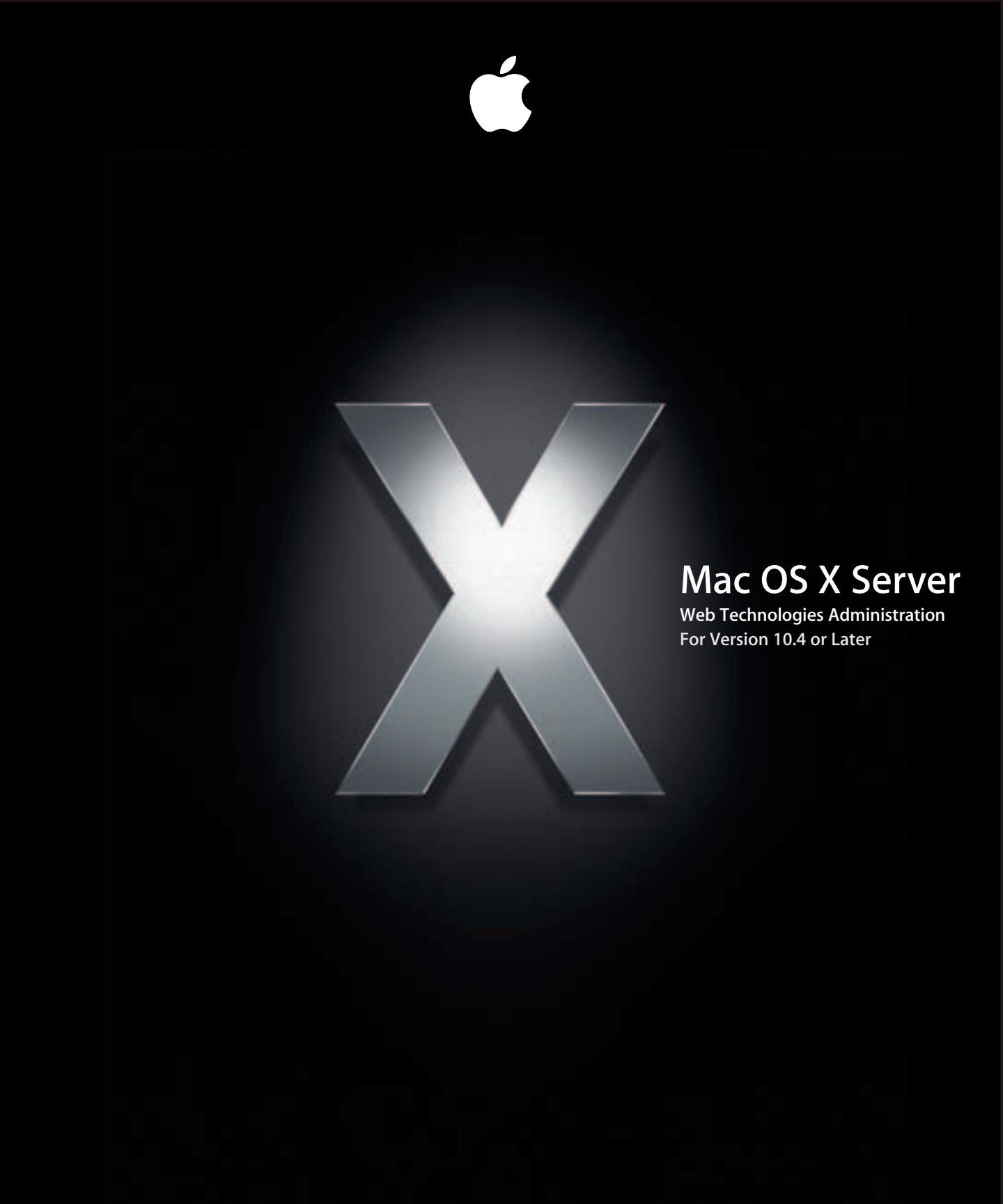
Mac OS X Server
Web Technologies Administration
For Version 10.4 or Later

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

019-0171/03-24-05

Contents

About This Guide

This guide tells you how to set up and manage a web server, websites, and use open source web technologies.

What's New in Version 10.4

- **Weblog service.** Mac OS X Server provides a multiuser weblog server that complies with the RSS and Atom XML standards. Weblog service supports Open Directory authentication. For additional safety, users can access Weblog service using a website that's SSL-enabled. Detailed coverage of weblogs is provided in Chapter 3, "Weblog Service," of the collaboration services administration guide.
- **Certificate Management.** Server Admin makes it easy to manage SSL certificates that can be used by web, mail, Open Directory, and other services that support them. You can create a self-signed certificate, and generate a Certificate Signing Request (CSR) to obtain an SSL certificate from an issuing authority and install the certificate. Certificate management controls are part of the mail component in Mac OS X Server.
- **WebObjects.** Mac OS X Server includes the WebObjects run-time libraries and an unlimited deployment license, making it the ideal platform for your J2EE-compatible WebObjects applications. Also provided are easy-to-use graphical tools for configuring and monitoring WebObjects from within the Server Admin application.

What's in This Guide

This guide is organized as follows:

- Chapter 1, "Web Technologies Overview," highlights key concepts and provides basic information about configuring a server, setting up websites, and understanding specialized web components.
- Chapter 2, "Managing Web Technologies," describes how to set up your web server for the first time and manage web settings and components.
- Chapter 3, "Managing Websites," provides instructions for setting up and managing websites.
- Chapter 4, "WebMail," tells you how to enable and use WebMail on your web server.

- Chapter 5, "Working With WebObjects and Web-Related Open Source Applications," provides information and instructions related to WebObjects and open source components Apache, JBoss, Tomcat, and MySQL.
- Chapter 6, "Installing and Viewing Web Modules," describes the modules included in Mac OS X Server and explains how to install, enable, and view modules.
- Chapter 7, "Solving Problems," helps you address issues with web technologies and websites.
- The Glossary defines terms you'll encounter as you read this guide.

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

Using This Guide

The chapters in this guide are arranged in the order that you're likely to need them when setting up Mac OS X Server to provide Windows services.

- Review Chapter 1 to acquaint yourself with basic concepts and components for web technologies.
- Follow the instructions in Chapter 2 to set up your web server and configure its technologies.
- Follow the instructions in Chapter 3 to set up and modify websites.
- For additional information about web technologies and instructions for specialized features, consult chapters 4–6.
- Review Chapter 7 if you encounter problems with web technologies.
- Consult Chapter 8 for additional resources.

Using Onscreen Help

You can view instructions and other useful information from this and other documents in the server suite by using onscreen help.

On a computer running Mac OS X Server, you can access onscreen help after opening Server Admin or Workgroup Manager. From the Help menu, select one of the options:

- *Server Admin Help* or *Workgroup Manager Help* displays information about the application.
- *Mac OS X Server Help* displays the main server help page, from which you can search or browse for server information.
- *Documentation* takes you to www.apple.com/server/documentation, from which you can download server documentation.

You can also access onscreen help from the Finder or other applications on a server or on an administrator computer. (An administrator computer is a Mac OS X computer with server administration software installed on it.) Use the Help menu to open Help Viewer, then choose Library > Mac OS X Server Help.

To see the latest server help topics, make sure the server or administrator computer is connected to the Internet while you're using Help Viewer. Help Viewer automatically retrieves and caches the latest server help topics from the Internet. When not connected to the Internet, Help Viewer displays cached help topics.

The Mac OS X Server Suite

The Mac OS X Server documentation includes a suite of guides that explain the services and provide instructions for configuring, managing, and troubleshooting the services. All of the guides are available in PDF format from:

www.apple.com/server/documentation/

This guide ...	tells you how to:
Mac OS X Server Getting Started for Version 10.4 or Later	Install Mac OS X Server and set it up for the first time.
Mac OS X Server Upgrading and Migrating to Version 10.4 or Later	Use data and service settings that are currently being used on earlier versions of the server.
Mac OS X Server User Management for Version 10.4 or Later	Create and manage users, groups, and computer lists. Set up managed preferences for Mac OS X clients.
Mac OS X Server File Services Administration for Version 10.4 or Later	Share selected server volumes or folders among server clients using these protocols: AFP, NFS, FTP, and SMB/CIFS.
Mac OS X Server Print Service Administration for Version 10.4 or Later	Host shared printers and manage their associated queues and print jobs.
Mac OS X Server System Image and Software Update Administration for Version 10.4 or Later	Use NetBoot and Network Install to create disk images from which Macintosh computers can start up over the network. Set up a software update server for updating client computers over the network.
Mac OS X Server Mail Service Administration for Version 10.4 or Later	Set up, configure, and administer mail services on the server.
Mac OS X Server Web Technologies Administration for Version 10.4 or Later	Set up and manage a web server, including WebDAV, WebMail, and web modules.
Mac OS X Server Network Services Administration for Version 10.4 or Later	Set up, configure, and administer DHCP, DNS, VPN, NTP, IP firewall, and NAT services on the server.
Mac OS X Server Open Directory Administration for Version 10.4 or Later	Manage directory and authentication services.
Mac OS X Server QuickTime Streaming Server Administration for Version 10.4 or Later	Set up and manage QuickTime streaming services.

This guide ...	tells you how to:
Mac OS X Server Windows Services Administration for Version 10.4 or Later	Set up and manage services including PDC, BDC, file, and print for Windows computer users.
Mac OS X Server Migrating from Windows NT to Version 10.4 or Later	Move accounts, shared folders, and services from Windows NT servers to Mac OS X Server.
Mac OS X Server Java Application Server Administration For Version 10.4 or Later	Configure and administer a JBoss application server on Mac OS X Server.
Mac OS X Server Command-Line Administration for Version 10.4 or Later	Use commands and configuration files to perform server administration tasks in a UNIX command shell.
Mac OS X Server Collaboration Services Administration for Version 10.4 or Later	Set up and manage weblog, chat, and other services that facilitate interactions among users.
Mac OS X Server High Availability Administration for Version 10.4 or Later	Manage link aggregation, load balancing, and other hardware and software configurations to ensure high availability of Mac OS X Server services.
Mac OS X Server Xgrid Administration for Version 10.4 or Later	Manage computational Xserve clusters using the Xgrid application.
Mac OS X Server and Storage Glossary	Interpret terms used for server and storage products.

Getting Documentation Updates

Periodically, Apple posts new onscreen help topics, revised guides, and solution papers. The new help topics include updates to the latest guides.

- To view new onscreen help topics, make sure your server or administrator computer is connected to the Internet and click the Late-Breaking News link on the main Mac OS X Server help page.
- To download the latest guides and solution papers in PDF format, go to the Mac OS X Server documentation webpage: www.apple.com/server/documentation.

Getting Additional Information

For more information, consult these resources:

Read Me documents—important updates and special information. Look for them on the server discs.

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

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

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

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

Apple mailing list directory —subscribe to mailing lists so you can communicate with other administrators using email.
www.lists.apple.com/

Samba website —information about Samba, the open source software on which the Windows services in Mac OS X Server are based.
www.samba.org

Consider obtaining some of the following reference materials. They contain background information, explanations of basic concepts, and ideas for getting the most out of your network.

- *Teach Yourself Networking Visually,* by Paul Whitehead and Ruth Maran (IDG Books Worldwide, 1998).
- *Internet and Intranet Engineering,* by Daniel Minoli (McGraw-Hill, 1997).

If You're an Experienced Server Administrator

If you're already familiar with network administration and you've used, Linux, UNIX, or a similar operating system, you may find these additional references useful.

- *Read Me documents*—important updates and special information. Look for them on the server discs.
- *Apple Service & Support website*—access to hundreds of articles from Apple's support organization.
  www.apple.com/support
- *Apple mailing list directory/*—subscribe to mailing lists so you can communicate with other administrators using email.
  www.lists.apple.com
- You can obtain a variety of relevant books from O'Reilly & Associates See the O'Reilly & Associates website: www.ora.com.

For detailed information about Apache, go to: www.apache.org/.

Web Technologies Overview

1

Become familiar with web technologies and understand the major components before setting up your services and sites.

Web Technologies Overview

Web technologies in Mac OS X Server offer an integrated Internet server solution. Web technologies—also called web service in this guide—are easy to set up and manage, so you don't need to be an experienced web administrator to set up multiple websites and configure and monitor your web server.

Web technologies in Mac OS X Server are based on Apache, an open source HTTP web server. A web server responds to requests for HTML webpages stored on your site. Open source software allows anyone to view and modify the source code to make changes and improvements. This has led to Apache's widespread use, making it the most popular web server on the Internet today.

Web administrators can use Server Admin to administer web technologies without knowing anything about advanced settings or configuration files. Web administrators proficient with Apache can choose to administer web technologies using Apache's advanced features.

In addition, web technologies in Mac OS X Server include a high-performance, front-end cache that improves performance for websites that use static HTML pages. With this cache, static data doesn't need to be accessed by the server each time it is requested.

Web service also includes support for Web-based Distributed Authoring and Versioning, known as WebDAV. With WebDAV capability, your client users can check out webpages, make changes, and then check the pages back in while the site is running. In addition, the WebDAV command set is rich enough that client computers with Mac OS X installed can use a WebDAV-enabled web server as if it were a file server.

