Apple Help Programming Guide

User Experience > Help Technologies

2007-10-31

Contents

Chapter 3   **Help Book Registration   55**

Chapter 4   **Opening Your Help Book in Help Viewer   61**

Appendix A   **Apple Help Meta Tag Properties   67**

Appendix B   **Apple Help URLs   69**

Appendix C   **Apple Help Segments   71**

Document Revision History   73

Figures, Tables, and Listings

Introduction to Apple Help Programming Guide

> **Note:** This document supersedes the information in the older document *Providing User Assistance With Apple Help*, which has been moved to the Legacy Documents area of the ADC Reference Library (Reference Library > Legacy Documents > User Experience). This document describes development of Apple Help for Mac OS X v10.4 and later. For information on providing Apple Help for earlier versions of the operating system, see *Providing User Assistance With Apple Help*.

This document describes Apple Help, the HTML-based system for providing user assistance in Mac OS X. Apple Help is the primary help system for Mac OS X and is designed to deliver online topic-based user help, such as is often provided in user manuals and lists of frequently asked questions (FAQ). Carbon, Cocoa, and Java applications can use Apple Help in Mac OS X.

> **Note:** In addition to Apple Help, Mac OS X includes another help technology, help tags. Help tags, also known as tooltips, are short contextual help messages that appear onscreen when the user hovers the pointer over an element in an application's user interface. Help tags are documented in *Providing Help Tags in Carbon*. You can use Interface Builder to add help tags to Carbon or Cocoa applications. In Cocoa Interface Builder, help tags are referred to as tooltips. In Carbon, they're referred to as Help and Extended Help (displayed by holding down the Option key).

Apple Help offers significant advantages over static help documents, such as Read Me files or manuals in PDF. The benefits of adopting Apple Help for user assistance include these:

■ Searchability. Apple Help offers users sophisticated search capabilities, including full-text searching and the ability to search on synonyms and common misspellings.

■ Full support for QuickTime and any other media for which plug-ins are installed. Using QuickTime or other authoring tools, you can create animated sequences that showcase hidden or complex features of your software product and play these sequences as part of your help content.

■ AppleScript automation. Using AppleScript, you can automate tasks and run them from your help content to guide users through a complex operation step by step.

■ Ease of updating. Apple Help makes it simple to revise and expand your help content, page by page or all at once. Using Apple Help, you can even maintain up-to-date help pages and search indexes on your own server and have them downloaded via the Internet to update your help content.

■ Ease of adoption. Many developers have recognized the advantages of browser-based help and implemented HTML solutions to provide user assistance. Apple Help makes it easy for you to adapt previously created HTML pages into the form used by Apple Help.

■ User experience. The Help Viewer application is optimized for delivering onscreen help. By using Apple Help, you'll ensure the user experience for your application is consistent with users' expectations of a Mac OS X application.

■ Spotlight For Help. Starting with Mac OS X v10.5, Apple Help integrates Spotlight For Help, which lets users search the contents of your menus and application help directly from your application's Help menu. If you don't use Apple Help, then Spotlight For Help returns results only from Mac Help rather than from your application's help.

When you use Apple Help, you can supply HTML-based user assistance and integrate it into your application with relatively little effort. Apple Help manages and displays **help books**; a help book is the collection of HTML files that constitute the user help for your software product, plus a help index file generated by the Help Indexer utility. When you supply a help book and register it with Apple Help, users can access your help from your user interface and view it in Help Viewer without any additional work on your part.

The Apple Help system includes these components:

■ The Help Viewer application. This is the default application for viewing user assistance in Mac OS X. Help Viewer displays your HTML-based help book.

■ The Apple Help application programming interface (API). This is a set of functions provided by Apple Help that allow you to access and load help in Help Viewer. You do not need to use these functions if you are providing only a basic help interface; however, if you wish to implement advanced help features (such as contextual help), you need to call the Apple Help functions. The Apple Help API is available to all Carbon, Cocoa, and Java developers.

■ The Help Indexer utility. This is a developer tool provided by Apple for indexing your help book. When you run the Help Indexer utility on your help book, the tool generates an index file that Help Viewer uses to make your help searchable.

Who Should Read This Document?

If you are creating an application, plug-in, or other software product for Mac OS X with a user interface, you should read this document to learn how to create an Apple Help help book and display it in Help Viewer.

Organization of This Document

This document includes the following chapters and appendixes:

■ "Apple Help Segments" (page 71) lists the commands you can use to divide HTML help pages into multiple segments.

Availability

Help Viewer and the Apple Help API are available in Mac OS X v10.0 and later. The Help Indexer tool is available in Mac OS X v10.4 and later in `/Developer/Applications/Utilities` when the Developer package is installed.

See Also

For more information about help technologies, you can refer to these other documents in the ADC Reference Library (http://developer.apple.com/referencelibrary/index.html):

■ For a detailed description of the Apple Help application programming interface, see the *Apple Help Reference*.

■ For information on Carbon help tags, see *Providing Help Tags in Carbon* and the *Carbon Help Manager Reference*.

