



Mac OS X Server

Extending Your Wiki Server
For Version 10.5 Leopard

© 2008 Apple 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 Inc. is not responsible for printing or clerical errors.

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

Apple

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

The Apple logo is a trademark of Apple Inc., registered in the U.S. and other countries. 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, iCal, iTunes, Leopard, Mac, Mac OS, and Macintosh are trademarks of Apple Inc., registered in the U.S. and other countries. Finder, Safari, and Time Machine are trademarks of Apple Inc.

UNIX is a registered trademark of The Open Group.

Other company and product names mentioned herein are trademarks of their respective companies. Mention of third-party products is for informational purposes only and constitutes neither an endorsement nor a recommendation. Apple assumes no responsibility with regard to the performance or use of these products.

019-1252/2008-04-24

Contents

5	Preface: About This Guide
5	Getting Additional Information
7	Chapter 1: Customizing How the Wiki Looks
7	Theme File Structure Overview
9	Variant Theme File Structure Overview
10	Wireframe Theme File Structure Overview
12	Creating a Theme
13	Editing CSS Files
14	Editing Property List Files
14	About Property List Editor
15	About Property List Files
17	Changing General Theme Settings
18	Creating Sidebars
22	Overview of the Example Sidebars
23	Using the Example Sidebars
25	Adding JavaScript and XSL Files
26	About JavaScript
27	About XSL
30	Chapter 2: Allowing Specific Protocols, CSS Styles, and HTML Tags and Attributes
31	Creating a Custom Whitelist
32	About the Custom Whitelist's Structure
32	Allowing Specific URL Protocols
32	Allowing Specific CSS Styles
33	Allowing Specific HTML Tags and Attributes
34	About the Built-in Whitelist
34	Protocols Allowed in the Built-in Whitelist
35	CSS Styles Allowed in the Built-in Whitelist
35	HTML Tags and Attributes Allowed in the Built-in Whitelist

37	Chapter 3: Managing Wiki Content
37	Migrating Wiki Content From Other Websites
37	Backing Up the Wiki Server
39	Wiki File Structure Overview
40	Manually Editing Raw Content
41	Viewing a Wiki's Administration Settings
44	Viewing a Page's Metainformation
45	Managing Attachments

About This Guide

This guide explains how to create custom themes, allow specific protocols, CSS styles, and HTML tags and attributes, and manage wiki content.

The default Wiki Server configuration makes it easy for groups to collaborate and communicate. Users can create and edit wiki pages, tag and cross-reference material, upload files and images, add comments, and search content.

This guide shows you how to customize and extend the wiki to suit your needs. For example, you'll learn how to:

- Create custom themes
- Enable JavaScript, YouTube, and Flash
- Automate migration from other sources, such as wikis and non-wiki websites

The tasks described in this guide can be used on standard, workgroup, and advanced configurations of Leopard Server. These tasks require administrator access and in some cases programming skill. You must be able to edit files on the server, which aren't accessible to most users. Be careful when editing these files, as an incorrect or untested change could cause problems.

Getting Additional Information

Information about setting up, administering, and using Wiki Server is covered in detail in several places.

For information about setting up web service, see:

- *Getting Started*—if you're using Server Preferences
- *Web Technologies Administration*—if you're using Server Admin

You can find both guides at the Mac OS X Server Resources website: www.apple.com/server/macosx/resources

For information about administering wikis, see Directory Help. You can view Directory Help by opening Directory (located in /Applications/Utilities/) and choosing Help > Directory Help.

For information about using the wiki, see Wiki help. You can view Wiki help by clicking the Help link located at the bottom of any Apple Wiki page.

This chapter describes how to customize a wiki's appearance.

You can use a wiki's admin settings to do the following:

- Change the wiki's title
- Change the wiki's theme
- Add a custom tag-search sidebar to the main page
- Choose who can make comments and which comments are moderated
- Create a podcast and integrate it with iTunes

These settings are sufficient for most wikis. However, if you have special needs—for example, if you want to use a specific organization's style, or if you want to change the appearance and names of the default sidebars on the front page—you must edit files located on the server.

Theme File Structure Overview

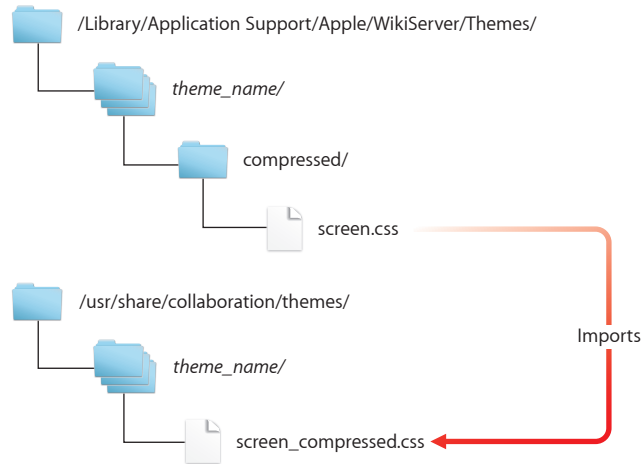
A theme is composed of Cascading Style Sheet (.css) files, property list (.plist) files, image files (usually .png and .jpg), and sometimes JavaScript (.js) and Extensible Stylesheet Language (.xsl) files.

All themes use the CSS files located at:
`/usr/share/collaboration/css/required/`

However, CSS files located in individual theme folders take precedence and override the settings in this folder.

Individual theme folders are subfolders with a `.wikitheme` extension that are located within this folder:

`/Library/Application Support/Apple/WikiServer/Themes/`

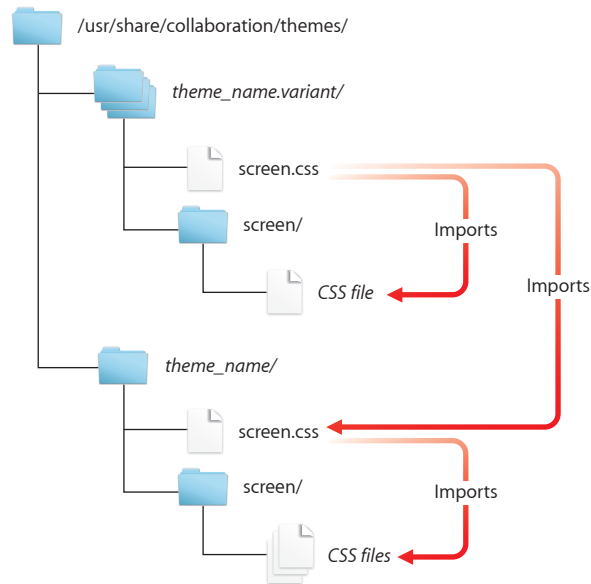


If the theme has a `/compressed` subfolder, it imports the CSS files located at:
`/usr/share/collaboration/themes/theme_name/screen_compressed.css`

The `screen_compressed.css` file combines the many individual CSS files located in the `theme_name/screen/` folder. These are provided in the prebuilt themes to improve overall performance. The `screen_compressed.css` file isn't created automatically or dynamically, so any changes to `.css` files located in the `theme_name/screen/` folder don't affect the `screen_compressed.css` file.

Variation Theme File Structure Overview

Prebuilt themes have several variants, which all share the same layout but might have different banners or color schemes. Shared CSS files are in the main theme's folder, while variants have separate CSS files located in other folders.