Since web service in Mac OS X Server is based on Apache, you can add advanced features with plug-in modules. Apache modules allow you to add support for Simple Object Access Protocol (SOAP), Java, and CGI languages such as Python.

Key Web Components

Web technologies in Mac OS X Server consist of several key components, which provide a flexible and scalable server environment.

Apache Web Server

Apache is an open source HTTP web server that administrators can configure with the Server Admin application.

Apache has a modular design, and the set of modules enabled by default is adequate for most uses. Server Admin can control a few optional modules. Experienced Apache users can add or remove modules and modify the server code. For information about modules, see "Apache Modules" on page 71.

Apache version 1.3 is installed in Mac OS X Server. Apache version 2 is provided with the server software for evaluation purposes; it is located in /opt/apache2/.

WebDAV

Web-based Distributed Authoring and Versioning (WebDAV) is particularly useful for updating content on a website. Users who have WebDAV access to the server can open files, make changes or additions, and save those revisions.

You can also use the realms capability of Apache to control access to WebDAV folders containing all or part of a website's content.

You can also use WebDAV as if it were a file server, so that authorized users in different locations and on different platforms can read, copy, post, and modify files on a website.

For more about using WebDAV for file sharing, see "Using WebDAV to Share Files" on page 45.

CGI Support

The Common Gateway Interface (CGI) provides a means of interaction between the server and clients. For example, CGI scripts allow users to place an order for a product offered on a website or submit responses to information requests.

You can write CGI scripts in any of several scripting languages, including Perl and Python. The folder /Library/WebServer/CGI-Executables is the default location for CGI scripts.

SSL Support

Web service includes support for Secure Sockets Layer (SSL), a protocol that encrypts information being transferred between the client and server. SSL works in conjunction with a digital certificate that provides a certified identity for the server by establishing a secure, encrypted exchange of information.

Dynamic Content With Server Side Includes (SSI)

Server side includes provide a method for using the same content on multiple pages in a site. They also can tell the server to run a script or insert specific data into a page. This feature makes updating content much easier, because you revise information in only one place and the SSI command displays that revised information on many pages.

See "Enabling Server Side Includes (SSI)" on page 49 for more information about SSI.

Front-End Cache

The web server includes a high-performance cache that increases performance for websites that serve static pages. The static content stays in the cache once used, so the server can quickly retrieve this content when it is requested again.

See "Improving Performance of Static Websites (Performance Cache)" on page 37 for more about using the performance cache.

Weblogs and RSS Support

The web server provides weblogs (blogs) as an option for each website. The weblogs comply with RSS and Atom XML standards and allow Open Directory authentication. Log users can choose from several techniques for working with templates and style sheets. Weblogs in Mac OS X Server are based on Blojsom, an open source application.

Note: When you turn on weblogs, they are on for every site on the server.

Detailed coverage of weblogs is provided in Chapter 3, "Weblog Service," of the collaboration services administration guide.

Before You Begin

This section provides information you need to know before you set up your web server for the first time. You should read this section even if you are an experienced web administrator, as some features and behaviors may be different from what you expect.

Configuring Your Web Server

You can use Server Admin to set up and configure most features of your web server. If you are an experienced Apache administrator and need to work with features of the Apache web server that aren't included in Server Admin, you can modify the appropriate configuration files. However, Apple does not provide technical support for modifying Apache configuration files. If you choose to modify a file, be sure to make a backup copy first. Then you can revert to the copy should you have problems.

For more information about Apache modules, see the Apache Software Foundation website at http://www.apache.org.

Providing Secure Transactions

If you want to provide secure transactions on your server, you should set up Secure Sockets Layer (SSL) protection. SSL lets you send encrypted, authenticated information across the Internet. If you want to allow credit card transactions through your website, for example, you can use SSL to protect the information that's passed to and from your site.

Important: You can't use the performance cache for a website if SSL is enabled for that site. See "Understanding the Effect of Using a Web Service Performance Cache" on page 37 for more information.

For instructions on how to set up secure transactions, see "Enabling SSL" on page 50.

Setting Up Websites

Before you can host a website, you must:

- Register your domain name with a domain name authority
- Create a folder for your website on the server
- Create a default page in the folder for users to see when they connect
- Verify that DNS is properly configured if you want clients to access your website by name

When you are ready to publish, or enable, your site, you can do this using Server Admin. The Sites pane in the Settings window lets you add a new site and select a variety of settings for each site you host.

See Chapter 3, "Managing Websites," on page 33 for more information.

Hosting More Than One Website

You can host more than one website simultaneously on your web server. Depending on how you configure your sites, they may share the same domain name, IP address, or port. The unique combination of domain name, IP address, and port identifies each separate site. Your domain names must be registered with a domain name authority such as InterNIC. Otherwise, the website associated with the domain won't be visible on the Internet. (There is a fee for each additional name you register.)

If you configure websites using multiple domain names and one IP address, older browsers that do not support HTTP 1.1 or later (that don't include the "Host" request header), will not be able to access your sites. This is an issue only with software released prior to 1997 and does not affect modern browsers. If you think your users will be using very old browser software, you'll need to configure your sites with one domain name per IP address.

See "Managing Multiple Sites on One Server" for more about multiple sites.

Understanding WebDAV

If you use WebDAV to provide live authoring on your website, you should create realms and set access privileges for users. Each site you host can be divided into a number of realms, each with its own set of users and groups that have either browsing or authoring privileges.

Defining Realms

When you define a *realm,* which is typically a folder (or directory), the access privileges you set for the realm apply to all the contents of that directory. If a new realm is defined for one of the folders within the existing realm, only the new realm privileges apply to that folder and its contents. For information about creating realms and setting access privileges, see "Setting Access for Websites" on page 43.

Setting WebDAV Privileges

The Apache process running on the server needs to have access to the website's files and folders. To provide this access, Mac OS X Server installs a user named "www" and a group named "www" in the server's Users & Groups List. The Apache processes that serve webpages run as the www user and as members of the www group. You need to give the www group read access to files within websites so that the server can transfer the files to browsers when users connect to the sites. The Apache process runs with effective user id and group id of www and needs access to the files and directories in the WebDAV realm, and to the /var/run/davlocks directory.

Understanding WebDAV Security

In Mac OS X Server 10.4, WebDAV lets you use a web server as a file server. Clients use their browsers from any location, on any type of computer, to access and share files on the server. See "Using WebDAV" for more information about using WebDAV for file sharing.

WebDAV also lets users update files in a website while the site is running. When WebDAV is enabled, the web server must have write access to the files and folders within the site users are updating.

Both features of WebDAV—providing a file server with browser access and website updating—have significant security implications when other sites are running on the server, because individuals responsible for one site may be able to modify other sites.

You can avoid this problem by carefully setting access privileges for the site files using the Sharing module of the Workgroup Manager application. Mac OS X Server uses a predefined group www, which contains the Apache processes. You need to give the www group Read & Write access to files within the website. You also need to assign these files Read & Write access by the website administrator (Owner) and No Access to Everyone.

Understanding Multipurpose Internet Mail Extension

Multipurpose Internet Mail Extension (MIME) is an Internet standard for specifying what happens when a web browser requests a file with certain characteristics. You can choose the response you want the web server to make based on the file's suffix. Your choices will depend partly on what modules you have installed on your web server. Each combination of a file suffix and its associated response is called a *MIME type mapping*.

MIME Suffixes

A *suffix* describes the type of data in a file. Here are some examples:

- txt for text files
- cgi for Common Gateway Interface files
- gif for GIF (graphics) files
- php for PHP: Hypertext Preprocessor (embedded HTML scripts) used for WebMail, and so on
- tiff for TIFF (graphics) files

Mac OS X Server includes a default set of MIME type suffixes. This set includes all the suffixes in the mime.types file distributed with Apache, with a few additions. If a suffix you need is not listed, or does not have the behavior you want, use Server Admin to add the suffix to the set or to change its behavior.

Note: Do not add or change MIME suffixes by editing configuration files.

Web Server Responses (Content Handlers)

When a file is requested, the web server handles the file using the response specified for the file's suffix. Responses, also known as content handlers, can be either an action or a MIME type. Possible responses include:

- Return file as MIME type (you enter the mapping you want to return)
- Send-as-is (send the file exactly as it exists)
- Cgi-script (run a CGI script you designate)
- Imap-file (generate an IMAP mail message)
- Mac-binary (download a compressed file in MacBinary format)

MIME type mappings are divided into two subfields separated by a forward slash, such as text/plain. Mac OS X Server includes a list of default MIME type mappings. You can edit these and add others using the Server Admin application.

When you specify a MIME type as a response, the server identifies the type of data requested and sends the response you specify. For example, if the browser requests a file with the suffix "jpg," and its associated MIME type mapping is image/jpeg, the server knows it needs to send an image file and that its format is JPEG. The server doesn't have to do anything except serve the data requested.

Actions are handled differently. If you've mapped an action to a suffix, your server runs a program or script, and the result is served to the requesting browser. For example, if a browser requests a file with the suffix "cgi," and its associated response is the action cgi-script, your server runs the script and returns the resulting data to the requesting browser.

Managing Web Technologies 2

Use Server Admin to set up web technologies initially and to manage web settings and components.

If you are familiar with web servers and their content, you can use these summary steps to get your web server started. If you'd like more detailed instructions for these tasks, see the similar topics in Chapter 3, "Managing Websites," on page 33.

Setting Up Your Web Server for the First Time

Setting up your web server involves these procedures.

Step 1: Set up the Documents folder

When your server software is installed, a folder named Documents is set up automatically in the WebServer directory. Put any items you want to make available through a website in the Documents folder. You can create folders within the Documents folder to organize the information. The folder is located in the directory /Library/WebServer/Documents.

In addition, each registered user has a Sites folder in the user's own home directory. Any graphics or HTML pages stored in the user's Sites folder will be served from the URL http://server.example.com/~username/.

Step 2: Create a default page

Whenever users connect to your website, they see the default page. When you first install the software, the file index.html in the Documents folder is the default page. You'll need to replace this file with the first page of your website and name it index.html. If you want to call the file something else, make sure you add that name to the list of default index files and move its name to the top of the list in the General pane of the site settings window of Server Admin. See "Setting the Default Page for a Website" on page 36 for instructions on specifying default index file names.

For more information about all website settings, see Chapter 3, "Managing Websites," on page 33.

Step 3: Assign privileges for your website

The Apache processes that serve webpages must have read access to the files, and read/execute access to the folders. (In the case of folders, execute access means the ability to read the names of files and folders contained in that particular folder.) Those Apache processes run as user www—a special user created specifically for Apache when Mac OS X Server is installed. The user www is a member of the group www. So for the Apache process to access the content of the website, the files and folders need to be readable by user www.

Consequently, you need to give the www group at least read-only access to files within your website so that it can transfer those files to browsers when users connect to the site. You can do this by:

- Making the files and folders readable by everyone regardless of their user or group ownership
- Making www the owner of files and folders and making sure that the files and folders are readable by the owner
- Making the group www the owner of the files and folders and making sure that the files and folders are readable by the group
- Making sure the files and folders are readable by world, regardless of their ownership and group settings. This is the default case.

For information about assigning privileges, see the file services administration guide.

Step 4: Configure your web server

The default configuration works for most web servers that host a single website, but you can configure all the basic features of web service and websites using Server Admin. For more advanced configuration options, see Chapter 5, "Working With WebObjects and Web-Related Open Source Applications," on page 61.

To host user websites, you must configure at least one website.

To configure a site:

1. Open Server Admin.
2. Click Web in the list for the server you want.
3. Click Settings in the button bar.
4. In the Sites pane, click the Enabled button for the site you want to turn on.
5. Double-click the site name and choose the configuration options you want for the site.

   For information about these settings, see Chapter 3, "Managing Websites," on page 33.

Step 5: Start web service

1. Open Server Admin and click Web in the list below the server name.
2. Click Start Service in the toolbar.

Important: Always use Server Admin to start and stop the web server. You can start the web server from the command line, but Server Admin won't show the change in status for several seconds. Server Admin is the preferred method to start and stop the web server and modify web server settings.

Step 6: Connect to your website

To make sure the website is working properly, open your browser and try to connect to your website over the Internet If your site isn't working correctly, see Chapter 7, "Solving Problems," on page 75.

Using Server Admin to Manage Your Web Server

The Server Admin application lets you set and modify most options for your web server.

To access the web settings window:

1. In Server Admin, click Web in the list for the server you want.
2. Click Settings in the button bar.

   Note: Click one of the five buttons at the top to see the settings in that pane.
3. Make the changes you want in settings.
4. Click Save.

   The server restarts when you save your changes.

Starting or Stopping Web Service

You start and stop web service from the Server Admin application.

To start or stop web service:

1. In Server Admin, click Web in the list for the server you want.
2. Click Start Service or Stop Service in the toolbar.

   If you stop web service, users connected to any website hosted on your server are disconnected immediately.

   Important: Always use Server Admin to start and stop the web server. You can start the web server from the command line, but Server Admin won't show the change in status for several seconds. Server Admin is the preferred method to start and stop the web server and modify web server settings.