■ For information on help tags, or tooltips, in Cocoa applications, see *Online Help* and *Providing Help Tags in Carbon*.

■ For guidelines on how to use help effectively within your application, see "User Assistance" in the Using Mac OS X Technologies chapter of *Apple Human Interface Guidelines*.

See Also

Apple Help Concepts

This chapter introduces the Help Viewer application and the Apple Help application programming interface. It describes the Help Viewer interface, how Help Viewer displays your help book, and how users access help from your application. All Carbon, Cocoa, and Java developers authoring user help for a Mac OS X application should be familiar with the concepts presented here.

Help Viewer

Users view online help in Help Viewer, a browser-like application designed to display HTML help content. Help Viewer displays files adhering to the HTML 4.01 specification. In addition, Help Viewer supports QuickTime media and AppleScript automation.

The Help Viewer Window

Users typically launch Help Viewer by choosing the application help item from the Help menu, or by typing a query in the Spotlight For Help text field in the Help menu (see "How Users Access Your Help" (page 17)). When Help Viewer launches, it brings up a window displaying help for the application from which the user requested assistance. In Figure 1-1, the Help Viewer window displays the system help book, Mac Help.

Figure 1-1   The Help Viewer window
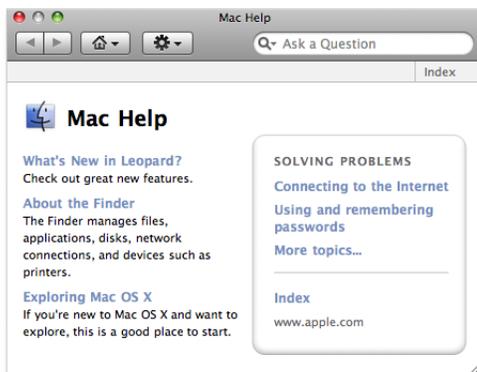
There are four user interface elements in the toolbar at the top of the Help Viewer window: the Back and Forward buttons, the Home button, the Action menu button, and the search field. The Back and Forward buttons on the left side of the toolbar allow users to view their navigation history and navigate through previously visited help pages. Clicking the Home button opens the landing page of the currently open help book; if the user presses this button, a menu of all the available help books appears. The Action menu provides items such as Make Text Larger (or smaller), Find (which finds text strings on the current page), and Print. The search field allows users to search all help in the current book or all help on the system—they click the

magnifying glass at the left of the search field to choose which. The magnifying-glass menu also allows the user to choose whether to include product support searches—that is, search results from Apple's Knowledge Base support web site or from your own support site.

Searching in Help Viewer

One of the primary advantages of Help Viewer for viewing online help is its ability to quickly and accurately search an installed set of help content. Users often arrive at online help with an idea of what they want to accomplish; help should allow them to get the information they need as quickly as possible and get on with their tasks. Especially for large help systems, searching is often the most efficient and effective way for users to obtain help.

The text entry field at the right side of the Help Viewer window's toolbar allows users to search the available help content on the system. Users enter the term or concept for which they want to obtain help into the text field. Figure 1-2 shows a query entered in the text field of the Help Viewer window, with Search All Help selected.

Figure 1-2   A query entered in the search field of Help Viewer
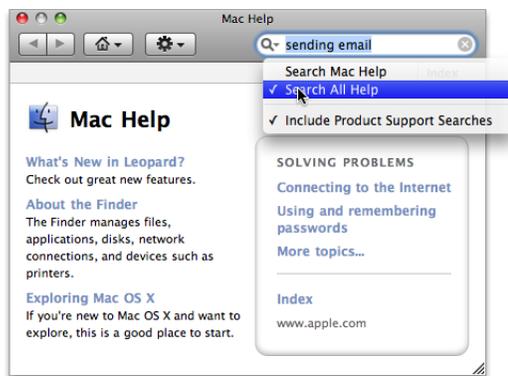
When the user presses Return, Help Viewer searches the help books installed on the system (if the user selected Search All Help) or in the selected help book (if they selected that option) for the relevant term or terms. It returns up to 25 help topics and up to 25 support articles (if the user selected that option) from the AppleCare website or from your support site (if you've set up support site searching—see "Providing Your Own Online Support Articles" (page 52)). Figure 1-3 shows the search results returned by Help Viewer in response to the query entered in Figure 1-2. Help Viewer sorts the help topics in order of relevancy (as determined by Search Kit). Support articles are listed alphabetically by title.