For example, the block theme has its shared CSS files located in:

```
/usr/share/collaboration/themes/block/screen/
```

Variants of the block theme all use these shared CSS files. Each variant also uses CSS files located in another folder such the following:

```
/usr/share/collaboration/themes/block.green.banner/screen/
```

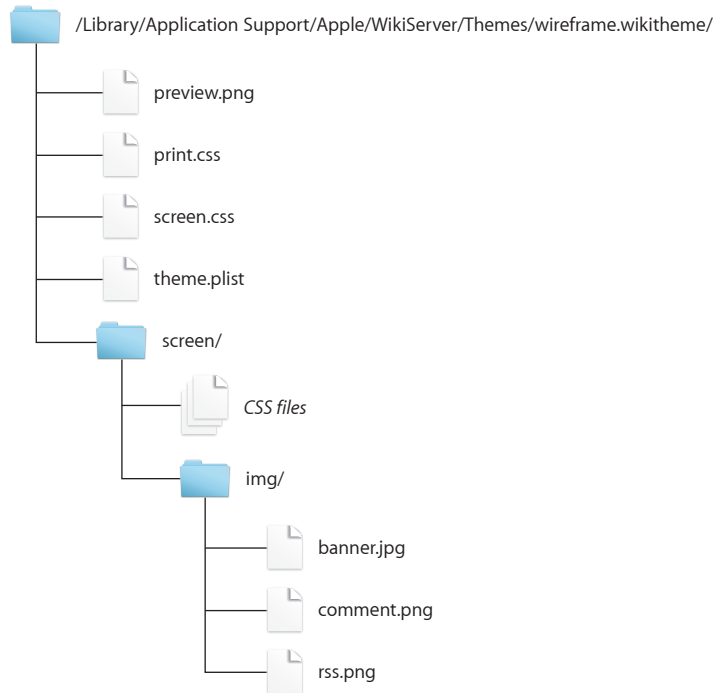
```
/usr/share/collaboration/themes/block.red/screen/
```

```
/usr/share/collaboration/themes/block.red.banner/screen/
```

These CSS files all serve only as examples, since the prebuilt themes use the screen_compressed.css file instead of these individual files.

Wireframe Theme File Structure Overview

The wireframe theme is a good example theme to customize, because unlike the other built-in themes, its CSS files are all located in its main theme folder (in the /Library location and not in the /usr location). It was designed as a bare-bones theme, which you can easily add to without having to remove unwanted elements.



The wireframe theme's folder includes the following files and folders:

File or Folder Name	Purpose
preview.png	Displayed when users are choosing which theme to use. Its dimensions are 128 x 128 pixels.
print.css	Applied when wiki pages are printed. If you don't remove the import line from this file, printing uses the print.css file located at: <code>/usr/share/collaboration/default/print.css</code>
screen.css	Applied when wiki pages are displayed on screen. This file simplifies organization and file structure by using the CSS files in /screen instead of containing all the CSS code in one file.
theme.plist	Affects general settings such as the theme's name, and sidebar attributes such as the maximum number of entries and types of sidebars shown.
screen/	Includes all of the CSS files used when displaying the wiki onscreen.

File or Folder Name	Purpose
screen/calendar.css	Sets the colors for the web calendar.
screen/comments.css	Sets various appearance attributes (such as padding, borders, and positioning) for comment entry and display.
screen/footer.css	Sets footer attributes, such as the margin, padding, color, and font size.
screen/forms.css	Sets the margin for the search field.
screen/general.css	Sets general blog and wiki display attributes, such as the font size and color, background colors, heading margins, and list styles.
screen/grouphome.css	Sets display attributes for the sidebar located on the main wiki page, such as its size, margins, and font size.
screen/header.css	Sets display attributes (such as the font size, margin, and padding) for the wiki, blog, calendar, and mail links located at the top of each page.
screen/layout.css	Sets display attributes (such as the margin and size) for the wiki and blog editing interface. Customizing the editing interface is an advanced task; depending on what you're trying to do, you may need to write JavaScript.
screen/metadata.css	Sets display attributes for fields in page histories, such as the margin and font size.
screen/tables.css	Sets display attributes for tables and table headers, such as the color, border styles, and font size.
screen/tags.css	Sets display attributes for the tag list shown on each page, such as the margins, padding, and separators.
screen/thumbnail.css	Sets display attributes for image thumbnails, such as the border styles, color, padding, and the magnification graphic.
screen/toc.css	Disabled. Don't edit.
screen/toolbar.css	Sets the editing toolbar's color, background color, and margin.
screen/weblog.css	Sets blog display attributes, such as which elements are displayed.
screen/img/	Contains graphics used in the theme.
screen/img/banner.jpg	Default banner for the theme. Its dimensions are 950 x 140 pixels.
screen/img/comment.png	Displayed in the upper right of every comment. Its dimensions are 16 x 15 pixels.
screen/img/rss.png	Displayed in the upper right of the What's Hot and Recent Changes sidebars on the main page. Its dimensions are 12 x 12 pixels.

Creating a Theme

The easiest way to create a theme is to take the existing wireframe theme, make a copy of it with permissions intact, and then customize the copy.

If you edit the included themes instead of copying them to new themes, your edits may be overwritten later by a software update.

To create a copy of the wireframe theme with permissions intact:

- 1 In Terminal, enter the following commands:

```
$ cd /Library/Application\ Support/Apple/WikiServer/Themes/  
$ sudo ditto wireframe.wikitheme/ theme_name.wikitheme/
```

Replace *theme_name* with the name of the theme you're creating.

If prompted, enter your administrator account password.

- 2 If you're going to use Property List Editor or a non-command-line tool, copy the theme.plist file to your Documents folder.

To copy this file, enter:

```
$ cp theme_name.wikitheme/theme.plist /Users/user_name/Documents/
```

Replace *user_name* with the short name of the user you're logged in as.

For information about Property List Editor, see "About Property List Editor."

- 3 Open theme.plist in Property List Editor or a text editor.

To open this file in a command-line text editor, enter:

```
$ sudo nano theme_name.wikitheme/theme.plist
```

If prompted, enter your administrator account password.

You can use any command-line text editor, as long as you precede the command to run the text editor with `sudo`.

If you're using Property List Editor or a non-command-line tool, open the theme.plist located in your Documents folder.

- 4 Remove the selectable key and its value, and then save the plist.

To remove the key using Property List Editor, select the selectable key and then click Delete.

To remove the key using a text editor, remove the following lines:

```
<!-- remove this selectable key from your theme -->  
<key>selectable</key>  
<false/>
```

- 5 If you copied theme.plist to your Documents folder, copy it back to the theme folder.

To copy this file, enter:

```
$ cp /Users/user_name/Documents/theme.plist theme_name.wikitheme/
```

6 Restart the web service by doing one of the following:

- Restart the server.
- In Server Preferences, click Web, move the slider to Off, and then to On.
- In Server Admin, open the disclosure triangle for your server, select Web, click Stop Web, and then click Start Web.
- Enter the following in Terminal:

```
$ sudo serveradmin stop teams; sudo serveradmin start teams;
```

If prompted, enter your administrator account password.

Restarting the server and restarting web service can disrupt users, so send out a notification that the server will be temporarily down before doing this.