   You can also use the serveradmin command-line tool to start or stop web service. Connect to the server and enter one of the commands below.

```
serveradmin start web
serveradmin stop web
```

Starting Web Service Automatically

Web service is set to start automatically (if it was running at shutdown) when the server starts up. This will ensure that your websites are available if there's been a power failure or the server shuts down for any reason.

When you start web service in the Server Admin toolbar, the service starts automatically each time the server restarts. If you turn off web service and then restart the server, you must turn web service on again.

Modifying MIME Mappings and Content Handlers

Multipurpose Internet Mail Extension (MIME) is an Internet standard for describing the contents of a file. The MIME Types pane lets you set up how your web server responds when a browser requests certain file types. For more information about MIME types and MIME type mappings, see "Understanding Multipurpose Internet Mail Extension" on page 18.

Content handlers are Java programs used to manage different MIME type-subtype combinations, such as text/plain and text/richtext.

The server includes the MIME type in its response to a browser to describe the information being sent. The browser can then use its list of MIME preferences to determine how to handle the information.

The server's default MIME type is text/html, which specifies that a file contains HTML text.

The web server is set up to handle the most common MIME types and content handlers. You can add, edit, or delete MIME type and content handler mappings. In the Server Admin application, these files are displayed in two lists: MIME Types and Content Handlers. You can edit items in each list and add or delete items in either list.

To add or modify a MIME type or content handler mapping:

1. In Server Admin, click Web in the list for the server you want.
2. Click Settings in the button bar.
3. In the MIME Types pane, click the Add button below the appropriate list to add a new mapping, or select a mapping and click the Delete or Edit button. (If you choose Delete, you've finished.)

4 In the new sheet that appears, do one of the following:

For a new MIME type, type each part of the name (separated by a slash), select the suffix and type its name, use the Add button to add any suffixes you want, then click OK.

For a new content handler, type a name for the handler, select the suffix and type its name, use the Add button to add any suffixes you want, then click OK.

To edit a MIME type or content handler, change its name as desired, select the suffix and change it as desired, add any suffixes you want using the Add button, then click OK.

If you add or edit a handler that has Common Gateway Interface (CGI) script, make sure you have enabled CGI execution for your site in the Options pane of the Settings/Sites window.

5 Click Save.

Managing Weblogs (Blogs)

You can turn on the weblog component of web service and set a format for weblogs. Users can then modify the weblog format and set an email domain from within their weblog.

To turn weblogs on or off and manage weblogs:

1 In Server Admin, click Web in the list for the server you want.

2 Click Settings in the button bar.

3 In the Weblogs pane, click the Enable Weblogs to turn this option on or off.

4 Choose a default theme from the pop-up menu.

5 Type a path for the weblog folder.

You can also use the browse button to locate the weblog folder.

6 Type a domain name to use for the default email address for new weblogs.

The email address is set automatically; each weblog's user can change it as desired.

Note: When you turn on weblogs, they are on for all sites on the web server.

Detailed coverage of weblogs is provided in Chapter 3, "Weblog Service," of the collaboration services administration guide.

Managing Connections

You can limit the period of time that users are connected to the server. In addition, you can specify the number of connections to websites on the server at any one time.

Setting Simultaneous Connections for the Web Server

You can specify the number of simultaneous connections to your web server. When the maximum number of connections is reached, new requests receive a message that the server is busy.

Simultaneous connections are concurrent HTTP client connections. Browsers often request several parts of a webpage at the same time, and each of those requests is a connection. So a high number of simultaneous connections can be reached if the site has pages with multiple elements and many users are trying to reach the server at once.

To set the maximum number of connections to your web server:

1. In Server Admin, click Web for the server you want.
2. Click Settings in the button bar.
3. In the General pane, enter a number in the "Maximum simultaneous connections" field.

   The range for maximum simultaneous connections is 1 to 2048. The default maximum is 500, but you can set the number higher or lower, taking into consideration the desired performance of your server.
4. Click Save.

Setting Persistent Connections for the Web Server

You can set up your web server to respond to multiple requests from a client computer without closing the connection each time. Repeatedly opening and closing connections isn't very efficient and decreases performance.

Most browsers request a persistent connection from the server, and the server keeps the connection open until the browser closes the connection. This means the browser is using a connection even when no information is being transferred. You can allow more persistent connections—and avoid sending a Server Busy message to other users—by increasing the number of persistent connections allowed.

Important: Persistent connections are not compatible with the performance cache.

To set the number of persistent connections:

1. In Server Admin, click Web in the list for the server you want.
2. Click Settings in the button bar.
3. In the General pane, click Allow Persistent Connections if it is not checked.
4. Enter a number in the "Maximum persistent connections" field.

   The range for maximum persistent connections is 1 to 2048.
5. Click Save.

   Web service restarts when you save the changes.

   Note: The Apache documentation refers to persistent connects as "Keep-Alive."

Setting a Connection Timeout Interval

You can specify a time period after which the server will drop a connection that is inactive.

To set the connection timeout interval:

1. In Server Admin, click Web in the list for the server you want.
2. Click Settings in the button bar.
3. In the General pane, enter a number in the "Persistent connection timeout" field to specify the amount of time that can pass between requests before the session is disconnected by the web server.

   The range for connection timeout is 0 to 9999 seconds.
4. Click Save.

Specifying Who Has Access to Web Service

You can allow all users to have access to the web server, or you can specify that only certain users or groups can connect to the server.

To specify who has access to web service:

1. In Server Admin, click the server you want in the list.
2. Click Settings in the button bar.
3. In the Access pane, click to remove the checkmark from "Use same access for all services" if necessary.
4. Click Web in the list of services.
5. Select "Allow only the users listed below" to limit access.
6. Click the Add button to display a list of users and groups.
7. Specify the users and groups who will have access by dragging them from the Users and Groups list to the Name field.
8. Click Save.

Setting Up Proxy Caching

A proxy lets users check a local server for frequently used files. You can use a proxy to speed up response times and reduce network traffic. The proxy stores recently accessed files in a cache on your web server. Browsers on your network check the cache before retrieving files from more distant servers.

To take advantage of this feature, client computers must specify your web server as their proxy server in their browser preferences.

If you want to set up a web proxy, make sure you create and enable a website for the proxy. You may wish to disable logging on the proxy site, or configure the site to record its access log in a separate file from your other sites' access logs. The site does not have to be on port 80, but setting up web clients is easier if it is because browsers use port 80 by default.

Important: If you don't restrict access to your server as a proxy, anyone may have access, which can be a security risk. This is particularly true is your server hosts both internal and external websites.

To set up a proxy:

1. In Server Admin, click Web for the server you want.
2. Click Settings in the button bar.
3. In the Proxy pane, click Enable Proxy.
4. Click Control Access To Proxy to limit access and type the domain name to allow.

   Generally, if you want to limit who can use your web server as a proxy, limit access of a specific domain. Users within that domain will have access.
5. Set the maximum cache size.

   When the cache reaches this size, the oldest files are deleted from the cache folder.
6. Type the pathname for the cache folder in the "Cache folder" field.

   You can also click the Browse button and browse for the folder you want to use.

   If you are administering a remote server, file service must be running on the remote server to use the Browse button.

   If you change the folder location from the default, you will have to select the new folder in the Finder, choose File > Get Info, and change the owner and group to www.
7. Click the Add button to add a host you want to block and type its URL.

   Continue to add names of hosts to block as necessary.
8. Click Save.

   Note: If proxy is enabled, any site on the server can be used as the proxy.

Blocking Websites From Your Web Server Cache

If your web server is set up to act as a proxy, you can prevent the server from caching objectionable websites.

Important: To take advantage of this feature, client computers must specify your web server as their proxy server in their browser preferences.

You can import a list of websites by dragging it to list of sites. The list must be a text file with the host names separated by commas or tabs (also known as csv and tsv strings). Make sure that the last entry in the file is terminated with a carriage return/line feed, or it will be overlooked.

To block websites:

1. In Server Admin, click Web for the server you want.
2. Click Settings in the button bar.
3. In the Proxy pane, click Enable Proxy.
4. Do one of the following:
   - Click the Add button and type the URL of the website you want to block.
   - Drag a list of websites (text file in comma-separated or tab-separated format) to the "Blocked hosts" field.
5. Click Save.

Using Secure Sockets Layer (SSL)

Secure Sockets Layer (SSL) provides security for a site and for its users by authenticating the server, encrypting information, and maintaining message integrity.

About SSL

The SSL protocol is on a layer below application protocols (HTTP, for example) and above TCP/IP. This means that when SSL is operating in the server and the client's software, all information is encrypted before being sent.

The Apache web server in Mac OS X Server supports SSLv2, SSLv3, and TLSv1. More information about these protocol versions is available at www.modssl.org.

The Apache server in Mac OS X Server uses a public key-private key combination to protect information. A browser encrypts information using a public key provided by the server. Only the server has a private key that can decrypt that information.

When SSL is implemented on a server, a browser connects to it using the https prefix in the URL, rather than http. The "s" indicates that the server is secure.

When a browser initiates a connection to an SSL-protected server, it connects to a specific port (443) and sends a message that describes the encryption ciphers it recognizes. The server responds with its strongest cipher, and the browser and server then continue exchanging messages until the server determines the strongest cipher both it and the browser recognize. Then the server sends its certificate (the Apache web server uses an ISO X.509 certificate) to the browser; this certificate identifies the server and uses it to create an encryption key for the browser to use. At this point a secure connection has been established and the browser and server can exchange encrypted information.

Using WebDAV

Web-based Distributed Authoring and Versioning (WebDAV) allows you or your users to make changes to websites while the sites are running. You enable WebDAV for individual sites, and you also need to assign access privileges for the sites and for the web folders. See "Enabling WebDAV on Websites" on page 44 for details.

Using Tomcat

Tomcat adds Java servlet and JavaServer Pages (JSP) capabilities to Mac OS X Server. Java servlets are Java-based applications that run on your server, in contrast to Java applets, which run on the user's computer. JavaServer Pages allows you to embed Java servlets in your HTML pages.

You can set Tomcat to start automatically whenever the server starts up. This ensures that the Tomcat module starts up after a power failure or after the server shuts down for any reason. You can use the Server Admin or Terminal application to enable Tomcat; see "Tomcat" on page 68 for details.

Using WebObjects

WebObjects is the Apple solution for rapid development and deployment of ecommerce and other Internet applications. WebObjects applications can connect to multiple databases and dynamically generate HTML content. WebObjects offers a comprehensive suite of tools and run-time libraries that facilitate developing standards-based web services and Java server applications.

You can set WebObjects to start automatically whenever the server starts up. This ensures that the WebObjects modules starts up after a power failure of after the server shuts down for any reason. You use the Server Admin application to turn WebObjects on or off; see "WebObjects" on page 61 for details.

Using JBoss

JBoss is a widely used full-featured Java application server. It provides a full Java 2 Platform, Enterprise Edition (J2EE) technology stack with features such as:

- An Enterprise Java Bean (EJB) container
- Java Management Extensions (JMX)
- Java Connector Architecture (JCA)

You can set JBoss to start automatically whenever the server starts up. This ensures that the JBoss module starts up after a power failure or after the server shuts down for any reason. You use the Server Admin or Terminal application to enable JBoss; see "JBoss" on page 66 for details.

You can use Server Admin or the command-line tool to enable the Tomcat module. See "Tomcat" on page 68 for more information about Tomcat and how to use it with your web server.

Viewing Web Service Status

In Server Admin you can check the current state of the Apache server and which server modules are active.

Web Service Overview

The overview in Server Admin shows server activity in summary form.

To view web service status overview:

1. Open Server Admin.
2. Click Overview in the button bar.

   The Start/Stop Status Messages field displays a summary of server activity and the server's start date and time.

   You can also view activity logs for each site on your server.

   See "Viewing Website Settings" on page 49 for more information.

Web Service Modules in Use

You can view a list of modules in use on the server as well as modules that are available but not in use.

To see which modules are enabled:

1. In Server Admin, click Web in the list for the server your want.
2. Click Settings in the button bar.
3. In the Modules pane, scroll to see the entire set of modules in use or available for use in the server.

Viewing and Searching Logs of Web Service Activity

Web service in Mac OS X Server uses the standard Apache log format, so you can also use any third-party log analysis tool to interpret the log data.

To view the log files:

1. In Server Admin, click Web in the list for the server you want.
2. Click Logs in the button bar.
3. Select the log you want to view in the list.

To search the log files, display the log you want to search, enter search text in the Filter field at the bottom of the log window, and press Return.

You can enable an access log and an error log for each site on the server. See "Enabling Access and Error Logs for a Website" on page 39 for more information.

Managing Websites

3

Use the Server Admin application to set up and manage the essential components of web service.

You administer websites on your server with Server Admin, an application that allows you to establish settings, specify folders and paths, enable a variety of options, and view the status of sites.

Using Server Admin to Manage Websites

The Sites pane in Server Admin lists your websites and provides some basic information about each site. You use the Sites pane to add new sites or change settings for existing sites.

To access the Sites pane:

- In Server Admin, click Web in the list for the server you want, click Settings in the button bar, then click Sites.