Figure 1-3   Search results displayed in Help Viewer
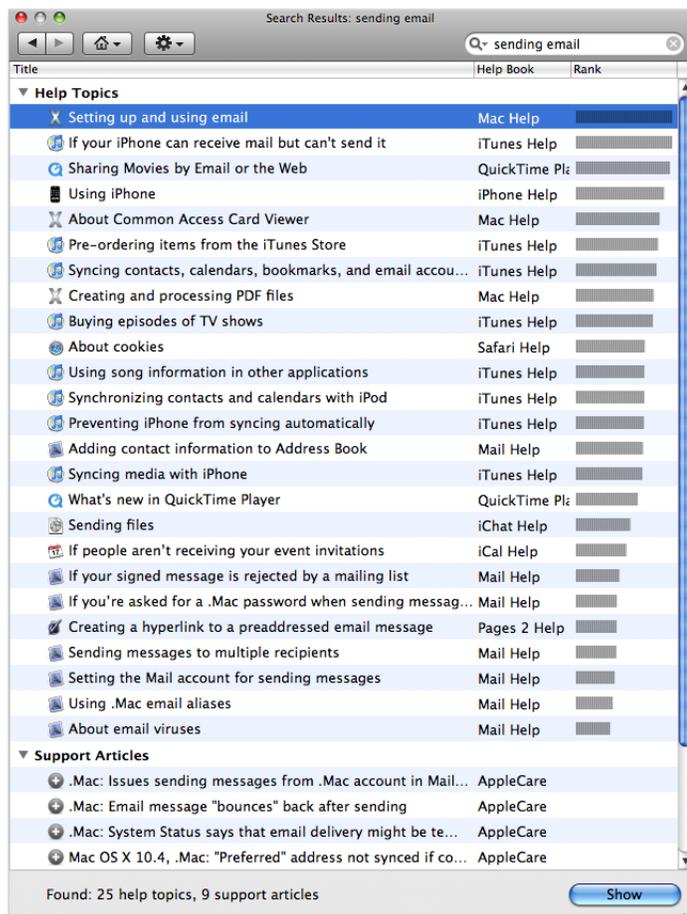
Help Viewer displays the title of each relevant help topic in a table of search results, along with the help book in which that topic is found (if the user selected Search All Help). When the user moves their cursor over a topic from the search results, an abstract of the help topic appears, if available. The user can view the selected topic by double-clicking the topic in the search results table or by clicking the Show button at the bottom of the Help Viewer window. Figure 1-4 shows the abstract for a help topic returned as a search result for the query entered in Figure 1-2 (page 12). For information on how to provide abstracts for help topics, see "Adding Abstracts" (page 37).

Figure 1-4   A topic abstract displayed for a search result in Help Viewer
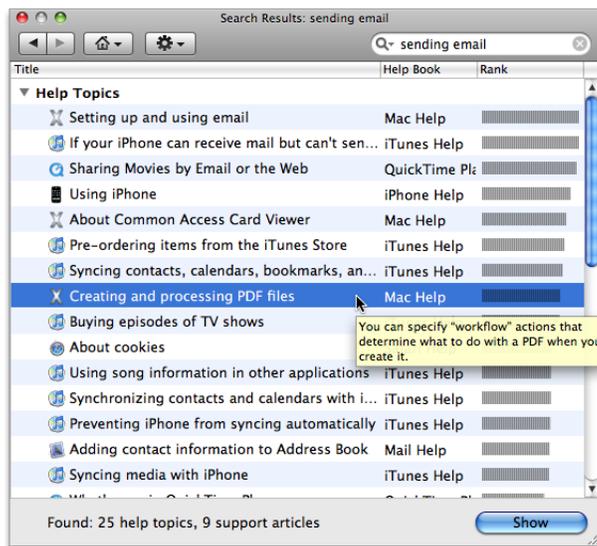
To ensure that certain queries return the most relevant results regardless of the search term's occurrence in the text, you can provide a list of search terms and corresponding search results to be displayed. This is referred to as *exact match* searching. You can use an exact match search list, for example, to provide responses to search terms too short to normally be used in a search (such as "CD"), to make sure that a particular help topic is included in the responses when there are likely to be more than 25 matches to a query, or to make sure that the most relevant results receive the highest relevancy rating regardless of the algorithm used to rank results. Exact match searches are described in more detail in "Setting Up Exact Match Searching" (page 51).

Cross-References and Index Lists

There are Apple Help–specific URLs that you can use to create links to help pages in your help book. You can create individual links to specific locations in your book, and you can generate lists of links to use for index pages or for "see also" lists at the end of help topics. See "Apple Help URLs" (page 69) for a list of these URLs and cross-references to their descriptions in this document.

The Library Menu

The Library menu lists all of the help books currently registered on the user's system. A help book is registered when an application calls the `AHRegisterHelpBook` function during application startup or (if the help book is listed in the application's `Info.plist` file) when the user chooses application help from the Help menu or from a help button.

With the Library menu, users can easily access and browse all of the help available to them. The Library menu is accessed by pressing the Home button, as shown in Figure 1-5.

Figure 1-5   The Library menu

Because the Library menu is available to the user at any time, no matter what help book they are currently viewing, users can conveniently switch between help books without switching applications. When the user chooses a help book from the Library menu, Help Viewer loads that help book.

