




# Mac OS X Server Web Technologies Administration

For Version 10.3 or Later



 Apple Computer, Inc.  
© 2003 Apple Computer, Inc. All rights reserved.

The owner or authorized user of a valid copy of Mac OS X Server software may reproduce this publication for the purpose of learning to use such software. No part of this publication may be reproduced or transmitted for commercial purposes, such as selling copies of this publication or for providing paid for support services.

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

Use of the "keyboard" Apple logo (Option-Shift-K) for commercial purposes without the prior written consent of Apple may constitute trademark infringement and unfair competition in violation of federal and state laws.

Apple, the Apple logo, Mac, Mac OS, Macintosh, and Sherlock are trademarks of Apple Computer, Inc., registered in the U.S. and other countries.

Adobe and PostScript are trademarks of Adobe Systems Incorporated.

Java and all Java-based trademarks and logos are trademarks or registered trademarks of Sun Microsystems, Inc. in the U.S. and other countries.

Netscape Navigator is a trademark of Netscape Communications Corporation.

UNIX is a registered trademark in the United States and other countries, licensed exclusively through X/Open Company, Ltd.

034-2350/09-20-03

# Contents

## Chapter 1

- 7 **Web Technologies Overview**
- 8 Key Web Components
- 8 Apache Web Server
- 8 WebDAV
- 8 CGI Support
- 8 SSL Support
- 8 Dynamic Content With Server-Side Includes (SSI)
- 9 Front-End Cache
- 9 Before You Begin
- 9 Configuring Your Web Server
- 9 Providing Secure Transactions
- 9 Setting Up Websites
- 10 Hosting More Than One Website
- 10 Understanding WebDAV
- 11 Understanding Multipurpose Internet Mail Extension

## Chapter 2

- 13 **Managing Web Technologies**
- 13 Setting Up Your Web Server for the First Time
- 15 Using Server Admin to Manage Your Web Server
- 15 Starting or Stopping Web Service
- 16 Modifying MIME Mappings and Content Handlers
- 17 Managing Connections
- 17 Setting Simultaneous Connections for the Web Server
- 18 Setting Persistent Connections for the Web Server
- 18 Setting a Connection Timeout Interval
- 19 Setting Up Proxy Caching
- 20 Blocking Websites From Your Web Server Cache
- 20 Using Secure Sockets Layer (SSL)
- 20 About SSL
- 21 Using WebDAV
- 21 Using Tomcat
- 22 Viewing Web Service Status
- 22 Web Service Overview

- 22 Web Service Modules in Use
- 22 Viewing Logs of Web Service Activity

### Chapter 3

- 23 **Managing Websites**
- 23 Using Server Admin to Manage Websites
- 23 Setting Up the Documents Folder for a Website
- 24 Enabling a Website on a Server
- 25 Changing the Default Web Folder for a Site
- 25 Setting the Default Page for a Website
- 26 Changing the Access Port for a Website
- 26 Improving Performance of Static Websites (Performance Cache)
- 26 Understanding the Effect of Using a Web Service Performance Cache
- 27 Enabling Access and Error Logs for a Website
- 28 Setting Up Directory Listing for a Website
- 29 Creating Indexes for Searching Website Content
- 29 Connecting to Your Website
- 30 Enabling WebDAV on Websites
- 31 Setting Access for WebDAV-Enabled Sites
- 32 WebDAV and Web Content File and Folder Permissions
- 32 Enabling Integrated WebDAV Digest Authentication
- 32 WebDAV and Web Performance Cache Conflict
- 33 Enabling a Common Gateway Interface (CGI) Script
- 33 Enabling Server Side Includes (SSI)
- 34 Viewing Website Settings
- 34 Setting Server Responses to MIME Types and Content Handlers
- 35 Enabling SSL
- 36 Setting Up the SSL Log for a Website
- 36 Enabling PHP
- 36 User Content on Websites
- 37 Web Service Configuration
- 37 Default Content
- 37 Accessing Web Content

### Chapter 4

- 41 **WebMail**
- 41 WebMail Basics
- 41 WebMail Users
- 42 WebMail and Your Mail Server
- 42 WebMail Protocols
- 42 Enabling WebMail
- 43 Configuring WebMail