  The pane shows a list of sites on the server.

- To edit a site, double-click the site name.

Setting Up the Documents Folder for a Website

To make files available through a website, you put the files in the Documents folder for the site. To organize the information, you can create folders inside the Documents folder. The folder is located in the directory /Library/WebServer/Documents/.

In addition, each registered user has a Sites folder in the user's own home directory. Any graphics or HTML pages stored here will be served from the URL: http://server.example.com/~username/.

To set up the Documents folder for your website:

1. Open the Documents folder on your web server.

   If you have not changed the location of the Documents folder, it's in this directory: /Library/WebServer/Documents/.

2 Replace the index.html file with the main page for your website.

Make sure the name of your main page matches the default document name you set in the General pane of the site's Settings window. See"Setting the Default Page for a Website" on page 36 for details.

3 Copy files you want to be available on your website to the Documents folder.

Enabling a Website on a Server

Before you can enable a website, you must create the content for the site and set up your site folders.

To enable the website:

1 In Server Admin, click Web in the list for the server you want.

2 Click Settings in the button bar.

3 In the Sites pane, click the Add button to add a new site or click the Enabled button for the site in the list that you want to enable. (If the site is already listed, you're finished.)

4 Double-click the site name to edit it.

5 In the General pane, type the fully qualified DNS name of your website in the Domain Name field.

Note: You can leave the domain name blank and the IP address set to "any" and the site will still operate.

6 Enter the IP address and port number for the site.

The default port number is 80. If you are using SSL, the port is 443. Make sure that the number you choose is not already in use by another service on the server.

Important: In order to enable your website on the server, the website must have a unique name, IP address, and port number combination. See "Hosting More Than One Website" on page 16 for more information.

7 Enter the path to the folder you set up for this website.

You can also click the Browse button and browse for the folder you want to use.

8 Enter the file name of your default document (the first page users see when they access your site).

9 Make any other settings you want for this site, then click Save.

10 Click the back button at the top left side of the editing window.

11 Click the Enabled box next to the site name in the Sites pane, if necessary.

12 Click Save.

Managing Multiple Sites on One Server

You can create multiple sites on the same web server, at the same IP address (virtual hosts) or at separate, secondary IP addresses (multihoming).

Virtual hosts are multiple sites on the same server. These sites can be named-based (such as www.example.com) or use IP addresses (such as 10.201.42.73). You can use the Server Admin application to manage both named-based, and IP-based virtual hosts.

A multihomed site is a site that has more than one connection to the public Internet. Multihoming is typically done to improve reliability and performance. Those multiple connections might be through the same Internet service provider (ISP) or through multiple ISPs, and they might involve multiple IP addresses or one address.

An Example of Using Aliases to Have a Site Respond to Multiple Names

If you want a website to respond to multiple names, choose one name as primary and add the other names as aliases. To set up a website this way, use the primary name as the site's name in Server Admin (double-click the site and enter the primary name in the General pane for the site, then add the other desired names in the Aliases pane for that site).

For instance, if you want your website to respond to example.com, www.example.com, and widget.example.com, you could set it up as follows (the names and IP addresses are examples only):

Primary name: www.example.com (entered in the General pane for the site).

Secondary names: example.com and widget.example.com (entered in the Web Server Aliases table of the site's Aliases pane).

Ensure that your DNS server aliases your web server's address to all three domain names as well.

Virtual Hosts and Multiple Network Interfaces

By default, the web server is configured with a single "wildcard" virtual host. Such a virtual host is useful for these reasons:

- It responds on all network interfaces and on all IP addresses on all those interfaces
- It responds to any DNS name that maps to any of those addresses

Other virtual hosts can be added using the Server Admin application. When virtual host are added, the administrator can associate a specific IP address or a wildcard address with each virtual host. (Note that this association with an IP address, not a network interface name.)

In terms of virtual host listener configuration, Apache knows nothing about network interface names such as en0; it only knows IP addresses and virtual host names.

If the web server has multiple interfaces and multiple addresses, configuring Apache to use them is simply a matter of configuring virtual hosts to listen on the desired addresses. An even simpler scenario is to allow the wildcard virtual host to respond to all the addresses, which it does by default.

Setting Up a Web Folder for a Site

A site's default web folder is used as the root for the site (called DocumentRoot in Apache). In other words, the default folder is the top level of the directory structure for the site.

To change the default web folder for a site hosted on your server:

1. Log in to the server you want to administer.

   You need access to the file system on the server.
2. Drag the contents of your previous web folder to your new web folder.
3. In Server Admin, click Web in the list for the server where the website is located.
4. Click Settings in the button bar.
5. In the Sites pane, double-click the site in the list.
6. Type the path to the web folder in the Web Folder field, or click the Browse button and navigate to the new web folder location.
7. Click Save.

Setting the Default Page for a Website

The default page appears when a user connects to your website by specifying a directory or host name instead of a file name.

You can have more than one default page (called a default index file in Server Admin) for a site. If multiple index files are listed for a site, the web server uses the first one listed in the site's folder.

To set the default webpage:

1. In Server Admin, click Web in the list for the server you want.
2. Click Settings in the button bar.
3. In the Sites pane, double-click the site in the list.
4. In the General pane, click the Add button and type a name in the "Default index files" field. (Do not use any spaces in the name.)

   A file with this name must be in the website folder.
5. To set the file as the one the server displays as its default page, drag that file to the top of the list.
6. Click Save.

   Note: If you plan to use only one index page for a site, you can leave index.html as the default index file and change the content of the existing file with that name in /Library/WebServer/Documents.

Changing the Access Port for a Website

By default, the server uses port 80 for connections to websites on your server. You may need to change the port used for an individual website, for instance, if you want to set up a streaming server on port 80. Make sure that the number you choose does not conflict with ports already being used on the server (for FTP, Apple File Service, SMTP, and others). If you change the port number for a website you must change all URLs that point to the web server to include the new port number you choose.

Note: If you turn SSL on for a site, the port for that site is automatically changed to 443. If you turn SSL off, the port will change to 80, regardless of what it was previously. A message on the screen alerts you to the port change when you turn off SSL.

To set the port for a website:

1. In Server Admin, click Web in the list for the server you want.
2. Click Settings in the button bar.
3. In the Sites pane, double-click the site in the list.
4. In the General pane, type the port number in the Port field.
5. Click Save.

Improving Performance of Static Websites (Performance Cache)

If your websites contain static files (such as images), and you expect high usage of the pages, you can enable the performance cache to improve server performance. For example, for a site that has static images that are displayed on dynamically generated pages, a good way to use the performance cache is to store all images on a separate virtual host that has the performance cache turned on, while the generated pages come from the main virtual host.

The performance cache is enabled by default.

You should disable the performance cache if:

- You do not anticipate heavy usage of your website.
- Most of the pages on your website are generated dynamically.

Understanding the Effect of Using a Web Service Performance Cache

Web service's performance cache is enabled by default and significantly improves performance for some websites. Sites that benefit most from the performance cache contain mostly static content and can fit entirely in RAM. Website content is cached in system RAM and is accessed very quickly in response to client requests.

Enabling the performance cache does not always improve performance. For example, when the amount of static web content exceeds the physical RAM of your server, using a performance cache increases memory swapping, which degrades performance.

Note: If you turn on SSL, the performance cache is automatically turned off.

Also note that when your server is running other services that compete for physical RAM, such as AFP, the web performance cache may be less effective or may impact the performance of those other services.

Consider these points when determine whether to use the performance cache for a website.

- When the performance cache is enabled for any virtual hosts, a process named webperfcache runs and takes over the listener on the configured TCP port (usually port 80). The web server is then configured to listen on a different TCP port (usually port 16080). All requests are received by the webperfcache process, and they are either served from the in-memory cache or relayed to the web server. In the latter case, the performance cache then receives a response from the web server, saves it in its in-memory cache (if it can be cached and caching is enabled for that virtual host), and returns the response to the requester.
- The performance cache is not compatible with Apache's connection Keep-Alive mechanism. By default, the Keep-Alive mechanism is disabled; you can turn it on in Server Admin by checking Allow Persistent Connections in the General pane for Web Service.
- The performance cache is also incompatible with SSL (as noted above), cookies, and WebDAV.
- If a request contains cookie headers, the performance cache considers the request uncacheable and does not cache any responses for that request.

The performance cache respects the expiration times of cacheable items such as images, audio, and video files. By default these are all configured to expire after one hour. If you are running a busy website with the performance cache enabled and you notice a heavier load on your web server once per hour, it may be the result of all the image, audio, and video files expiring at once. You may be able to flatten the load profile by staggering the expiration times of these file types. These are controlled by Apache's ExpiresByType directive. This directive is not configurable with the Server Admin application, so you'd need to use a text editor to edit /etc/httpd/httpd.conf.

Additional information about the performance cache can be found in its configuration file: /etc/webperfcache/webperfcache.conf.

To enable or disable the performance cache for a website:

1. In Server Admin, click Web in the list for the server you want.
2. Click Settings in the button bar.

3 In the Sites pane, double-click the site in the list.

4 In the Options pane, click Performance Cache to change its state.

5 Click Save.

You can also improve server performance by disabling the access log.

Interaction Between the Performance Cache and Apache

The Apache web server uses port 16080 when the performance cache is enabled. Blocking this port could interfere with web service. In previous versions of Mac OS X Server it was necessary to keep this port open in the firewall, because server redirection caused the port number to be appended to the URLs used by web browsers.

The performance cache communicates on port 80, which is normally used for Web service. Each port can accommodate only one service. Since the performance cache process is a front-end http server by itself, Apache is reconfigured to run on port 16080. That is, when the performance cache is on, Apache adds 1600 to the port number. Performance cache is enabled by default.

The performance cache stores data that can be reserved to clients who request it again. However, not all data can be cached. In this case, the performance cache retrieves the data by querying the Apache server locally on port 16080. You may sometimes see clients sending requests to port 16080 directly, as the result of a redirect sent to the clients by the server. Otherwise the performance cache process should transmit all request-response cycles.

Note: Because of this design, the performance cache process introduces a small performance penalty when serving dynamic content.

Enabling Access and Error Logs for a Website

You can set up error and access logs for individual websites that you host on your server. However, enabling the logs can slow server performance.

To enable access and error logs for a website:

1 In Server Admin, click Web in the list for the server you want.

2 Click Settings in the button bar.

3 In the Sites pane, double-click the site in the list.

4 In the Logging pane, check Enable Access Log to enable this log.

5 Set how often you want the logs to be archived by clicking the checkbox and typing a number of days.

6 Type the path to the folder where you want to store the logs.

You can also click the Browse button to locate the folder you want to use.

If you are administering a remote server, file service must be running on the remote server to use the Browse button.

7 Choose a log format from the Format pop-up menu.

8 Edit the format string, if necessary.

Note: The Help button next to the format string opens the Apache documentation web page (http://httpd.apache.org/docs/mod/mod_log_config.html#formats), which explains the parameters for format strings.

9 Enter archive, location, and level choices for the error log as desired.

10 Click Save.

Considerations for Naming Website Logs

You have the option of naming separate logs for multiple websites. If you have many virtual sites on your server, include the virtual host name in the log name for easy recognition of the logs.

If you have just two virtual hosts, you may want to use a single log (with the default name the server uses).

Analyzing Log Content

You can search the contents of a log by typing a search term in the Filter field.

You can also use various third-party tools to analyze logs of web server activity. One useful application is Analog, which can be downloaded free from www.analog.cx.

Using a Log to Detect Suspicious Activity

In some instances, you may discover virus activity by studying a site's log. For example, an unusual entry such as "winNT.<xxx>" may indicate that a virus is trying to propagate itself.

Some warning messages in a log are benign; some are not.

Understanding the Web Service access_log Format

In version 10.4 of Mac OS X Server, the web performance cache does not prevent a remote client's IP address from being logged in the access_log. The web performance cache process now adds an HTTP header named "PC-Remote-Addr" that contains the client's IP address before passing a request to the Apache web server.

With the performance cache disabled, the standard log format string on the CustomLog directive in httpd.conf remains the same as in earlier versions:

%h %l %u %t "%r" %>s %b

When the performance cache is enabled (default) the "%h" item will extract the local machine's IP address. To extract the remote client's IP address, the log format string needs to be modified as follows:

%{PC-Remote-Addr}i %l %u %t "%r" %>s %b

When you use the Server Admin application to enable and disable web performance cache for each site (virtual host), the CustomLog directive in httpd.conf for each site is adjusted automatically so your access logs should always contain the correct remote client address.

For more information about log format strings, see the information at http://httpd.apache.org/docs/mod/mod_log_config.html#formats.

Setting Up Directory Listing for a Website

When users specify the URL for a directory, you can display either a default webpage (such as index.html) or a list of the directory contents. You can display a folder list. To set up directory listing, you need to enable indexing for the website.

Note: Folder listings are displayed only if no default document is found.

To enable indexing for a website:

1. In Server Admin, click Web in the list for the server you want.
2. Click Settings in the button bar.
3. In the Sites pane, double-click the site in the list.
4. In the Options pane, select Folder Listing.
5. Click Save.