Editing CSS Files

Cascading Style Sheet (.css) files control how the wiki looks. By editing these CSS files, you can easily change display attributes for specific page elements. For example, you can change:

- Color
- Background color
- Margin
- Padding
- Font size

If you're skilled at editing CSS files, you can completely overhaul the look of the wiki by changing its layout, removing unwanted visual elements, and adding your own visual elements.

When you edit a CSS file, your changes occur immediately; you don't have to restart the server or web service after editing. Because you might disrupt users by changing CSS files, you should create, edit, and test them with a theme that isn't currently in use.

There are many online sources of information about Cascading Style Sheets. These websites should help you get started:

Website	URL
World Wide Web Consortium	w3.org
CSS3	css3.info
QuirksMode	quirksmode.org
Position is Everything	positioniseverything.net
The WebKit OpenSource Project	webkit.org
A List Apart	alistapart.com
CSS Zen Garden	csszengarden.com

Here are some published CSS guides:

- *CSS: The Definitive Guide, 3rd Edition*, by Eric Meyer. O'Reilly Media, Inc. 2006.
- *CSS Mastery: Advanced Web Standards Solutions*, by Andy Budd, Simon Collison, and Cameron Moll. Friends of ED. 2006.

Editing Property List Files

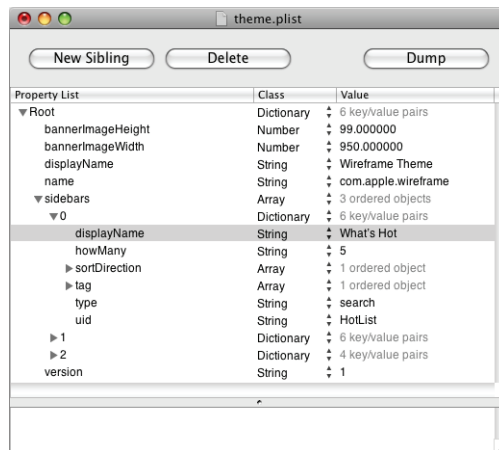
Every theme has a theme.plist file, which controls:

- The name of the theme
- Whether the theme is publicly available
- The height and width of the banner
- Sidebar properties such as the name, type and number of links, and sort order

After making changes to the theme.plist file, you must restart the web service or restart the server. Because this can disrupt users, you should create, edit, and test themes on a test computer and then copy them over to the server.

About Property List Editor

Property List Editor is a specialized editor for property list files.



In order to edit a theme.plist file using Property List Editor, you must copy it to your Documents folder. When you finish editing the theme.plist file, you must copy it back to `/Library/Application Support/Apple/WikiServer/Themes/theme_name.wikithe.me/`.

Property List Editor is located on the *Mac OS X Server v10.5 Administration Tools CD*, in `/Utilities/`. You can run it from the CD or copy it to a different location.

Property List Editor doesn't show comments, so if you want to see the example sidebars provided in the wireframe's theme.plist, you'll need to open the theme.plist file in a text editor and remove the comment markers surrounding the example sidebars.

About Property List Files

The theme.plist file is structured like an XML file. If you understand XML, editing the theme.plist is straightforward. If not, read the following sections for basic editing information.

About Property List Keys and Values

Every setting controlled by the theme.plist file has a key, and an associated value or an array (a collection) of associated values. To change the setting, you change the value or values, not the key.

For example, to halve the height of the banner image in Property List Editor, double-click the value 99 next to bannerImageHeight and replace it with 50.

To halve the height of the banner image in a text editor, change the following lines in the theme.plist:

```
<key>bannerImageHeight</key>  
<real>99</real>
```

to:

```
<key>bannerImageHeight</key>  
<real>50</real>
```

About Strings

Most of the values in the theme.plist file are character sequences called *strings*.

About Dictionaries

A *dictionary*, or *dict* when viewed in a text editor, is an assortment of values of different classes. In the theme.plist file, a sidebar is represented by a dictionary.

Because a dictionary can contain several values, Property List Editor provides a disclosure triangle that allows you to view or hide the contents of the dictionary.

About Arrays

Arrays are collections or lists of similar classes. The theme.plist file includes two types of arrays: a list of dictionaries and a list of strings.

The sidebars array is an example of a dictionary array. Because the sidebars array is commented out by default, you won't be able to see it in Property List Editor without first removing the comment markers in a text editor.

Because an array can contain several values, Property List Editor provides a disclosure triangle that allows you to view or hide the contents of the array.

Here is the sidebars array in a plain text editor:

```
<key>sidebars</key>
<array>
  <dict>
    sidebar keys and values
  </dict>
  <dict>
    sidebar keys and values
  </dict>
</array>
```

Here is an example of an array with a single string:

```
<array>
  <string>value</string>
</array>
```

Here is an example of an array with multiple strings:

```
<array>
  <string>value</string>
  <string>value 2</string>
</array>
```

About Comments

The theme.plist file includes comments to help provide editing instructions and example keys and values. Comments are ignored when the theme.plist file is read by the wiki server.

Comments start with “<!--” and end with “-->.” Everything in between is ignored, even if the comment contains many lines of keys and values.

For example, the wireframe theme’s theme.plist file includes the following comment to indicate where the example sidebars section starts:

```
<!-- example (non-default) sidebars -->
```

The sidebars array is preceded by a <!-- comment marker. The end comment marker doesn’t appear until after the array. Because the comment markers surround the array, the code for these sidebars in the array is ignored unless the comment markers are removed.

About Errors

Make sure all required values are present. If you don’t include required keys and values in the theme.plist file (for example, a displayName or a uid for a sidebar), loading the wiki’s main page will display an “Internal Server Error” page. Wiki pages that don’t have sidebars won’t display this error.

Changing General Theme Settings

The wireframe theme's theme.plist includes keys and values for the following general settings:

Key	Default Value	Possible Values	Optional or Required	Description
bannerImageHeight	99	Any number, 0 or more	Optional	<p>The amount in pixels allotted to the height of the banner image. Images you upload are scaled to this size.</p> <p>If you define bannerImageHeight but not bannerImageWidth, the width scales proportionally.</p> <p>This value affects the banner image's height, not the banner's height. To change the banner's height, edit the wireframe theme's layout.css file.</p>
bannerImageWidth	950	Any number, 0 or more	Optional	<p>The amount in pixels allotted to the width of the banner image. Images you upload are scaled to this size.</p> <p>If you define bannerImageWidth but not bannerImageHeight, the height scales proportionally.</p> <p>This value affects the banner image's width, not the banner's width. To change the banner's width, edit the wireframe theme's layout.css file.</p>
displayName	Wireframe Theme	Any string (such as a word or phrase)	Required	The name of the theme as shown in the theme chooser accessed from a wiki's admin settings.
name	com.apple.wireframe	Any string with no spaces	Required	Uniquely identifies the theme.
selectable	<false/>	<false/>	Optional	Displays a theme in the theme chooser, if the selectable key and <false/> value are both removed.
version	1	A number	Required	Don't change this value.

Creating Sidebars

If you haven't created a sidebar before, it might be easier to customize an example sidebar to suit your needs.