### Chapter 5

- 45 **Secure Sockets Layer (SSL)**
- 45 Setting Up SSL

	45	Generating a Certificate Signing Request (CSR) for Your Server
	46	Obtaining a Website Certificate
	47	Installing the Certificate on Your Server
	47	Enabling SSL for the Site
	48	Web Server SSL Password Not Accepted When Manually Entered
<b>Chapter 6</b>	<b>49</b>	<b>Working With Open-Source Applications</b>
	49	Apache
	50	Location of Essential Apache Files
	50	Editing Apache Configuration Files
	51	Starting and Stopping Web Service Using the apachectl Script
	51	Enabling Apache Rendezvous Registration
	55	Experimenting With Apache 2
	56	JBoss
	58	Backing Up and Restoring JBoss Configurations
	58	Tomcat
	59	MySQL
	60	Installing MySQL
<b>Chapter 7</b>	<b>61</b>	<b>Installing and Viewing Web Modules</b>
	61	Apache Modules
	61	Macintosh-Specific Modules
	61	mod_macbinary_apple
	62	mod_sherlock_apple
	62	mod_auth_apple
	62	mod_hfs_apple
	62	mod_digest_apple
	62	mod_rendezvous_apple
	62	Open-Source Modules
	62	Tomcat
	63	PHP: Hypertext Preprocessor
	63	mod_perl
<b>Chapter 8</b>	<b>65</b>	<b>Solving Problems</b>
	65	Users Can't Connect to a Website on Your Server
	66	A Web Module Is Not Working as Expected
	66	A CGI Will Not Run
<b>Chapter 9</b>	<b>67</b>	<b>Where to Find More Information</b>
<b>Glossary</b>	<b>69</b>	
<b>Index</b>	<b>73</b>	



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

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 61.

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 WebDAV to control access to all or part of a website’s content.

### 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 only revise information in one place and the SSI command displays that revised information on many pages.



## 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.

## 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.

For instructions on how to set up secure transactions, see Chapter 5, "Secure Sockets Layer (SSL)," on page 45.

## 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 23 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.

## 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 WebDAV-Enabled Sites" on page 31.

### 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. If you're using WebDAV, the www user and www group both need write access to the files and folders in the websites. In addition, the www user and group need write access to the /var/run/davlocks directory.

### Understanding WebDAV Security

WebDAV 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. This has 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.

If you are concerned about website security, you may choose to leave WebDAV disabled and use Apple file service or FTP service to modify the contents of a website instead.

## 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.

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.

## 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 "Using Server Admin to Manage Your Web Server" on page 15 and Chapter 3, "Managing Websites," on page 23.

### Setting Up Your Web Server for the First Time

#### **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 `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 25 for instructions on specifying default index file names.

For more information about all website settings, see Chapter 3, "Managing Websites," on page 23.

### 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

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 6, “Working With Open-Source Applications,” on page 49.

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 “Using Server Admin to Manage Your Web Server” on page 15 and Chapter 3, “Managing Websites,” on page 23.

### 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 8, "Solving Problems," on page 65.

## 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.

## 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 11.

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.
- 5 Click Save.

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.

## 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.

Web service restarts.

## 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.

### 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, enter a number in the “Maximum persistent connections” field and type a new number.

The range for maximum persistent connections is 1 to 2048. The default setting of 500 provides better performance.

- 4 Click Save.

Web service restarts.

## 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 “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.

Web service restarts.

## 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.

### 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 Set the maximum cache size.

When the cache reaches this size, the oldest files are deleted from the cache folder.

- 5 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.

- 6 Click Save.

Web service restarts.

**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, type the URL of the website you want to block in the Add field, and click Add.
  - Drag a list of websites (text file in comma-separated or tab-separated format) to the “Blocked hosts” field.
- 5 Click Save.

Web service restarts.

## 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

SSL was developed by Netscape and uses authentication and encryption technology from RAS Data Security, Inc. For detailed information about the SSL protocol, see:

- [www.netscape.com/eng/ssl3/draft302.txt](http://www.netscape.com/eng/ssl3/draft302.txt)
- <http://developer.netscape.com/misc/developer/conference/proceedings/cs2/index.html>

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](http://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 30 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 will ensure that the Tomcat module starts up after a power failure or after the server shuts down for any reason.

You can use Server Admin or the command-line tool to enable the Tomcat module. See “Tomcat” on page 58 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 34 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 you 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 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.

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 27 for more information.

## 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 25 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 In the General pane, type the fully qualified DNS name of your website in the Name field.
- 5 Enter the IP address and port number (any number up to 8999) for the site.  
The default port number is 80. 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 10 for more information.
- 6 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.
- 7 Enter the file name of your default document (the first page users see when they access your site).
- 8 Make any other settings you want for this site, then click Save.
- 9 Click the back button at the top right side of the editing window.
- 10 Click the Enabled box next to the site name in the Sites pane.
- 11 Click Save.  
Web service restarts.