Creating Indexes for Searching Website Content

The mod_spotlight_apple Apache module allows web browsers to search the content of your website. This module replaces the mod_sherlock_apple module used in previous versions of Mac OS X Server. The indexfolder command-line tool is no longer supported.

Note: The entire Spotlight mechanism is turned off by default in Mac OS X Server version 10.4. You must turn it on to provide Spotlight-based search capability.

Step 1: Enable Spotlight by editing a config file

To enable Spotlight, you open the file /etc/hostconfig and replace SPOTLIGHT=-NO- with SPOTLIGHT=-YES-. Then either reboot or, using the Terminal application, run the System Startup Item that normally launches the Spotlight processes:

```
sudo SystemStarter start "MetaData Search"
```

This may have performance implications.

Step 2: Import your web content
Import your web content into the Spotlight MetaData store. The /Library/WebServer/Documents folder is not normally monitored by the MetaData Import processes, so you need to import the data manually, both initially and whenever you want any changes to be reflected.

It's a good practice to create different folders to hold the content of your virtual hosts. You need to manually import and update those as well.

You use the mdimport command-line too to import the data. Type the following command in Terminal:

```
sudo /System/Library/Frameworks/CoreServices.framework/Frameworks/
    MetaData.frameworks/Resources/mdimport -f /Library/WebServer/
    Documents/example-vhost/
```

The "-f" option is necessary to force the mdimport tools to process the contents of the specified folder. Note that it may take some time to import large amounts of content. See the man page for the mdimport tool for more details.

A variety of mdimporter extensions are preinstalled for most common content formats. If you have custom document formats, they may not be supported.

Step 3: Enable the mod_spotlight_apple module
Use the Modules pane in Server Admin for your server to enable mod_spotlight_apple and make sure web service is running.

Step 4: Copy the template.spotlight file
Copy the template.spotlight file from /Library/WebServer/Documents into the Document Root of each virtual host for which you want the Spotlight search to be available. You may customize the title, maximum allowed hits, and other aspects of the presentation by modifying a copy of this file.

Step 5: Advise web clients to use ".spotlight" in the URL
To access the search capability, your web clients must append ".spotlight" to the URL for virtual hosts. A sample URL is:
httpd:/vhost1.example.com/.spotlight

This present a simple search page that searches the contents of the DocumentRoot for the virtual host. Results are sorted with the most relevant hits first, although no relevance score is presented.

Connecting to Your Website

Once you configure your website, it's a good idea to view the site with a web browser to verify that everything appears as intended.

To make sure a website is working properly:

1. Open a web browser and type the web address of your server.

   You can use either the IP address or the DNS name of the server. If you've enabled SSL, be sure to use "https" in the URL instead of "http."

2. Type the port number, if you are not using the default port.

3. If you've restricted access to specific users, enter a valid user name and password.

Setting Access for Websites

You can use realms to control access and provide security for websites by specifying who has access to them. Realms are locations within a site (or the site itself) that users can view. If WebDAV is enabled, users who have authoring privileges can also make changes to content in the realm. You set up the realms and specify what users and groups have access to them.

To set access using a realm:

1. In Server Admin, click Web in the list for the server you want.

2. Click Settings in the button bar.

3. In the Sites pane, double-click the site in the list.

4. In the Realms pane, select the realm you want to edit.

   If no realm names are listed, create one using the instructions in "Enabling WebDAV on Websites" on page 44.

5. To set access for all users, do one of the following:

   If you want all users to browse or author, or both, select Can Browse or Can Author for Everyone.
   - When you select privileges for Everyone, you have these options:
   - Can Browse allows everyone who can access this realm to see it. You can add additional users and groups to the User or Group list to enable authoring for them.
   - Can Browse and Can Author together allow everyone who has access to this realm to see and make changes to it.

   If you want to assign access to specific users (and not to all users), do not select Can Browse or Can Author for Everyone.

6. To specify access for individual users and groups, click Users & Groups to open a drawer listing users and groups.

7. Click Users or Groups in the drawer's button bar to show the list you want.

8. Drag user names to the Users field or group names to the Groups field.

   Note: You can also use the add (+) button to open a sheet in which you type a user or group name and select access options.

9 Select Can Browse and Can Author for each user and group as desired.

Note: When users or members of a group you've added to the realm connect to the site, they must supply their user name and password.

10 Click Save.

Use the Realms pane to delete a user or group by selecting the name and clicking the Delete (-) button.

Enabling WebDAV on Websites

Web-based Distributed Authoring and Versioning (WebDAV) allows you or your users to make changes to websites while the sites are running. If you enable WebDAV, you also need to assign access privileges for the sites and for the web folders.

To enable WebDAV for a site:

1 In Server Admin, click Web in the list for the server you want.

2 Click Settings in the button bar.

3 In the Sites pane, double-click the site in the list.

4 In the Options pane, select WebDAV and click Save.

5 Click Realms. Double-click a realm to edit it, or click the Add button to create a new realm.

The realm is the part of the website users can access.

6 Type the name you want users to see when they log in.

7 If you want digest authentication for the realm, choose Digest from the Authorization pop-up menu, or choose Kerberos for Kerberos authentication.

If you want Kerberos authorization for the realm the server must be joined to a Kerberos domain, and SSL must be on for the site. (Because credentials are sent in the clear, Server Admin requires that SSL be on.)

Basic authorization is on by default.

8 Type the path to the location in the website to which you want to limit access, and click OK.

You can also click the Browse button to locate the folder you want to use.

9 Click Save when you have finished creating realms.

See "Setting Access for Websites" on page 43 for instructions on specifying access to realms.

Note: If you have turned off the WebDAV module in the Modules pane of Server Admin, you must turn it on again before WebDAV takes effect for a site. This is true even if the WebDAV option is checked in the Options pane for the site. See "Apache Modules" on page 71 for more about enabling modules.

Using WebDAV to Share Files

You can use WebDAV to allow authorized users to connect to a website on the server and to share files on that site. The steps below provide a brief example of setting up and using shared files using WebDAV.

- Turn on WebDAV for the site in Server Admin.

See "Enabling WebDAV on Websites" on page 44 for details.

- Set up Realms for the site in Server Admin to control access to the site.

See "Setting Access for Websites" on page 43 for details.

For example, you could create a folder for shared documents inside the website's folder and give specific people browse and author access to that folder.

- Tell authorized users how to connect to the site using the WebDav client built into Mac OS X (or Mac OS X Server).

Users can connect to the website using a WebDAV-enabled application, such as the Finder in Mac OS X, Adobe GoLive, Macromedia Dreamweaver, or Microsoft Explorer. Browsers are not generally WebDAV-enabled. But a browser can access a WebDAV-enabled site and do read operations (limited only by realm permissions configured on the web server), because WebDAV is a superset of HTTP. Write operations cannot be performed by a web browser; they require a WebDAV client, such as Goliath, or the one built into the Mac OS X file system and typically used via the Finder.

Note: To connect from another platform, see the platform-specific documentation for the appropriate WebDAV client. Microsoft platforms use an authentication mechanism that may make it difficult or impossible to mount WebDAV volumes from Mac OS X.

The URL for connecting through such an application is http://<serverURL>:<server port>/<folder or directory where collaborative files are stored>.

WebDAV and Web Content File and Folder Permissions

Mac OS X Server imposes the following constraints on web content files and folders (which are located by default in /Library/WebServer/Documents):

- For security reasons, web content files and folders should not be writable by world.
- Web content files and folders are owned by user root and group admin by default, so they are modifiable by any administrator but not by user or group www.
- To use WebDAV, web content files must be readable and writable by user or group www, and folders must be readable, writable, and executable by user or group www.
- If you need to modify web content files and folders while you are logged in as an administrator, those files or folders need to be modifiable by the administrator.

If you want to use WebDAV, you need to enable it in Server Admin and manually change the web content files' or folders' ownership to user and group www. If you are using WebDAV and you want to make changes to web content files or folders while logged in as an administrator, you need to change the web content file and folder permissions to admin, make your edits, and then restore the file and folder permissions to www.

To add sites to your web server while using WebDAV:

1. Change the group privileges of the folder containing your websites to admin (default folder location is: /Library/Webserver/Documents).
2. Add your new site folder.
3. Change the group privileges of the folder containing your websites back to www.

Enabling Integrated WebDAV Digest Authentication

You can enable digest authentication for WebDAV realms in the Realms pane of Server Admin. See "Setting Access for Websites" on page 43 for more information.

WebDAV and Web Performance Cache Conflict

If you enable both WebDAV and the web performance cache on one or more virtual hosts (sites), WebDAV clients may encounter problems if they try to upload multiple files in the Finder—the upload may fail to complete.

To avoid this problem, disable the web performance cache for virtual hosts with WebDAV enabled.

See "Improving Performance of Static Websites (Performance Cache)" on page 37 for more information about the performance cache.

Managing Access to Sites Using Aliases

You can manage access to websites by using aliases and redirect commands.

An alias is an alternate name for a website, which can be useful in simplifying the name users must enter to connect to the site. You can have multiple aliases for a single site.

Note: Server aliases and virtual hosts must be DNS names, and they must resolve to the IP address of the website.

A redirect command specifies that when a user asks for a specific directory or file on a site, their browser is sent to a different location that you designate.

Sample Aliases and Redirects

The examples below show aliases and redirects.

- Alias:

  For a host named "example.com" you might want to provide a server alias named "www.example.com."

- Redirect:

  Pattern: /images/boats.jpg

  Path: http://www.apple.com

  In this example, if the user enters the URL <your website>/images/boats.jpg and the site has a folder "images" containing the file "boats.jpg," the user's browser will be redirected to www.apple.com.

To create or edit aliases the site responds to:

1. In Server Admin, click Web in the list for the server you want.
2. Click Settings in the button bar.
3. In the Sites pane, double-click the site in the list.
4. In the Aliases pane, click the Add button under Web Server Aliases to create a new aliases or select an alias and click the Edit button.
5. Type the alias you want and click OK.
6. Click Save.

You can also redirect commands for a website.

To manage aliases and redirect commands for the site:

1. In Server Admin, click Web in the list for the server you want.

   Click Settings in the button bar.

   In the Sites pane, double-click the site in the list.

   In the Aliases pane, click the Add button under URL Aliases and Redirects to create a new redirect or select a redirect and click the Edit button.

2. Choose the an option from the Type pop-up menu.

   An alias maps from the URL term to a specific place in the file system.

   An Alias Match maps a regular expression pattern for a path to a specific path in the file system.

   A Redirect maps a specific URL term to redirect to another server.

   A Redirect Match maps a regular expression pattern for a path to redirect to another server.

3. Type the pattern for the alias or redirect.

   This is the pattern input from the incoming URL.

4 Type the path for the alias or redirect and click OK.

This is the path in the file system or the redirect that gets sent back to the querier.

5 Click Save.

Sample Aliases and Redirects

The examples below show aliases and redirects.

- Alias:

  pattern: /images

  path: /Volumes/Data/imgs

  If you have made a file system change but you don't want to have to change all of the image URLs in your HTML files, this will translate http://www.example.com/images/boat.jpg to grab the file from /Volumes/Data/boat.jpg.

- Alias match:

  pattern: ^/(.*)\.gif

  path: /Library/WebServer/Documents/gifs$1.jpg

  If you want all gifs to be stored in a particular directory but to be referenced from the web server root, this will alias http://www.example.com/logo.gif to serve the file located at /Library/WebServer/Documents/gifs/logo.gif.

- Redirect

  Pattern: /webstore

  Path: https://secure.example.com/webstore

  This redirects all queries for webstore to the secure server.

- Redirect match:

  pattern: (.*)\.jpg

  path: http://imageserver.example.com$1.jpg

  If you plan to host static content such as images on a new server, this will redirect all requests for files ending in .jpg to a different server.

Additional information and other examples of aliases and redirects are available at http://httpd.apache.org/docs/mod/mod_alias.html.

Enabling a Common Gateway Interface (CGI) Script

Common Gateway Interface (CGI) scripts (or programs) send information back and forth between your website and applications that provide different services for the site. If a CGI is to be used by only one site, install the CGI in the Documents folder for the site. The CGI name must end with the suffix ".cgi."

If a CGI is to be used by all sites on the server, install it in the /Library/WebServer/CGI-Executables folder. In this case, clients must include /cgi-bin/ in the URL for the site. For example, http://www.example.com/cgi-bin/test.cgi.

Make sure the file permissions on the CGI allow it to be executed by the user named "www." Since the CGI typically isn't owned by www, the file should be executable by everyone.

To enable a CGI for a website:

1. In Server Admin, click Web in the list for the server you want.
2. Click Settings in the button bar.
3. In the Sites pane, double-click the site in the list.
4. In the Options pane, select CGI Execution.
5. Click Save.

Note: Disabling CGIs for a site does not disable any CGIs in the CGI-Executables directory.

Enabling Server Side Includes (SSI)

Enabling Server Side Includes (SSI) allows a chunk of HTML code or other information to be shared by different webpages on your site. SSIs can also function like CGIs and execute commands or scripts on the server.

