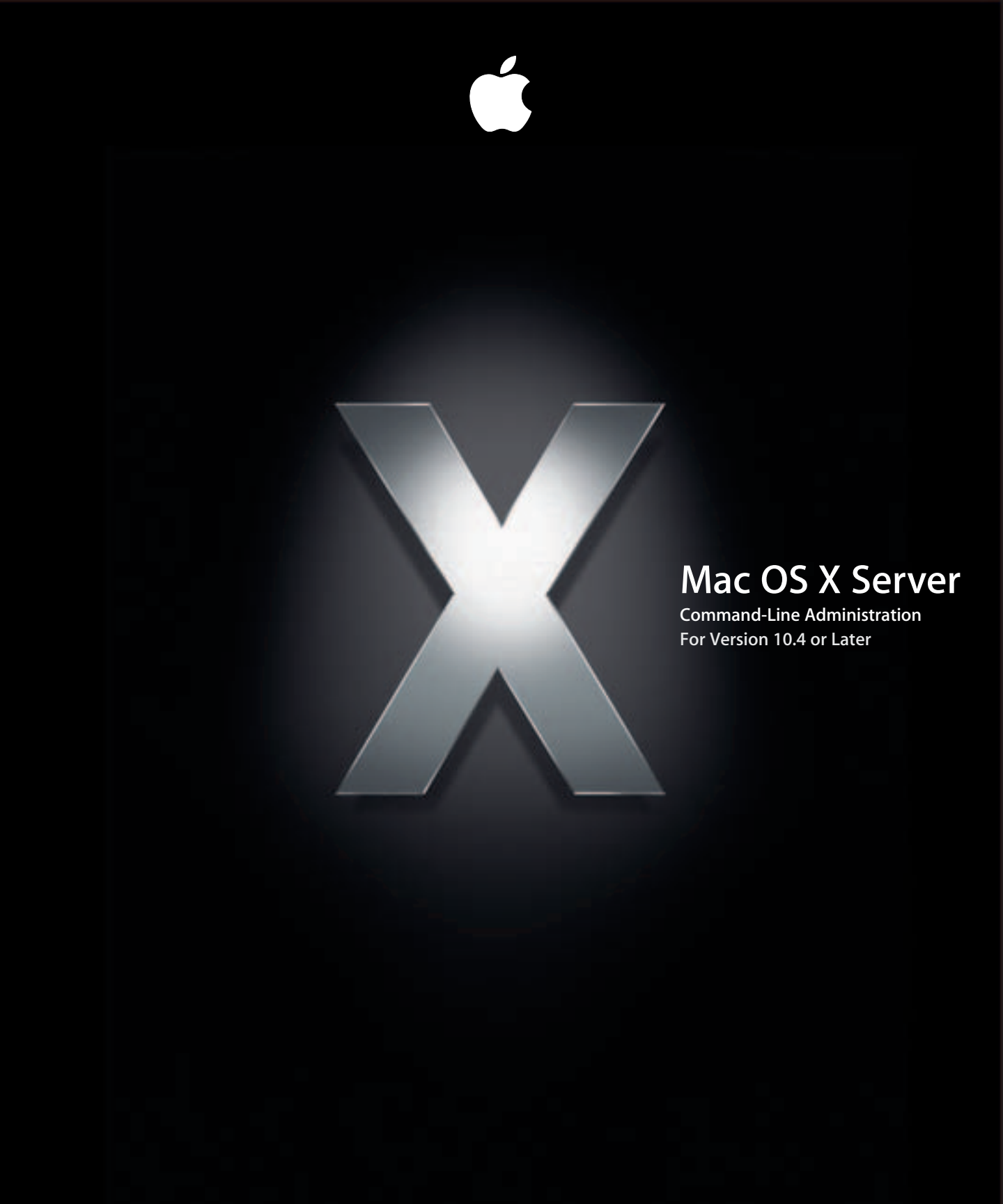

Mac OS X Server

Command-Line Administration
For Version 10.4 or Later

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

019-0160/03-24-2005

Contents

Preface 13 **About This Guide**
13 What Does This Guide Cover?
13 What's New?
14 Notation Conventions
14 Summary
14 Commands and Other Terminal Text
14 Command Parameters and Options
15 Default Settings
15 Commands Requiring Root Privileges

Chapter 1 17 **Typing Commands**
17 Using Terminal
18 Correcting Typing Errors
18 Repeating Commands
18 Including Paths Using Drag-and-Drop
19 Commands Requiring Root Privileges
20 Sending Commands to a Remote Server
20 Sending a Single Command
21 How SSH Works
21 Password-Less Logins Using SSH keys
22 Updating SSH Key Fingerprints
22 What is an SSH Man-in-The-Middle Attack?
23 Controlling Access to SSH Service
23 Notes on Communication Security and `servermgrd`
23 Using Telnet
24 Getting Onscreen Help for Commands
25 Notes About Specific Commands and Tools
25 `serversetup`
25 `serveradmin`

Chapter 2 27 **Installing Server Software and Finishing Basic Setup**
27 Installing Server Software
27 Locating Servers for Installation

28 Automating Server Setup
28 Creating a Configuration File Template
29 Creating Customized Configuration Files from the Template File
32 Naming Configuration Files
32 Storing a Configuration File in an Accessible Location
33 Remote Configuration of The Server From The Command-Line
33 Changing Server Settings
33 Viewing, Validating, and Setting the Software Serial Number
34 Updating Server Software
35 Moving a Server

Chapter 3 37 **Restarting or Shutting Down a Server**
37 Restarting a Server
37 Examples
37 Automatic Restart
38 Changing a Remote Server's Startup Disk
38 Shutting Down a Server
38 Examples
38 Open Firmware NVRAM Variables
38 Example
39 `launchd` Replaces `watchdog`

Chapter 4 41 **Setting General System Preferences**
41 Computer Name
41 Viewing or Changing the Computer Name
41 Date and Time
42 Viewing or Changing the System Date
42 Viewing or Changing the System Time
42 Viewing or Changing the System Time Zone
43 Viewing or Changing Network Time Server Usage
43 Energy Saver Settings
43 Viewing or Changing Sleep Settings
43 Viewing or Changing Automatic Restart Settings
44 Power Management Settings
45 Startup Disk Settings
45 Viewing or Changing the Startup Disk
45 Sharing Settings
45 Viewing or Changing Remote Login Settings
45 Viewing or Changing Apple Event Response
46 International Settings
46 Viewing or Changing Language Settings
46 Login Settings
46 Disabling the Restart and Shutdown Buttons

Chapter 5 47 **Network Preferences**
47 A Note on `ifconfig`
47 Network Interface Information
47 Viewing Port Names and Hardware Addresses
48 Viewing or Changing MTU Values
48 Viewing or Changing Media Settings
48 Network Port Configurations
49 Creating or Deleting Port Configurations
49 Activating Port Configurations
49 Changing Configuration Precedence
50 TCP/IP Settings
50 Changing a Server's IP Address
50 Viewing or Changing IP Address, Subnet Mask, or Router Address
51 Viewing or Changing DNS Servers
52 Enabling TCP/IP
52 Working with VLANs
53 IEEE 802.3ad Ethernet Link Aggregation
54 AppleTalk Settings
54 Enabling and Disabling AppleTalk
55 Proxy Settings
55 Viewing or Changing FTP Proxy Settings
55 Viewing or Changing Web Proxy Settings
55 Viewing or Changing Secure Web Proxy Settings
56 Viewing or Changing Streaming Proxy Settings
56 Viewing or Changing Gopher Proxy Settings
56 Viewing or Changing SOCKS Firewall Proxy Settings
56 Viewing or Changing Proxy Bypass Domains
56 AirPort Settings
56 Viewing or Changing Airport Settings
57 Computer, Host, and Bonjour Name
57 Viewing or Changing the Computer Name
57 Viewing or Changing the Local Hostname
58 Viewing or Changing the Bonjour Name
58 Preference Files and `configd`

Chapter 6 61 **Working With Disks and Volumes**
61 Disks and Partitions
61 Mounting and Unmounting Volumes
61 Mounting Volumes
62 Unmounting Volumes
62 Checking for Disk Problems
62 Monitoring Disk Space
62 `diskspacemonitor`

63 df
63 Reclaiming Disk Space Using Log Rolling Scripts
64 Erasing, Partitioning, and Formatting Disks
64 pdisk
65 newfs
65 newfs_hfs
65 disktool
65 diskutil
66 Managing Disk Journaling
66 Checking to See if Journaling is Enabled
66 Turning on Journaling for an Existing Volume
67 Enabling Journaling When You Erase a Disk
67 Disabling Journaling
67 Setting Up a Case-Sensitive HFS+ File System
68 Enabling and Disabling Spotlight
68 Enabling and Disabling Indexing
69 Managing RAID Volumes
69 View a List of Available RAID Sets
69 Create an Unpaired Mirrored RAID From a Single Filesystem Disk
69 Repair a Failed Mirror
70 Imaging and Cloning Volumes Using ASR

Chapter 7 71 **Working With Users and Groups**
71 Creating Server Administrator Users
72 Importing Users and Groups
73 Creating a Character-Delimited User Import File
76 Checking a Server User's Name, UID, or Password
77 Creating a User's Home Directory
77 Mounting a User's Home Directory
78 Editing Group Records
78 Creating a Group Folder
78 Checking a User's Administrator Privileges
78 lookupd

Chapter 8 79 **Working With File Services**
79 Share Points
79 Listing Share Points
80 Creating a Share Point
81 Modifying a Share Point
81 Disabling a Share Point
81 AFP Service
81 Starting and Stopping AFP Service
81 Checking AFP Service Status

81 Viewing AFP Settings
82 Changing AFP Settings
82 List of AFP Settings
86 List of AFP `serveradmin` Commands
86 Listing Connected Users
87 Sending a Message to AFP Users
87 Disconnecting AFP Users
88 Canceling a User Disconnect
89 Listing AFP Service Statistics
90 Viewing AFP Log Files
90 NFS Service
90 Starting and Stopping NFS Service
90 Checking NFS Service Status
90 Viewing NFS Settings
91 Changing NFS Service Settings
91 FTP Service
91 Starting FTP Service
91 Stopping FTP Service
91 Checking FTP Service Status
91 Viewing FTP Settings
91 Changing FTP Settings
92 FTP Settings
93 List of FTP `serveradmin` Commands
94 Viewing the FTP Transfer Log
94 Checking for Connected FTP Users
94 Windows (SMB/CIFS) Service
94 Starting and Stopping SMB/CIFS Service
94 Checking SMB/CIFS Service Status
94 Viewing SMB/CIFS Settings
95 Changing SMB/CIFS Settings
96 List of SMB/CIFS Service Settings
98 List of SMB/CIFS `serveradmin` Commands
98 Listing SMB/CIFS Users
99 Disconnecting SMB/CIFS Users
99 Listing SMB/CIFS Service Statistics
100 Updating Share Point Information
100 Viewing SMB/CIFS Service Logs
101 ACLs
101 Using `chmod` to Modify ACLs

Chapter 9 103 **Working With Print Service**
104 Commands Used to Manage the Print Service
104 Starting and Stopping Print Service

105 Checking the Status of Print Service
105 Viewing Print Service Settings
105 Changing Print Service Settings
106 Print Service Settings
107 Queue Data Array
110 Print Service `serveradmin` Commands
110 Listing Queues
110 Pausing a Queue
111 Listing Jobs and Job Information
111 Holding a Job
112 Viewing Print Service Log Files

Chapter 10 113 **Working With NetBoot Service and System Images**
113 Commands to Manage NetBoot Service
113 Starting and Stopping NetBoot Service
113 Checking NetBoot Service Status
113 Viewing NetBoot Settings
114 Changing NetBoot Settings
114 NetBoot Service Settings
114 General Settings
115 Storage Record Array
115 Filters Record Array
116 Image Record Array
117 Port Record Array
117 Updating an Image
118 Booting From an Image
118 Working With System Images
118 Using `hdiutil` to Work With System Images
119 Using `asr` To Restore System Images
119 Choosing a Boot Device Using `systemsetup`

Chapter 11 121 **Working With Mail Service**
121 Mail Service Introduction
121 Postfix
122 Cyrus
122 Mailman
123 Commands To Manage Mail Service
123 Starting and Stopping Mail Service
123 Checking the Status of Mail Service
123 Viewing Mail Service Settings
123 Changing Mail Service Settings
124 Mail Service Settings
136 Mail `serveradmin` Commands

136 Listing Mail Service Statistics
137 Viewing the Mail Service Logs
138 Backing Up the Mail Files
139 Reconstructing the Mail Database
140 Setting Up SSL for Mail Service
140 Generating a CSR and Creating a Keychain
142 Obtaining an SSL Certificate
142 Importing an SSL Certificate Into the Keychain
142 `certadmin`
143 Creating a Passphrase File

Chapter 12 145 **Working With Web Technologies**
145 Introduction
146 Commands for Managing Web Service
146 Starting and Stopping Web Service
146 Checking Web Service Status
146 Viewing Web Settings
147 Changing Web Settings
147 `serveradmin` and Apache Settings
147 Changing Settings Using `serveradmin`
148 Web `serveradmin` Commands
148 Listing Hosted Sites
148 Viewing Service Logs
149 Viewing Service Statistics
150 Example Script for Adding a Website
151 Performance Tuning and `ab`
152 Tomcat
152 JBoss
153 MySQL

Chapter 13 155 **Working With Network Services**
155 DHCP Service
155 Starting and Stopping DHCP Service
155 Checking the Status of DHCP Service
155 Viewing DHCP Service Settings
156 Changing DHCP Service Settings
156 DHCP Service Settings
157 DHCP Subnet Settings Array
159 Adding a DHCP Subnet
160 Adding a DHCP Static Map
161 List of DHCP `serveradmin` Commands
161 Viewing the DHCP Service Log
162 DNS Service

162 Starting and Stopping the DNS Service
162 Checking the Status of DNS Service
162 Viewing DNS Service Settings
162 Changing DNS Service Settings
162 DNS Service Settings
163 List of DNS `serveradmin` Commands
163 Viewing the DNS Service Log
163 Listing DNS Service Statistics
164 `xinetd`
164 IP Forwarding
165 Firewall Service
165 Starting and Stopping Firewall Service
165 Checking the Status of Firewall Service
165 Viewing Firewall Service Settings
165 Changing Firewall Service Settings
166 Firewall Service Settings
167 Defining Firewall Rules
169 `ipfilter` Rules Array
169 Firewall `serveradmin` Commands
170 Viewing Firewall Service Log
170 Using Firewall Service to Simulate Network Activity
170 NAT Service
170 Starting and Stopping NAT Service
170 Checking the Status of NAT Service
170 Viewing NAT Service Settings
171 Changing NAT Service Settings
171 NAT Service Settings
172 NAT `serveradmin` Commands
172 Port Mapping
173 Viewing the NAT Service Log
173 VPN Service
173 Starting and Stopping VPN Service
173 Checking the Status of VPN Service
174 Viewing VPN Service Settings
174 Changing VPN Service Settings
175 List of VPN Service Settings
178 List of VPN `serveradmin` Commands
178 Viewing the VPN Service Log
178 Site-to-Site VPN
179 Configuration
180 Adding a VPN Key Agent User
180 IP Failover
180 Requirements

181 Failover Operation
181 Enabling IP Failover
182 Configuring IP Failover
184 Enabling PPP Dial-In

Chapter 14 185 **Working With Open Directory**
185 Overview
185 General Directory Tools
185 Testing Your Open Directory Configuration
185 Modifying an Open Directory Node
185 Testing Open Directory Plugins
186 Registering URLs With Service Location Protocol (SLP)
186 Changing Open Directory Service Settings
187 LDAP
187 Configuring LDAP
189 A Note on Using `ldapsearch`
189 Idle Rebinding Options
189 Additional Information About LDAP
190 NetInfo
190 Configuring NetInfo
190 Password Server
190 Working With the Password Server
190 Viewing or Changing Password Policies
190 Enabling or Disabling Authentication Methods
191 Kerberos and Single Sign-On
191 Backing Up the Kerberos Database
192 Principal Management
193 Directory Service Tools
193 `dscl`
194 `lookupd`
194 `dseditgroup`
195 `dsconfigldap`
195 `dsconfigad`

Chapter 15 197 **Working With QuickTime Streaming Server**
197 Starting QTSS Service
197 Stopping QTSS Service
197 Checking QTSS Service Status
198 Viewing QTSS Settings
198 Changing QTSS Settings
199 QTSS Settings
202 QTSS `serveradmin` Commands
202 Listing Current Connections

203 Viewing QTSS Service Statistics
204 Viewing Service Logs
204 Forcing QTSS to Re-Read its Preferences
205 Preparing Older Home Directories for User Streaming
205 Security
205 Resetting the Streaming Server Admin User Name and Password
205 Controlling Access to Streamed Media
206 Creating an Access File
207 What Clients Need to Access Protected Media
208 Adding User Accounts and Passwords
208 Adding or Deleting Groups
208 Making Changes to the User or Group File
208 Manipulating QuickTime and MP4 Movies
209 Creating Reference Movies

Appendix 211 **PCI RAID Card Command Reference**
212 `megaraid` Commands Functions

Glossary 215

Index 225

About This Guide

What Does This Guide Cover?