In the wireframe theme's `theme.plist` file, there is a `sidebars` section that contains an array of sidebars you can customize:

```
<key>sidebars</key>
<array>
  <dict>
    sidebar keys and values
  </dict>
  <dict>
    sidebar keys and values
  </dict>
</array>
```

Understanding the Types of Sidebars

There are several types of sidebars, each with a specific purpose and set of keys:

Sidebar Type	Description	Related Section
static	A list of URLs.	"Static Unique Keys," on page 19
search	A list of search results using tag-based or text-based searching.	"Search Unique Keys," on page 20
calendar	A list of upcoming events.	"Calendar Unique Keys," on page 21
tag	A list of search results using tag-based searching.	"Tag Unique Keys," on page 22
recentActivity	A list of pages sorted by recent viewing frequency. Based on the last 100 page views in a wiki, this list displays the most-viewed page first, followed by the next most-viewed, and so on.	"Recent Activity Unique Keys," on page 22
admin	A sidebar that appears automatically if someone logs in as an admin. This sidebar isn't customizable.	Not applicable

Shared Keys

All sidebars require the following keys:

Key	Example Values	Possible Values	Optional or Required	Description
<code>displayName</code>	What's Hot, What's New, Useful Links, Upcoming Events	Any string	Required	Name of the sidebar.
<code>type</code>	search, static, calendar	static, search, calendar, tag, recentActivity, admin	Required	Type of sidebar used.
<code>uid</code>	HotList, RecentAddsList, UsefulLinks	Any string	Required	Identifier for the sidebar. This value must be unique.

Static Unique Keys

Static sidebars list predefined URLs. “Useful Links” is an example static sidebar.

Static sidebars have the following keys:

Key	Example Value	Possible Values	Optional or Required	Description
urls	Not applicable	An array of dictionary entries, each representing a single link	Required	Because links include several values, such as the name of the link and the URL itself, every link is represented by a dictionary entry.
headerURL	<code>http://example.com/groups/groupname/search/?sort=modifiedDate&kind=wiki&kind=weblog&kind=mailinglist&sortDirection=reverse&excludePages=wiki/welcome</code>	Fully qualified URL	Optional	A URL that is linked to from the sidebar’s header.
feedURL	<code>feed://example.com/groups/groupname/search/index.rss?sort=modifiedDate&kind=wiki&kind=weblog&kind=mailinglist&sortDirection=reverse&excludePages=wiki/welcome</code>	Fully qualified URL	Optional	An RSS or Atom feed that is associated with the sidebar.

Each dictionary entry has the following keys:

Key	Example Value	Possible Values	Optional or Required	Description
kind	url	url, wiki, weblog, mailinglist, calendar	Required	Not used in any themes. Use this to assign a class to a sidebar. You can then use the class selector in CSS to customize sidebars, such as by adding small calendar icons next to calendar sidebar links or small Safari icons next to URL links.
subtitle	Your server’s homepage, List of groups with wikis on this server	Any string	Required	Appears under the name of the link.
title	Server Home, Groups	Any string	Required	Name of the link.
uid	serverhome, groups	Any string	Required	Identifier for the URL. This value must be unique.
url	<code>/, /groups/</code>	A relative or absolute path to a web page	Required	Contents of the link.

Key	Example Value	Possible Values	Optional or Required	Description
snippet	Click here to see your server's homepage.	Any string	Optional	Used for the link's tool tip. To enable tool tips, comment out the following block in the theme's grouphome.css file: <pre>.grouphome .snippet{ display:none; }</pre>

Search Unique Keys

Search sidebars list search results using tag-based or text-based searching. “What’s Hot” and “What’s New” are examples of search sidebars.

Search sidebars have the following unique keys:

Key	Example Value	Possible Values	Optional or Required	Description
howMany	5	Any number, 1 or more	Required	Maximum number of entries in the sidebar.
tag	hot	One or more strings all in the same array	Optional	The tags associated with this sidebar. Pages using all of these tags are added to this sidebar.
kind	wiki, weblog, mailinglist	One or more of the following in the same array: <ul style="list-style-type: none"> • all • wiki • weblog • mailinglist • calendar 	Optional	The kinds of pages associated with this sidebar. A search only lists pages of these types. If this key isn't provided, all types of pages can be shown in the sidebar.
q	annejohnson, wiki, 2008-03-17	One or more strings all in the same array	Optional	The text query used to search for entries.
fields	title, content, author	One or more of the following in the same array: <ul style="list-style-type: none"> • title • author • authorLongName • lastModifiedAuthor • lastModifiedAuthorLongName • createDate • modifiedDate • content • strippedContent • kind 	Optional	The fields the text query is applied to. If the query matches any of these fields, the entry appears in the sidebar.

Key	Example Value	Possible Values	Optional or Required	Description
sort	modifiedDate, createDate, title, author	One or more of the following in the same array: <ul style="list-style-type: none"> • title • author • authorLongName • lastModifiedAuthor • lastModifiedAuthor LongName • createDate • modifiedDate • content • strippedContent • kind 	Optional	Entries are sorted based on these fields. The first field listed is sorted first, then the entries are resorted by subsequent fields. The last field listed takes precedence over all previously listed fields.
sortDirection	reverse	forward or reverse	Optional	Sorting order of entries. reverse—most recently modified page is listed first forward—least recently modified page is listed first
emptyListString	There are no matches.	Any string	Optional	Shown when there are no matching search results.

Calendar Unique Keys

Calendar sidebars list upcoming events. “Upcoming Events” is an example calendar sidebar.

Calendar sidebars have the following unique keys:

Key	Example Value	Possible Values	Optional or Required	Description
howManyDays	5	Any number, 1 or more	Required	Maximum number of days until an event for it to be listed in the sidebar. Events occurring after this number of days aren’t shown in the sidebar.
emptyListString	There are no matches.	Any string	Optional	Shown when there are no matching calendar events.

Tag Unique Keys

Tag sidebars list wiki, blog, and mailing list entries that use a specific tag or set of tags. The most recently modified entries are listed first.

Tag sidebars have the following unique keys:

Key	Example Value	Possible Values	Optional or Required	Description
tag	hot	A single string or several strings all in the same array	Required	Tags associated with the sidebar. Pages using all of these tags are added to the sidebar.
howMany	5	Any number, 1 or more	Required	Maximum number of entries in the sidebar.
emptyListString	There are no matches.	Any string	Optional	Shown when there are no matching search results.

Recent Activity Unique Keys

Recent activity sidebars list pages that have had the most readers recently.

Recent activity sidebars have the following unique keys:

Key	Example Value	Possible Values	Optional or Required	Description
howMany	5	Any number, 1 or more	Required	Maximum number of entries in the sidebar.
emptyListString	There are no matches.	Any string	Optional	Shown when there are no matching search results.

Overview of the Example Sidebars

The wireframe theme's theme.plist provides several example sidebars:

Sidebar Name	Description
What's Hot	Lists the five most recently modified pages tagged with "hot."
What's New	Lists the five most recently created pages. This includes wiki, weblog, and mailing list pages.
Useful Links	Lists several predefined links.
Upcoming Events	Lists all events occurring within the next five days.

Using the Example Sidebars

The wireframe theme's `theme.plist` file includes several sidebar examples. All of these examples are commented out by default. You'll need to remove the comments marks if you want to use the example sidebars as custom sidebars. If you don't add any custom sidebars, the default sidebars are used. If you use at least one custom sidebar, the default sidebars don't appear.