To enable SSI in Server Admin:

1. In Server Admin, click Web in the list for the server you want.
2. Click Settings in the button bar.
3. In the Sites pane, double-click the site in the list.
4. In the Options pane, select Server Side Includes (SSI).
5. Click Save.

Viewing Website Settings

You can use the Sites pane of Server Admin to see a list of your websites. The Sites pane shows:

- Whether a site is enabled
- The site's DNS name and IP address

- The port being used for the site

Double-clicking a site in the Sites pane opens the site details window, where you can view or change the settings for the site.

Setting Server Responses to MIME Types and Content Handlers

Multipurpose Internet Mail Extension (MIME) is an Internet standard for specifying what happens when a web browser requests a file with certain characteristics. Content handlers are similar and also use suffixes to determine how a file is handled. A file's suffix describes the type of data in the file. Each suffix and its associated response together is called a MIME type mapping or a content handler mapping. See "Understanding Multipurpose Internet Mail Extension" on page 18 for more information.

To set the server response for a MIME type or content handler:

1. In Server Admin, click Web in the list for the server you want.
2. Click Settings in the button bar.
3. In the MIME Types or content Handlers pane, click the Add button, or select the item in the list you want to edit and click the Edit button.
4. If necessary, type a name for a new MIME type or content handler, then type the file suffix associated with this mapping in the Suffixes field.

   If you use the suffix cgi, make sure you've enabled CGI execution for the website.
5. Click Save.

Enabling SSL

If you want to provide secure transactions on your server, such as allowing users to purchase items from a website, you should set up Secure Sockets Layer (SSL) protection. SSL lets you send encrypted, authenticated information across the Internet. If you want to allow credit card transactions through a website, for example, you can protect the information that's passed to and from that site.

Before you can enable Secure Sockets Layer (SSL) protection for a website, you have to obtain the proper certificates. When you have obtained a certificate, you can set up SSL for a site. For detailed information about certificates and their management, see the Appendix, "Certificates and Security," in the mail service administration guide.

To set up SSL for a website:

1. In Server Admin, click Web in the list for the server you want.
2. Click Settings in the button bar.
3. In the Sites pane, double-click the site in the list.
4. In the Security pane, select Enable Secure Sockets Layer (SSL).

   When you turn on SSL, a message notes that the port is changed to 443.
5. Type the location of the SSL log file in the SSL Log File field.

   You can also click the Browse button to locate the folder you want to use.

   If you are administering a remote server, file service must be running on the remote server to use the Browse button.
6. Choose the certificate you want in the pop-up menu.

   The name of the certificate must match the virtual host name if the certificate is protected by a passphrase. If the names don't match, web service won't restart.

   Note: For details on editing the certificate details, see the Appendix, "Certificates and Security," in the mail service administration guide.
7. If you choose Custom Configuration or want to edit a certificate, you may need to do the following:

   a. Click the Edit button and supply the correct information in each field for the certificate.
   b. If you received a ca.crt file from the certificate authority, click the Edit button and paste the text from the ca.crt file in the Certificate Authority File field.

      Note: The ca.crt file may be required but not sent directly to you. This file should be available on the website of the certificate authority.
   c. Type a passphrase in the Private Key Passphrase field and click OK.
8. Click Save.
9. Confirm that you want to restart web service.

   Server Admin allows you to enable SSL with or without saving the SSL password. If you did not save the passphrase with the SSL certificate data, the server prompts you for the passphrase upon restart, but won't accept manually entered passphrases. Use the Security pane for the site in Server Admin to save the passphrase with the SSL certificate data.

   Note: Detailed information about certificates and their management is in Appendix, "Certificates and Security," of the mail service administration guide.

Using a Passphrase With SSL Certificates

If you manage SSL certificates using the Server Admin application, and you use a passphrase for your certificates, Server Admin ensures that the passphrase is stored in the system keychain. When a website is configured to use the certificate and that web server is started, the getsslpassphrase(8) utility extracts the passphrase from the system keychain and passes it to the web server, as long as the certificate name matches the virtual host name.

If you would perfer not to rely on this mechanism, you can instead arrange for the Apache web server to prompt you for the passphrase when you start or restart it. To do so, enter the following from the command line:

```
sudo serveradmin settings web:IfModule:_array_id:mod_ssl.c:SSL
    PassPhraseDialog=builtin
```

Note: With this method, you must use the command line rather than Server Admin to start Apache, with the command:

```
sudo serveradmin start web
```

You will then be prompted for the certificate passphrase.

Setting Up the SSL Log for a Website

If you are using Secure Sockets Layer (SSL) on your web server, you can set up a file to log SSL transactions and errors.

To set up an SSL log:

1. In Server Admin, click Web for the server you want.
2. Click Settings in the button bar.
3. In the Sites pane, double-click the site you want to edit.
4. In the Security pane, make sure Enable Secure Sockets Layer is checked, then enter the pathname for the folder where you want to keep the SSL log in the SSL Log File field.

   You can also use the Browse button to navigate to the folder.
5. Click Save.

Enabling PHP

PHP (PHP: Hypertext Preprocessor) is a scripting language embedded in HTML that is used to create dynamic webpages. PHP provides functions similar to those of CGI scripts, but supports a variety of database formats and can communicate across networks via many different protocols. The PHP libraries are included in Mac OS X Server, but are disabled by default.

See “Open Source Modules” on page 72 for more information on PHP.

To enable PHP:

1. In Server Admin, click Web for the server you want.
2. Click Settings in the button bar.
3. In the Modules pane, scroll to php4_module in the module list and click Enabled for the module, if necessary.
4. Click Save.

User Content on Websites

Mac OS X client has a Personal Web Sharing feature, where a user may place content in the Sites folder of his or her home directory and have it visible on the web. Mac OS X Server has much broader web service capability, which can include a form of personal web sharing, but there are important differences between Mac OS X client and Mac OS X Server.

Web Service Configuration

By default, on Mac OS X Server:

- Web service ignores any files in the /etc/httpd/users/ folder.
- Workgroup Manager does not make any web service configuration changes.
- Folder listings are not enabled for users.

All folder listings in web service use Apache's FancyIndexing directive, which makes folder listings more readable. In Server Admin, the Sites/Options pane for each site has a Folder Listing checkbox. This setting enables folder listings for a specific virtual host by adding a "+Indexes" flag to Apache's Options directive for that virtual host. If folder listings are not explicitly enabled for each site (virtual host), file indexes are not shown.

The site-specific settings do not apply outside the site; therefore site-specific settings do not apply to users' home directories. If you want users to have folder-indexing capability on their home directories, you need to add suitable directives to Apache's configuration files. For a specific user, you add the following directives inside the <IfModule mod_userdir.c> block in the httpd.conf file:

```
<Directory "/Users/refuser/Sites">
    Options Indexes MultiViews
    AllowOverride None
    Order allow,deny
    Allow from all
</Directory>
```

Default Content

The default content for the user's Sites folder is an index.html file along with a few images. It is important to note that this index.html file has text that describes the Personal Web Sharing feature of Mac OS X client. The user should replace that index.html file with one suited to the content of his or her Sites folder.

Accessing Web Content

Once the home directory is created, the content of the Sites folder within the user's home directory is visible whenever web service is running. If your server is named example.com and the user's short name is refuser, the content of the Sites folder can be accessed at the URL http://example.com/~refuser.

If the user has multiple short names, any of those can also be used after the tilde to access that same content.

If the user has placed a content file named foo.html in his or her Sites folder, that file should be available at http://example.com/~refuser/foo.html.

If the user has placed multiple content files in his or her Sites folder, and cannot modify the index.html to include links to those files, the user may benefit from the automatic folder indexing described previously. If the "Enable folder listing" setting is enabled, an index listing of file names will be visible to browsers at http://example.com/~refuser.

Indexing settings also apply to subfolders placed in the user's Sites folder. If the user adds a content subfolder named Example to the Sites folder, and either an index.html file is present inside the Example folder, or folder indexing is enabled for that user's site, then the folder will be available to browsers at http://example.com/~refuser/Example.

The Module mod_hfs_apple Protects Web Content Against Case Insensitivity in the HFS File System

Mac OS X Server version 10.4 provides case-sensitive coverage for HFS file names. This feature should mean that the extra protection of mod_hfs_apple (discussed below) is not necessary.

The HFS Extended volume format commonly used for Mac OS X Server preserves the case of file names but does not distinguish between a file or folder named "Example" and one named "eXaMpLe." Were it not for mod_hfs_apple, this would be a potential issue when your web content resides on such a volume and you are attempting to restrict access to all or part of your web content using security realms. If you set up a security realm requiring browsers to use a name and a password for read-only access to content within a folder named "Protected," browsers would need to authenticate in order to access the following URLs:

http://example.com/Protected

http://example.com/Protected/secret

http://example.com/Protected/sECreT

But they could bypass it by using something like the following:

http://example.com/PrOtECted

http://example.com/PrOtECted/secret

http://example.com/PrOtECted/sECreT

Fortunately, mod_hfs_apple prevents those types of efforts to bypass the security realm, and this module is enabled by default.

Note: mod_hfs_apple operates on folders; it is NOT intended to prevent access to individual files. A file named "secret" can be accessed as "seCREt". This is correct behavior, and does not allow bypassing security realms.

You can verify that mod_hfs_apple is operating correctly by creating a security realm and attempting to bypass it with a case-variant of the actual URL. You will be denied access and your attempt will be logged in the web service error log with messages similar to the following:

[Wed Jul 31 10:29:16 2002] [error] [client 17.221.41.31] Mis-cased URI: /Library/WebServer/Documents/PrOTecTED/secret, wants: /Library/WebServer/Documents/Protected/

WebMail

4

Enable WebMail for the websites on your server to provide access to basic email operations by means of a web connection.

WebMail adds basic email functions to your website. If your web service hosts more than one website, WebMail can provide access to mail service on any or all of the sites. The mail service looks the same on all sites.

WebMail Basics

The WebMail software is included in Mac OS X Server, but is disabled by default.

The WebMail software is based on SquirrelMail (version 1.4.1), which is a collection of open source scripts run by the Apache server. For more information on SquirrelMail, see the website www.squirrelmail.org.

WebMail Users

If you enable WebMail, a web browser user can:

- Compose messages and send them
- Receive messages
- Forward or reply to received messages
- Maintain a signature that is automatically appended to each sent message
- Create, delete, and rename folders and move messages between folders
- Attach files to outgoing messages
- Retrieve attached files from incoming messages
- Manage a private address book
- Set WebMail Preferences, including the color scheme displayed in the web browser

To use your WebMail service, a user must have an account on your mail server. Therefore, you must have a mail server set up if you want to offer WebMail on your websites.

Users access your website's WebMail page by appending /WebMail to the URL of your site. For example, http://mysite.example.com/WebMail/.

Users log in to WebMail with the name and password they use for logging in to regular mail service. WebMail does not provide its own authentication. For more information on mail service users, see the mail service administration guide.

When users log in to WebMail, their passwords are sent over the Internet in clear text (not encrypted) unless the website is configured to use SSL. For instructions on configuring SSL, see "Enabling SSL" on page 50.

WebMail users can consult the user manual for SquirrelMail at www.squirrelmail.org/wiki/UserManual.

WebMail and Your Mail Server

WebMail relies on your mail server to provide the actual mail service. WebMail merely provides access to the mail service through a web browser. WebMail cannot provide mail service independent of a mail server.

WebMail uses the mail service of your Mac OS X Server by default. You can designate a different mail server if you are comfortable using the Terminal application and UNIX command-line tools. For instructions, see "Configuring WebMail" on page 59.

WebMail Protocols

WebMail uses standard email protocols and requires your mail server to support them. These protocols are:

- Internet Message Access Protocol (IMAP) for retrieving incoming mail
- Simple Mail Transfer Protocol (SMTP) for exchanging mail with other mail servers (sending outgoing mail and receiving incoming mail)

Note: The SquirrelMail configuration script allows setting the IMAP server type. The setting "macosx = Mac OS X Mailserver" refers to the older Apple MailServer in Mac OS X Server version 10.2. In Mac OS X version 10.3 and version 10.4, the correct setting (and the one set by default) is "cyrus = Cyrus IMAP Server."

WebMail does not support retrieving incoming mail via Post Office Protocol (POP). Even if your mail server supports POP, WebMail does not.

Enabling WebMail

You can enable WebMail for the website (or sites) hosted by your web server. Changes take effect when you restart web service.

To enable WebMail for a site:

1. Make sure your mail service is started and configured to provide IMAP and SMTP service.
2. Make sure IMAP mail service is enabled in the user accounts of the users you want to have WebMail access.

   For details on mail settings in user accounts, see the user management guide.
3. In Server Admin, click Web in the list for the server you want.
4. Click Settings in the button bar.
5. In the Sites pane, double-click the site in the list.
6. In the Options pane, select WebMail.
7. Click Save.

Note: When you turn WebMail on, the PHP module is enabled (if it was not already on). If you turn WebMail off, PHP remains on until you turn it off. See "Enabling PHP" on page 52 for details.