Beneath the appealing, easy-to-use interface of Mac OS X is a rock-solid foundation that is engineered for stability, reliability, and performance. This foundation 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 4.4BSD (Berkeley Software Distribution), 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 book focuses on commands developed by Apple to allow administrators to perform GUI functions from the command-line. The book highlights BSD commands that were modified or extended to support Apple specific functionality. Finally, important commands commonly used by UNIX system administrators will be covered to the extent possible.

Note that this manual is not a definitive guide to BSD commands that by virtue of Darwin's definition are also Darwin commands. For a more extensive treatment of such commands, consult a book that deals specifically with BSD.

System administrators should keep up with discussion forums since they provide a wealth of knowledge and shared expertise. Discussion lists on www.apple.com/support along with kbase articles are especially recommended. There are a variety of other useful sites, such as afp548.com, macosxhints.com, and macosxsecurity.com.

What's New?

This update of the Command-Line administration guide covers new features in Tiger Server and augments some sections from the previous edition. Wherever applicable, an effort has been made to explain the context, benefits, and disadvantages of using certain command-line tools.

Notation Conventions

The following conventions are used throughout this book.

Summary

Notation	Indicates
`monospaced` font	A command or other terminal text
`$`	A shell prompt
`[text_in_brackets]`	An optional parameter
`(one|other)`	Alternative parameters (type one or the other)
`underlined`	A parameter you must replace with a value
`[...]`	A parameter that may be repeated
`<anglebrackets>`	A displayed value that depends on your server configuration

Commands and Other Terminal Text

Commands or command parameters that you might type, 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
```

However, to use this command, type it without the dollar sign and the space in a Terminal window, then press the Return key. (Terminal is found in the Applications folder.)

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 Type as Shown

If you need to type 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 the above example, type the entire line as shown (without the `$` and space).

Parameter Values You Provide

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

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

In the above example, you need to replace `hh` with the hour and `mm` with the minute, as shown in the previous example.

Optional Parameters

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

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

To use the command in the above example, type either `doit` or `doit -w later`. The result might vary but the command will be performed either way.

Alternative Parameters

If you need to type 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 the command, you must type 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 other choices you've made (such as the name or IP address of your server, for example), 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`.

Typing Commands

1

How to use Terminal to execute commands, connect to a remote server, and view online information about commands and utilities.

To access a UNIX shell command prompt, open the Terminal application. In Terminal, you can use the `ssh` command to log in to other servers. You can use the `man` command to view online documentation for most common commands.

Using Terminal

To enter shell commands or run server command-line tools and utilities, 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 (in /Applications/Utilities).

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

```
server1:~ admin$
```

you're logged in to a computer named "server1" as the user named "admin" and your current directory is the admin's home directory (`~`).

Throughout this manual, wherever a command is shown as you might type it, the prompt is abbreviated as `$`.

To type a command:

- Wait for a prompt to appear in the Terminal window, then type the command and press Return.

If you get the message `command not found`, check your spelling. If the error recurs, the program you're trying to run might not be in your default search path. Add the path before the program name or change your working directory to the directory that contains the program. For example:

```
[server:/] admin$ serversetup -getAllPort
serversetup: Command not found.
[server:/] admin$ /System/Library/ServerSetup/serversetup -getAllPort
1
Built-in Ethernet
[server:/] admin$ cd /System/Library/ServerSetup
[server:/System/Library/ServerSetup] admin$ ./serversetup -getAllPort
1
Built-in Ethernet
[server:/System/Library/ServerSetup] admin$ cd /
[server:/] admin$ PATH = "$PATH:/System/Library/ServerSetup"
[server:/] admin$ serversetup -getAllPort
1
Built-in Ethernet
```

Correcting Typing Errors

To correct a typing error before you press Return to issue the command, use the Delete key or press Control-H to erase unwanted characters and retype.

To ignore what you have typed and start again, press Control-U.

Repeating Commands

To repeat a command, press Up-Arrow until you see the command, then press Return.

To repeat a command with modifications, press Up-Arrow until you see the command, press Left-Arrow or Right-Arrow to skip over parts of the command you don't want to change, press Delete to remove characters, type regular characters to insert them, then press Return to execute the command.

Including Paths Using Drag-and-Drop

To include a fully qualified file name or directory path in a command, stop typing where the item is required in the command and drag the folder or file from a Finder window into the Terminal window.

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 issue a single command as the root user, begin the command with `sudo`. 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 type `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 are certain about what you're doing. Logging in as an administrative user and using `sudo` selectively might prevent you from making unintended changes.

Throughout this guide, commands that require root privileges begin with `sudo`.

Sending Commands to a Remote Server

Secure Shell (SSH) lets you send secure, encrypted commands to a server over the network remotely, as if you were using the server's console. You can use the `ssh` command in Terminal to open a command-line connection to a remote server. While the connection is open, commands you type are performed on the remote server.

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

To open a connection to a remote server:

1. Open Terminal.
2. Type the following command to log in to the remote server:

   ```
   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. For example:

   ```
   ssh -l admin 10.0.1.2
   ```

3. If this is the first time you've connected to the server, you're prompted to continue connecting after the remote computer's RSA fingerprint is displayed. Type `yes` and press Return.
4. When prompted, type the user's password (the user's password on the remote server) and press Return.

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

   ```
   [10.0.1.2:~] admin$
   ```

5. To send a command to the remote server, type the command and press Return.

To close a remote connection:

- Type `logout` and press Return.

Sending a Single Command

You can authenticate and send a command using a single typed line by appending the command you want to execute to the basic `ssh` command.

For example, to delete a file you could type:

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

or

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

You're prompted for the user's password.

How SSH Works

SSH works similar to SSL by setting up encrypted channels using public and private keys. The following is a description of an SSH session.

- The client and server exchange their public keys. If the client machine has never encountered a given public key before, both SSH and most web browsers ask the user whether to accept the unknown key.
- The client and the server use the public keys to negotiate a session key that is used to encrypt all subsequent session data.
- The server attempts to authenticate the client using RSA or DSA certificates. If this is not possible, the client is prompted for a standard user name/password combination.
- After successful authentication the session begins: either a remote shell, a secure file transfer, a remote command, or so on, is begun over the encrypted tunnel.

You should note the following SSH tools:

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

Password-Less Logins Using SSH keys

There are two main methods for SSH authentication: the standard user name and password, and Identity key pair.

Identity key pair authentication allows you to log in to the server without having to supply a password. Here is how it works. You generate a private and public key associated with a username to establish that user's authenticity. When you attempt to log in as that user the user name is sent to the server. Next, the server looks in the user's .ssh directory for the user's public key. 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 this happens the user is logged in without the need for a password. This is especially useful when automating remote scripts.

To generate the public key pair use the following command on your client machine:

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

When prompted for a passphrase and verification, press Return both times without entering a passphrase. Copy the resultant public key to the user's home directory in .ssh/ on the server machine. The next time you log into the server from the client machine you won't need to enter a password.

Updating SSH Key Fingerprints

The first time you connect to a remote server using SSH, the local computer asks if it can add the remote server's "fingerprint" (a security 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. Make 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 that 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
- Perform a clean install of the server software
- Start up from a Mac OS X Server CD

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 server are instead routed to the attacker who impersonates the server to the client and the client to the server. Here's a typical scenario: A user connects to the server using SSH. By means of spoofing techniques, the attacker poses as the server and receives the information from the client. The attacker then relays the information to the intended server, receives a response and then relays the server's response to the client. 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 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 of the server 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 to Mac OS X Server using the ssh command in Terminal. Users with server administrator privileges are always allowed to open a connection using SSH. The ssh command uses the Secure Shell (SSH) service. For information on controlling access to the SSH service, see the Open Directory administration guide.

Notes on Communication Security and `servermgrd`

When you use the Server Admin GUI application or the `serveradmin` command-line tool, you're communicating with a local or remote `servermgrd` process.

- `servermgrd` uses SSL for encryption and client authentication but not for user authentication, which uses HTTP basic authentication along with Directory Services.
- `servermgrd` uses a self-signed (test) SSL certificate installed by default in /etc/servermgrd/ssl.crt/. You can replace this with an actual certificate.
- The default certificate format for SSLeay/OpenSSL is PEM, which actually is Base64 encoded DER with header and footer lines (from www.modssl.org).
- `servermgrd` checks the validity of the SSL certificate only if the "Require valid digital signature" option is checked in Server Admin preferences. If this option is enabled, the certificate must be valid and not expired or Server Admin will refuse to connect.
- The SSLOptions and SSLRequire settings determine what SSL encryption options are used. By default, they're set as shown below but can be changed at any time by editing /etc/servermgrd/servermgrd.conf, port 311.

```
SSLCertificateFile /private/etc/servermgrd/ssl.crt/server.crt
SSLCertificateKeyFile /private/etc/servermgrd/ssl.key/server.key
SSLCipherSuite
   ALL:!ADH:!EXPORT56:RC4+RSA:+HIGH:+MEDIUM:+LOW:+SSLv2:+EXP:+eNULL
SSLOptions +StdEnvVars
```

Using Telnet

Because it isn't as secure as SSH, Telnet access isn't enabled 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, username 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 programs can effortlessly sniff the network and take possession of your user name and password. If you run something as superuser during your Telnet session, your super user account will be compromised as well.

Getting Onscreen Help for Commands

Onscreen help is available for most commands and utilities.

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

To view onscreen information about a command, try one of the following:

- Type the command without any parameters or options. This will often pop up a list of options and parameters you can use with the command. For example:

```
$ sudo serveradmin
```

- Type `man command`, where `command` is the command you're curious about. This usually displays detailed information about the command, its options, parameters, and proper use. For example:

```
$ man serveradmin
```

  For help using the `man` command, type:

```
$ man man
```

- Type the command followed by a `-help`, `-h`, `--help`, or `help` parameter. For example:

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

Additionally, you should check updates and articles published on the Apple site. AppleCare Service & Support website at www.apple.com/support/ routinely publishes known issues and their resolutions as well as solutions to common problems. You can also check for new onscreen Mac OS X Server help topics that update this guide. To view new help topics, make sure your server or administrator computer is connected to the Internet. Choose Help > Mac Help in the Finder, choose Library > Mac OS X Server Help in Help Viewer, and then click "Late-breaking news."

Notes About Specific Commands and Tools

serversetup

The `serversetup` utility is located in /System/Library/ServerSetup. To run this command, you can type the full path, for example:

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

Or, if you want to use the utility to perform several commands, you can change your working directory and type a shorter command:

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

Or add the directory to your search path for this session and type an even shorter command:

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

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

serveradmin

You can use the `serveradmin` tool to perform many service-related tasks. You'll see it used throughout this manual.

Determining Whether a Service Needs to be Restarted

Some services need to be restarted after you change certain settings. If a change you make using a service's `writeSettings` command requires that you 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.

Installing Server Software and Finishing Basic Setup

2

Commands you can use to install, set up, and update Mac OS X Server software on local or remote computers.

Installing Server Software

You can use the command-line installer `/usr/sbin/installer` to install Mac OS X Server or other software on a computer. You can use the `installer` command locally or remotely. The command-line installer requires at least two arguments: the package to install, and the target destination of the installed package. For a standard install, your target would be the root drive. Here is an example installation command:

```
install -pkg OSInstall.mpkg -target /
```

Other useful options include:

- `lang`—The OS 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.

For more information, see the man page.

Locating Servers for Installation

If you are installing a remote server from Terminal, you will first want to establish an SSH session as the root user with the target server. To do so, you need the remote server's working IP address and serial number. The serial number will be on a label on the server. You can use the `sa_srchr` command to identify all servers ready for installation on your subnet.

The `sa_srchr` command uses the broadcast address 224.0.0.1 to request a response (via `sa_rspndr`) from all computers ready for installation or set up. The command is:

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

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 response looks like this:

```
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. Note that for this to work, you need to have booted the computer form the installation CD.

Automating Server Setup

Normally, when you install Mac OS X Server on a computer and restart, the Server Assistant opens and asks you to provide the basic information necessary to get the server up and running (for example, the name and password of the administrator user, the TCP/IP configuration information for the server's network interfaces, and how the server uses directory services). You can automate this initial setup task by providing a configuration file that contains these settings. Servers starting up for the first time look for this file and use it to complete initial server setup without user interaction.

Creating a Configuration File Template

An easy way to prepare configuration files to automate the setup of a group of servers is to start with a file saved using the Server Assistant. You can save the file as the last step when you use the Server Assistant to set up the first server, or you can run the Server Assistant later to create the file. You can then use that first file as a template for creating configuration files for other servers. You can edit the file directly or write scripts to create customized configuration files for any number of servers that use similar hardware.

To save a template configuration file during server setup:

1. In the final pane of the 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.

   So you can later edit the file, don't select "Save in Encrypted Format."
3. Choose a location to save the file and click Save.

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

1. Open the Server Assistant (in /Applications/Server).
2. In the Welcome pane, choose "Save setup information in a file or directory record" and click Continue.

3 Enter settings on 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.

So you can later edit the file, don't select "Save in Encrypted Format."

5 Choose a location to save the file and click Save.

Creating Customized Configuration Files from the Template File

After you create a template 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 servers.

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

The following example shows the basic structure and contents of a configuration file for a server with the following configuration:

- An administrative 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

```
<?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>a-123-bcd-456-efg-789-hij-012-klm-345-n</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. This is one reason you should start from a template configuration file created on a computer similar to those you plan to set up.

Naming Configuration Files

The Server Assistant recognizes configuration files with these names:

- `MAC-address-of-server.plist`
- `IP-address-of-server.plist`
- `hardware-serial-number-of-server.plist`
- `full-host-name-of-server.plist`
- `generic.plist`

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

Storing a Configuration File in an Accessible Location

The Server Assistant looks for configuration files in the following locations:

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

where `vol` is any device volume mounted in the /Volumes directory.