Consider commenting out all sidebars you don't want to use, and leaving uncommented all sidebars you want to use. Doing this allows you to choose which sidebars you want and allows you to easily change sidebar settings later on. To do this, you must start a comment before the `<dict>` tag for a sidebar, and end the comment after the `</dict>` tag for that sidebar.

To comment out specific sidebars:

- 1 In Terminal, enter the following command:

```
$ cd /Library/Application\ Support/Apple/WikiServer/Themes/
```

- 2 If you want to use a non-command-line tool, copy the `theme.plist` file to your Documents folder.

To copy this file, enter:

```
$ cp theme_name.wikitheme/theme.plist /Users/user_name/Documents/
```

Replace *theme_name* with the name of the theme. Replace *user_name* with the short name of the user you're logged in as.

- 3 Open `theme.plist` in a text editor.

To open this file in a command-line text editor, enter:

```
$ sudo nano theme_name.wikitheme/theme.plist
```

If prompted, enter your administrator account password.

You can use any command-line text editor, as long as you precede the command to run the text editor with `sudo`.

If you're using a non-command-line tool, open the `theme.plist` located in your Documents folder.

- 4 If there is a `<!--` comment marker before `<keys>sidebars</key>`, remove it and the `-->` comment marker before `<key>version</key>`.

5 For each sidebar that you want to comment out, do the following:

a Add `<!--` before the `<dict>` tag of the sidebar you want to comment out.

For sidebars, `<dict>` tags are followed by the `displayName` key.

For example, to comment out the “What’s Hot” sidebar, you’d change this:

```
<dict>
<key>displayName</key>
<string>What’s Hot</string>
```

to this:

```
<!--
<dict>
<key>displayName</key>
<string>What’s Hot</string>
```

b Add `-->` after the `</dict>` tag of the sidebar you want to comment out.

The corresponding `</dict>` tag is the first one found at the same indentation level of the sidebar’s `<dict>` tag.

For example, to finish commenting out the “What’s Hot” sidebar, you’d look for the first `</dict>` tag at the same indentation level as the `What’s Hot <dict>` tag level, and then you’d change this:

```
</dict>
```

to this:

```
</dict>
-->
```

6 If you copied `theme.plist` to your Documents folder, copy it back to the theme folder.

To copy this file, enter:

```
$ cp /Users/user_name/Documents/theme.plist theme_name.wikitheme/
```

7 Restart the web service by doing one of the following:

- Restart the server.
- In Server Preferences, click Web, move the slider to Off, and then to On.
- In Server Admin, open the disclosure triangle for your server, select Web, click Stop Web, and then click Start Web.
- Enter the following in Terminal:

```
$ sudo serveradmin stop teams; sudo serveradmin start teams;
```

If prompted, enter your administrator account password.

Restarting the server and stopping and starting web service can disrupt users, so send out a notification that the server will be temporarily down before doing this.

Adding JavaScript and XSL Files

The wiki server supports the use of custom JavaScript and Extensible Stylesheet Language (XSL) files. JavaScript and XSL change the content you're displaying, whereas CSS controls the presentation of content. For example, you could use JavaScript or XSL to add a "digg it" link to your blog entries, which would then tie the blog entries to digg.com.

Carefully choose whether you want to use JavaScript, XSL, or both.

JavaScript has these advantages:

- Unlike custom XSL files, JavaScript files continue to work after doing a software update. The wiki server includes XSL files, which can be changed during a software update. If these files change, then an error occurs because the file versions of included files no longer match the file versions of custom XSL files.
- JavaScript can allow you more control over a document than XSL if you have access to information on the server that isn't provided in the XSL context.

XSL has this advantage:

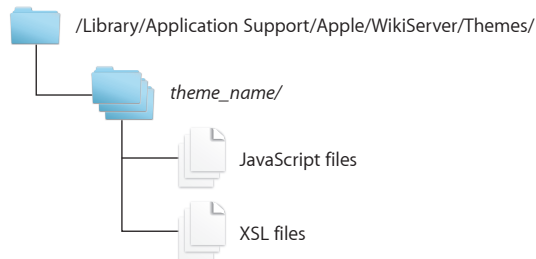
- XSL transformations are performed on the wiki server, so the user's browser isn't required to support and run it.

To add JavaScript files or XSL files to a theme:

- 1 Put JavaScript (.js) files or XSL (.xsl) files in this folder:

`/Library/Application Support/Apple/WikiServer/Themes/theme_name/`

Replace *theme_name* with the name of the theme.



2 Restart the web service by doing one of the following:

- Restart the server.
- In Server Preferences, click Web, move the slider to Off, and then to On.
- In Server Admin, open the disclosure triangle for your server, select Web, click Stop Web, and then click Start Web.
- Enter the following in Terminal:

```
$ sudo serveradmin stop teams; sudo serveradmin start teams;
```

If prompted, enter your administrator account password.

Restarting the server and restarting web service can disrupt users, so send out a notification that the server will be temporarily down before doing this.

About JavaScript

JavaScript is a lightweight, object-oriented, cross-platform scripting language.

All wiki server pages automatically include the frameworks `prototype.js` and `script.aculo.us`, which enhance and streamline the use of JavaScript.

For more information, see the following:

Topic	URL
JavaScript	developer.mozilla.org/en/docs/JavaScript quirksmode.org/js/contents.html crockford.com/javascript/javascript.html
<code>prototype.js</code>	prototypejs.org/api
<code>script.aculo.us</code>	wiki.script.aculo.us

Here are some published JavaScript guides:

- *JavaScript: The Definitive Guide, 5th Edition*, by David Flanagan. O'Reilly Media, Inc. 2006.
- *Beginning JavaScript with DOM Scripting and Ajax: From Novice to Professional*, by Christian Heilmann. Apress. 2006.

JavaScript Example

The following example JavaScript function adds a header link next to the magnifier button that points to www.apple.com, and changes the “Other Wikis” link to point to www.apple.com:

```
Event.observe(window, 'load', function() {
    // add another header link (to apple.com) by manipulating the DOM
    // using script.aculo.us Builder
    // we'll just insert it before the search button
    if ($('#linkSearch')) {
        $('#linkSearch').parentNode.insertBefore(Builder.node('li',
            {id:'linkApple'}, [
                Builder.node('a', {href:'http://www.apple.com'}, 'apple')
            ]), $('#linkSearch'));
    }
    // change the Other Wikis link to something else
    if ($('#groups_users_button')) {
        $('#groups_users_button').down('a').href = 'http://www.apple.
        com/';
    }
});
```

To use this example JavaScript code, save it as a .js file in this folder:

`/Library/Application Support/Apple/WikiServer/Themes/theme_name/`

About XSL

XSL defines XML document transformation and presentation. For information about XSL, see:

www.w3.org/Style/XSL/

The wiki server uses a set of XSL files located at:

`/usr/share/wikid/lib/apple_templates/`

Because a software update can overwrite files in `/usr/share/wikid/lib/apple_templates/`, edit the copies of these XSL files located in:

`/Library/Application Support/Apple/WikiServer/Themes/theme_name/`

XSL files in this location override XSL files in `/usr/share/wikid/lib/apple_templates/`.

If a software update overwrites the files in `/usr/share/wikid/lib/apple_templates/`, you'll need to manually merge the changed contents of your customized XSL files with the updated XSL files. If you don't merge the contents, the versions won't match and an error will occur.