Configuring WebMail

After enabling WebMail to provide basic email functions on your website, you can change some settings to integrate WebMail with your site. You can do this by editing the configuration file /etc/squirrelmail/config/config.php, or by using the Terminal application to run an interactive configuration script with root privileges. Either way, you actually change the settings of SquirrelMail, which is open source software that provides WebMail service for the Apache web server of Mac OS X Server.

SquirrelMail, hence WebMail, has several options that you can configure to integrate WebMail with your site. The options and their default settings are as follows:

- **Organization Name** is displayed on the main WebMail page when a user logs in. The default is Mac OS X Server WebMail.
- **Organization Logo** specifies the relative or absolute path to an image file.
- **Organization Title** is displayed as the title of the web browser window while viewing a WebMail page. The default is Mac OS X Server WebMail.
- **Trash Folder** is the name of the IMAP folder where mail service puts messages when the user deletes them. The default is Deleted Messages.
- **Sent Folder** is the name of the IMAP folder where mail service puts messages after sending them. The default is Sent Messages.
- **Draft Folder** is the name of the IMAP folder where mail service puts the user's draft messages. The default is Drafts.

You can configure these and other settings—such as which mail server provides mail service for WebMail—by running an interactive Perl script in a Terminal window, with root privileges. The script operates by reading original values from the config.php file and writing new values back to config.php.

Important: If you use the interactive configuration script to change any SquirrelMail settings, you must also use the script to enter your server's domain name. If you fail to do this, WebMail will be unable to send messages.

The WebMail configuration settings apply to all websites hosted by your web service.

To configure basic WebMail options:

1. In the Terminal application, type the following command and press Return:

   ```
   sudo /etc/squirrelmail/config/conf.pl
   ```

2. Follow the instructions displayed in the Terminal window to change SquirrelMail settings as desired.
3. Change the domain name to your server's real domain name, such as example.com.

   The domain name is the first item on the SquirrelMail script's Server Settings menu.

   The script operates by reading original values from config.php and writing new values back to config.php.

   If you don't enter the server's actual domain name correctly, the interactive script replaces the original value, getenv(SERVER_NAME), with the same value but enclosed in single quotes. The quoted value no longer works as a function call to retrieve the domain name, and as a result WebMail can't send messages.

   WebMail configuration changes do not require restarting web service unless users are logged in to WebMail.

   To further customize the appearance (for example, to provide a specific appearance for each of your websites), you need to know how to write PHP scripts. In addition, you need to become familiar with the SquirrelMail plug-in architecture and write your own SquirrelMail plug-ins.

Working With WebObjects and Web-Related Open Source Applications

5

Become familiar with WebObjects and the open source applications Mac OS X Server uses to administer and deliver web services.

The application server component of Mac OS X Server offers versatile tools that allow you to extend your web server in a variety of ways.

The Apple web development tool WebObjects is represented in the application server. In addition, several open source applications provide essential features of web service. These applications include:

- Apache web server
- JBoss application server
- Tomcat servlet container
- MySQL database

WebObjects

Mac OS X Server includes the WebObjects run-time libraries and an unlimited deployment license, making it the ideal platform for your J2EE-compatible WebObjects applications. You can optionally purchase the WebObjects development tools from the Apple Store (store.apple.com), Apple's retail stores, and authorized Apple resellers.

For more information and documentation on WebObjects, go to www.apple.com/webobjects/ or developer.apple.com/documentation/WebObjects/.

Starting or Stopping WebObjects

In Server Admin:

1. Click WebObjects in the list for the server you want.
2. Click Start Service or Stop Service.

In Terminal:

1. Open the Terminal application.
2. Type one of the following commands to start or stop WebObjects.

```
serveradmin start webobjects
serveradmin stop webobjects
```

Changing the WebObjects Configuration

You use Server Admin to change the WebObjects configuration.

1. In Server Admin click WebObjects in the list for the server you want.
2. Click Settings in the button bar.
3. Specify the wotaskd (WebObjects Task Daemon) port or the Monitor port, as desired, or turn Monitor on.
4. Click Save.

Opening the Java Monitor Application

The Java Monitor helps you configure WebObjects applications.

1. In Server Admin click WebObjects in the list for the server you want.
2. Click Settings in the button bar.
3. Click Turn Monitor on.
4. Click Save.
5. Click Overview in the button bar.
6. Click the Running link for Monitor to open it in the browser.

Note: To get more information about the Monitor, wotaskd, and other WebObjects components, open the Monitor and click the Help tab.

Apache

Apache is the http web server provided with Mac OS X Server. You can use the Server Admin application to manage most server operations, but in some instances you may want to add or change parts of the open source Apache server. In such situations, you need to modify Apache configuration files and change or add modules.

Note: Mac OS X Server contains two versions of the Apache web server—Apache 1.3 and Apache 2.0. Version 1.3 is supported in the Server Admin application; version 2.0 is for evaluation.

Location of Essential Apache Files

Locations of key Apache files are as follows:

- The Apache configuration file for web service is located in the directory /etc/httpd/.
- The site configuration files are located in the directory /etc/httpd/sites.
- The Apache error log, which is very useful for diagnosing problems with the configuration file, is located in the directory /var/log/httpd/ (with a symlink that allows the directory to be viewed as /Library/Logs/WebServer/).
- Temporarily disabled virtual hosts are in the directory /etc/httpd/'sites_disabled/.

Note: All files in /etc/httpd/sites/ are read and processed by Apache when it does a hard or soft (graceful) restart. Each time you save changes, the server does a graceful restart. If you edit a file using a text editor that creates a temporary or backup copy, the server restart may fail because two files with almost identical names are present. To avoid this problem, delete temporary or backup files created by editing files in this folder.

Editing Apache Configuration Files

You can edit Apache configuration files if you need to work with features of the Apache web server that aren't included in Server Admin. To edit configuration files, you should be an experienced Apache administrator and familiar with text-editing tools. Be sure to make a copy of the original configuration file before editing it.

The configuration file httpd.conf handles all directives controlled by the Server Admin application. You can edit this file, as long as you follow the conventions already in place there (as well as the comments in that file). This file also has a directive to include the sites/ directory. In that directory are all of the virtual hosts for that server. The files are named with the unique identifier of the virtual host (for example, 0000_17.221.43.127_80_www.example.com.conf). You disable specific sites by moving them to the sites_disabled directory and then restarting web service. You can also edit site files as long as the conventions in the file are followed.

One hidden file in the sites_disabled folder is named "default_default.conf." This file is used as the template for all new virtual hosts created in Server Admin. An administrator can edit the template file to customize it, taking care to follow the conventions already established in the file.

For more information about Apache and its modules, see "Apache Modules" on page 71.

Starting and Stopping Web Service Using the apachectl Script

The default way to start and stop Apache on Mac OS X Server is to use the web module of Server Admin.

If you want to use the apachectl script to start and stop web service instead of using Server Admin, be aware of the following behaviors:

- The web performance cache is enabled by default. When web service starts, both the main web service process (httpd) and a webperfcache process start. (The webperfcache process serves static content from a memory cache and relays requests to httpd when necessary.) The apachectl script that comes with Mac OS X Server is unaware of webperfcache. So if you have not disabled the performance cache, you also need to use the webperfcachectl script to start and stop webperfcache.

- The apachectl script does not increase the soft process limit beyond the default of 100. Server Admin raises this limit when it starts Apache. If your web server receives a lot of traffic and relies on CGI scripts, web service may fail to run when it reaches the soft process limit.
- The apachectl script does not start Apache automatically when the server restarts.

Because of the issue noted above, if you need to control Apache from a script, the recommended approach is to use the serveradmin command-line tool. To start Apache, and the performance cache if appropriate, and mark /etc/hostconfig to start web service on reboot, issue the following command from a script:

```
serveradmin start web
```

To stop Apache, and to stop the performance cache if appropriate, and to mark /etc/hostconfig not to start web service on reboot, issue the following command from a script:

```
serveradmin stop web
```

Understanding apachectl and the Web Service Soft Process Limit

When Apache is started using the apachectl script, the soft process limit is 100, the default limit.

When you use CGI scripts, this limit may not be high enough. In this case, you can start web service using Server Admin, which sets the soft process limit to 2048. Alternatively, you can type "ulimit -u 2048" before using apachectl.

About Apache Multicast DNS Registration

Apache multicast DNS registration should not be used with the server.

Important: Do not try to turn on Apache multicast DNS (mdns) registration for the server. It does not support virtual hosts, and the server uses virtual hosts.

Using Apache Axis

You can use Apache Axis by writing web applications that use the Axis libraries and then deploy the applications in Tomcat or JBoss. Unlike JBoss and Tomcat, Axis is not usually used as an application server.

Mac OS X Server version 10.4 includes a preinstalled version of Apache Axis (1.1), which operates in conjunction with the preinstalled Tomcat 4.1.x. Apache Axis is an implementation of Simple Object Access Protocol (SOAP). More about SOAP can be found at http://www.w3.org/TR/SOAP/. More about Axis can be found at: http://ws.apache.org/axis/.

The Axis libraries can be found in /System/Library/Axis. By default, Apple installs a sample Axis web application into Tomcat. The web application known as axis can be found in /Library/Tomcat/webapps/axis.

After you enable Tomcat using the Application Server section of Server Admin, you can validate the preinstalled Apache Axis by browsing the following:

http://example.com:9006/axis/

Replace "example.com" in the URL above with your host name. Note the nonstandard Tomcat port.

The first time you exercise the preinstalled Axis by browsing http://example.com:9006/axis/ and selecting the link entitled "Validate the local installation's configuration," you should expect to see the following error messages:

- Warning: could not find class javax.mail.internet.MimeMessage from file mail.jar Attachments will not work.

  See http://java.sun.com/products/javamail/

- Warning: could not find class org.apache.xml.security.Init from file xmlsec.jar XML Security is not supported

  See http://xml.apache.org/security/

Follow the instructions that accompany the warning messages if you require those optional components.

Consult the Axis User's Guide on the Apache Axis website to learn more about using Axis in your own web applications.

Experimenting With Apache 2

Version 10.4 of Mac OS X Server includes Apache 2 for evaluation purposes in addition to the operational version of Apache 1.3. By default, Apache 2 is disabled, and all Server Admin operations work correctly with Apache 1.3.

If you want to experiment with Apache 2, note the following:

- It is installed in a separate location in the file system: /opt/apache2.
- It is not connected to Server Admin.
- It serves webpages from /opt/apache2/htdocs.
- Its configuration is in /opt/apache2/conf/httpd.conf. Apple modified this file by configuring it to run the httpd processes as user and group www. If you enable WebDAV with Apache 2, note that although your WebDAV clients using version 10.1 of Mac OS X or Mac OS X Server will be able to mount Apache2 WebDAV volumes, they will not have write access; they will have read-only access. WebDAV clients using version 10.2 or later will not have this problem.
- It is controlled by its own version of the apachectl script, so to start it, type

  ```
  sudo /opt/apache2/bin/apachectl start
  ```

- Although it's possible to run both versions of Apache, you should be cautious when doing so. Make sure the two versions do not attempt to listen on the same port. Apache 2 is configured by default to listen on port 8080, so it is possible have both Apache 1.3 and Apache 2 running at the same time.

JBoss

JBoss (version 3.2.3) is an open source application server designed for J2EE applications; it runs on Java 1.4.2. JBoss is a widely used, full-featured Java application server. It provides a full Java 2 Platform, Enterprise Edition (J2EE) technology stack with features such as:

- An Enterprise Java Bean (EJB) container
- Java Management Extensions (JMX)
- Java Connector Architecture (JCA)

By default, JBoss uses Tomcat as its web container, but you can use other web containers, such as Jetty, if you wish.

You can use the Application Server section of Server Admin and the command-line tools in the Terminal application to manage JBoss. Server Admin integrates with the launchd process to ensure continuous availability of JBoss once JBoss has been started. For more information about the launchd process, consult the man page by opening Terminal and typing the following command.

```
man launchd
```

You can use Server Admin to start one of the available JBoss configurations, stop JBoss, and view the log files.

Two web-based tools for working with JBoss are also included with Mac OS X Server, one for management and configuration of the JBoss server and one for deployment of existing applications. Both tools are located in /Library/JBoss/Applications.

For detailed information about JBoss, J2EE, and the tools, see these guides:

- Java application server administration guide, which explains how to deploy and manage J2EE applications using JBoss in Mac OS X Server
- Java enterprise applications guide, which explains how to develop J2EE applications

Both guides are available from Apple developer publications.

Additional information about these Java technologies is available online.

- For JBoss, see www.jboss.org/.
- For J2EE, see java.sun.com/j2ee/.

To open the JBoss management tool:

- In Server Admin, click Application Server in the list for the server you want.

To start or stop JBoss using Server Admin:

You work with JBoss in Server Admin.

1. In Server Admin, click Application Server in the list for the server you want.
2. Click Settings in the button bar.
3. Select one of the JBoss options. (Do not select Tomcat Only.)

4 Click Start Service or Stop Service.

To start or stop JBoss using Terminal:

1 Open the Terminal application.

2 Type the following commands.

```
cd /Library/JBoss/3.2/bin
./run.sh
```

JBoss is preconfigured to use a local configuration.