## Changing the Default Web Folder for a Site

A site's default web folder is used as the root for the site. 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.
- 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 (if accessing this server remotely, file service must be turned on to do this; see the file services administration guide for more information).
- 7 Click Save.

Web service restarts.

## 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 displays the one highest in the list that is 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.

Web service restarts.

**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.

**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.

Web service restarts.

## Improving Performance of Static Websites (Performance Cache)

If your websites contain static HTML files, and you expect high usage of the pages, you can enable the performance cache to improve server performance. 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 most 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.

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.

**To enable or disable the performance cache for your web server:**

- 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.

Web service restarts.

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

## 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.
- 9 Enter archive, location, and level choices for the error log as desired.
- 10 Click Save.

Web service restarts.

## Understanding the New Web Service access\_log Format

In version 10.3 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.

## 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.
- Web service restarts.

## Creating Indexes for Searching Website Content

Version 10.3 of Mac OS X Server continues to support the `mod_sherlock_apple` Apache module, which allows web browsers to search the content of your website. As in previous versions of the server, you must produce a content index before content searching is possible.

Content indexes in earlier server versions had to be created in Sherlock. Now, you can create content indexes using the Finder. Select the folder containing the files you want to index, then choose `File > GetInfo`. Click `Content Index`, then click `Index Now`. (You can remove an index by using the `Delete Index` button in the Info window.)

In addition, there are new constraints that restrict the creation of index files. To create an index, you must be the owner of the folder and must own any files in that folder that are to be indexed. In the case of content in the `/Library/WebServer/Documents` folder, the folder and all the files within it are owned by `root`. Even though the folder and files are writable by members of the `admin` group, you must still be logged in as `root` to create a content index.

Creating an index remotely or on a headless server is done using a command-line tool named `indexfolder`. See the man pages for usage details. The operation of `indexfolder` is affected by the login window. If nobody is logged in at the login window, the tool must be run as `root`. If an administrator is logged in at the login window, the tool must be run as that administrator. Otherwise, the tool will fail with messages similar to these:

```
kCGErrorIllegalArgument : initCGDisplayState: cannot map display interlocks.  
kCGErrorIllegalArgument : CGSNewConnection cannot get connection port.
```

Whether done from the Finder or the `indexfolder` tool, content indexing creates a folder named `“.FBCIndex”` either in the folder to be indexed or in one of its parent folders.

## 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.
- 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.

## 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.  
The default realm name is “untitled.”
- 7 If you want digest authentication for the realm, choose Digest from the Authorization pop-up menu.  
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.  
If you are administering a remote server, file service must be running on the remote server to use the Browse button.
- 9 Click Save.

Web service restarts.

**Note:** If you have turned off the WebDAV modules 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 61 for more about enabling modules.

## Setting Access for WebDAV-Enabled Sites

You create realms to provide security for websites. *Realms* are locations within a site that users can view or make changes to when WebDAV is enabled. When you define a realm, you can assign browsing and authoring privileges to users of the realm.

### To add users and groups to 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 30.
- 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:

**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.

**Browse** and **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.
- 10 Click Save.

Web service restarts.

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

## 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 WebDAV-Enabled Sites” on page 31 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 26 for more information about the performance cache.



## 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.

Web service restarts.

## 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.

**Note:** Enabling SSI requires making changes to UNIX configuration files in the Terminal application. To enable SSI, you must be comfortable with typing UNIX commands and using a UNIX text editor.

### To enable SSI:

- 1 In the Terminal application, use the sudo command with a text editor to edit as the super user (root):
- 2 Add the following line to each virtual host for which you want SSI enabled:

```
Options Includes
```

Each site is in a separate file in /etc/httpd/sites/.

To enable SSI for all virtual hosts, add the line outside any virtual host block.

- 3 In Server Admin for the server you want, click Settings in the button bar.
- 4 In the Sites pane, double-click one of the virtual host sites.

- 5 In the General pane, add `index.shtml` to the set of default index files for that site. Repeat this procedure for each virtual host site that uses SSI. (See “Setting the Default Page for a Website” on page 25 for more information.)

By default, the `/etc/httpd/httpd.conf` file maintained by Server Admin contains the following two lines:

```
AddHandler server-parsed shtml
AddType text/html shtml
```

You can add MIME types in Server Admin from the MIME Types pane.

The changes take effect when you restart web service.