Devices you can use to provide configuration files include:

- A partition on one of the server'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 directory

Remote Configuration of The Server From The Command-Line

It's possible to remotely configure the server from the command-line. However, it's a complicated process. Finding another Macintosh to remotely run Server Assistant on it is usually worth the effort. But if you do decide to set up your server from the command-line, you'll need the following tools:

- `nicl`—Use `nicl` to create an admin account. `nicl` works at the level of individual fields, so creating a user requires that you know all of the NetInfo fields and the associated data for a user account.
- `systemsetup`—Use `systemsetup` to set a number of system-wide preferences. If you were going through the Setup Assistant, you would have to select the proper Keyboard and time zone. systemsetup can configure both these preferences, and more.
- `networksetup`—Anything that you can configure in the Network pane of System Preferences can also be configured using `networksetup`.

You can view the man pages for these tools for more information. You can also check their usage in the later chapters of this guide.

Changing Server Settings

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

For information on changing general system preferences, see Chapter 4, "Setting General System Preferences," on page 41.

For information on changing network settings, see Chapter 5, "Network Preferences," on page 47.

For information on changing service-specific settings, see the chapter that covers the service.

Viewing, Validating, and Setting the Software Serial Number

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

To display the server's software serial number:

```
$ serversetup -getServerSerialNumber
```

To set the server software serial number:

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

Parameter	Description
serialnumber	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:

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

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

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:

```
/System/Library/ServerSetup/serversetup -issitelicensedserialnumber
```

Updating Server Software

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

To check for available updates:

```
$ softwareupdate --list
```

To install an update:

```
$ 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.

To view command help:

```
$ softwareupdate --help
```

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 directories 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 that other computers depend on.
- Wait to import final user accounts. Limit accounts to test accounts so you minimize the user-specific network information (such as home directory 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 directories on the server. See "Changing a Server's IP Address" on page 50. 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.

Restarting or Shutting Down a Server

3

Commands you can use to shut down or restart a local or remote server.

Restarting a Server

You can use the `reboot` or `shutdown -r` command to restart a server at a specific time. For more information, see the man pages.

Examples

To restart the local server:

```
$ shutdown -r now
```

To restart a remote server immediately:

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

To restart a remote server at a specific time:

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

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

Automatic Restart

You can also use the `systemsetup` command to set up the server to start automatically after a power failure or system freeze. See "Viewing or Changing Automatic Restart Settings" on page 43.

Changing a Remote Server's Startup Disk

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

To change the startup disk:

Log in to the remote server using SSH and type:

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

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

For information on using SSH to log in to a remote server, see "Sending Commands to a Remote Server" on page 20.

Shutting Down a Server

You can use the `shutdown` command to shut down a server at a specific time. For more information, see the man page.

Examples

To shut down a remote server immediately:

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

To shut down the local server in 30 minutes:

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

Parameter	Description
server	The IP address or DNS name of the server.

Open Firmware NVRAM Variables

You can use the nvram command to manipulate Open Firmware NVRAM variables. Note that if you modify a value with nvram, the value would be saved only if the computer cleanly restarts or shuts down. For more information on nvram review the man page.

Example

To view the different NVRAM variables:

```
$ nvram -p
```

launchd Replaces watchdog

In the pervious version of Mac OS X, a daemon called watchdog watched critical services and immediately started them once they failed or once a system restarted either from a crash or from a reboot. The watchdog daemon relied on a configuration file watchdog.conf in /etc.

In Mac OS X Server 10.4, watchdog has been replaced by launchd. The launchd daemon manages other daemons, both for the system as a whole and for individual users. The launchd daemon allows you to configure daemons to launch on demand, based on criteria specified in their respective XML property lists.

During the boot process launchd is the first process invoked by the kernel to run and setup the rest of the system. In Darwin it is preferable to have your daemon launch via 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 highly encouraged to work with launchd to implement whatever changes they require and not modify rc or create a SystemStarter Startup Item. The rc command script will be phased out in the future.

The configuration files are located in the following directories:

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

Setting General System Preferences

4

Commands you can use to set system preferences, usually set using the System Preferences GUI application.

Computer Name

You can use the `systemsetup` command to view or change a server's 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.

Viewing or Changing the Computer Name

To display the server's computer name:

```
$ sudo systemsetup -getcomputername
```

or

```
$ sudo networksetup -getcomputername
```

To change the computer name:

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

or

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

Date and Time

You can use the `systemsetup` or `serversetup` command to view or change:

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

These settings would otherwise 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
```

Energy Saver Settings

You can use the `systemsetup` command to view or change a server's energy saver settings, which would otherwise be set 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)
```

Power Management Settings

You can use the `pmset` command 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

`pmset` allows you to configure different settings for the different power modes. There are four flags you can use: `-a`, `-b`, `-c`, `-u`. (`-b`) apply the settings to battery operation, (`-c`) to charger (wall power), UPS (`-u`) or all (`-a`).

To set disk spindown timer for all modes of operation you would use:

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

Parameter	Description
minutes	Must be a multiple of 30 seconds.

To display the current settings:

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

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

Startup Disk Settings

You can use the `systemsetup` command to view or change a server's computer startup disk, which would otherwise be set using the Startup Disk pane of System Preferences.

Viewing or Changing the Startup Disk

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
```

Sharing Settings

You can use the `systemsetup` command to view or change settings that would otherwise 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 23.

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)
```

International Settings

You can use the `serversetup` command to view or change language settings that would otherwise be set using the Sharing pane of System Preferences.

Viewing or Changing Language Settings

To view the current primary language:

```
$ serversetup -getPrimaryLanguage
```

To view the installed primary language:

```
$ serversetup -getInstallLanguage
```

To change the install language:

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

To view the script setting:

```
$ serversetup -getPrimaryScriptCode
```

Login Settings

Disabling the Restart and Shutdown Buttons

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

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

0 disables the buttons.

1 enables the buttons.

To view the current setting:

```
$ serversetup -getDisableRestartShutdown
```

Network Preferences

5

Commands you can use to change a server's network settings.

A Note on `ifconfig`

Mac OS X Server includes the standard UNIX tool for configuring network interfaces, `ifconfig`. 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. If you use `ifconfig`, your system will be out of sync and will revert back to the contents of preferences.plist after a reboot. You can still use `ifconfig` to view the entire interface configuration. This is particularly beneficial when your system is using an auto-negotiated Ethernet connection.

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

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 "Network Port Configurations" on page 48.

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

You can use these commands to change the maximum transmission unit (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] [...]
```

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 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]
    [...]
```

TCP/IP Settings

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` command.

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 directory system, you must include the path to the directory domain (directory node). For a standalone server, type "-" 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.

For more information or examples, see the man page.

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

3 Restart the server.

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

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

Important: Changing a server's IP address isn't as simple as changing the TCP/IP settings. You must first run the `changeip` utility to make sure necessary changes are made throughout the system. See "Changing a Server's IP Address" on page 50.

To list TCP/IP settings for a configuration:

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

Example:

```
$ 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

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

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

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

`ifconfig`

Apple introduced the implementation of the IEEE 802.3ad Ethernet Link Aggregation standard as part of the `ifconfig` tool. 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 applications and externally to other clients on the Internet. Any application or server that rely on your IP address will continue to work seamless without any modifications. The advantages of aggregation are 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, 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 re-routed to use one of the remaining ports. This failover is completely transparent to the application software using the connection. This feature provides increased bandwidth and automatic failover for the server environment.

To add an Ethernet interface to a bond virtual device (pseudo device) use the following command:

```
$ 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 re-programmed 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 again 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) use the following command:

```
$ 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 802.3ad Link Aggregation state, use the `-b` option.

For more information review the man page of `ifconfig`.

networksetup

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
```

AppleTalk Settings

Enabling and Disabling 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
```

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 [...]
```

AirPort Settings

Viewing or Changing 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]
```

Computer, Host, and Bonjour Name

Viewing or Changing the Computer Name

To display the server's 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
```

Viewing or Changing the Local Hostname

To display the server's local hostname:

```
$ serversetup -getHostname
```

To change the server's local hostname:

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

Viewing or Changing 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 a `0` if the name was changed.

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

Preference Files and `configd`

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 `/Library/Preferences/SystemConfiguration/preferences.plist`.

Network configuration is handled by `configd`, the configuration daemon. `configd` reads the network configuration information and stores it with the current state of the machine's networking information. This storage is in the form of key-value pars. 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 set monitors on them using the scutil command. This can be especially valuable when you are trying to debug your network configuration from the command-line. The following is a typical `scutil` session.

```
$ scutil
> open
```

This opens a session with `configd`

```
> list
```

Each item on the resultant 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 machine. File indicates stored information according to the last time the file was updated.

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

`scutil` stores the information from the get command in a local variable called `d`.

```
> q
```

You can also watch or monitor a variable, such that if its state changes, `scutil` will alert you. For more information read the main page for `scutil` and type help at the `scutil` prompt.

Working With Disks and Volumes

6

Commands you can use to initialize, and test disks and volumes.

Disks and Partitions

Mac OS X, as with standard UNIX, 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/diskn`, 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 system 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.

Mounting and Unmounting Volumes

You can use the `mount_afp` command to mount an AFP volume. For more information, type `man mount_afp` to see the man page.

Mounting Volumes

You can use the `mount` command 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

For more information, see the related man pages.

Unmounting Volumes

You can use the `umount` command to unmount a volume. For more information, see the man page.

Checking for Disk Problems

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

Monitoring Disk Space

`diskspacemonitor`

When you need more vigilant monitoring of disk space than the log rolling scripts provide, you can use the `diskspacemonitor` command-line 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. You can enable it by opening a Terminal window and typing `sudo diskspacemonitor on`. You may be prompted for your password. Type `man diskspacemonitor` for more information about the command-line options.

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 thresholds to use for determining when to take the actions in the scripts. By default, disks are checked every 10 minutes, an alert script executed when disks are 75% full, and a recovery script 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 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 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, you can. 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 server using SSH.

`df`

The tool `/bin/df` 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. 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.

Reclaiming Disk Space Using Log Rolling Scripts

Three predefined scripts are executed automatically 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. Refer to 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` command-line tool to reclaim disk space.

Erasing, Partitioning, and Formatting Disks

The following are command-line tools that you can use to erase, partition, and format disks:

`pdisk`

The `pdisk` command (`/usr/sbin/pdisk`) lets you edit the disk partition table. You can initialize the disk, create partitions, and delete partitions. The `pdisk` command is a menu-driven command-line program, which means that after you launch it, the program prompts you 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. Type `?` 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 have not been implemented.

newfs

You can use the `newfs` command (`/sbin/newfs`) to create UFS volumes. There are many parameters you can set when formatting disks such as block and clump size, as well as b-tree attribute and catalog node sizes. Extreme care should be taken to ensure a successful format when modifying the settings beyond the default. The man page discusses the options in detail.

newfs_hfs

You can use the `newfs_hfs` command (`/sbin/newfs_hfs`) for creating HFS plus volumes. For more information read the man page.

disktool

The `disktool` command (`/usr/sbin/disktool`) allows you to mount, unmount, and rename volumes. `disktool` is also a menu-driven command-line program. To see a full list of functions, type `disktool` (with no arguments) at a command prompt. You can use disktool `-l` command to list the disks currently known and available on the system. If your system is an Xserve, you can use this command to determine which drive is in which bay.

diskutil

You can use `diskutil` to modify, verify, and repair disks. This tool 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 levels 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. Following are some examples of using `diskutil`.

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 with 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).
part1Format	HFS+ or UFS.
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).

is the device name (such as disk0), is , and

Managing Disk Journaling

Checking to See if Journaling is Enabled

You can use the mount command 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)
```

Turning on Journaling for an Existing Volume

You can use the diskutil command 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 command before you turn on journaling.

To enable journaling:

```
$ diskutil enableJournal volume
```

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

Example

```
$ 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` command 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.

Setting Up a Case-Sensitive HFS+ File System

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

For more information, see the man page for `diskutil`.

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 `/etc/hostconfig` 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.

Enabling and Disabling Indexing

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

To enable indexing on a given volume:

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

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

To disable indexing on a given volume:

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

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

See the man page of the `mdutil` command for more information.

Managing RAID Volumes

In addition to standard drive management options, `diskutil` has the ability to manage software RAID volumes. For example, you can create a RAID set by typing:

`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.

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

View a List of Available RAID Sets

`checkRAID device`

Parameter	Description
device	Device file.

Create an Unpaired Mirrored RAID From a Single Filesystem Disk

`diskutil enableRAID mirror device`

Parameter	Description
mirror	Name of the mirror RAID set.

where mirror is the

Repair a Failed Mirror

`diskutil repairMirror device slicenumber fromDisk toDisk`

Parameter	Description
slicenumber	Specifies the slice number to replace.
fromDisk	Specifies the mirror source.
toDisk	Specifies the repaired mirror destination.

Imaging and Cloning Volumes Using ASR

You can use Apple Software Restore (ASR) to copy a disk image onto a volume or 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` utility 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/write disk images.

To image a boot volume:

1. Install and configure Mac OS X on the volume as you want it.
2. Restart from a different volume.
3. Make sure the volume you're imaging has permissions enabled.
4. Use `hditutil` to make a read-write disk image of the volume.
5. Mount the disk image.
6. Remove cache files, host-specific preferences, and virtual memory files. You can find example files to remove on the `asr` man page.
7. Unmount the volume and convert the read-write image to a read-only compressed image.

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

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

   ```
   sudo asr -imagescan compressedimage
   ```

To restore a volume from an image:

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

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

Working With Users and Groups

7

Commands you can use to set up and manage users and groups.

Creating Server Administrator Users

You can use the `serversetup` command to create administrator users for a server. To create regular users, see "Importing Users and Groups" on page 72. `serversetup` 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 a user:

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

The name, short name, and password must be typed in the order shown. If the full name includes spaces, type it in quotes.

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

To create a user with a specific UID:

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

The name, short name, password, and UID must be typed in the order shown. If the full name includes spaces, type it in quotes.

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

To create a user with a specific UID and home directory:

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

The name, short name, password, and UID must be typed in the order shown. If the full name includes spaces, type it in quotes.

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

Importing Users and Groups

You can use the dsimport command to import user and group accounts.

This command is in /Applications/Server/Workgroup Manager.app/Contents/Resources.

For information on the formats of the files you can import, see "Creating a Character-Delimited User Import File" on page 73.

```
$ dsimport (-g|-s|-p) file directory (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.
directory	The path to the Open Directory node 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 directory: O: Overwrite the matching record. M: Merge the records. Empty attributes in the directory 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 directory administrator.
password	The password of the directory 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 73.

Open Directory supports up to 200,000 records. For local NetInfo databases, make sure the file contains no more than 10,000 records.

2 Log in as the administrator of the directory domain into which you want to import accounts.

3 Open the Terminal application and type the `dsimport` command. The tool is located in `/Applications/Utilities/Workgroup Manager.app/Contents/Resources`.

To include the space in the path name, precede it with a backslash (`\`). For example:

```
/Applications/Utilities/Workgroup\ Manager.app/Contents/Resources
    /dsimport -h
```

4 If you want, use the `createhomedir` tool to create home directories for imported users. See "Creating a User's Home Directory" on page 77.