With JBoss turned on, you can use the management tool to configure your server.

For details of configuring JBoss and using the command-line tools for it, see the Java application server administration guide, which explains how to deploy and manage J2EE applications using JBoss in Mac OS X Server. This guide is available from Apple developer publications.

To change the JBoss configuration in use:

1 In Server Admin, click Application Server in the list for the server you want.

2 Click Settings in the button bar.

3 Click Use Local Configuration and choose a configuration from the pop-up menu.

To manage JBoss:

1 In Server Admin, click Application Server.

2 Click Settings in the button bar.

3 Click Manage JBoss.

Note: The JBoss management tool must already be running. You can use the Terminal application to set it as a startup item.

4 Make the adjustments you want in the management console.

Backing Up and Restoring JBoss Configurations

You use the Application Server section of Server Admin to back up and restore JBoss configurations.

To back up or restore a JBoss configuration:

1 In Server Admin, click Application Server in the list for the server you want.

2 Click Settings in the button bar at the bottom of the window.

3 Click Backup at the top of the window.

4 Click either Backup or Restore and navigate to the location where you want to store or have stored configurations.

The current configuration is backed up.

Tomcat

Tomcat is the open source servlet container that is used as the official Reference Implementation for the Java Servlet and JavaServer Pages technologies. The Java Servlet and JavaServer Pages specifications are developed by Sun under the Java Community Process.

The current production series is the Tomcat 4.1.x series and it implements Java Servlet 2.3 and JavaServer Pages 1.2 specifications. More information is available from the following sources:

- For Java Servlet specifications, see java.sun.com/products/servlets
- For Java ServerPages specifications, see java.sun.com/products/jsp

In Mac OS X Server 10.4, you use the Application Server section of Server Admin to manage Tomcat. Once Tomcat is started its life cycle is managed by Server Admin, which ensures that Tomcat starts up automatically after a power failure or after the server shuts down for any reason.

For more information about Tomcat and documentation for this software, see http://jakarta.apache.org/tomcat/

For information about Java Servlets that you can use on your server, see:

- http://java.sun.com/products/servlet/
- http://java.sun.com/products/jsp/

If you want to use Tomcat, you must activate it. You can use Server Admin or the command-line tool to start Tomcat.

Note: The weblog application, Blojsom, uses a separate instance of Tomcat. Therefore, you can use Tomcat without interfering with weblogs.

Setting Tomcat as the Application Container

To start Tomcat using Server Admin:

1. In Server Admin, click Application Server in the list for the server you want.
2. Click Settings in the button bar.
3. Click Tomcat Only.
4. Click Start Service.

To start Tomcat using Terminal:

1. Open the Terminal application.
2. Type the following commands:

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

To verify that Tomcat is running, use a browser to access port 9006 of your website by entering the URL for your site followed by :9006. If Tomcat is running, this URL will display the Tomcat home page.

MySQL

MySQL provides a relational database management solution for your web server. With this open source software, you can link data in different tables or databases and provide the information on your website.

The MySQL Manager application simplifies setting up the MySQL database on Mac OS X Server. You can use MySQL Manager to initialize the MySQL database, and to start and stop the MySQL service. MySQL Manager is located in /Applications/Server. You use it to install required files (the first time you use the application), turn on MySQL service, enter a root password, and allow network connections. The first time you run MySQL manager, you may need to unlock it and supply an administrator password before making changes.

MySQL is preinstalled on Mac OS X Server, with its various files already in the appropriate locations. At some point you may wish to upgrade to a newer version of MySQL. You can install the new version in /usr/local/mysql, but MySQL Manager will not be aware of the new version of MySQL and will continue to control the pre-installed version. If you do install a newer version of MySQL, use MySQL Manager to stop the preinstalled version, then start the newer version via the config file.

Installing MySQL

Mac OS X Server version 10.4 includes the latest MySQL, version 4.1. Since it's preinstalled, you won't find it in /usr/local/mysql. Instead, its elements are distributed in the file system according to standard UNIX file layout, with executables in /usr/sbin and /usr/bin, man pages in /usr/share/man, and other parts in /usr/share/mysql. When installed, the MySQL database resides in /var/mysql.

At some point a newer version of MySQL will be posted to http://www.mysql.com. At that time you may consider downloading the source and building it yourself (if you have the developer packages installed) or downloading the appropriate binary distribution and installing it yourself, following the instructions posted on that website. By default, such installations reside in /usr/local/mysql/. So if you install your own version of MySQL, you'll have two versions of MySQL present on your system. This should do no harm as long as you don't try to run both the old one and the new one. Just be sure to prefix any commands intended for the new version with the full path (starting with /usr/local/mysql), or make sure your shell's path variable is set to search in your local directory first.

Note that the MySQL Manager application works only with the preinstalled version of MySQL; it does not work with MySQL installed elsewhere. The paths to the various preinstalled components of MySQL are stored in the following plist file:

/Applications/Server/MySQL Manager.app/Contents/Resources/tool_strings.

If You Are Updating from Mac OS X Server 10.2 or Earlier and Use MySQL

Previous versions of the server contained MySQL 3.23.x; the version now installed is 4.1, which is the latest production version. This version is the one recommended by mysql.com.

Your MySQL 3.23.x databases should work with the new version of MySQL, but it's a good idea to back them up before updating.

When using MySQL 4.1, there are several commands you can use with your old databases to remove dependency on the ISAM table format, which has been deprecated over time.

- Use mysql_fix_privilege_tables to enable new security privilege features.
- Use mysql_convert_table_format (if all existing tables are ISAM or MyISAM) or use ALTER TABLE table_name TYPE=MyISAM on all ISAM tables to get away from the degraded ISAM table format.

Refer to the instructions provided on the MySQL website at www.mysql.com/doc/en/Upgrading-from-3.23.html before using these commands.

For more information about MySQL, see www.mysql.com.

Installing and Viewing Web Modules

6

Become familiar with the modules that provide key features and controls for web service.

The Apache web server includes a series of modules that control the server's operation. In addition, Mac OS X Server provides some modules with specialized functions for the Macintosh.

Apache Modules

Modules "plug in" to the Apache web server software and add functionality to your website. Apache comes with some standard modules, and you can purchase modules from software vendors or download them from the Internet. You can find information about available Apache modules at the website www.apache.org/docs/mod.

To work with Apache modules:

- To view a list of web modules installed on your server, in Server Admin click Web in the list for the server you want, choose Settings in the button bar, and click Modules.
- To enable a module, select the Enabled box beside its name, and click Save.
- To install a module, follow the instructions that came with the module software. The web server loads modules from the directory /usr/libexec/httpd/.

Macintosh-Specific Modules

Web service in Mac OS X Server installs some modules specific to the Macintosh. These modules are described in this section.

mod_macbinary_apple

This module packages files in the MacBinary format, which allows Macintosh files to be downloaded directly from your website. A user can download a MacBinary file using a regular web browser by adding ".bin" to the URL used to access the file.

mod_spotlight_apple

This module lets Apache perform relevance-ranked searches of the website using Spotlight. Once you index your site, you can provide a search field for users to search your website.

Clients must add .spotlight to your website's URL to access a page that allows them to search your site. For example, http://www.example.com/.spotlight.

mod_auth_apple

This module allows a website to authenticate users by looking for them in directory service domains within the server's search policy. When authentication is enabled, website visitors are prompted for a user name and password before they can access information on the site.

mod_hfs_apple

This module requires users to enter URLs for HFS volumes using the correct case (lowercase or uppercase). This module adds security for case-insensitive volumes. If a restriction exists for a volume, users receive a message that the URL is not found.

mod_digest_apple

This module enables digest authentication for a WebDAV realm.

mod_bonjour

The mod_bonjour module allows administrators to control how websites are registered with multicast DNS.

Open Source Modules

Mac OS X Server includes these popular open source modules: Tomcat, PHP: Hypertext Preprocessor, and mod_perl.

Tomcat

The Tomcat module, which uses Java-like scripting, is the official reference implementation for two complementary technologies developed under the Java Community Process. For more information about Tomcat, see "Tomcat" on page 68.

If you want to use Tomcat, you must activate it first. You use the Application Server section of Server Admin to start Tomcat. See "Tomcat" on page 68 for instructions.

PHP: Hypertext Preprocessor

PHP lets you handle dynamic web content by using a server side HTML-embedded scripting language resembling C. Web developers embed PHP code within HTML code, allowing programmers to integrate dynamic logic directly into an HTML script rather than write a program that generates HTML.

PHP provides CGI capability and supports a wide range of databases. Unlike client-side JavaScript, PHP code is executed on the server. PHP is also used to implement WebMail on Mac OS X Server. For more information about this module, see www.php.net.

mod_perl

This module integrates the complete Perl interpreter into the web server, letting existing Perl CGI scripts run without modification. This integration means that the scripts run faster and consume fewer system resources. For more information about this module, see perl.apache.org.

Solving Problems

7

If you experience a problem with web service or one of its components, check the tips and strategies in this chapter.

From time to time you may encounter a problem when setting up or managing web services. Some of the situations that may cause a problem for administering web service or for client connections are outlined here.

Users Can't Connect to a Website on Your Server

Try these strategies to uncover the problem:

- Make sure that web service is turned on and the site is enabled.
- Check the Web Service Overview window to verify that the server is running.
- Check the Apache access and error logs. (If you are not sure what the messages mean, you'll find explanations on the Apache website at www.apache.org.)
- Make sure users are entering the correct URL to connect to the web server.
- Make sure that the correct folder is selected as the default web folder. Make sure that the correct HTML file is selected as the default document page.
- If your website is restricted to specific users, make sure those users have access privileges to your website.
- Verify that users' computers are configured correctly for TCP/IP. If the TCP/IP settings appear correct, use a "pinging" utility that allows you to check network connections.
- Verify that the problem is not a DNS problem. Try to connect with the IP address of the server instead of its DNS name.
- Make sure your DNS server's entry for the website's IP address and domain name are correct.

A Web Module Is Not Working as Expected

Try several strategies to uncover the problem.

- Check the error log in Server Admin for information about why the module might not be working correctly.
- If the module came with your web server, check the Apache documentation for that module and make sure the module is intended to work the way you expected.
- If you installed the module, check the documentation that came with the web module to make sure it is installed correctly and is compatible with your server software.

For more information on supported Apache modules for Mac OS X Server, see Chapter 6, "Installing and Viewing Web Modules," on page 71 and the Apache website at www.apache.org/docs/mod/.

A CGI Will Not Run

Try this strategy to uncover the problem.

- Check the CGI's file permissions to make sure the CGI is executable by www. If not, the CGI won't run on your server even if you enable CGI execution in Server Admin.

The Server Is Not Working Correctly or Performance Is Slow

Try this strategy to uncover the problem.

- Turn off the performance cache if it is enabled. See "Improving Performance of Static Websites (Performance Cache)" on page 37 for more about the cache.

Glossary

Apache An open source HTTP server integrated into Mac OS X Server. You can find detailed information about Apache at www.apache.org.

application server Software that runs and manages other applications, usually web applications, that are accessed using a web browser. The managed applications reside on the same computer where the application server runs.

blog See **weblog**.

Blojsom The open-source project on which Weblog service is based.

certificate Sometimes called an "identity certificate" or "public key certificate." A file in a specific format (Mac OS X Server uses the x.509 format) that contains the public key half of a public-private keypair, the user's identity information such as name and contact information, and the digital signature or either a *Certificate Authority* (CA) or the key user.

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.

Common Gateway Interface See **CGI**.

everyone Any user who can log in to a file server: a registered user or guest, an anonymous FTP user, or a website visitor.

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.

Internet Protocol See **IP**.

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 address A unique numeric address that identifies a computer on the Internet.

JavaScript A scripting language used to add interactivity to webpages.

Jboss A full-featured Java application server that provides support for Java 2 Platform, Enterprise Edition (J2EE) applications.

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.

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" (e.g, 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.

multicast DNS A protocol developed by Apple for automatic discovery of computers, devices, and services on IP networks. This proposed Internet standard protocol is sometimes referred to as "ZeroConf." 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**.

MySQL An open source relational database management tool frequently used by web servers.

open source A term for the cooperative development of software by the Internet community. The basic principle is to involve as many people as possible in writing and debugging code by publishing the source code and encouraging the formation of a large community of developers who will submit modifications and enhancements.

PHP PHP Hypertext Preprocessor (originally Personal Home Page). A scripting language embedded in HTML that's used to create dynamic webpages.

port A sort of virtual mail slot. A server uses port numbers to determine which application should receive data packets. Firewalls use port numbers to determine whether data packets are allowed to traverse a local network. "Port" usually refers to either a TCP or UDP port.

protocol A set of rules that determines how data is sent back and forth between two applications.

proxy server A server that sits between a client application, such as a web browser, and a real server. The proxy server intercepts all requests to the real server to see if it can fulfill the requests itself. If not, it forwards the request to the real server.

realm See **WebDAV realm**, **Kerberos realm**.

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).

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.

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**.

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.

weblog A webpage that hosts chronologically ordered entries. It functions as an electronic journal or newsletter. Weblog service lets you create weblogs that are owned by individual users or by all members of a group.

Index