## 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 11 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.  
Web service restarts.

## Enabling SSL

Before you can enable Secure Sockets Layer (SSL) protection for a website, you have to obtain the proper certificates. For more information, see “Secure Sockets Layer (SSL)” on page 45. When you have obtained a certificate, you can set up SSL for a site.

**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.
- 5 Type a password in the Pass Phrase field.
- 6 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.
- 7 Type the location of the location of each certificate file in the appropriate field (if necessary), or use the Browse button to choose the location.
- 8 Click the Edit button for the Certificate File, Key File, and CA File fields and paste the contents of the appropriate certificate or key in the text field for each. Click OK each time you paste text.
- 9 Click Save.
- 10 Click Stop Service, wait a moment, and then click Start Service.

## 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.

Web service restarts.

## 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 “Installing and Viewing Web Modules” on page 61 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.

Web service restarts.

## 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 10.3 has a new feature that provides case-sensitive coverage for HFS file names. This new 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.

Because of the warning message that appears in the web service error log about `mod_hfs_apple`, there have been questions about the function of `mod_hfs_apple`. The warning messages do not indicate a problem with the correct function of `mod_hfs_apple`.

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/
```





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](http://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 35.

WebMail users can consult the user manual for SquirrelMail at [www.squirrelmail.org/wiki/UserManual](http://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 43.

## 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)

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.

Web service restarts.

## 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.

## Use Secure Sockets Layer to provide secure transactions and encrypted communication to users of the websites on your server.

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.

### Setting Up SSL

When you generate a certificate signing request (CSR), the certificate authority sends you a certificate that you install on your server. They may also send you a CA certificate (ca.crt). Installing this file is optional. Normally, CA certificates reside in client applications such as Internet Explorer and allow those applications to verify that the server certificate originated from the right authority. However, CA certificates expire or evolve, so some client applications may not be up to date.

### Generating a Certificate Signing Request (CSR) for Your Server

The CSR is a file that provides information needed to set up your server certificate.

#### To generate a CSR for your server:

- 1 Log in to your server using the root password and open the Terminal application.
- 2 At the prompt, type these commands and press Return at the end of each one:  

```
cd  
dd if=/dev/random of=rand.dat bs=1m count=1  
openssl genrsa -rand rand.dat -des 1024 > key.pem
```
- 3 At the next prompt, type a passphrase, then press Return.

The passphrase you create unlocks the server's certificate key. You will use this passphrase when you enable SSL on your web server.

- 4 If it doesn't already exist on your server, create a directory at the location `/etc/httpd/ssl.key`.

Make a copy of the `key.pem` file (created in step 2) and rename it `server.key`. Then copy `server.key` to the `ssl.key` directory.

- 5 At the prompt, type the following command and press Return:

```
openssl req -new -key key.pem -out csr.pem
```

This generates a file named `csr.pem` in your home directory.

- 6 When prompted, enter the following information:
  - **Country:** The country in which your organization is located.
  - **State:** The full name of your state.
  - **Locality:** The city in which your organization is located.
  - **Organizational name:** The organization to which your domain name is registered.
  - **Organizational unit:** Usually something similar to a department name.
  - **Common name of your web server:** The DNS name, such as `server.apple.com`.
  - **Email address:** The email address to which you want the certificate sent.

The file `csr.pem` is generated from the information you provided.

- 7 At the prompt, type the following, then press Return:

```
cat csr.pem
```

The `cat` command lists the contents of the file you created in step 5 (`csr.pem`). You should see the phrase "Begin Certificate Request" followed by a cryptic message. The message ends with the phrase "End Certificate Request." This is your certificate signing request (CSR).

## Obtaining a Website Certificate

You must purchase a certificate for each website from an issuing authority.

Keep these important points in mind when purchasing your certificate:

- You must provide an InterNIC-registered domain name that's registered to your organization.
- If you are prompted to choose a software vendor, choose Apache Freeware with SSLeay.
- You have already generated a CSR, so when prompted, open your CSR file using a text editor. Then copy and paste the contents of the CSR file into the appropriate text field on the issuing authority's website.
- You can have an SSL certificate for each IP address on your server. Because certificates are expensive and must be renewed each year, you may want to purchase a certificate for one host name and use the URL with host name followed by domain name to avoid having to purchase multiple certificates. For example, if your domain name is `mywidgets.com`, you could purchase a certificate for the host name "buy" and your customers would connect to the URL `https://buy.mywidgets.com`.

- The default certificate format for SSLeay/OpenSSL is PEM, which actually is Base64 encoded DER with header and footer line. For more about the certificate format, see [www.modssl.org](http://www.modssl.org).

After you've completed the process, you'll receive an email message that contains a Secure Server ID. This is your server certificate. When you receive the certificate, save it to your web server's hard disk as a file named `server.crt`.

**Important:** Be sure to make a copy of the certificate message or file.

## Installing the Certificate on Your Server

You can use Server Admin or the command-line tool to specify the certificates for a site. For instructions on using Server Admin for this purpose, see "Enabling SSL" on page 35.

### To install an SSL certificate using the command-line tool in the Terminal application:

- 1 Log in to your server as the administrator or super user (also known as root).
- 2 If it doesn't already exist on your server, create a directory with this name:  
`/etc/httpd/ssl.crt`
- 3 Copy `server.crt` (the file that contains your Secure Server ID) to the `ssl.crt` directory.

## Enabling SSL for the 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 where you plan to use the certificate.
- 4 In the Security pane, select Enable Secure Socket Layer.
- 5 Type the password from your CSR in the Pass Phrase field.
- 6 Set the location of the log file that will record SSL transactions.
- 7 Click the Edit button and paste the text from your certificate file (the certificate you obtained from the issuing authority) in the Certificate File field.
- 8 Click the Edit button and paste the text from your key file (the file `key.pem`, which you set up earlier) in the Key File field.
- 9 Click the Edit button and paste the text from the `ca.crt` file in the CA File field. (This is an optional file that you may have received from the certificate authority.)
- 10 Click Save.
- 11 Stop and then start web service.

## Web Server SSL Password Not Accepted When Manually Entered

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.



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

Several open-source applications provide essential features of web service. These applications include:

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

## 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.

## Location of Essential Apache Files

Apache configuration files and locations have been simplified in Mac OS X Server 10.3. Locations of key 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, `10.201.42.7410_80_17.221.43.127_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 61.

## 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.

### 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.

### Enabling Apache Rendezvous Registration

Starting with version 10.2.4 of Mac OS X and Mac OS X Server, the preinstalled Apache 1.3 web service has the capability to register sites with Rendezvous. This feature, which allows Rendezvous-enabled browsers such as Safari to find sites by name, is implemented using a new Apache module, mod\_rendezvous\_apple. This module is different from the mod\_rendezvous available from a third party. (Apache Rendezvous is not supported on the preinstalled Apache 2 web service.)

The module mod-rendezvous\_apple allows administrators to control how websites are registered with Rendezvous. Mod\_rendezvous\_apple is disabled by default on Mac OS X Server.

### To enable `mod_rendezvous_apple` on Mac OS X Server:

- To enable the module, use the Modules pane in Server Admin.

### To set up `mod_rendezvous_apple` on Mac OS X Server:

- To cause additional logging, which may be helpful if you discover a problem, find the `LogLevel` directive in `httpd.conf` and change it to a more verbose setting, such as “info.”

**Note:** Whenever new users are added, restart web service so that their sites are registered.

As always, follow the guidelines Apple has added as comments in configuration files. They explain safe procedures for modifying those files.

Note that a user's home directory, which would include a Sites folder, might not be present if the administrator added the user without creating a home directory for that person. There are several ways to create a home directory, such as adding the home directory in the Workgroup Manager application or using the command-line `createhomedir` tool to create the directory.