Creating a Character-Delimited User Import File

You can create a character-delimited file by using Workgoup Manager to export accounts from the LDAP directory of an Open Directory master or a NetInfo Domain to 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 directory on the user's computer)
- 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

Here is an example of a record description:

```
0x0A 0x5C 0x3A 0x2C dsRecTypeStandard:Users 7
RecordName Password UniqueID PrimaryGroupID
RealName NFSHomeDirectory UserShell
```

Here is an example of a record encoded using the above description:

```
jim:Adl47E$:408:20:J. Smith, Jr.,
     M.D.:/Network/Servers/somemac/Homes/jim:/bin/csh
```

The record consists of values, delimited by colons. Use a double colon (`::`) to indicate a value is missing.

Here 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 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:

```
jim:Ad147E$:408:20:J. Smith, Jr.,
     M.D.:/Network/Servers/somemac/Homes/jim:/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
```

Here is an example of a record encoded using the description:

```
students:Ad147:88:jones,alonso,smith,wong
```

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.

Note: These tasks apply only to the local directory 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 the directory, `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 the directory, `0` if it isn't.

To see if a UID is already in use:

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

The command displays a `1` if the UID is already in the directory, `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, `0` if it isn't.

To view the names associated with a UID:

```
$ sudo /System/Library/ServerSetup/serversetup -getNamesByID userid
```

No response means UID not valid.

To generate the default UNIX short name for a user long name:

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

The Windows net command for advanced configuration of PDC and AD

Apple provides a command-line `command:net` which is essentially a clone of the Windows net command. The net command enables administrators to perform advanced customization of the PDC and mapping domain privileges to UNIX groups. For more information check the man pages.

Creating a User's Home Directory

Normally, you can create a user's home directory by clicking the Create Home Now button on the Homes pane of Workgroup Manager. You can also create home directory folders using the `createhomedir` tool. Otherwise, Mac OS X Server creates the user's home directory when the user logs in for the first time.

You can use `createhomedir` to create

- A home directory for a particular user (`-u` option)
- Home directories for all users in a directory domain (`-n` or `-l` option)
- Home directories for all users in all domains in the directory search path (`-a` option)

For more information, type `man createhomedir` to view the man page.

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

To create a home directory for a particular user:

```
$ sudo createhomedir [(-a|-l|-n domain)] -u userid
```

To create a home directory for users in the local domain:

```
$ sudo createhomedir -l
```

To create a home directory for users in the local domain:

```
$ sudo createhomedir [(-a|-l|-n domain)] -u userid
```

You can also create a user's home directory using the `serversetup` tool.

To create a home directory for a particular user:

```
$ sudo /System/Library/ServerSetup/serversetup -createHomedir userid
```

The command displays a `1` if the user ID you specify doesn't exist.

Mounting a User's Home Directory

You can use the `mnthome` command to mount a user's home directory. For more information, see the man page.

Editing Group Records

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

Examples

To display the information about a particular group:

```
dseditgroup mygroup
```

To delete a group:

```
dseditgroup -o delete -d /LDAPv3/ldap.example.com -u username -P
    password groupname
```

For more information about `dseditgroup` review the man page.

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` command to create the actual group folder. You issue the command as follows:

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

For more information see the man page.

Checking a User's Administrator Privileges

To see if a user is a server administrator:

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

The command displays a `0` if the user has administrator privileges, `1` if the user doesn't.

`lookupd`

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 find out more of the `lookupd` commands type `?` and you will be presented with all the different commands you can run `lookupd` with.

Working With File Services

8

Commands you can use to create share points and manage file services.

Share Points

You can use the `sharing` tool to list, create, and modify share points.

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 directory 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 directory, 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 directory. The digits represent, from left to right, AFP, FTP, and SMB/CIFS. 1=shared, 0=not shared.
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 directory 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 on 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 on strict locking, see the file services administration guide.

Examples

```
$ sharing -a /Volumes/100GB/Art
```

Creates a share point named Art, shared using AFP, FTP, and SMB/CIFS, and using the name Art for all three types of clients.

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

Shares the directory named Windows Docs on the disk 100GB. The share point is named WinDocs for server management purposes, but SMB/CIFS users see it as Documents. It's 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 80.

Disabling a Share Point

To disable a share point:

```
$ sharing -r sharepointname
```

Parameter	Description
sharepointname	The current name of the share point.

AFP Service

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, type `serveradmin settings afp` or see "List of AFP Settings" on this page.

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` command.

To change a setting:

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

Parameter	Description
`setting`	An AFP service setting. To see a list of available settings, type `$ sudo serveradmin settings afp` or see "List of AFP Settings" on this page.
`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`

Parameter (afp:)	Description
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 true to force administrative users on Mac OS X to see share points instead of all volumes. Default = yes
adminGetsSp	Set to true to force administrative 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
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 directories. 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

Parameter (afp:)	Description
guestAccess	Allow guest users access to the server. Default = yes
idleDisconnectFlag: adminUsers	Enforce idle disconnect for administrative 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
idleDisconnectTime	Idle time (in minutes) allowed before disconnect. Default = 10
kerberosPrincipal	Kerberos server principal name. Default = "afpserver"
loggingAttributes: logCreateDir	Record directory 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)

Parameter (afp:)	Description
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
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 administrative users super 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 users volume. Default = yes

Parameter (afp:)	Description
useAppleTalk	Don't modify. Internal use only.
useHomeDirs	Default = no

List of AFP serveradmin Commands

In addition to the standard start, stop, status, and settings commands, you can use serveradmin to issue the following service-specific AFP commands.

Command (afp:command=)	Description
cancelDisconnect	Cancel a pending user disconnect. See "Canceling a User Disconnect" on page 88.
disconnectUsers	Disconnect AFP users. See "Disconnecting AFP Users" on page 87.
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 89.
getLogPaths	Display the locations of the AFP service activity and error logs.
sendMessage	Send a text message to connected AFP users. See "Sending a Message to AFP Users" on page 87.
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 "Determining Whether a Service Needs to be Restarted" on page 25.

Listing Connected Users

You can use the serveradmin getConnectedUsers command 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:

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

Output

The following array of settings is 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>
```

Sending a Message to AFP Users

You can use the `serveradmin sendMessage` command 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.
`sessionid`*n*	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 86.

Disconnecting AFP Users

You can use the `serveradmin disconnectUsers` command 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.
`minutes-until`	The number of minutes between the time the command is issued and the users are disconnected.
`sessionid`*n*	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 86.

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 issued.
<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 86.
<status>	A command status code: 0 = command successful

Canceling a User Disconnect

You can use the serveradmin cancelDisconnect command to cancel a disconnectUsers command. Users receive an announcement that they're no longer scheduled to be disconnected.

To cancel a 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 issued 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.

Output

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

Value	Description
<time>	The time at which the command was issued.
<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 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`.

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 `serveradmin getLogPaths` command to see where the current AFP error and activity logs are located.

To display the log paths:

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

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`

NFS 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 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` command to change settings for the NFS service.

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

FTP Service

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 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 Settings

You can change FTP service settings using the `serveradmin` application.

To change a setting:

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

Parameter	Description
setting	An FTP service setting. To see a list of available settings, type `$ sudo serveradmin settings ftp` or see "FTP 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
```

FTP Settings

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

Parameter (`ftp:`)	
`administratorEmailAddress`	Default = `"user@hostname"`
`anonymous-root`	Default = `"/Library/FTPServer/FTPRoot"`
`anonymousAccessPermitted`	Default = `no`
`authLevel`	Default = `"STANDARD"`

`bannerMessage`	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`	Default = `"/Library/FTPServer/FTPRoot"`
`logCommands:anonymous`	Default = `no`
`logCommands:guest`	Default = `no`

Parameter (ftp:)	
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
showWelcomeMessage	Default = yes
welcomeMessage	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 application to manage FTP service.

Command (ftp:command=)	Description
getConnectedUsers	List connected users. See "Checking for Connected FTP Users" on page 94.
getLogPaths	Show location of the FTP transfer log file. See "Viewing the FTP Transfer Log" on page 94.
writeSettings	Equivalent to the standard serveradmin settings command, but also returns a setting indicating whether the service needs to be restarted. See "Determining Whether a Service Needs to be Restarted" on page 25.

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
```

The default location of `log-file` is `/Library/Logs/FTP.transger.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
```

Windows (SMB/CIFS) Service

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 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, type `$ sudo serveradmin settings smb` or see "List of SMB/CIFS Service Settings" on page 96.

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 Settings

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

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, type `$ sudo serveradmin settings smb` or see "List of SMB/CIFS Service Settings" on page 96.
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 96.

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` command to change settings for the SMB/CIFS service.

Parameter (`smb:`)	Description
`adminCommands:homes`	Whether home directories 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` Corresponds to the "Enable virtual share points" checkbox in the Advanced pane of Window service settings in the Server Admin GUI application.

`adminCommands:serverRole`	The authentication role played by the server. Can be set to:
	`"standalone"`
	`"domainmember"`
	`"primarydomaincontroller"`
	Corresponds to the Role pop-up menu in the General pane of Windows service settings in the Server Admin GUI application.

`domain master`	Whether the server is providing domain master browser service. Can be set to: `yes` | `no` Corresponds to the Domain Master Browser checkbox in the Advanced pane of Window service settings in the Server Admin GUI 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) Corresponds to the Code Page pop-up menu on the Advanced pane of Windows service settings in the Server Admin GUI application.

Parameter (`smb:`)	Description
`local master`	Whether the server is providing workgroup master browser service. Can be set to: `yes` | `no` Corresponds to the Workgroup Master Browser checkbox in the Advanced pane of Window service settings in the Server Admin GUI 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) Corresponds to the Log Detail pop-up menu in the Logging pane of Window service settings in the Server Admin GUI application.
map to guest	Whether guest access is allowed. Can be set to: `"Never"` (No guest access) `"Bad User"` (Allow guest access) Corresponds to the "Allow Guest access" checkbox in the Access pane of Window service settings in the Server Admin GUI application.
`max smbd processes`	The maximum allowed number of smb 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 GUI application.
`netbios name`	The server's NetBIOS name. Can be set to a maximum of 15 bytes of UTF-8 characters. Corresponds to the Computer Name field in the General pane of the Windows service settings in the Server Admin GUI 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. Corresponds to the Description field in the General pane of the Windows service settings in the Server Admin GUI application.
`wins support`	Whether the server provides WINS support. Can be set to: `yes` | `no` Corresponds to the WINS Registration "Off" and "Enable WINS server" selections in the Advanced pane of the Windows service settings in the Server Admin GUI application.

Parameter (smb:)	Description
wins server	The name of the WINS server used by the server. Corresponds to the WINS Registration "Register with WINS server" selection and field in the Advanced pane of the Windows service settings in the Server Admin GUI application.
workgroup	The server's workgroup. Can be set to a maximum of 15 bytes of UTF-8 characters. Corresponds to the Workgroup field in the General pane of the Windows service settings in the Server Admin GUI application.

List of SMB/CIFS serveradmin Commands

You can use these commands with the serveradmin tool to manage SMB/CIFS service.

Command (smb:command=)	Description
disconnectUsers	Disconnect SMB/CIFS users. See "Disconnecting SMB/CIFS Users" on page 99.
getConnectedUsers	List users currently connected to an SMB/CIFS service. See "Listing SMB/CIFS Users" on this page.
getHistory	List connection statistics. See "Listing SMB/CIFS Service Statistics" on page 99.
getLogPaths	Show location of service log files. See "Viewing SMB/CIFS Service Logs" on page 100.
syncPrefs	Update the service to recognize changes in share points. See "Updating Share Point Information" on page 100.
writeSettings	Equivalent to the standard serveradmin settings command, but also returns a setting indicating whether the service needs to be restarted. See "Determining Whether a Service Needs to be Restarted" on page 25.

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 to disconnect users.

To list connected users:

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

Output

The following array of settings is displayed for each connected user:

```
smb:usersArray:_array_index:i:disconnectID = <disconnectID>
smb:usersArray:_array_index:i:sessionID = <sessionID>
smb:usersArray:_array_index:i:connectAt = <connect-time>
smb:usersArray:_array_index:i:service = <service>
smb:usersArray:_array_index:i:loginElapsedTime = <login-elapsed-time>
smb:usersArray:_array_index:i:name = "<name>"
```

```
smb:usersArray:_array_index:i:ipAddress = "<ip-address>"
```

Value returned by `getConnectedUsers` (`smb:usersArray:_array_index:`*n*`:`)	Description
`<sessionID>`	An integer that identifies the user session.
`<connect-time>`	The date and time when the user connected to the server.
`<service>`	The share point the user is accessing.
`<login-elapsed-time>`	The elapsed time since the user connected.
`<name>`	The user's name.
`<ip-address>`	The user's IP address.

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
`sessionid`*n*	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 98.

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 `serveradmin getHistory` command to display a log of periodic samples of the number of SMB/CIFS connections. Samples are taken once each minute.

To list samples:

```
$ sudo serveradmin command
smb:command = getHistory
smb:variant = v1
smb:timeScale = scale
```

```
Control-D
```

Parameter	Description
v1	The number of connected users (average during sampling period).
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 `smb:timeScale = 1800`.

Output

```
smb:nbSamples = <samples>
smb:samplesArray:_array_index:0:vn = <sample>
smb:samplesArray:_array_index:0:t = <time>
smb:samplesArray:_array_index:1:vn = <sample>
smb:samplesArray:_array_index:1:t = <time>
[...]
smb:samplesArray:_array_index:i:vn = <sample>
smb:samplesArray:_array_index:i:t = <time>
smb:v1Legend = "CONNECTIONS"
smb: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.

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
```

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`

ACLs

An ACL is a list of access control entries (ACEs), each specifying the permissions to be granted or denied to a group or user, and how these permissions are propagated throughout a folder hierarchy. ACLs in Mac OS X Server let you set file and folder access permissions to 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. For a more expanded discussion of 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 commands 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

Permission	Description
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 directories:

Permission	Description
list	List entries
add_file	Add a file
add_sudirectory	Add a subdirectory
delete_child	Delete an object

To grant a user write permissions for a file:

```
chmod +a "user1 allow write" file1
```

To deny a guest read permissions to a file:

```
chmod +a "guest deny read" file1
```