Help Books

To display help in Help Viewer, you must create and register a help book. A help book is the collection of HTML files that constitute your help content plus a help index file. In addition to HTML files, help books can contain graphics, AppleScript scripts, QuickTime movies, and other resources used in the help pages. A help book can be simple or complex, depending upon the complexity of the software product it describes. In addition to any help content, a help book must contain these two items:

- A default, or title, page. This is the XHTML page that is displayed by default when Help Viewer first opens the help book. The title page is identified by the `AppleTitle` meta tag in its header. The `AppleTitle` meta tag is one of the help-specific meta tags described in "Help-Specific Meta Tags" (page 19). Without this page, Help Viewer cannot locate and automatically access the help book. Title pages are described in more detail in "Creating a Title Page" (page 32).

- An index file. For a help book to be searchable, it must have an index file. You can create index files using the Help Indexer utility, described in "Indexing Your Help Book" (page 35).

All of the content referenced by your help book must reside in a single folder, although a help book folder can contain any number of subfolders. If you localize your help book, you need a folder for each language—see "Localizing Your Help Book" (page 53) for more information.

> **Note:** If you keep all or portions of your help content on a remote server, the help folder you store locally need not contain all of the content used by your help book. For more information, see "Storing Pages on Remote Servers" (page 42).

When you register your help book, Help Viewer locates your help book folder, searches the folder for the title page and index file or files for your help book, and caches the location of those files. When users choose your help book in the Library menu or choose the application help item from the Help menu in your application, Help Viewer opens the title page of your help book. When users enter a search in Help Viewer, Help Viewer searches the index files in your help book and displays the relevant results in the table view shown in Figure 1-3 (page 13).

> **Important:** You must register your help book for it to appear in the Library menu, be searchable in Help Viewer, and be accessible through the Apple Help API. For information on registering a help book, see "Help Book Registration" (page 55).

Internet-Based Help Book Content

Help Viewer supports Internet-based help book content: You can store help pages on a remote server and Help Viewer downloads them as needed. This allows you to keep your help content up to date easily and without inconvenience to your users. There are several strategies you can use to provide help:

- Internet-only: Only a small percentage of the pages are installed locally. If the Internet isn't available, the help is not available. This strategy is used, for example, by Mac OS X Server Help.

- Internet-primary: Most or all help pages are installed locally, but Help Viewer checks the server to determine whether a newer version of each page is available before displaying the page. If the server isn't available, or if the page has not been updated, then the local version is used. This strategy is used by Mac Help and most Apple application help books.

- Local-primary: Local pages are used if available. If the requested page is not available locally, then Help Viewer looks for it on the remote server.

When you index your help book, you specify the server from which Help Viewer should retrieve content. You also specify whether Internet-based help pages should take priority over local help pages. When the user opens your help book, Help Viewer loads your title page. When the user follows a link to another page, Help Viewer checks the index to see whether you specified local-primary or Internet-primary help. If you specified local-primary help, Help viewer loads the page in the local help book folder. If the page is not present in the local help folder, then Help Viewer contacts the server specified in the index and downloads the page. On the other hand, if you specified Internet-primary help, Help Viewer checks first for content on your remote server and downloads the page from there. If the page is not present on the server, but is present locally, then Help Viewer loads the page from the local help book folder.

Local-primary help has the advantage of loading quickly while still allowing you to add help content by providing new topics on an Internet server. Using Internet-primary help, you can update help content at any time without requiring the user to install an updated version of the help book, while still providing a complete help book locally in case the user is not connected to a network. With Internet-only help, the user must be connected to the Internet to access help. This method is most appropriate for applications that must be connected to the Internet in order to function.

Help Viewer also supports updating indexes over the Internet.

To learn more about supplying content remotely, see

How Users Access Your Help

Users access your application's help book— and other help resources—in any of the following three ways:

■ Choosing the application help item from the Help menu or typing a query into the Spotlight For Help text field in the Help menu. This is the most visible means of accessing user help, and your application must provide Help menu access to its help book. If you register your help book, the system automatically makes your help book accessible from the Help menu and automatically indexes your help book in Spotlight For Help.

■ Clicking a help button. Where appropriate—for example, in a dialog requesting a user action—your application may supply a help button. Clicking this button should bring up a relevant page in your application's user help.

■ Choosing a help item from a contextual menu. If contextually relevant help exists for a part of your application's user interface, the first item in a contextual menu should be a help item. If the user chooses the help item, your application should bring up the relevant help.

The Help Menu

Users access your application's help book and other help resources through the Help menu, normally the rightmost menu in the application region of the menu bar. Although your application may also provide additional means of obtaining help, such as contextual menu items or help buttons, the Help menu is the most obvious means of obtaining assistance for the majority of users. Because the Help menu is easily visible, remains in a consistent location, and is always accessible, it is the first place users go when they have a question. Help menu items are not context-sensitive.

Figure 1-6 shows the Help menu in Mail.

Figure 1-6   The Help menu
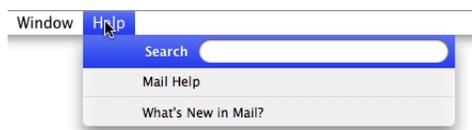
The Help menu is supplied by the system. If you have registered your help book, the system not only creates the Help menu, it also adds an application help menu item and opens Help Viewer to the home page of your application's registered help book when the user chooses your application help. In Mac OS X v10.5 and later, the user can also use the Spotlight For Help search field to search the content of your help book. When they do so and choose one of the returned search items, the system opens the appropriate page in your help file.