If you edit a copy of the default.xml file, you must also make copies of all of the XSL files that include the original default.xml file, and then change them to include the edited default.xml file.

Viewing Dynamic XML

The wiki server dynamically generates XML content, which is called *context*. Context is transformed by XSL files into HTML.

Before you view context, familiarize yourself with the XSL files located in:

```
/usr/share/wikid/lib/apple_templates/
```

The XSL files correspond to specific pages in the wiki. For example, a group's wiki homepage uses grouphome.xml, while a typical wiki page uses content_entry.xml. To view context, create an XSL file that displays context and put it in your theme's folder. This overrides the XSL files in /usr/share/wikid/lib/apple_templates/ and changes the output for the pages that they correspond to.

To view context:

- 1 Create a plain text file with the following content:

```
<?xml version="1.0" encoding="UTF-8" ?>
<xsl:stylesheet version="1.0"
    xmlns:xsl="http://www.w3.org/1999/XSL/Transform">
  <xsl:output
    method="html"
    version="1.0"
    encoding="UTF-8"
    indent="yes"
    doctype-public="-//W3C//DTD HTML 4.01//EN"
    doctype-system="http://www.w3.org/TR/html4/strict.dtd"/>
  <xsl:template match="page">
    <xsl:copy-of select="context"/>
  </xsl:template>
</xsl:stylesheet>
```

- 2 Save the file in this location:

```
/Library/Application Support/Apple/WikiServer/Themes/theme_name/
```

Use the same filename as any of the XSL files located in:

```
/usr/share/wikid/lib/apple_templates/
```

Choose a filename based on the page with the context you want to view. For example, to view the context for the group's wiki homepage, change the filename to grouphome.xml.

3 Restart the web service by doing one of the following:

- Restart the server.
- In Server Preferences, click Web, move the slider to Off, and then to On.
- In Server Admin, open the disclosure triangle for your server, select Web, click Stop Web, and then click Start Web.
- Enter the following in Terminal:

```
$ sudo serveradmin stop teams; sudo serveradmin start teams;
```

If prompted, enter your administrator account password.

Restarting the server and restarting web service can disrupt users so send out a notification that the server will be temporarily down before doing this.

4 In a web browser, load a page that uses the XSL file, and then view the page's source.

To view a page's source in Safari, load the page and choose View > View Source.

Allowing Specific Protocols, CSS Styles, and HTML Tags and Attributes

This chapter describes how to enable specific protocols, CSS styles, and HTML tags and attributes.

The default wiki server setup simplifies administration by automatically removing potentially harmful protocols, CSS styles, and HTML tags and attributes. The wiki server is capable of allowing all protocols, CSS styles, and HTML tags and attributes.

The wiki server uses two whitelist files (a built-in whitelist and a custom whitelist) to determine allowed protocols, CSS styles, and HTML tags and attributes. Elements that appear in either of these whitelists are allowed, and all other elements are disallowed.

The built-in whitelist includes common, usually harmless, elements. It doesn't include potentially harmful tags like `embed`, `param`, `object`, and `script`. To embed Flash or YouTube in your site, you'll need to include some of these tags. If you create a custom whitelist, you can allow these elements, along with new styles (such as `font-size`) and protocols (such as `irc` and `scp`).

These whitelists affect all wikis on the server.

WARNING: Some protocols, HTML tags and attributes can compromise your server's security and integrity, or harm users who connect to your server. Make sure you understand the implications of whatever you enable. For example, allowing JavaScript introduces security vulnerabilities such as cross-site scripting. For information about cross-site scripting, see http://en.wikipedia.org/wiki/Cross-site_scripting.

Creating a Custom Whitelist

To create a custom whitelist, create a plain text file named `whitelist.plist` in:
`/Library/Application Support/Apple/WikiServer/`

The following sections describe this example `whitelist.plist` in detail:

```
<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE plist PUBLIC "-//Apple//DTD PLIST 1.0//EN" "http://www.apple.
    com/DTDs/PropertyList-1.0.dtd">
<plist version="1.0">
<dict>
    <key>protocols</key>
    <array>
        <string>irc</string>
        <string>scp</string>
    </array>
    <key>styles</key>
    <array>
        <string>font-size</string>
    </array>
    <key>tags</key>
    <dict>
        <key>object</key>
        <array>
            <string>width</string>
            <string>height</string>
        </array>
        <key>param</key>
        <array>
            <string>name</string>
            <string>value</string>
        </array>
        <key>embed</key>
        <array>
            <string>src</string>
            <string>type</string>
            <string>width</string>
            <string>height</string>
        </array>
    </dict>
</dict>
</plist>
```

About the Custom Whitelist's Structure

The following lines are required in the custom whitelist:

```
<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE plist PUBLIC "-//Apple//DTD PLIST 1.0//EN" "http://www.apple.
    com/DTDs/PropertyList-1.0.dtd">
<plist version="1.0">
<dict>
    <key>protocols</key>
    <array>
        <string>allowed protocols</string>
    </array>
    <key>styles</key>
    <array>
        <string>allowed CSS styles</string>
    </array>
    <key>tags</key>
    <dict>
        <string>allowed HTML tags and attributes</string>
    </dict>
</dict>
</plist>
```

The dict contains several key/value blocks that allow specific protocols, styles, tags, or attributes.

Allowing Specific URL Protocols

To allow specific URL protocols, add strings with the protocols' names to the protocols block.

For example, to allow entry of URLs that start with "irc://" and "scp://", create the following block:

```
<key>protocols</key>
<array>
    <string>irc</string>
    <string>scp</string>
</array>
```

Allowing Specific CSS Styles

To allow specific CSS style attributes, add strings with the style attributes' names to the styles block.

For example, to allow the use of the font-size style attribute, create the following block:

```
<key>styles</key>
<array>
  <string>font-size</string>
</array>
```

To allow the HTML “style” attribute, add the “style” attribute to the “_all_” tag, which is described in the following section.

Allowing Specific HTML Tags and Attributes

YouTube, Flash, and other advanced web formats usually require HTML tags and attributes that aren’t allowed by default. For example, YouTube requires the following tags and attributes:

Tag	Attributes
object	width, height
param	name, value
embed	src, type, width, height

Because these HTML tags or attributes aren’t included in the built-in whitelist, you must add them to the custom whitelist.

To allow specific HTML tags and attributes:

- 1 Create a tags key and follow it with a dict.
- 2 In the dict, create keys named after the HTML tags.
To allow a specific attribute for all HTML tags, create a key named “_all_”.
- 3 After each key, create string arrays listing allowed tag attributes.

For example, to support YouTube, create the following block:

```
<key>tags</key>
<dict>
  <key>object</key>
  <array>
    <string>width</string>
    <string>height</string>
  </array>
  <key>param</key>
  <array>
    <string>name</string>
    <string>value</string>
  </array>
  <key>embed</key>
```

```
<array>
  <string>src</string>
  <string>type</string>
  <string>width</string>
  <string>height</string>
</array>
</dict>
```

About the Built-in Whitelist

You can't change the built-in whitelist. However, a software update could change the allowed elements in the built-in whitelist.