To view the ACL of a 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
```

For more information view the man page for `chmod`.

Working With Print Service

9

Understanding the Print Process

Apple's printing infrastructure is built on CUPS. CUPS uses open standards such as IPP and PostScript Printer Description files (PPDs). Command-line utilities 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 command-line utilities communicate with the daemon using the Internet Printing Protocol (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 command-line utilities, you modify cupsd.conf or printers.conf, respectively.

To prepare files for printing, cupsd invokes other programs, called filters and backends. These reside in subdirectories 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 need not enable web sharing to use it. You can find the CUPS documentation at this URL.

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, in `/var/log/cups`, include `access_log` containing all HTTP requests processed by CUPS server, `error_log` containing messages from the scheduler (errors, warnings, and so on), `page_log` containing summary of each page sent to a printer.

In addition to using the GUI to add a queue, you can use the `lpadmin` command, or the CUPS web interface. When you add a printer or create a printer pool, you create a CUPS print queue. A PPD file, which defines the attributes of that queue, is placed in `/etc/cups/ppd/`. The name of the PPD file corresponds with the name of the queue (either the name of a printer or the name of a class). PPD is an abbreviation for PostScript Printer Description, but 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 subdirectories. The standard location is in the following folders: `/Library/Printers/PPDs/Contents/Resources/` and `/System/Library/Printers/PPDs/Contents/Resources/`. The `lpadmin` utility can use only PPD files in `/usr/share/cups/model` and its subdirectories.

When you initiate a print job, you generate a CUPS spool file and an IPP attributes file in the folder `/var/spool/cups`. The `lp` or `lpr` command 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 command-line utilities 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 on CUPS and command-line tools specific to CUPS, review the documentation available at: 127.0.0.1:631/documentation.html. You can also read the man pages for the CUPS commands: `accept`, `backend`, `cancel`, `disable`, `enable`, `filter`, `lp`, `lpadmin`, `lpinfo`, `lpoptions`, `lpq`, `lpr`, `lpstat`, and `reject`.

The following section discusses the commands you can issue using the serveradmin command-line tool. These commands conveniently interact with CUPS to implement the desired changes.

Commands Used to Manage the Print Service

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, type `$ sudo serveradmin settings print` or see "Print Service Settings" on page 106.
`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` command 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 107.
`serverLogArchiveSizeMB`	Default = `1`; maximum log size Range = 1–512 MB
`logLevel`	Default = `info`; for details view 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 PAP 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 Panther Printer Sharing, now integrated with X Server.

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:0:sharingEnable`	Default = `no`; `yes` = share printing via the IPP service.
`queuesArray:_array_id:<id>:sharingList:_array_index:1:service`	Default = `"LPR"`; service name for Unix Line Printer.
`queuesArray:_array_id:<id>:sharingList:_array_index:1:sharingEnable`	Default = `no`; `yes` = share printing via the LPR service.
`queuesArray:_array_id:<id>:sharingList:_array_index:2:service`	Default = `"SMB"`; service name for Windows SMB/CIFS.
`queuesArray:_array_id:<id>:sharingList:_array_index:2:sharingEnable`	Default = `no`; `yes` = share printing via the SMB/CIFS service .
`queuesArray:_array_id:<id>:sharingList:_array_index:3:service`	Default = `"PAP"`; service name for AppleTalk/PAP.
`queuesArray:_array_id:<id>:sharingList:_array_index:3:sharingEnable`	Default = `no`; `yes` = share printing via the AppleTalk service.
`queuesArray:_array_id:<id>:shareable`	Default = `yes`. Cannot be changed.
`queuesArray:_array_id:<id>:defaultJobPriority`	Not used. Default = `"NORMAL"`
`queuesArray:_array_id:<id>:printerName`	Default = `"<printer-name>"` Cannot be changed using `serveradmin`.

Parameter (`print:`)	Description
`queuesArray:_array_id:<id>: defaultJobState`	Not used. Default = `"PENDING"`
`queuesArray:_array_id:<id>: printerURI`	Default = `<uri>`; CUPS printer device info. Format depends on type of printer. Cannot be changed using serveradmin.
`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.
`defaultCoverPage`	Name of assigned cover page.
`accessControlMode`	Default = `ALLOWALL`; possible values: `NONE`, `ALLOWLIST`, `ALLOWALL`
`DENYLIST`	Selected Access Control List enforcement setting.

Here is an example of a queue array parameter block:

```
print:queuesArray:_array_id:29D3ECF3-17C8-16E5-A330-
     84CEC733F249:quotasEnforced = no
print:queuesArray:_array_id:29D3ECF3-17C8-16E5-A330-
     84CEC733F249:sharingList:_array_index:0:service = "LPR"
print:queuesArray:_array_id:29D3ECF3-17C8-16E5-A330-
     84CEC733F249:sharingList:_array_index:0:sharingEnable = no
print:queuesArray:_array_id:29D3ECF3-17C8-16E5-A330-
     84CEC733F249:sharingList:_array_index:1:service = "SMB"
print:queuesArray:_array_id:29D3ECF3-17C8-16E5-A330-
     84CEC733F249:sharingList:_array_index:1:sharingEnable = no
print:queuesArray:_array_id:29D3ECF3-17C8-16E5-A330-
     84CEC733F249:sharingList:_array_index:2:service = "PAP"
print:queuesArray:_array_id:29D3ECF3-17C8-16E5-A330-
     84CEC733F249:sharingList:_array_index:2:sharingEnable = no
print:queuesArray:_array_id:29D3ECF3-17C8-16E5-A330-84CEC733F249:shareable =
     yes
print:queuesArray:_array_id:29D3ECF3-17C8-16E5-A330-
     84CEC733F249:defaultJobPriority = "NORMAL"
print:queuesArray:_array_id:29D3ECF3-17C8-16E5-A330-84CEC733F249:printerName
     = "Room 3 Printer"
print:queuesArray:_array_id:29D3ECF3-17C8-16E5-A330-
     84CEC733F249:defaultJobState = "PENDING"
print:queuesArray:_array_id:29D3ECF3-17C8-16E5-A330-84CEC733F249:printerURI
     = "pap://*/Room%203%20Printer/LaserWriter"
print:queuesArray:_array_id:29D3ECF3-17C8-16E5-A330-
     84CEC733F249:registerRendezvous = yes
print:queuesArray:_array_id:29D3ECF3-17C8-16E5-A330-84CEC733F249:printerKind
     = "HP LaserJet 4100 Series "
print:queuesArray:_array_id:29D3ECF3-17C8-16E5-A330-84CEC733F249:sharingName
     = "Room 3 Printer"
```

Print Service `serveradmin` Commands

You can use the following commands with the `serveradmin` application to manage Print service.

Command (`print:command=`)	Description
`getJobs`	List information about the jobs waiting in a queue. See "Listing Jobs and Job Information" on page 111.
`getLogPaths`	Finding the locations of the Print service and job logs. See "Viewing Print Service Log Files" on page 112.
`getQueues`	List Print service queues. See "Listing Queues" on this page.
`setJobState`	Hold or release a job. See "Holding a Job" on page 111.
`setQueueState`	Pauses or release a queue. 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 "Determining Whether a Service Needs to be Restarted" on page 25.

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 this page.

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 110.

For each job, the command lists:

- Document name
- Document size
- Job ID
- Submitting user
- Submitting host
- Job name
- Job state
- Printing protocol
- 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:namesArray:_array_index:0:printer = queue
print:namesArray:_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 110.
`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 this page.

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:namesArray:_array_index:0:printer = queue
print:namesArray:_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/error_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 PAP 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
```

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"
```

Working With NetBoot Service and System Images

10

Commands to Manage NetBoot Service

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` command.

To change a setting:

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

Parameter	Description
`setting`	A NetBoot service setting. To see a list of available settings, type `$ sudo serveradmin settings netboot` or see "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
```

NetBoot Service Settings

General Settings

Use the following parameters with the `serveradmin` command 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 install images. For a description, see "Storage Record Array" on page 115.
`netBootFiltersRecordsArray...`	An array of values for each computer explicitly allowed or disallowed access to images. For a description, see "Filters Record Array" on page 115.
`netBootImagesRecordsArray...`	An array of values for each boot or install image stored on the server. For a description, see "Image Record Array" on page 116.
`netBootPortsRecordsArray...`	An array of values for each server network port used to deliver boot or install images. For a description, see "Port Record Array" on page 117.

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.
`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 Install 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 a 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` command-line 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.

Updating an Image

You can use the installer command 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:

```
$ install -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` command or by booting into open firmware mode.

To boot from an image follow the following steps:

1. Boot into open firmware by clicking (command+option+o+f) as you boot.
2. At the prompt type 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
```

Working With System Images

Using `hdiutil` to Work With System Images

You can use the `hdiutil` command to manipulate disk images. You can perform many functions using `hdiutil` such as creating images, compressing them, mounting and unmouting, resizing them, display image info, burning images on CDs and others. For a complete treatment of `hdiutil` and the commands you can use with it view the man pages.

Here's a sample of what you can do with `hdiutil`:

To verify an image against its internal checksum:

```
hdiutil verify myimage.img
```

To segment an image into three segments:

```
hdiutil segment -segmentSize 10m -o /tmp/aseg 30m.dmg
```

creates 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 command-line tool `asr` 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.

Choosing a Boot Device Using `systemsetup`

You can use the `systemsetup` command 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 type:

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

Working With Mail Service

11

Mail Service Introduction

The Mail services in Mac OS X Server consists of three components, all based on open standards with full support for Internet mail protocols. The components are:

- Postfix, the SMTP mail transfer agent
- Cyrus, which supports IMAP and POP
- Mailman, which provides mailing list management features

These three components combine to form all the required parts of a full-featured mail system. It will enable you to send and receive email to and from the rest of the world, allow your users to view their email, and allow you to have mailing lists.

Postfix

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 programs. Postfix does not even 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 program 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 mindful 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`. For more information about postfix, visit www.postfix.org

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 will 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 program 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.

For more information on Cyrus visit: `asg.web.cmu.edu/cyrus/`.

Mailman

Mailman is a Mailing List service with support for built-in archiving, automatic bounce processing, content filtering, digest delivery, spam filters, and some 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 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`.

Commands To Manage Mail Service

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 mail services.

For information on Postfix, visit www.postfix.org.

For information on Cyrus IMAP/POP, visit asg.web.cmu.edu/cyrus.

You can also use Sherlock or Google to search the web for information on Postfix or Cyrus.

Mail Service Settings

Use the following parameters with the `serveradmin` command 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"`

Parameter (mail:)	Description
postfix:sendmail_path	Default = "/usr/sbin/sendmail"
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 = "+="

Parameter (mail:)	Description
postfix:resolve_dequoted_address	Default = yes
postfix:cleanup_service_name	Default = "cleanup"
postfix:header_address_token_limit	Default = 10240
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"

Parameter (mail:)	Description
postfix:smtpd_timeout	Default = "300s"
postfix:require_home_directory	Default = no
postfix:smtpd_error_sleep_time	Default = "1s"
postfix:helpful_warnings	Default = yes
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"

Parameter (mail:)	Description
postfix:smtpd_sender_restrictions	Default = ""
postfix:delay_warning_time	Default = "0h"
postfix:alias_maps	Default = "hash:/etc/aliases"
postfix:sender_canonical_maps	Default = ""
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

Parameter (mail:)	Description
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
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

Parameter (mail:)	Description
postfix:default_delivery_slot_cost	Default = 5
postfix:default_privs	Default = "nobody"
postfix:smtp_bind_address	Default = no
postfix:nested_header_checks	Default = "$header_checks"
postfix:canonical_maps	Default = no
postfix:debug_peer_level	Default = 2
postfix:in_flow_delay	Default = "1s"
postfix:smtpd_junk_command_limit	Default = 100
postfix:program_directory	Default = "/usr/libexec/postfix"
postfix:smtp_quit_timeout	Default = "300s"
postfix:smtp_mail_timeout	Default = "300s"
postfix:minimal_backoff_time	Default = "1000s"
postfix:queue_file_attribute_count_limit	Default = 100
postfix:body_checks	Default = no
postfix:smtpd_client_restrictions:_array_index:0	Default = ""
postfix:mydestination:_array_index:0	Default = "$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

Parameter (mail:)	Description
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"
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"

Parameter (mail:)	Description
postfix:smtp_destination_recipient_limit	Default = "$default_destination_recipient_limit"
postfix:smtpd_restriction_classes	Default = ""
postfix:mime_nesting_limit	Default = 100
postfix:virtual_mailbox_maps	Default = ""
postfix:bounce_service_name	Default = "bounce"
postfix:header_size_limit	Default = 102400
postfix:strict_8bitmime	Default = no
postfix:virtual_transport	Default = "virtual"
postfix:berkeley_db_create_buffer_size	Default = 16777216
postfix:broken_sasl_auth_clients	Default = no
postfix:home_mailbox	Default = no
postfix:content_filter	Default = ""
postfix:forward_path	Default = "$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"

Parameter (mail:)	Description
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"
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

Parameter (mail:)	Description
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
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

Parameter (mail:)	Description
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
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

Parameter (mail:)	Description
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 application 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 page 136.
getLogPaths	Display the locations of the Mail service logs. See "Viewing the Mail Service Logs" on page 137.
writeSettings	Equivalent to the standard serveradmin settings command, but also returns a setting indicating whether the service needs to be restarted. See "Determining Whether a Service Needs to be Restarted" on page 25.

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.

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
```

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`
`<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 mail boxes 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 store 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 would then copy the following files/directories 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 directories. Each mailbox directory 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. 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 itself.
- `cyrus.index`—This file contains fixed-length information about the mailbox itself 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 program can be used to recover from corruption in mailbox directories. 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 themselves. The state reconstruct attempts to preserve includes the flag names, flag state, and internal date. reconstruct derives all other information 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 mailboxes file, run the `reconstruct -m` command. The `reconstruct` program, 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 directories to put in the mailboxes file.

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

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

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 passphrase 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 command-line tool `certtool`. 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, type the following two commands:

```
$ cd /private/var/root/Library/Keychains/
$ /usr/bin/certtool r csr.txt k=certkc c
```

   This use of the `certtool` command begins an interactive process that generates a Certificate Signing Request (CSR) in the file csr.txt and creates a keychain named certkc.
3. In the New Keychain Passphrase dialog that appears, enter a passphrase or password for the keychain you're creating, enter the password or passphrase a second time to verify it, and click OK.

   Remember this passphrase, because later you must supply it again.
4. When "Enter key and certificate label:" appears in the Terminal window, type a one-word key, a blank space, and a one-word certificate label, then press Return.

   For example, you could type your organization's name as the key and `mailservice` as the certificate label.
5. Type `r` when prompted to select a key algorithm, 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. Type a key size at the next prompt, 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 Type `y` when prompted to confirm the algorithm and key size, then press Return.

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

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

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

9 Type `s` when prompted to select a signature algorithm, 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 Type `y` when asked to confirm the selected algorithm, 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, 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), pressing 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 Type `y` when asked to confirm the information you entered, 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 the file is plain text, not rich text, and contains only the certificate text.

Importing an SSL Certificate Into the Keychain

To import an SSL certificate into a keychain, use the command-line tool `certtool`. 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 directory where the saved certificate file is located.

   For example: type `cd /private/var/root/Desktop` and press Return if the certificate file is saved on the desktop of the root user.
4. Type the following command and 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.
   ```

certadmin

Server Admin keeps a centralized store of all of your server's certificates for ease of use and management. `certadmin` allows you to access this information from the command-line. `certadmin` manipulates the list of certificates stored in the System keychain. It recognizes two commands:

- `list`: lists the certificates stored in the System keychain. By default, `certadmin` will print the "Common Name" field of each certificate separated by newlines. Adding the option (`-x` or `--xml`) will print to screen the certificate list as an XML property list (plist).
- `export`: exports the given certificate to OpenSSL.

For more information view the man pages for `certadmin`.

Creating a Passphrase File

To create a passphrase file, you will use TextEdit, then change the privileges of the file using the Terminal application. This file contains the passphrase you specified when you created the keychain. Mail service will automatically use the passphrase file to unlock the keychain that contains the SSL certificate. This concludes configuring Mail service for automatic SSL connections.

To create a passphrase file:

1. Log in to the server as root (if you're not already logged in as root).
2. In TextEdit, create a new file and type the passphrase exactly as you entered it when you created the keychain.