> **Note:** If you do not register your help book, you need to add an *ApplicationName* Help item to the Help menu, where *ApplicationName* is the name of your software product. When the user chooses this item from the menu, you should open Help Viewer to the first page of your application's primary help book. In this case, Spotlight For Help cannot return items from your help book in response to a query, and the user will see only choices returned from Mac Help and other help books on their system.

In addition to your application's primary help book, you may want to include in the Help menu items for other help resources, such as links to support or customer service websites or tutorials.

Help Buttons

A help button is a small round button displaying the standard help icon—a question mark. This button is available in Interface Builder.

When contextually relevant help is available, you can use a help button in a window or dialog to take users directly to the pertinent information. For example, you can use a help button in a Preferences window to take users directly to information about setting preferences for your application. The button is customarily placed in the lower-right corner of the window when there are no other buttons in that location and in the lower-left corner otherwise.

Figure 1-7 shows a typical help button, in the lower-right corner of a Mail preferences window.

Figure 1-7   A help button
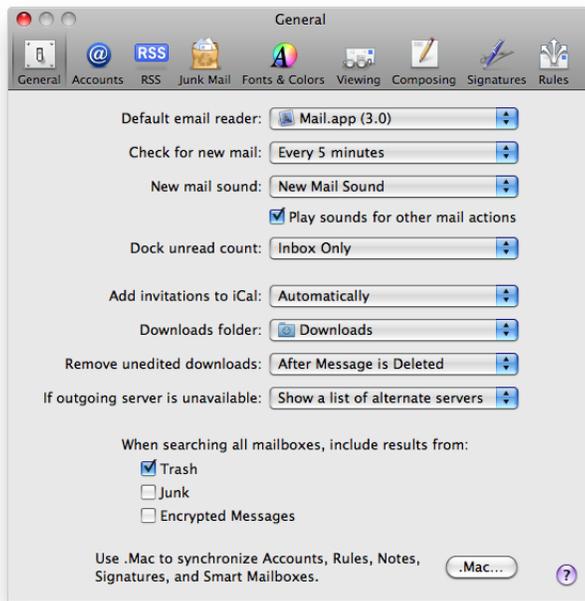
You should use help buttons only when contextually relevant help exists for the current window or dialog. Help buttons should allow users to obtain assistance for the task at hand without searching through your help book.

When the user clicks the help button, your application should load the relevant help topic in Help Viewer, using the Apple Help function `AHLookupAnchor`. See "The Apple Help Application Programming Interface" (page 20) for more information on using this function.

Help Viewer HTML Items

Help Viewer recognizes a number of help-specific items in the HTML file, which you can use in your help book to control how Help Viewer displays and accesses your help. These items fall into three categories:

- Metadata. Help Viewer uses this metadata to recognize your help book and determine how it should be displayed.

- Help URLs. Help Viewer handles a number of help-specific URLs, which you can use to load particular help content in Help Viewer.

- Segment commands. These are HTML comments that Help Viewer interprets to mark and identify subsections within an HTML page.

Help-Specific Meta Tags

Help Viewer recognizes a set of properties for the standard HTML `meta` element, which you can use to control how your help book appears in the Library menu and in search results. You can also use these meta tags to control how your help is indexed. "Authoring Apple Help" (page 23) describes many of these meta tags and how you can use them. For a complete list of the help-specific metadata recognized by Help Viewer, see Table A-1 (page 67).

> **Note:** Most of the metadata recognized by Help Viewer is optional. However, you must include the `AppleTitle` meta tag on one page in the root of your help book for Help Viewer to properly identify and display your help book.

Help URLs

There are several URLs, using the Help Viewer `help:` protocol, that you can use in your help book to create links to additional help content. You use the standard `a href` syntax for a source link with these help URLs. Although you can use help URLs to link to HTML help pages, you can use standard hyperlinks for that purpose as well. The main advantage of the help URLs is that they allow you to create hyperlinks that, when clicked, initiate searches in Help Viewer, run AppleScript scripts, and perform anchor lookup.

The help URLs are listed in Table B-1 (page 69). That table includes cross-references to discussions of these URLs in this document.

Apple Help Segments

Help Viewer also recognizes certain HTML comments that you can use to split your help book into subsections, called segments. Segments can be returned as individual hits in a search, each with its own description, title, and search keywords. You may find it useful to use segments if you have a particularly large help book and

wish to reduce the file count without loading too much help content into a single help page. "Creating Segments in Help Files" (page 39) describes segments in more detail. See Table C-1 (page 71) for a complete list of the comments denoting segments.

VoiceOver Summaries

To support readers with visual disabilities, the W3C document *HTML Techniques for Web Content Accessibility Guidelines 1.0* (http://www.w3.org/TR/2000/NOTE-WCAG10-HTML-TECHS-20001106/) defines a `summary` attribute for the `table` element. You can use this, for example, to provide VoiceOver with a description of the relationships among cells in a table. The Help Viewer application supports the `summary` attribute.