Protocols Allowed in the Built-in Whitelist

The built-in whitelist allows these protocols:

Protocol
afp
feed
feeds
fax
ftp
gopher
http
https
itms
itpc
ldap
mailto
news
nfs
nntp
rdar
rtsp
sip
sips
sftp
smb
ssh
svn
svn+ssh

Protocol
tel
telnet
vnc
webcal
xmpp

CSS Styles Allowed in the Built-in Whitelist

The built-in whitelist allows these CSS styles:

Style
text-decoration
font-weight
font-style

HTML Tags and Attributes Allowed in the Built-in Whitelist

The built-in whitelist allows these HTML tags and attributes:

Tag	Attributes
all	class, title
a	href, name, ref
b	
blockquote	cite
body	
br	
caption	
dd	
div	
dl	
dt	
em	
h1	
h2	
h3	
h4	
h5	
h6	
html	
i	
img	src, alt, name, width, height, longdesc
li	

Tag	Attributes
node	
ol	
p	
pre	
q	cite
span	
strong	
table	
tbody	
tfoot	
th	colspan, rowspan
thead	
td	colspan, rowspan
tr	
u	
ul	

This chapter describes how to manage wiki content using tools and the command-line.

Using tools and the command line allows you to migrate, back up, and edit raw wiki content.

Migrating Wiki Content From Other Websites

AppleWikiImporter is a tool that can traverse a website that has a page listing all other pages in the site, and create wiki pages based on the website contents. If the website uses tags, AppleWikiImporter can import those tags.

To use this tool, you'll need some knowledge of CSS selectors, and, depending on the website you're importing, some knowledge of JavaScript.

AppleWikiImporter is located at:

<https://connect.apple.com/cgi-bin/WebObjects/MemberSite.woa/wa/getSoftware?bundleID=19994>

For information about using AppleWikiImporter, see its readme file.

Backing Up the Wiki Server

If you installed Mac OS X Server using a standard or workgroup configuration, you can use Time Machine to automatically back up the wiki server. If you installed Mac OS X Server using an advanced configuration and haven't changed the default wiki and calendar data store locations, Time Machine also automatically backs up your wiki server.

If you don't use Time Machine, the following information describes how you can manually back up the wiki server.

All wiki content is stored in a data store. The default location is `/Library/Collaboration/`. This location can be changed using Server Admin (in Web service).

To manually back up the wiki, copy the wiki data store with its permissions intact. For example, to copy the wiki data store to a folder on a different volume, enter the following in Terminal and then authenticate as an administrator when requested:

```
sudo ditto /Library/Collaboration/ /Volumes/volumename/foldername/
```

If you created or edited themes, copy the theme folder with its permissions intact. To copy the themes folder to a folder on a different volume, enter the following in Terminal and then authenticate as an administrator when requested:

```
sudo ditto /Library/Application\ Support/Apple/WikiServer/ /Volumes/  
volumename/foldername/
```

If you're using the web calendar, you must back up the calendar data store. The default location is `/Library/CalendarServer/Documents/`. This location can be changed using Server Admin (in iCal service).

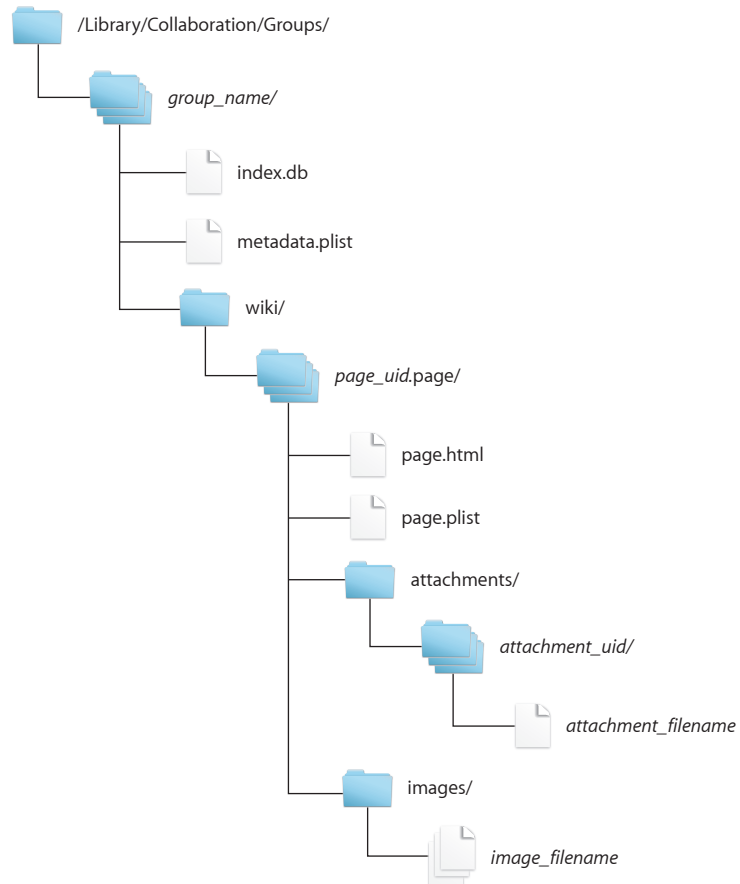
To manually back up the calendar, copy the calendar data store with its permissions intact. For example, to copy the calendar data store to a folder on a different volume, enter the following in Terminal and then authenticate as an administrator when requested:

```
sudo ditto /Library/CalendarServer/Documents/ /Volumes/volumename/  
foldername/
```

Wiki File Structure Overview

By default, the wiki server stores group wikis in `/Library/Collaboration/Groups/`.

The illustration below shows group wiki files and folders.



The Groups folder in the wiki data store includes the following critical wiki files and folders:

File or Folder Name	Purpose
<i>group_name/</i>	Enabling web services for a group creates a folder with the group's short name.
<i>group_name/index.db</i>	Automatically changed when you edit wiki content. If you manually edit raw content, erase this file so that it will regenerate based on raw content.
<i>group_name/metadata.plist</i>	Contains wiki administration settings.
<i>group_name/wiki/</i>	Stores all pages in the wiki. The blog uses a similar file structure. The blog stores all of its pages in <i>group_name/weblog/</i> .
<i>group_name/wiki/page_uid.page/</i>	Automatically created when you create a new wiki page. The name of the folder is a unique identifier that also appears in the page's URL: <i>http://domain_name/groups/group_name/wiki/page_uid/page_name</i>
<i>group_name/wiki/page_uid.page/page.html</i>	Contains the page's HTML content.
<i>group_name/wiki/page_uid.page/page.plist</i>	Contains meta-information for a page.
<i>group_name/wiki/page_uid.page/attachments/</i>	Stores subfolders for every attachment.
<i>group_name/wiki/page_uid.page/attachments/attachment_uid/</i>	Automatically created when a file is attached to a page. Every folder stores one attachment. The name of the folder is a unique identifier that also appears in the attachment's URL: <i>http://domain_name/groups/groupname/wiki/pageuid/attachments/attachment_uid/attachment_filename</i>
<i>group_name/wiki/page_uid.page/attachments/attachment_uid/attachment_filename</i>	This is an attachment.
<i>group_name/wiki/page_uid.page/images/</i>	Stores all images for a page.
<i>group_name/wiki/page_uid.page/images/image_filename</i>	This is an image.