   Don't press Return after typing the passphrase.
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 type `/private/var/root/Library/Keychains/` and click Go.
6. In the Terminal application, change the access privileges to the passphrase 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
   ```

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

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

Working With Web Technologies

12

Introduction

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 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 which 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 outlines the major directories.

Application binaries	`/usr/sbin`
CGI programs	`/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 directory.

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` directory.

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 `httpd.conf` file contains detailed instructions on how to safely edit its options.

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 on apache visit www.apache.org. The documentation is also included with Mac OS X Server at `127.0.0.1/manual/`.

Commands for Managing Web Service

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.

For information on Apache settings, visit www.apache.org.

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.

For information on Apache, visit www.apache.org.

`serveradmin` and Apache Settings

The parameters are written differently in the Apache configuration file than they are in `serveradmin`. For example, see 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>
```

appear as follows 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".
```

For information on Apache settings, visit www.apache.org.

Changing Settings Using `serveradmin`

You can change Web service settings using the `serveradmin` command.

To change a setting:

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

Parameter	Description
`setting`	A Web service setting. To see a list of available settings, type `$ 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` application to manage Web service.

Command (`web:command=`)	Description
`getHistory`	View Web service statistics. See "Viewing Service Statistics" on page 149.
`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`.

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 actual script you run. It accepts values for the site's IP address, port number, server name, and root directory and uses `sed` to substitute these values in the settings it reads from the second file (`addsite.in`) feeds to `serveradmin`.
- `addsite.in`—Contains the actual 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 directory for the site's files and subdirectories.

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 directory 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
```

Performance Tuning and `ab`

When trying to analyze the server's performance, keep in mind that a lot of factor can affect performance: CGI scripts growing too large, database queries exhausting your system'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, and other valuable statistics. The following command will simulate 1000 requests to the specified URL with the username and password provided.

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

Tomcat

Mac OS X Server comes with Tomcat, the open source servlet container developed by Sun Microsystems. Tomcat runs as part of the Java process.

You can start Tomcat from the command-line using the following command:

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

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

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

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

JBoss

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 in the `/Library/JBoss/3.2/server/deploy-standalone/` folder.

You can start JBoss in Terminal with the following command:

```
/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.

Monitor the JBoss logs by reading the logs in the /Library/Logs/JBoss/ folder.

To stop JBoss in Terminal, enter the `/Library/JBoss/3.2/bin/shutdown.sh` command or terminate the running `run.sh` command. JBoss uses Tomcat as its default web server and servlet container.

MySQL

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 Mac OS X Server Web Technologies Administration manual.

Mac OS X Server stores the files of the pre-installed 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 system reboots you need to 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 be performed from the command-line.

To install the default database:

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

To set 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 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 config file `/etc/mysqlManager.plist`. It uses the Apple-provided `mysqld_manager_options(8)` tool to do this.

Following are useful command-line tools distributed with MySQL, each covered with 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, visit www.mysql.org.

Working With Network Services

13

Commands you can use to manage DHCP, DNS, Firewall, NAT, and VPN service in Mac OS X Server.

DHCP Service

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 change a setting:

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

Parameter	Description
setting	A DHCP service setting. To see a list of available settings, type `$ sudo serveradmin settings dhcp` or see "DHCP Service Settings" on this page and "DHCP Subnet Settings Array" on page 157.
value	An appropriate value for the setting.

To change several settings:

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

DHCP Service Settings

Use the following parameters with the `serveradmin` command 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 GUI 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 this 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 GUI 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 GUI 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 GUI 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 GUI application.
dhcp_ldap_url:_array_index:*n*	The URL of the LDAP directory to be used by clients. Corresponds to the Lease URL field in the LDAP pane of the subnet settings in the Server Admin GUI 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 GUI 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 GUI 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 GUI 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 GUI 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 GUI 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 GUI 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 GUI 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 GUI 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 GUI 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 GUI 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 GUI 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 155.

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 157.

Adding a DHCP Static Map

A static DHCP map allows you to map a specific IP address to a physical machine based on the machine's Ethernet address. You can add a static map to the DHCP configuration by using serveradmin to add an array of the DHCP parameters described below.

Static Map Parameter (dhcp:static_maps:_array_id:<host name>:<mapID>:)	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.

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
```

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

The static map entries are stored in the local NetInfo database in the machines record, so they can also be manipulated with NetInfo utilities such as `nidump`. (See the man page for `nidump` and related utilities for details.) For example, the static map created by the `servreadmin` command 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` application 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
```

Output

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

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

DNS Service

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:

Type only as much of the name as you want, stopping at a colon (`:`), then type 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
```

Sample Output

```
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
```

xinetd

Mac OS X Server uses the process `xinetd` 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 and both of them can cohabit peacefully. `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` provides a mapping of services to the executable that should be run in order 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 into the machine, `xinetd` receives the request, then launches `ftpd` to handle it. In this way, `xinetd` can keep the number of services running on a particular machine lower by launching only those that are requested by a client.

`inetd` and `xinetd` have different 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 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 on `xinetd`, visit www.xinetd.org.

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` command 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 boots up, you must set the `IPFORWARDING` flag in the `/etc/hostconfig` file to `-YES-` to enable IP forwarding during the startup process.

Firewall Service

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. `ipfw` can be configured from the command-line and can take a variety of commands. For more information consult the man page.

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:

Type only as much of the name as you want, stopping at a colon (`:`), then type 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 166.
`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` command 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 this page.
`rules:_array_id:<rule>:...`	Arrays of rule settings, one array per defined rule. See "`ipfilter` Rules Array" on page 169.
`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 GUI application to implement the IP Address groups on 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 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`

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.

For more information, read the `ipfw` man page.

The unmodified `ipfw.conf` file:

```
# ipfw.conf.default - Installed by Apple, never modified by Server Admin app
#
# ipfw.conf - The servermgrd process (the back end of Server Admin app)
# creates this from ipfw.conf.default if it's absent, but does not modify
# it.
#
# Administrators can place custom ipfw rules in ipfw.conf.
#
# Whenever a change is made to the ipfw rules by the Server Admin
# application and saved:
#    1. All ipfw rules are flushed
#    2. The rules defined by the Server Admin app (stored as plists)
#         are exported to /etc/ipfilter/ipfw.conf.apple and loaded into the
#         firewall via ipfw.
#    3. The rules in /etc/ipfilter/ipfw.conf are loaded into the firewall
#         via ipfw.
# Note that the rules loaded into the firewall are not applied unless the
# firewall is enabled.
#
# The rules resulting from the Server Admin app's IPFirewall and NAT panels
# are numbered:
#    10 - from the NAT Service - this is the NAT divert rule, present only
#         when he NAT service is started via the Server Admin app.
#    1000 - from the "Advanced" panel - the modifiable rules, ordered by
#         their relative position in the drag-sortable rule list
#    12300 - from the "General" panel - "allow"" rules that punch specific
#         holes in the firewall for specific services
#    63200 - from the "Advanced" panel - the non-modifiable rules at the
#         bottom of the panel's rule list
#
# Refer to the man page for ipfw(8) for more information.
#
# The following default rules are already added by default:
#
#add 01000 allow all from any to any via lo0
```

```
#add 01010 deny all from any to 127.0.0.0/8
#add 01020 deny ip from 224.0.0.0/4 to any in
#add 01030 deny tcp from any to 224.0.0.0/4 in
#add 12300 ("allow" rules from the "General" panel)
#...
#add 65534 deny ip from any to any
```

For more information, read the `ipfw` man page.

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 169.

Example:

```
$ 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 168).

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 application to manage Firewall (ipfilter) 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 GUI application.
writeSettings	Equivalent to the standard serveradmin settings command, but also returns a setting indicating whether the service needs to be restarted. See "Determining Whether a Service Needs to be Restarted" on page 25.

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
```

Output

```
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 on Dummynet, visit ai3.asti.dost.gov.ph/sat/dummynet.html or use Google or Sherlock to search the web. Also, check out the man page for ipfw(8).

NAT Service

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, type `$ sudo serveradmin settings nat` or see "NAT Service Settings" on this page.
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` command 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"`

Parameter (nat:)	Description
unregistered_only	yes | no Default = no
same_ports	yes | no Default = yes

NAT serveradmin Commands

You can use the following commands with the serveradmin application 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 "Determining Whether a Service Needs to be Restarted" on page 25.

Port Mapping

You can configure port mapping by adding a redirect_port directive to the config file passed to the natd process. You can accomplish this by editing the plist version of the config file etc/nat/natd.plist. This file is in turn processed by the serveradmin command, and used to create the config file passed to the natd process, /etc/nat/natd.conf.apple. See the man page for natd for details about configuring natd.

Note: You must not 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` command:

```
$ 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
```

Output

```
nat:natLog = <nat-log>
```

Value	Description
`<nat-log>`	The location of the NAT service log. Default = `/var/log/alias.log`

VPN Service

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, type `$ sudo serveradmin settings vpn` or see "List of VPN Service Settings" on page 175.
`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` command 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"`
`com.<name>.ppp.l2tp:IPv4:DestAddressRanges`	Default = `_empty_array`
`com.<name>.ppp.l2tp:IPv4:OfferedRouteMasks`	Default = `_empty_array`
`com.<name>.ppp.l2tp:IPv4:OfferedRouteAddresses`	Default = `_empty_array`
`com.<name>.ppp.l2tp:IPv4:OfferedRouteTypes`	Default = `_empty_array`
`com.<name>.ppp.l2tp:IPv4:ConfigMethod`	Default = `"Manual"`
`com.<name>.ppp.l2tp:DNS:OfferedSearchDomains`	Default = `_empty_array`
`com.<name>.ppp.l2tp:DNS:OfferedServerAddresses`	Default = `_empty_array`
`com.<name>.ppp.l2tp:Interface:SubType`	Default = `"L2TP"`

Parameter (vpn:Servers:)	Description
com.<name>.ppp.l2tp: Interface:Type	Default = "PPP"
com.<name>.ppp.l2tp: PPP:LCPEchoFailure	Default = 5
com.<name>.ppp.l2tp: PPP:ACSPEnabled	Default = 1
com.<name>.ppp.l2tp: PPP:VerboseLogging	Default = 1
com.<name>.ppp.l2tp: PPP:AuthenticatorACLPlugins	Default = DSACL
com.<name>.ppp.l2tp: PPP:AuthenticatorEAPPlugins	Default = EAP-KRB
com.<name>.ppp.l2tp: PPP:AuthenticatorPlugins: _array_index:*n*	Default = "DSAuth"
com.<name>.ppp.l2tp: PPP:LCPEchoInterval	Default = 60
com.<name>.ppp.l2tp: PPP:LCPEchoEnabled	Default = 1
com.<name>.ppp.l2tp: PPP:IPCPCompressionVJ	Default = 0
com.<name>.ppp.l2tp: PPP:AuthenticatorProtocol: _array_index:*n*	Default = "MSCHAP2"
com.<name>.ppp.l2tp: PPP:LogFile	Default = "/var/log/ppp/vpnd.log"
com.<name>.ppp.pptp: Server:VerboseLogging	Default = 1
com.<name>.ppp.pptp: Server:MaximumSessions	Default = 128
com.<name>.ppp.pptp: Server:LogFile	Default = "/var/log/ppp/vpnd.log"
com.<name>.ppp.pptp: IPv4:DestAddressRanges	Default = _empty_array
com.<name>.ppp.pptp: IPv4:OfferedRouteMasks	Default = _empty_array
com.<name>.ppp.pptp: IPv4:OfferedRouteAddresses	Default = _empty_array
com.<name>.ppp.pptp: IPv4:OfferedRouteTypes	Default = _empty_array
com.<name>.ppp.pptp: IPv4:ConfigMethod	Default = "Manual"

Parameter (vpn:Servers:)	Description
com.<name>.ppp.pptp: DNS:OfferedSearchDomains	Default = _empty_array
com.<name>.ppp.pptp: DNS:OfferedServerAddresses	Default = _empty_array
com.<name>.ppp.pptp: Interface:SubType	Default = "PPTP"
com.<name>.ppp.pptp: Interface:Type	Default = "PPP"
com.<name>.ppp.pptp: PPP:CCPProtocols:_array_index:*n*	Default = "MPPE"
com.<name>.ppp.pptp: PPP:LCPEchoFailure	Default = 5
com.<name>.ppp.pptp: PPP:MPPEKeySize128	Default = 1
com.<name>.ppp.pptp: PPP:ACSPEnabled	Default = 1
com.<name>.ppp.pptp: PPP:AuthenticatorACLPlugins	Default = DSACL
com.<name>.ppp.pptp: PPP:AuthenticatorEAPPlugins	Default = EAP-RSA
com.<name>.ppp.pptp: PPP:VerboseLogging	Default = 1
com.<name>.ppp.pptp: PPP:AuthenticatorPlugins: _array_index:*n*	Default = "DSAuth"
com.<name>.ppp.pptp: PPP:MPPEKeySize40	Default = 0
com.<name>.ppp.pptp: PPP:LCPEchoInterval	Default = 60
com.<name>.ppp.pptp: PPP:LCPEchoEnabled	Default = 1
com.<name>.ppp.pptp: PPP:CCPEnabled	Default = 1
com.<name>.ppp.pptp: PPP:IPCPCompressionVJ	Default = 0
com.<name>.ppp.pptp: PPP:AuthenticatorProtocol: _array_index:*n*	Default = "MSCHAP2"
com.<name>.ppp.pptp: PPP:LogFile	Default = "/var/log/ppp/vpnd.log"

List of VPN `serveradmin` Commands

You can use the following commands with the `serveradmin` application 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 "Determining Whether a Service Needs to be Restarted" on page 25.

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
```

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 `racoon` and `setkey`. `racoon` negotiates and configures a set of parameters of IPsec. `setkey` manipulates Security Association Database (SAD) entries as well as Security Policy Database (SPD) entries in the kernel. For more information about `racoon` and `setkey` you can read the man pages (section 8). `racoon` also has a webpage: www.kames.com/racoon. You might also find the man page for `ipsec` (section 4) helpful in getting more information.

Apple provides an interactive command-line tool, `s2svpnadmin`, located in `/usr/sbin/` that enables you to configure and set up site-to-site VPN. The `s2svpnadmin` tool access configuration information for the Client Server VPN tool 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 utilized only to configure 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. However, certain parameters must be kept common for a successful configuration.

`s2svpnadmin` must be run as root.

Configuration

To configure a site-to-site configuration, 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 tool. If 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.

Note: 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; or alternatively, you can edit the configuration file directly. The configuration file is a plist file and is located in:

```
/Library/Preferences/SystemConfiguration/com.apple.RemoteAccessServers.plist
```

Adding a VPN Key Agent User

To enable the PPTP protocol in your VPN server, you must add a keyagent user in the LDAP directory that hosts your users. If you have more than one directory that has VPN users, you must add a keyagent in each one of the directories.

The `vpnaddkeyagentuser` utility enables you to add the required VPN PPTP keyagent user to the directory of interest.

The utility will prompt you for the administrator username and password of the directory. It will then set up the keyagent user. This is a required step to be able to proceed with the configuration of the VPN PPTP server.

Note: You must run the `vpnaddkeyagentuser` command on the machine running the VPN service.

To add the keyagent user to the Open LDAP master on your local machine:

```
sudo vpnaddkeyagentuser /LDAPv3/127.0.0.1
```

If your Open LDAP master is not running on the local machine, you would replace `127.0.0.1` with the IP address of the Open LDAP master.

`vpnaddkeyagentuser` must be run as root. If no argument is specified, the keyagent user is added to the local netinfo node.

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 pages for more information.

Requirements

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 will need to set up the following hardware and software.

Hardware

IP failover requires the following hardware setup:

- Primary server
- Secondary server
- Public network (servers must be on same subnet)

- Private network between the servers (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. In addition, both servers must have IP addresses on the same subnet of the private network.