Here is a full description of the Apache configuration directives supported by `mod_rendezvous_apple`.

#### RegisterDefaultSite directive

- **Syntax:** `RegisterMachine [port | main]`
- **Default:** No registration if directive is absent. Port defaults to 80.
- **Context:** server config
- **Compatibility:** Apache 1.3.x; Mac OS X and Mac OS X Server only
- **Module:** `mod_rendezvous_apple`

This directive controls how the computer name is registered on the default site with Rendezvous.

The `RegisterDefaultSite` directive causes the registration of the default website under the computer name, as specified in the Sharing pane of System Preferences. A port number can be specified, or the keyword “main”; in the latter case, the port number of the “main server” (outside any virtual hosts) is used. On Mac OS X Server, do not specify “main,” because all externally visible sites are virtual hosts, and the main server is used only for status. If the argument is omitted, port 80 is used.

If the directive is absent, the computer name is not registered.

Rendezvous details: This directive results in a call to the registration function, with an empty string as the name (causing Rendezvous to use the computer name), with “\_http\_tcp” as the service type (indicating a web server), and with an empty string as the TXT parameter (indicating the default website).

## RegisterUserSite directive

- **Syntax:** RegisterUserSite *username* | all-users | customized users [ *registrationNameFormat* [*port* | main]
- **Default:** No registration if directive is absent; registration name defaults to longname. Port defaults to 80, host defaults to local.
- **Context:** server config
- **Compatibility:** Apache 1.3.x; Mac OS X and Mac OS X Server only
- **Module:** mod\_rendezvous\_apple

This RegisterUserSite directive causes the registration of the specified users' default website.

The required first argument is either an individual user's name or the keyword "all-users" or "customized-users." The "all-users" keyword causes all users in the hosts' directory to be considered for registration. Registration takes place if the user is a non-system user (user ID > 100), with an enabled website directory as specified in the UserDir directive, and only if that directory is accessible by the local host. Note that this may require a mount if the user's home directory is remote; if the home directory is not available, the user site is not registered. The "customized-users" keyword limits registration to those users who have an index.html file in their website directory that differs from the index.html file in the standard user template. In other words, it makes a reasonable attempt to limit registration to users who have customized their websites.

The optional second argument determines the form of the name under which the user site is registered. This takes the form of a format string, similar to the LogFormat directive. Certain directives in the format string are replaced with values:

%l - user's longname, such as Joe User

%n - user's short name, such as juser

%u - user's userid, such as 1234

%t - HTML title of user's index file (as determined by DirectoryIndex directive; by default it is index.html) from the user's default site folder (as determined by the UserDir directive; by default it is Sites). For Mac OS X Personal Web Sharing, the default title in a non-customized webpage is "Mac OS X Personal Web Sharing."

%c - computer name, as set in Sharing Preference panel

The default is %l, the longname. The second argument must be specified if the optional third argument is desired.

The optional third argument can be used to specify a port number under which the HTTP service is to be registered, or the keyword “main”; in the latter case, the port number of the “main server” (outside any virtual hosts) is used. In the case of Mac OS X Server, do not specify “main” for the port, because all externally visible sites are virtual hosts, and the main server is used only for status. If the port argument is omitted, port 80 is used.

If the directive is absent, no user site registration takes place. This directive is not processed if `mod_userdir` is not loaded. The `UserDir` and `DirectoryIndex` directives must precede the `RegisterUserSite` directive in the Apache config file.

Rendezvous details: This directive results in a call to the registration function, with a string like “Joe User” as the name, with “\_http\_tcp” as the service type (indicating a web server), and with a value like “path=/~juser/” as the TXT parameter (which, after expansion by `mod_userdir`, indicates the user’s default website), and with the appropriate port.

### RegisterResource directive

- **Syntax:** `RegisterResource name path [port | main]`
- **Default:** No registration if directive is absent. Port defaults to 80.
- **Context:** server config
- **Compatibility:** Apache 1.3.x; Mac OS X and Mac OS X Server only
- **Module:** `mod_rendezvous_apple`

The `RegisterResource` directive causes the registration of the specified resource path under the specified name.

The optional third argument can be used to specify a port number, or the keyword “main”; in the latter case, the port number of the “main server” (outside any virtual hosts) is used. On Mac OS X Server, do not specify “main,” because all externally visible sites are virtual hosts, and the main server is used only for status. If the third argument is omitted, port 80 is used.

Rendezvous details: This directive results in a call to the registration function, with the specified name, with “\_http\_tcp” as the service type (indicating a web server), with “path=/specifiedpath” as the TXT parameter, and with the appropriate port.