Manually Editing Raw Content

When you use the wiki to edit wiki pages, the wiki server automatically indexes your content. If you manually edit raw content, the wiki server doesn't automatically index. You must delete the *index.db* file so that the file is regenerated based on the changed raw content; otherwise, an error occurs.

You can avoid this problem by using the Atom Publishing Protocol, also known as *AtomPub*. You can use an AtomPub-enabled client to log in, publish, update, or delete wiki and blog entries. Configure AtomPub clients to use this service address:
http://domain_name/groups/group_name/atompub

For more information about AtomPub, see:

Website	URL
RFC 4287: The Atom Syndication Format	atompub.org/rfc4287.html
RFC 5023: The Atom Publishing Protocol	tools.ietf.org/html/rfc5023

To regenerate the index after editing raw wiki content:

- 1 Delete `/Library/Collaboration/Groups/group_name/index.db`.

To delete this file, enter:

```
sudo rm /Library/Collaboration/Groups/group_name/index.db
```

Replace `group_name` with the group's short name.

- 2 Restart the web service by doing one of the following:

- Restart the server.
- In Server Preferences, click Web, move the slider to Off, and then to On.
- In Server Admin, open the disclosure triangle for your server, select Web, click Stop Web, and then click Start Web.
- Enter the following in Terminal:

```
$ sudo serveradmin stop teams; sudo serveradmin start teams;
```

If prompted, enter your administrator account password.

Restarting the server and restarting web service can disrupt users, so send out a notification that the server will be temporarily down before doing this.

Viewing a Wiki's Administration Settings

The wiki's admin settings are stored in:

```
/Library/Collaboration/Groups/group_name/metadata.plist
```

Replace `group_name` with the group's short name.

For information about viewing or editing property list files, see "Editing Property List Files," on page 14.

The metadata.plist file includes the following keys:

Key	Example Value	Description
GUID	28CC6BE9-11C1-456C-A3B2-8FCF5520A1DB	Uniquely identifies the wiki. This value must correspond to the group record's GUID in the directory domain. To view a group record's GUID, use Workgroup Manager. For more information, see <i>User Management</i> at www.apple.com/server/macosx/resources/ .
LongName	Wiki Name	Name of the wiki. By default, this name is displayed in the top left of every wiki page.
allowComments	<true/>, <false/>	Set to true if comments are allowed.
allowUnauthenticatedComments	<true/>, <false/>	Set to true if comments from anyone are allowed. Set to false if only comments from authenticated users are allowed.
allowUnmoderatedComments	<true/>, <false/>	Set to true if comments from authenticated users are unmoderated.
allowUnmoderatedUnauthenticatedComments	<true/>, <false/>	Set to true if comments from unauthenticated users are unmoderated
bannerImage	/groups/group_name/public/image_filename	Path to the banner image.
indexVersion	2	Don't change this value.
name	group_name	Short name of the group.
podcastCategory	Sports, Music	Name of the podcast category.
podcastEnabled	<true/>, <false/>	Set to true if podcasts are enabled.
sidebars	An array	Contains custom sidebar settings. For more information about setting up sidebars, see "Creating Sidebars," on page 18.
themeName	com.apple.block.green.banner	Set to the unique identifier of the theme.

Use the admin settings page to change these settings instead of manually editing the metadata.plist file. If you manually edit the file, you'll first need to stop web service or your changed settings will be automatically overwritten.

To manually edit wiki administration settings:

1 Stop the web service by doing one of the following:

- In Server Preferences, click Web and move the slider to Off.
- In Server Admin, open the disclosure triangle for your server, select Web, and click Stop Web.
- Enter the following in Terminal:

```
$ sudo serveradmin stop teams;
```

If prompted, enter your administrator account password.

Stopping web service can disrupt users, so send out a notification that the server will be temporarily down before doing this.

2 If you're going to use Property List Editor or a non-command-line tool, copy the theme.plist file to your Documents folder.

To copy this file, enter:

```
$ sudo cp /Library/Collaboration/Groups/group_name/metadata.plist /Users/  
user_name/Documents/
```

Replace *group_name* with the group's short name. Replace *user_name* with the short name of the user you're logged in as.

If prompted, enter your administrator account password.

3 Open theme.plist in Property List Editor or in a text editor.

To open this file in a command-line text editor (in this example, nano), enter:

```
$ sudo nano theme_name.wikitheme/theme.plist
```

If prompted, enter your administrator account password.

You can use any text editor, as long as you precede the command to run the text editor with `sudo`.

If you're using Property List Editor or a non-command-line tool, open the theme.plist located in your Documents folder.

For information about Property List Editor, see "About Property List Editor," on page 14.

4 Edit theme.plist and save your changes.

5 If you copied theme.plist to your Documents folder, copy it back to the theme folder.

To copy this file, enter:

```
$ sudo cp /Users/user_name/Documents/theme.plist theme_name.wikitheme/
```

If prompted, enter your administrator account password.

6 Start the web service by doing one of the following:

- In Server Preferences, click Web and move the slider to On.
- In Server Admin, open the disclosure triangle for your server, select Web, and click Start Web.
- Enter the following in Terminal:

```
$ sudo serveradmin start teams;
```

If prompted, enter your administrator account password.

Viewing a Page's Metainformation

A page's metainformation, such as its title, last author, and last modification date, are stored in:

```
/Library/Collaboration/Groups/group_name/wiki/page_uid.page/page.plist
```

Replace *group_name* with the group's short name. Replace *page_uid* with the unique identifier of your page.

Blog and mailing list archive pages also each have their own page.plist file.

For information about viewing or editing property list files, see "Editing Property List Files," on page 14.

The page.plist file includes the following keys:

Key	Example Value	Description
author	annejohnson	Short user name of the original author.
commentUID	groups/group_name/discussion/ page_uid	Unique identifier for the comments page and its location.
createdDate	2008-02-19T18:58:47Z	Original page creation date and time.
edited	<true/>, <false/>	Originally set to false, this is set to true the first time the page is saved.
kind	wiki, weblog, mailinglist	Type of page.
lastModifiedAuthor	ravipatel	Short user name of the last person to edit the page.
modifiedDate	2008-03-20T23:49:19Z	Last page modification date and time.
title	Welcome	Title of the page.

Key	Example Value	Description
tombstoned	<true/>, <false/>	Originally set to false, this is set to true when a page is temporarily deleted. If a page is permanently deleted, the entire <i>page_uid.page/</i> folder is removed.
uid	<i>groups/group_name/wiki/page_uid/</i>	Unique identifier for the wiki page and its location.
versioned	<true/>, <false/>	Wiki pages are versioned; blog and mailing list entries aren't.

After manually editing this file, delete the group's `index.db` file and restart the server. For more information, see "Manually Editing Raw Content," on page 40.

Managing Attachments

When a user attaches a file to a wiki page, that file is stored in:

/Library/Collaboration/Groups/group_name/wiki/page_uid.page/attachments/attachment_uid/

Group_name is the group's short name. *Page_uid* is the unique identifier of the page that the file is attached to and *attachment_uid* is the unique identifier of the file. Both unique identifiers are included in the attachment's URL:

http://domain_name/groups/group_name/wiki/page_uid/attachments/attachment_uid/filename

If someone deletes or replaces an attachment in the wiki, the old attachment remains on the server. This allows use of the wiki's history feature to restore old versions of attached files.

If you delete the folder containing the attachment on the server, users won't be able to restore the attachment.