Software

IP failover requires the following software setup:

- Unique IP addresses for each network interface (public and private)
- Software to mirror primary server data to secondary server
- Scripts to control failover behavior on secondary server (optional)

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, 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. At 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 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. At 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, a colon, then the primary server's public IP address.

(Optional) 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 machine.

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, 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 directory 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 directory to control whether, in the event of a failover condition, the secondary server acquires the primary'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 non-zero 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 directory 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` directory on the secondary server, as previously discussed. The scripts use these four prefixes:

- `PreAcq`—Run before acquiring IP address from primary server
- `PostAcq`—Run after acquiring IP address from primary server
- `PreRel`—Run before relinquishing IP address back to primary server
- `PostRel`—Run after relinquishing IP address back to 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 their file names appear in a directory listing using the `ls` command.

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 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` command to set up Point-to-Point Protocol (PPP) dial-in service. For more information, see the man page. The "Examples" section of the man page shows an example of setting up dial-in service.

Working With Open Directory

14

Commands Used To Manage The Open Directory Service.

Overview

Open Directory 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 server. 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 normal 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 interpreted more easily with these systems.

This chapter includes descriptions of general directory tools and tools for working with LDAP, NetInfo, and the Password Server.

General Directory Tools

Testing Your Open Directory Configuration

You can use the `dscl` utility to test your directory services configuration. For more information, type `man dscl` to see the man page.

Modifying an Open Directory Node

You can also use the `dscl` utility to create, modify, or delete directory information in an Open Directory node.

Testing Open Directory Plugins

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.

For more information, type `man dsperfmonitor` to see the man page.

Directory services API support is provided by the `DirectoryService` daemon. For more information, type `man DirectoryService` to see the man page.

For information on the data types used by directory services, type `man DirectoryServiceAttributes` to see the man page.

Finally, for information on 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 Service Location Protocol (SLP)

You can use the `slp_reg` command to register service URLs using the Service Location Protocol (SLP).

For more information, type `man slp_reg` to see the man page.

SLP registration is handled by the SLP daemon `slpd`. For more information, type `man slpd` to see the man page.

Changing Open Directory Service Settings

Use the following parameters with the `serveradmin` command to change settings for the Open Directory service.

Be sure to add `dirserv:` to the beginning of any parameter you use. For example, to see the role that the server is playing in the directory hierarchy, you would type:

```
serveradmin settings dirserv:LDAPServerType
```

Parameter (`dirserv:`)	Description
replicationUnits	Default = `"days"`
replicaLastUpdate	Default = `""`
LDAPDataBasePath	Default = `""`
replicationPeriod	Default = `4`
LDAPSearchBase	Default = `""`
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"`

Parameter (dirserv:)	Description
LDAPDefaultPrefix	Default = "dc=<domain>,dc=com"
LDAPTimeoutUnits	Default = "minutes"
LDAPServerBackend	Default = "BerkeleyDB"

LDAP

Configuring LDAP

The OpenLDAP server daemon is /usr/libexec/slapd. slapd is launched automatically by the LDAP startup item. The primary configuration files for the OpenLDAP system are all kept in /etc/openldap/. In there you will find slapd.conf, which contain basic configuration information. Most of the configuration for Open Directory is kept in slapd_macosxserver.conf. There is an include statement in slapd.conf that includes slapd_macosxserver.conf.

The directives in these files are modified by the graphical user interface. It's advisable that you not modify these directives. You may 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 database—access controls do not apply to the root user. An example value for rootdn is uid=root,cn=users,dc=pretendco,dc=com.

An administrator user on the system can edit the slapd_macosxserver.conf file to add a new password hash, or plain text password, into 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.

The following tools are available for configuring LDAP. For more information, see the man page for each tool.

slapconfig

You can use the slapconfig utility to configure the slapd and slurpd LDAP daemons and related search policies. For more information, type man slapconfig to see the man page.

Standard Distribution Tools

A number of different command-line utilities come with OpenLDAP. These tools can be separated into two different types:

- Tools which operate directly on the LDAP databases—These tools begin with `slap`.
- Tools which go through the LDAP protocol—These tools begin with `ldap`.

The slap tools must be run directly on the computer hosting the LDAP database. It is also advisable to shut down the LDAP server when using the slap tools or else your database may become out of sync.

These tools are included in the standard LDAP distribution.

Program	Used to
`/usr/bin/ldapadd`	Add entries to the LDAP directory.
`/usr/bin/ldapcompare`	Compare a directory entry's actual attributes with known attributes.
`/usr/bin/ldapdelete`	Delete entries from the LDAP directory.
`/usr/bin/ldapmodify`	Change an entry's attributes.
`/usr/bin/ldapmodrdn`	Change an entry's relative distinguished name (RDN).
`/usr/bin/ldappasswd`	Set the password for an LDAP user. Apple recommends using `passwd` instead of `ldappasswd`. For more information, type `man passwd`.
`/usr/bin/ldapsearch`	Search the LDAP directory. See the usage note under "A Note on Using `ldapsearch`" on page 189.
`/usr/bin/ldapwhoami`	Obtain the primary authorization identity associated with a user.
`/usr/sbin/slapadd`	Add entries to the LDAP directory.
`/usr/sbin/slapcat`	Export LDAP Directory Interchange Format files.
`/usr/sbin/slapindex`	Regenerate directory indexes.
`/usr/sbin/slappasswd`	Generate user password. hashes.

Examples:

To query the LDAP server for all the user's information issue the following command:

```
ldapsearch -H ldap://127.0.0.1 -b cn=users,dc=pretendco,dc=com
```

To load an LDIF file into your LDAP server use the ldapadd command as follows:

```
ldapadd -H ldap://pantherserver.pretendco.com -f myusers.ldif
```

A Note on Using `ldapsearch`

The `ldapsearch` tool connects to an LDAP server, binds to it, finds entries, and returns attributes of the entries found.

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, include the `-x` option when you type the command. For example:

```
ldapsearch -h 192.168.100.1 -b "dc=ecxample,dc=com" -x
```

The `-x` option forces `ldapsearch` to use simple authentication instead of SASL.

Idle Rebinding Options

The following two LDAPv3 plug-in parameters are documented in the Open Directory administration guide. The parameters are in, or can be added to, 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 reconnect attempts.

```
<key>Delay Rebind Try in seconds<\key>
<integer>n<\integer>
```

You should find this parameter in the 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 plugin 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 it doesn't already exist in the plist file, you can add it near `<key>OpenClose Timeout in seconds<\key>`.

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.

NetInfo

Configuring NetInfo

You can use the following command-line utilities to manage the NetInfo directory. For more information about a utility, see the related man page.

Utility	Used to
NeST	Configure the directory system of a server.
`nicl`	Create, view, and modify entries in the NetInfo directory.
`nifind`	Search the NetInfo directory for a particular entry.
`nigrep`	Search the NetInfo directory for an expression.
`nidump`	Export NetInfo data to text or flat files.
`niload`	Import flat files into the NetInfo directory.
`nireport`	Print tables of NetInfo directory entries.

For example, you can use the `NeST -setprotocols` command to specify which authentication methods the server's Open Directory Password Server uses.

Password Server

Working With the Password Server

You can use the `mkpassdb` utility to create, modify, or back up the password database used by the Mac OS X Server Password Server. For more information, type `man mkpassdb` to read the man page.

Viewing or Changing Password Policies

You can use the `pwpolicy` command to view or change the authentication policies used by the Mac OS X Server Password Server. For more information, type `man pwpolicy` to see the man page.

Enabling or Disabling Authentication Methods

All password authentication methods supported by Open Directory Password Server are initially enabled. You can disable and enable 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)
```

Parameter	Description
`protocol`	Any of the protocol names listed by `NeST -getprotocols` (for example, `SMB-LAN-MANAGER`).

For information on the available methods, see the Open Directory administration guide.

Kerberos and Single Sign-On

The KDC process is `/usr/sbin/krb5kdc` and is started by the `launchd` daemon. Its user, host, and service principals are stored in `/var/db/krb5kdc/principal`. The principal database is encrypted with a key stored in a stash file called `.k5.GS.REALM.COM` located at `/var/db/krb5kdc`. The `kdc.conf` file specifies per-realm configuration data to be used by the KDC. All of these files are extremely sensitive.

The following tools are available for setting up your Kerberos and Apple Single Sign-On environment. For more information on 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 utility for maintaining the Kerberos database. It is problematic to back up the KDC while the `krb5kdc` process is running. `kdb5_util` 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.

To dump the KDC's database, use the following command:

```
sudo kdb5_util dump > /path/to/secure/backup
```

To load KDC data from a dumped file, use the following command:

```
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

Apple uses MIT's Kerberos administration architecture for principal management. The Kerberos administration daemon, `kadmind`, is responsible for making changes to the Kerberos database. Outside of Open Directory, `kadmind` is largely manipulated by the `kadmin` and `kadmin.local` command-line utilities. Generally, in Mac OS X, Apple's tools are responsible for telling the `kadmin` tools what to do and hence manual modifications are rarely needed.

The configuration files for both `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 called `principal.kadm5` is the `kadmind` process' policy database. It resides at `/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`.

Finally `principal.kadm5.lock` is a lock file used by `kadmind`. It acts essentially in reverse of most lock files though, as `kadmind` will not write to either the policy or principal database unless it exists.

The `/usr/sbin/kadmin` tool 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.

`kadmin.local`, unlike `kadmin`, 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).

Examples

To add a principal use the following command:

```
sudo kadmin.local -q "add_principal student1"
```

To add a service principal:

```
sudo kadmin.local -q "add_principal afpserver/server.example.com"
```

To delete a principal:

```
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 pre-Kerberizes many services for you, you can use Kerberos CLI tools to Kerberize additional services with Open Directory Kerberos.

The service type for most services is fixed, so often the server can assume that its principal name is `serviceType/fqdn@REALM`. However, the principal name is service specific and the primary place to get the info is from the service documentation.

To Kerberize a service do the following steps:

1. Use `kadmin` (or `kadmin.local`) to create the service principal

   ```
   sudo kadmin.local addprinc -randkey service-principal
   ```

2. Import the principal key into the `keytab` file

   ```
   sudo kadmin.local ktadd service-principal
   ```

3. Configure the service to use the new principal. This step is service specific. Make sure to check the documentation of your service on how to perform this step.

Directory Service Tools

The following are miscellaneous directory service tools that you can use to configure directory service and to troubleshoot any problems.

`dscl`

`dscl` is a general-purpose tool for operating on Directory Service directory nodes. Its commands allow one to create, read, and manage Directory Service data. If invoked without any commands, `dscl` runs in an interactive mode, reading commands from standard input.

The following are some samples usages for `dscl`.

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

For more information on `dscl` view the man page.

lookupd

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 Directory Services, allowing access to data from LDAP and other directory systems.

To lookup a user by name:

```
lookupd -q user -a name apple
```

This returns the user records that have a short name of "apple."

To run lookupd in interactive mode:

```
lookupd -d
>?
```

The ? displays all the possible commands for lookupd.

To list the attributes of a user:

```
> userWithName: warren
```

For more information on lookupd view the man page.

dseditgroup

dseditgroup allows manipulation of a single named group record on either the default local node or the specified Directory Service node. The following are some samples uses for dseditgroup.

To display the attributes of a group in the local node:

```
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 3600 -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
```

For more information on dseditgroup view the man page.

dsconfigldap

`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
```

dsconfigad

`dsconfigad` allows you to configure the Active Directory Plug-in in Directory Service from the command-line. `dsconfigad` has the same functionality for configuring the Active Directory plugin as the Directory Access application.

To add a computer to a directory:

```
dsconfigad -a ThisComputer -u "administrator" -ou
    "CN=Computers,OU=Engineering,DC=ads,DC=demo,DC=com" -domain
    domain.ads.apple.com
```

For more information about `dsconfigad` view the man page.

Working With QuickTime Streaming Server

15

Commands you can use to manage QTSS service in Mac OS X Server.

Starting QTSS Service

You can use the `serveradmin` command to start QTSS service, or you can use the `quicktimestreamingserver` command to specify additional service parameters when you start the service.

To start QTSS service:

```
$ sudo serveradmin start qtss
```

or

```
$ sudo quicktimestreamingserver
```

To see a list of `quicktimestreamingserver` command options, type:

```
$ sudo quicktimestreamingserver -h
```

Stopping QTSS Service

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 using the `serveradmin` command 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, type: `$ sudo serveradmin settings qtss` or see "QTSS Settings" on page 199.
`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` command to change settings for the QTSS service.

Descriptions of Settings

To see descriptions of most QTSS settings, you can look in the sample settings file `/Library/QuickTimeStreaming/Config/streamingserver.xml-sample`.

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".
```

QTSS parameters you might change:

Parameter (`qtss:`)	Description
`broadcaster:password`	Default = `""`
`broadcaster:username`	Default = `""`
`modules:_array_id:QTSSAccessLogModule: request_logfile_dir`	Default = `"/Library/QuickTime Streaming/Logs/"`
`modules:_array_id:QTSSAccessLogModule: request_logfile_interval`	Default = `7`
`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"`
`modules:_array_id:QTSSAccessModule: modAccess_usersfilepath`	Default = `"/Library/QuickTimeStreaming/Config/qtusers"`

Parameter (qtss:)	Description
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
modules:_array_id:QTSSMP3StreamingModule: mp3_streaming_enabled	Default = yes

Parameter (qtss:)	Description
modules:_array_id:QTSSReflectorModule: allow_broadcasts	Default = yes
modules:_array_id:QTSSReflectorModule: allow_non_sdp_urls	Default = yes
modules:_array_id:QTSSReflectorModule: BroadcasterGroup	Default = "broadcaster"
modules:_array_id:QTSSReflectorModule: broadcast_dir_list	Default = ""
modules:_array_id:QTSSReflectorModule: disable_overbuffering	Default = no
modules:_array_id:QTSSReflectorModule: enable_broadcast_announce	Default = yes
modules:_array_id:QTSSReflectorModule: enable_broadcast_push	Default = yes
modules:_array_id:QTSSReflectorModule: ip_allow_list	Default = "127.0.0.*"
modules:_array_id:QTSSReflectorModule: kill_clients_when_broadcast_stops	Default = no
modules:_array_id:QTSSReflectorModule: minimum_static_sdp_port	Default = 20000
modules:_array_id:QTSSReflectorModule: timeout_broadcaster_session_secs	Default = 20
modules:_array_id:QTSSRelayModule: relay_prefs_file	Default = "/Library/Quick TimeStreaming/Config/ relayconfig.xml"
server:authentication_scheme	Default = "digest"
server:auto_restart	Default = yes
server:default_authorization_realm	Default = "Streaming Server"
server:do_report_http_connection_ip_address	Default = no
server:error_logfile_dir	Default = "/Library/Quick TimeStreaming/Logs/"
server:error_logfile_name	Default = "Error"
server:error_logfile_size	Default = 256000
server:error_logfile_verbosity	Default = 2
server:error_logging	Default = yes
server:force_logs_close_on_write	Default = no
server:maximum_bandwidth	Default = 102400
server:maximum_connections	Default = 1000
server:module_folder	Default = "/Library/Quick TimeStreaming/Modules/"

Parameter (qtss:)	Description
server:movie_folder	Default = "/Library/Quick TimeStreaming/Movies/"
server:pid_file	Default = "/var/run/Quick TimeStreamingServer.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 = ""

QTSS serveradmin Commands

You can use the following commands with the serveradmin application to manage 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 203.
getLogPaths	Find the current location of the service logs. See "Viewing Service Logs" on page 204.

Listing Current Connections

You can use the serveradmin getConnectedUsers command to retrieve information about QTSS connections.

To list connected users:

```
$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`.