The Apple Help Application Programming Interface

The Apple Help application programming interface, declared in the `AppleHelp.h` header file in the Carbon framework, allows you to programmatically access and load help pages in Help Viewer. You can call this interface from Cocoa as well as Carbon applications. When users choose an item from the Help menu, click a help button, or choose Help from a contextual menu, your application must respond by displaying the appropriate help material. If this help material is in an Apple Help book, your application must open the relevant page of the help book in Help Viewer.

> **Note:** If your help book is registered, the system opens the book in Help Viewer for you when the user chooses your application's help from the Help menu. However, you are responsible for providing access to help through help buttons, contextual menus, or additional Help menu items. To register your help book with Apple Help, use the Apple Help function `AHRegisterHelpBook`, as described in "How to Register Your Help Book" (page 57). The `NSHelpManager` equivalents to `AHLookupAnchor` and `AHSearch` (see the following table) call `AHRegisterHelpBook` for you.

The Apple Help functions that open a help book in Help Viewer are listed in Table 1-1.

Table 1-1   Apple Help functions for accessing your help book

Function name	Action	NSHelpManager equivalent
`AHLookupAnchor`	Opens your help book to the section or page identified by the given anchor.	`openHelpAnchor: inBook:`
`AHSearch`	Searches your help book for a given text string.	`findString:inBook:`
`AHGotoPage`	Opens a help book page at a known location.	none

Whereas the `AHGotoPage` function requires that you know the full or partial path to the HTML file describing the desired help topic, `AHLookupAnchor` allows you to access a help topic with only the anchor name. In most cases, using an anchor is easier and more flexible than tracking the location of the file describing the topic. If there is not one particular topic that you wish to load in Help Viewer, you can instead use the `AHSearch` function to search your help book for all topics containing a particular string.

The Apple Help functions are described in detail in the *Apple Help Reference*. `NSHelpManager` methods are described in *NSHelpManager Class Reference*.

To learn how you can use the Apple Help functions to access your help book content from within your application, see "Opening Your Help Book in Help Viewer" (page 61).

Authoring Apple Help

This chapter describes how to author help content for Help Viewer and organize it into a help book. Anyone authoring user help for Mac OS X should be familiar with the basic requirements of creating a help book and with the general guidelines for writing user help.

These are the basic steps for creating a help book for Mac OS X:

1. Design the help content.

2. Author the HTML help pages.

3. Organize the help book. This includes creating the necessary auxiliary files that Help Viewer uses.

4. Index the help book.

In addition, this chapter describes how you can include additional content, such as QuickTime movies and AppleScript scripts, in your help book and how you can localize your help book for other languages.

Designing a Help Book

The first steps in authoring your help book are identifying the topics your help must cover and designing a layout for presenting these topics. To this end, you may find it useful to create a topic outline. If the software product for which you are creating help already has existing documentation, you may be able to base your outline on this material. If you are creating a help book from scratch, there are a number of ways you can approach the outline. A few examples:

- Walk through the steps of the main task sequence in order. If you are writing help for a larger application, there may be several different task sequences a user would perform. For example, a productivity suite may have different task sequences for word processing, using spreadsheets, and creating presentations.

- List topics alphabetically.

- Go through each menu and menu item in the application sequentially.

Each topic should be simple enough to be described in a few short paragraphs on a single HTML page. If a topic is lengthy, you should consider breaking it up into smaller subtopics.

Here are some tips to keep in mind when designing your help book.

- Divide the information into overview information and tasks. Overview information defines terms and explains concepts important to an understanding of your software product; task information gives step-by-step directions for accomplishing a particular goal. You should generally place these two kinds of information on separate help pages to give users quick access to the information they want. You can

link between pages containing overview and task information when appropriate. Avoid including "feature-oriented" pages, which describe application features but don't tell users what they can do or how.

■ Identify any information you think you'll need to give users more than once in a help book. You can write an individual help page to cover this information and link to it from other topics in the book to avoid duplication.

■ Build pages around four central questions:

❏ What can users do?

❏ Why do they want to do it?

❏ How can they do it?

❏ How can they solve problems doing it?

Depending upon the complexity of the task, a well-designed help page may cover each of the first three questions in one or two sentences and the fourth in two or three bullet points.

Authoring Help Pages

Once you have identified the subjects covered in your help book, you need to create HTML files for your help pages. To ensure that your help displays properly in Help Viewer, your help files should comply with the HTML 4.01 specification. Your main file—which contains the `AppleTitle` meta tag—should conform to the XHTML 1.0 specification. For a comprehensive description of the HTML 4.01 specification, see *HTML 4.01 Specification*, W3C Recommendation 24 December 1999 (http://www.w3.org/TR/1999/REC-html401-19991224/). XHTML is described in *XHTML 1.0 The Extensible HyperText Markup Language (Second Edition)*, W3C Recommendation 26 January 2000, revised 1 August 2002 (http://www.w3.org/TR/2002/REC-xhtml1-20020801/).