## 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 10.3 includes a preinstalled version of Apache Axis (1.1), which operates in conjunction with the preinstalled Tomcat 4.1.24-LE. 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 10.3 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.

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 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. Both are configured to listen on port 80, so either edit `/opt/apache2/conf/httpd.conf` to change the Listen directive or use the web section of Server Admin to change the port of all your virtual hosts to something other than 80. Also note that if the web performance cache is enabled, it may be the process that's actually listening on port 80.

## JBoss

JBoss is an open-source application server designed for J2EE applications; it runs on Java 1.4.1. JBoss is a widely used, full-featured Java application server. It provides a full Java 2Platform, 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 watchdog process to ensure continuous availability of JBoss once JBoss has been started. 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/Application`.



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/](http://www.jboss.org/).
- For J2EE, see [java.sun.com/j2ee/](http://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:**

- 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.

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 Do one of the following:
  - Click Load Remote Configuration and type the location of a JBoss NetBoot server.
  - 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](http://java.sun.com/products/servlets)
- For Java ServerPages specifications, see [java.sun.com/products/jsp](http://java.sun.com/products/jsp)

In Mac OS X Server 10.3, 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.

**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
./catalina.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 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 versions 10.3 includes the latest MySQL, version 4.0.14. 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.x and Use MySQL

Mac OS X Server version 10.3 contains a new version of MySQL. Previous versions of the server contain MySQL 3.23.x; the version now installed is 4.0.14, which is the latest production version. This version is the one recommended by [mysql.com](http://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 4.0.14, 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](http://www.mysql.com/doc/en/Upgrading-from-3.23.html) before using these commands.

For more information about MySQL, see [www.mysql.com](http://www.mysql.com).

# Installing and Viewing Web Modules

# 7

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](http://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. (Web service restarts automatically.)
- 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\_sherlock\_apple

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

- To index a folder's contents, choose Get Info from the file menu.

**Note:** You must be logged in as root for the index to be copied to the web directory in order to be searchable by a browser.

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

## 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

The new mod\_digest\_apple module enables digest authentication for a WebDAV realm.

## mod\_rendezvous\_apple

The new mod\_rendezvous\_apple module allows administrators to control how websites are registered with Rendezvous. See "Enabling Apache Rendezvous Registration" on page 51 for more information.

## 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 58.

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 58 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](http://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](http://perl.apache.org).





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](http://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

- 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 7, “Installing and Viewing Web Modules,” on page 61 and the Apache website at [www.apache.org/docs/mod/](http://www.apache.org/docs/mod/).

## A CGI Will Not Run

- 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.

For information about configuration files and other aspects of Apache web service, see these resources:

- *Apache: The Definitive Guide*, 3rd edition, by Ben Laurie and Peter Laurie (O'Reilly and Associates, 2002)
- *CGI Programming with Perl*, 2nd edition, by Scott Guelick, Shishir Gundavaram, and Gunther Birznieks (O'Reilly and Associates, 2000)
- *Java Enterprise in a Nutshell*, 2nd edition, by William Crawford, Jim Farley, and David Flanagan (O'Reilly and Associates, 2002)
- *Managing and Using MySQL*, 2nd edition, by George Reese, Randy Jay Yarger, Tim King, and Hugh E. Williams (O'Reilly and Associates, 2002)
- *Web Performance Tuning*, 2nd edition, by Patrick Killelea (O'Reilly and Associates, 2002)
- *Web Security, Privacy & Commerce*, 2nd edition, by Simson Garfinkel and Gene Spafford (O'Reilly and Associates, 2001)
- *Writing Apache Modules with Perl and C*, by Lincoln Stein and Doug MacEachern (O'Reilly and Associates, 1999)
- For more information about Apache, see the Apache website: [www.apache.org](http://www.apache.org)
- For an inclusive list of methods used by WebDAV clients, see RFC 2518. RFC documents provide an overview of a protocol or service that can be helpful for novice administrators, as well as more detailed technical information for experts. You can search for RFC documents by number at this website: [www.faqs.org/rfcs](http://www.faqs.org/rfcs)