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
```

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 Re-Read its Preferences

You can force QTSS to re-read its preferences without restarting the server. You must log in as root to perform this task.

To force QTSS to re-read 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 Directories for User Streaming

If you want to enable QTSS home directory streaming for home directories 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 directory. You can use the `createuserstreamingdir` tool to set up the needed /Sites/Streaming folder.

To set up /Sites/Streaming in older home directories:

```
$ createuserstreamingdir user
```

Parameter	Description
`user`	The user in whose home directory the /Sites/Streaming folder is created.

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, and type 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 .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 directory you selected as your Media Directory. If an access file is not present in the streaming server media directory, all clients are allowed access to the media in the directory.

To set up access control:

1. Use the `qtpasswd` command-line utility to create new user accounts with passwords.
2. Create an access file and place it in the media directory that you want to protect.
3. If you want to disable authentication for a media directory, 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 directory in which the access file is stored. The directory you use to store streamed media can contain other directories, and each directory 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 directory where the media file is located. If an access file is not found, it looks in the enclosing directory. 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 .rtf or any other file format).

`message` is 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` is 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` is 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 a lot of users, it may be easier to set up one or more groups, and then enter the group names, than to list each user.

`username` is 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` is 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 directory-by-directory basis.

If you have made customized 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 directory. Any customized modifications you may have made prior to this will not be preserved.

What Clients Need to Access 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. Log in to the server computer as root, open a terminal window, and type the following:

   ```
   qtpasswd -f <user filename> <user-name>
   ```

   Alternatively, use sudo to execute the command as root.

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, simply 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. Log in to the server computer as root, open a terminal window, and type the following:

   ```
   qtpasswd <user-name>
   ```

   Alternatively, use sudo to execute the command as root.

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` command to manipulate QuickTime and MP4 movies. You can add hint tracks, prepare for "fast-start," and edit annotations. For more information run the `qtmedia` and it will display the different command-line options.

Creating Reference Movies

You can use the `qtref` command to create reference movies that can be used to embed QuickTime content in Web pages. The following are the options you can use 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 on using `qtref` you can run the command without any arguments from the command-line and it will display usage information.

PCI RAID Card Command Reference

This chapter describes the megaraid commands you can use to work with your PCI Hardware RAID Card.

The commands are listed directly below. The following section explains the function and parameters of each command.

- `megaraid -alarm -on | -off | -silence`
- `megaraid -changepolicy ld [-writecache enable | disable][-readahead on | off | adaptive] [-iopolicy direct | cached]`
- `megaraid -changestate pd -online | -fail`
- `megaraid -chkcon ld -start | -stop | -status`
- `megaraid -create auto [-numld n]`
- `megaraid -create R0 | R1 | R5 -drive { 0 1 2 3} [-stripesize n] [-size x] [-writecache enable | disable] [-readahead on | off | adaptive][-iopolicy direct | cached]`
- `megaraid -destroyconfig [-yes]`
- `megaraid -flash flashFileName`
- `megaraid -initialize ld -start | -stop | -status`
- `megaraid -rebuild pd -start | -stop | -status`
- `megaraid -showadapter`
- `megaraid -showconfig [ld]`
- `megaraid -showdevices`
- `megaraid -spare pd -create | -delete`

Note: You can use the `man megaraid` command to see the online reference of the `megaraid` commands. You can also use all `megaraid` commands with a `[-log file]` parameter and all the displayed information is logged with date and time in the file you specify.

megaraid Commands Functions

`megaraid -alarm -on | -off | -silence`

Turns the alarm on, off, or to silence. When the alarm is set to silence, it turns off for current failure, but will turn on again for 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` must be 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, then the remaining size of the array will automatically be used. If `the stripesize` and `iopolicy` parameters are not specified then 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 system 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 system 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 system, 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 nor recover data from it. You can access all data on a degraded logical drive (without a failure) even if all 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 in the optimal state is in perfect condition.
`megaraid -showdevices [-log file]` Displays all drives connected to the PCI Hardware 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," and "not responding."
`megaraid -spare pd -create | -delete [-log file]` Creates or deletes a global hotspare. You can create hot spares from a ready pool of devices. After deletion, a hot spare drive becomes a ready drive. The parameter `pd` is the physical drive ID.

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 program 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 directory 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 directory A folder for a user's personal use. Mac OS X also uses the home directory, 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 directory A home directory that resides on disk on the computer a user is logged in to. It's accessible only by logging directly in to the computer where it resides unless you log in to the computer using SSH.

local hostname A name that designates a computer on a local subnet. It can be used without a global DNS system to resolve names to IP addresses. It consists of lowercase letters, numbers, or hyphens (except as the last characters), and ends with ".local" (For example, bills-computer.local). Although the 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. A program that allows applications on different computers to communicate within a local area network.

NetBoot server A Mac OS X server on which you've installed NetBoot software and have configured to allow clients to start up from disk images on the server.

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**.

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.

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.

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 directories, 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 TLS (Transport Level Security).

static IP address An IP address that's assigned to a computer or device once and is never changed.

subnet A grouping on the same network of client computers that are organized by location (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 directory 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

A

ab 151
ACL 101
AFP (Apple Filing Protocol)
  canceling user disconnect 88
  changing service settings 82
  checking service status 81
  disconnecting users 87
  listing connected users 86
  sending user message 87
  service settings 82
  starting service 81
  stopping service 81
  viewing service logs 90
  viewing service settings 81
  viewing service statistics 89
AirPort settings 56
Apache 145
  performance tuning 151
Apache web server 147
Apple Filing Protocol. *See* AFP
AppleTalk settings 54
asr
  restoring images 119

B

Backup
  mail files 138
bless command 38
Bonjour Name 58
Boot
  changing boot devices 119
BootP
  set server to use 51

C

case-sensitive file system 68
certadmin 142
certificate file 140–142
Certificate management 142
certificates, purchasing 142
certtool utility 140, 142
changeip tool 50
chmod
  ACL 101
command editing shortcuts 18
command not found message 18
command prompt 17
computer name 41, 57
configd 58
configuration file, server
  example 29
  naming 32
  saving 28
connections
  AFP 86
  FTP 94
  QTSS 202
  SMB 98
createhomedir 77
CSR (Certificate Signing Request) 140–142
CUPS
  logs 106
cups
  lp 103
  lpr 103
Cyrus 122

D

date 41, 42
delay rebinding options, LDAP 189
df 63
DHCP
  static map 160
DHCP (Dynamic Host Configuration Protocol)
  adding a subnet 159
  changing service settings 156
  checking service status 155
  service settings 156
  set server to use 51
  starting service 155
  stopping service 155
  viewing service logs 161
  viewing service settings 155
dial-in service, PPP 184

DirectoryServiceAttributes 186
DirectoryServiceAttributes 185
DirectoryService daemon 185
DirectoryService daemon 185
disk journaling 66
diskspacemonitor command 62
disktool 65
diskutil 65
DNS (Domain Name System)
  changing servers 51
  changing service settings 162
  checking service status 162
  service settings 162
  starting service 162
  stopping service 162
  viewing service logs 163
  viewing service settings 162
  viewing service statistics 163
Domain Name System. *See* DNS
dscl 193
dscl command 185
dsconfigad 195
dsconfigldap 195
dseditgroup 78, 194
dsimport command 72–73
dsperfmonitor command 185
dsRecTypeStandard 73
Dynamic Host Configuration Protocol. *See* DHCP

E

energy saver settings 43
error messages
  command not found 18

F

file system, case-sensitive 68
File Transfer Protocol. *See* FTP
fingerprint, RSA 22
Firewall service. *See* IPFilter service
fsck_hfs 62
fsck command 66
FTP (File Transfer Protocol)
  changing service settings 91
  checking connections 94
  checking service status 91
  service settings 92
  starting service 91
  stopping service 91
  viewing service logs 94
  viewing service settings 91
FTP proxy settings 55

G

Gopher proxy settings 56

H

hdiutil
  image management 118
home directory, creating 77
host name 57
hup signal 204

I

ifconfig 47
  link aggregation 53
install 27
  updating an image 117
installer 27
installer command 27
IP address
  changing server's address 50
  validating 51
IP Failover 180–183
ipfilter service
  changing settings 165
  checking status 165
  configuration file 167
  defining rules 167
  settings 166
  starting 165
  stopping 165
  viewing logs 170
  viewing settings 165
IP Forwarding 164
ipfw.conf file 167

J

JBoss 152
journaling 66

K

kadmind 192
kdcsetup utility 191
Kerberos 191
  backing up 191
  principal management 192
  tools and utilities 191
kerberosautoconfig tool 191
keychain 140
kill command 204
known_hosts file 22
krb5kdc 192

L

launchd 39
LDAP 187
LDAP (Lightweight Directory Access Protocol)
  and SASL 189
  configuration file 189
  delay rebinding options 189

idle timeout parameter 189
ldapsearch tool 189
parameter list 189
rebinding parameter 189
tools and utilities 188
tools for configuring 187
ldapadd tool 188
ldapcompare tool 188
ldapdelete tool 188
ldapmodify tool 188
ldapmodrdn tool 188
ldappasswd tool 188
ldapsearch tool 188, 189
ldapwhoami tool 188
Lightweight Directory Access Protocol. *See* LDAP
log files
AFP service 90
DHCP service 161
DNS service 163
FTP service 94
IPFilter service 170
Mail service 137
NAT service 173
Print service 112
QTSS 204
reclaiming space 63
SMB service 100
VPN service 178
Web service 148
login, enabling remote 45
lookupd 78, 194
lp 103
lpr 103

M

MAC address 47
Mail
backup 138
Mailman 122
Mail service
changing settings 123
checking status 123
settings 124
starting 123
stopping 123
viewing logs 137
viewing settings 123
viewing statistics 136
man command 24
man pages, viewing 24
mkpassdb utility 190
mount command 61
MySQL 153

N

NAT
port mapping 172
NAT (Network Address Translation)
changing service settings 171
checking service status 170
service settings 171
starting service 170
stopping service 170
viewing service logs 173
viewing service settings 170
NeST tool 190
NetBoot
enabling NetBoot 1.0 117
NetBoot service
changing settings 114
checking status 113
filters record array 115
general settings 114
image record array 116
port record array 117
starting 113
stopping 113
storage record array 115
viewing settings 113
NetInfo
tools and utilities 190
Network Address Translation. *See* NAT
Network File System. *See* NFS
network interface, settings 47
network port, settings 47
network port configurations 48
networksetup 33, 41, 47
network time server 41, 43
newfs 65
NFS (Network File System)
changing service settings 91
checking service status 90
starting and stopping service 90
viewing service settings 90
nicl 33
nicl tool 190
nidump tool 190
nifind tool 190
nigrep tool 190
niload tool 190
nireport tool 190
nvram 38
booting from an image 118

O

Open Directory
data types 185, 186
LDAP 187
modifying a node 185

NetInfo 190
settings 186
SLP 186
testing configuration 185
testing plugins 185
Open firmware
booting from an image 118

P

password server 190
pdisk 64
plugins, Open Directory 185
pmset command 44
Point-to-Point Protocol. *See* PPP
Postfix 121
power failure
automatic restart 43
power management 44
PPP (Point-to-Point Protocol)
enabling dial-in service 184
pppd command 184
pppd command 184
Print service
changing settings 105
checking status 105
holding jobs 111
listing jobs 111
listing queues 110
pausing queues 110
queue data array 107
settings 106
starting 104
stopping 104
viewing logs 112
viewing settings 105
prompt 17
proxy settings
FTP 55
Gopher 56
SOCKS firewall 56
streaming 56
web 55
ps command
listing QTSS processes 204

Q

qtmedia 208
qtpasswd 208
qtref 209
QTSS
access control 205
security 205
QTSS (QuickTime Streaming Server)
changing settings 198
checking status 197
commands for managing 197
listing connections 202
logs 204
settings 199
starting 197
statistics 203
stopping 197
viewing settings 198
QuickTime Streaming Server. *See* QTSS

R

RAID 69
rebinding options, LDAP 189
remote login, enabling 45
restart
automatic 43
checking if required 25
server 37
root privileges
su command 19
sudo command 19
RSA fingerprint 22

S

s2svpnadmin 179
sa_srchr 28
SASL
used by ldapsearch 189
scp 21
scripts
adding a website 150
scutil 58
Secure Sockets Layer. *See* SSL
serial number, server software 33
serveradmin utility
usage notes 25
server configuration file
example 29
naming 32
saving 28
Server Message Block. *See* SMB
servermgrd 23
serversetup 47
serversetup utility
usage notes 25
Service Location Protocol. *See* SLP
sftp 21
share points
creating 80
listing 79
updating SMB service after change 100
sharing command 79, 80
shell prompt 17
shortcuts
typing commands 18

shutdown command 38
  restarting a server 37
Single Sign-On 191
single sign-on 191
slapadd tool 188
slapcat tool 188
slapconfig utility 187
slapindex tool 188
slappasswd tool 188
sleep settings 43
SLP (Service Location Protocol)
  registering URLs 186
slp_reg command 186
SMB (Server Message Block)
  changing service settings 95
  checking service status 94
  disconnecting users 99
  listing service users 98
  service settings 96
  starting service 94
  stopping service 94
  viewing service logs 100
  viewing service settings 94
  viewing service statistics 99
SOCKS firewall proxy settings 56
softwareupdate command 34
Spotlight
  enabling and disabling 68
SSH 20
ssh command 20
sshd 21
ssh-keygen 21
SSL 21, 23
SSL (secure Sockets Layer)
  using with Mail service 140
SSLOptions 23
SSLRequire 23
sso_util utility 191
startup disk 45
statistics
  AFP 89
  DNS 163
  Mail service 136
  QTSS 203
  SMB 99
  Web service 149
streaming proxy settings 56
su 19
subnet mask
  validating 51
su command 19
sudo 19
sudo command 19
sysctl 164
systemsetup 33
systemsetup 41, 45

T

tail command
  viewing AFP service logs 90
  viewing DHCP service logs 161
  viewing DNS service logs 163
  viewing FTP service logs 94
  viewing IPFilter service logs 170
  viewing Mail service logs 137
  viewing NAT service logs 173
  viewing Print service logs 112
  viewing QTSS service logs 204
  viewing SMB service logs 100
  viewing VPN service logs 178
  viewing Web service logs 148
TCP/IP settings 50
Telnet 23
Terminal
  using 17
throughput. *See* statistics
time 41, 42
time server 41, 43
time zone 41, 42
Tomcat 152
tools
  networksetup 33
  nicl 33
  systemsetup 33

U

users
  checking admin privileges 78
  checking name, id, or password 76
  creating administrators 71
  creating home directory 77
  importing 72–75

V

Virtual Private Network. *See* VPN
volumes, mounting and unmounting 61
VPN
  key agent user 180
  site-to-site 178
VPN (Virtual Private Network)
  changing service settings 174
  checking service status 173
  service settings 175
  starting service 173
  stopping service 173
  viewing service logs 178
  viewing service settings 174
vpnaddkeyagentuser 180

W

watchdog 39
web proxy settings 55

Web service
changing settings 147
checking status 146
listing sites 148
script to add site 150
starting 146
stopping 146
viewing logs 148
viewing settings 146
viewing statistics 149
websites
script for adding 150
Windows service. *See* SMB service

X

`xinetd` 164