> **Compatibility Note:** If your help book is used only on Mac OS X v10.4 or later, use HTML 4.01 and use XHTML for the main page that contains the `AppleTitle` meta tag. Help Viewer references this file often and formatting it as XHTML improves performance. If your help book is also used on Mac OS X v10.3 or earlier, however, use HTML 3.2 and do not use XHTML for the main page. If you use XHTML in that case, your help book will not be indexed properly. For details about writing Apple Help for older versions of Mac OS X, see *Providing User Assistance With Apple Help* in the Legacy Documentation section of the Apple Developer Connection Reference Library.

For an example of a help book to use as a starting point, see the files for Mac Help in `/Library/Documentation/Help/MacHelp.help/Contents/Resources/`.

Authoring Tools

You can author your help book in any application that generates valid HTML 4 files. Likewise, you can view your help book in any HTML 4–qualified browser; however, you should always test your help book in Help Viewer to ensure that your help book displays properly.

Creating Topic Pages

Each help page should cover only one topic, which can be expressed in a few short paragraphs. As mentioned in the section "Designing a Help Book" (page 23), your help book may contain both overview and task information. Overview pages define terms and concepts important to your application or offer other general information that users may need to know to understand your software product. For example, the help book page shown in Figure 2-1 describes application menus.

Figure 2-1   A help book page containing overview information
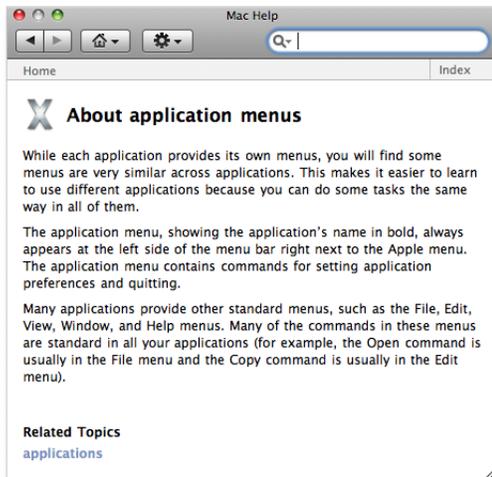
Task pages, on the other hand, offer a step-by-step description of the actions the user must take to perform a common task in your software product. The help book page shown in Figure 2-2 describes the steps necessary to change the background of a Finder window.

Figure 2-2 A task-oriented help book page
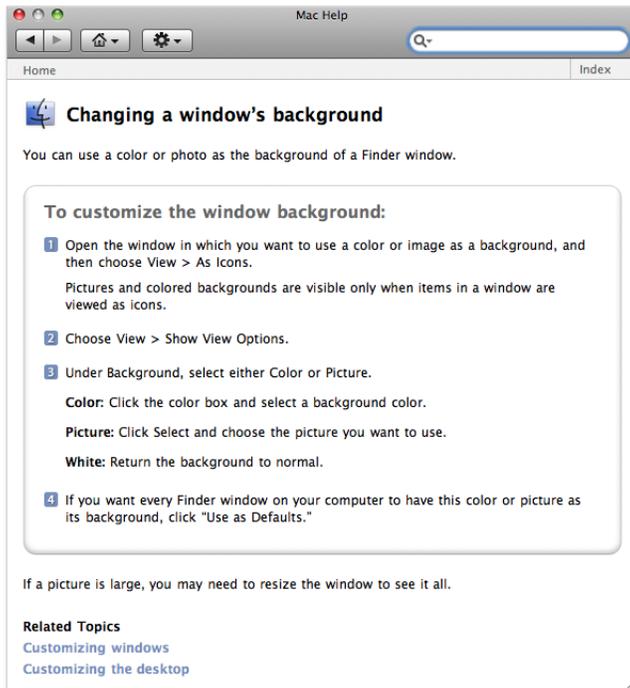
Topic pages typically include these elements:

- A title identifying the topic. In Figure 2-1 (page 25), "About application menus" identifies the topic and indicates that the topic is informational, not task oriented. In Figure 2-2 (page 26), "Changing a window's background" indicates (by beginning with a verb) that the topic describes a procedure or task.

- The topic introduction. For an overview page, this section describes what the user will learn about by reading this page. For a task-oriented page, the introduction indicates what the user will accomplish by performing the task.

- Requirements for performing the task. For a task page, any conditions that must be met in order for the task to succeed should be mentioned up front, before the user begins the task. For example, if the help topic is "Burning a CD," the system requirements—such as the presence of a CD burner—for burning the CD should be mentioned here.

- The task description. These are the steps that the user must perform to accomplish the given task. Overview pages typically do not contain this information.

- The topic wrap-up. This includes any information the user may need in order to wrap up any task described in the page. It is also a good place to include tips, shortcuts, troubleshooting information, and links to related help topics. For example, the last paragraph shown in Figure 2-2 (page 26) gives a hint the user might need in order to see a large picture in their Finder window.