**Apache** An open-source HTTP server that is integrated into Mac OS X Server. You can find detailed information about Apache at [www.apache.org](http://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.

**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. For example, if a user fills out a form on the site, a CGI could send the message to an application that processes the data and sends a response back to the user.

**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.

**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.

**Mac OS X Server** An industrial-strength server platform that supports Mac, Windows, UNIX, and Linux clients out of the box and provides a suite of scalable workgroup and network services plus advanced remote management tools.

**MySQL** An open-source relational database management tool for 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.

**owner** The person who created a file or folder and who therefore has the ability to assign access privileges for other users. The owner of an item automatically has read/write privileges for that item. An owner can also transfer ownership of an item to another user.

**PHP (PHP: Hypertext Preprocessor)** A scripting language embedded in HTML that is 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 or not 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*.

**short name** An abbreviated name for a user. The short name is used by Mac OS X for home directories, authentication, and email addresses.

**SSL (Secure Sockets Layer)** An Internet protocol that allows you to send encrypted, authenticated information across the Internet.

**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 is defined to provide access for WebDAV users and groups.





## A

- access privileges
  - setting for WebDAV 10
  - websites 11, 14
- Apache module 7, 9, 29, 51, 60, 61
- Apache web server 8, 61
  - configuration 9

## C

- CA certificate 45
- cache. *See* proxy cache
- certificate file 45–47
- CGI (Common Gateway Interface) 8
- CGI programs
  - problems with 66
- CGI scripts
  - enabling 33
  - installing 33
  - solving problems 66
- CSR (certificate signing request) 45–46

## D

- Documents folder 13

## F

- folders
  - Documents folder 13

## I

- Internet servers. *See* web servers

## J

- Java
  - JavaServer Pages (JSP) with Tomcat 21
  - servlet (with Tomcat) 21
  - Tomcat and 21

## L

- logs
  - access 27
  - error 27
  - SSL 35
  - web service 22

## M

- Macintosh-specific web modules 61
- MIME (Multipurpose Internet Mail Extension) 12
  - mappings 16
  - server response, setting 34
  - suffixes 11
  - type mapping 11
  - types 16
  - Types pane 16
  - understanding 11
  - web server responses 11
- mod\_auth\_apple module 62
- mod\_hfs\_apple module 62
- mod\_macbinary\_apple module 61
- mod\_perl module 63
- mod\_sherlock\_apple module 62
- Multipurpose Internet Mail Extension. *See* MIME
- MySQL Manager 59
- MySQL module 59

## O

- open source modules 60, 62, 63

## P

- Perl
  - mod\_perl 63
- PHP (PHP Hypertext Preprocessor) 63
  - Apache module 63
    - enabling 36
- PHP Hypertext Preprocessor (PHP) *See* PHP (PHP Hypertext Preprocessor)
- proxy 19
  - blocking websites with 20
- proxy cache
  - enabling 19
- proxy server 20

## R

- realms, WebDAV 10
- resources
  - web service 67

## S

- scripts
  - See* CGI scripts
- Secure Sockets Layer (SSL)
  - See* SSL (Secure Sockets Layer)
- security
  - WebDAV 10
  - websites 11
- Server Admin 23
  - configuring web server 9
    - mime\_macosxserver.conf file 34
    - modifying MIME type mappings 16
    - SSL, enabling 47
    - starting or stopping web service 15
    - starting Tomcat 22
    - viewing web service logs 22
    - viewing web service status 22
- servers
  - Apache web server 9
    - enabling SSL on 47
    - proxy servers 19, 20, 36
  - server side includes *See* SSI
- settings
  - MIME types 16
  - web service 15
- SQL 59
- SquirrelMail *See* WebMail
- SSI (server side includes) 8
  - enabling 33
- SSL (Secure Sockets Layer) 8
  - certificate signing request (CSR) 45
  - described 9
  - enabling 47
  - setting up 35, 45
  - website certificate 46

## T

- Tomcat module 62
  - Java and 21
  - Java servlet 21
  - JSP (JavaServer Pages) 21
    - starting 21
- troubleshooting
  - web service 65–66

## U

- Users 65

## W

- Web-based Distributed Authoring and Versioning (WebDAV) *See* WebDAV (Web-based Distributed Authoring and Versioning)
- web browsers 10
- WebDAV (Web-based Distributed Authoring and Versioning) 8
  - defining realms 10
  - described 7
  - enabling 21, 30
  - security 10
  - setting access 31
  - setting access privileges 10
  - setting up 21
  - understanding 10
- WebMail
  - about 41
  - configuring 43–44
  - enabling 42
  - logging in 42
  - mail server and 42
  - protocols 42
  - security limitations 42
  - SquirrelMail 41
- web modules 60, 61
  - Mac-specific 61
  - open-source 62
- webpages
  - default 13
- web servers
  - Apache web server 9
    - certificate for 46–47
- web service 7
  - configuring 9, 14
  - default page 13
  - described 7
  - Documents folder 13
  - limiting simultaneous 17
  - logs, viewing 22
  - monitoring 22
  - more information 67
  - MySQL 59
  - persistent connections 18

- problems with 65–66
- resources 67
- secure transactions 9, 45–47
- settings for 15
- setting up 13–15
- setting up websites 9
- solving problems 65
- SSL, enabling 20–36
- starting 15
- stopping 15
- Tomcat 21
- WebDAV 21
- WebMail, managing 42–44
- website privileges 14
- websites 23–36
  - access privileges 11
  - assigning privileges 14
  - connecting to 15
  - connection problems 65
  - default page 13, 25
  - default Web Folder 25
  - directory listing 28
  - documents Folder 23
  - enabling 24
  - hosting 10, 14
  - improving performance 26
  - information about 23
  - logs 27
  - MIME, configuring 35
  - monitoring 34
  - security of 11
  - setting access port 26
  - setting up 9
  - setting up SSL 35
  - solving problems 65–66
- web technologies
  - about 7
  - preparing for setup 7–12