- Related topics. At the end of a topic is a list of links to other topics that are related to this one and thus might be of special interest to the user.

Printing A Page's URL

While debugging a help book or examining a help book to learn how it was made, it can be difficult to figure out which file is the source of the page being displayed. To find the URL of the page of a help book, execute the following command in the Terminal application:

```
defaults write com.apple.helpviewer PrintURLInFooter YES
```

This command causes every page's URL to be listed in the footer when the page is printed. The onscreen version of the page does not change. Figure 2-3 shows a printed help page with a URL in the footer.

Figure 2-3   URL printed in footer
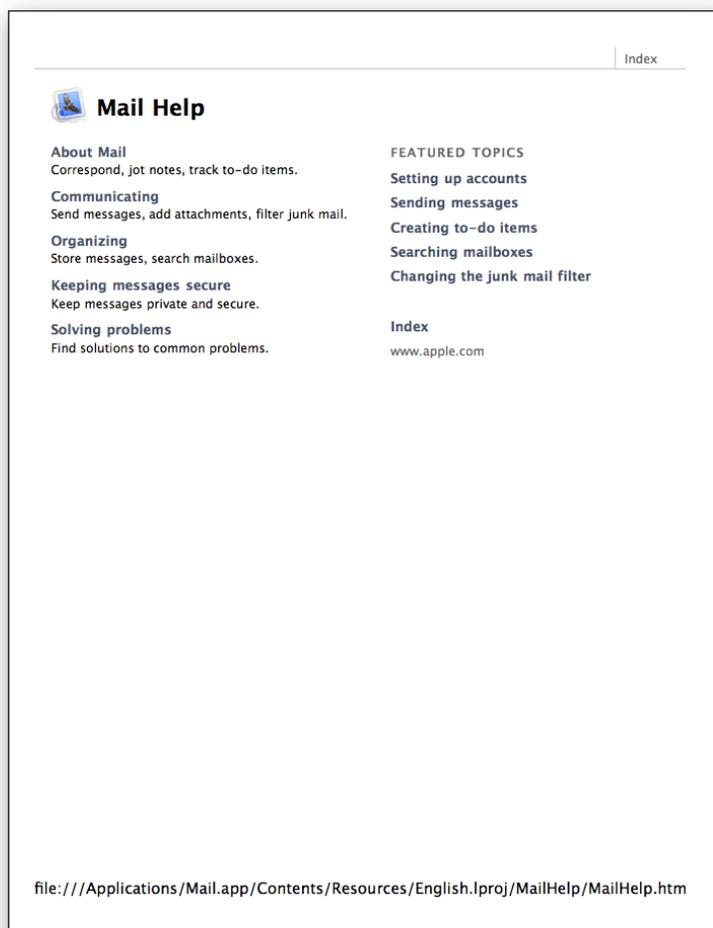
Creating Navigation Pages

In addition to topic pages, you may need to create navigation pages for your help book. Users should be able to find most of the information they need by searching and navigating through links in your topic pages. However, navigation pages, such as tables of content, allow users to browse your help book and navigate to topics they want to learn more about without having a particular search topic in mind. You may consider providing a table of contents at the following levels:

- Top level

- Chapter level

- Topic level

Including a table of contents on the title page, at the top level of your help book, allows the user to select a starting point within your help book. A title-page table of contents gets the user started in finding help, even if they do not quite know what they are looking for. Figure 2-4 shows the top-level table of contents for the Mail application.

Figure 2-4 The title-page table of contents for Mail Help
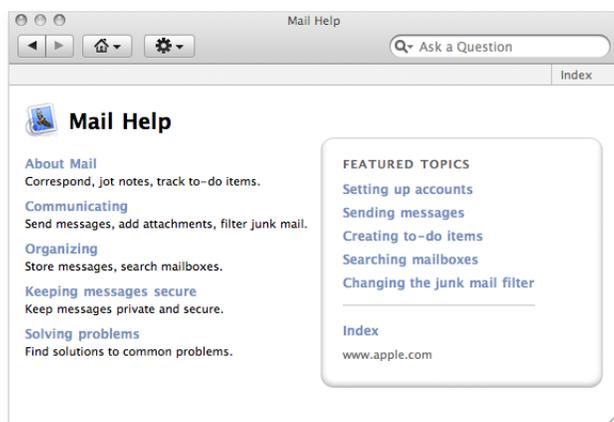
As mentioned in "Designing a Help Book" (page 23), you should break complex topics with lengthy descriptions into smaller subtopics in order to keep each help topic short and focused. However, it may not be appropriate to include all of the subtopics directly in your main table of contents.

You can create navigation pages that contain links to sets of related subtopics. Figure 2-5 shows a high-level TOC page from Mac Help. If the user clicks one of the topics, a list of subtopics appears, giving information about each (Figure 2-6 (page 30)). By clicking one of those subtopics, the user can get an information or task page for that subtopic (Figure 2-7 (page 30)). Typically, this page also contains links to further information about this subtopic and to pages for related subtopics.

Figure 2-5   A high-level table of contents in Mac Help
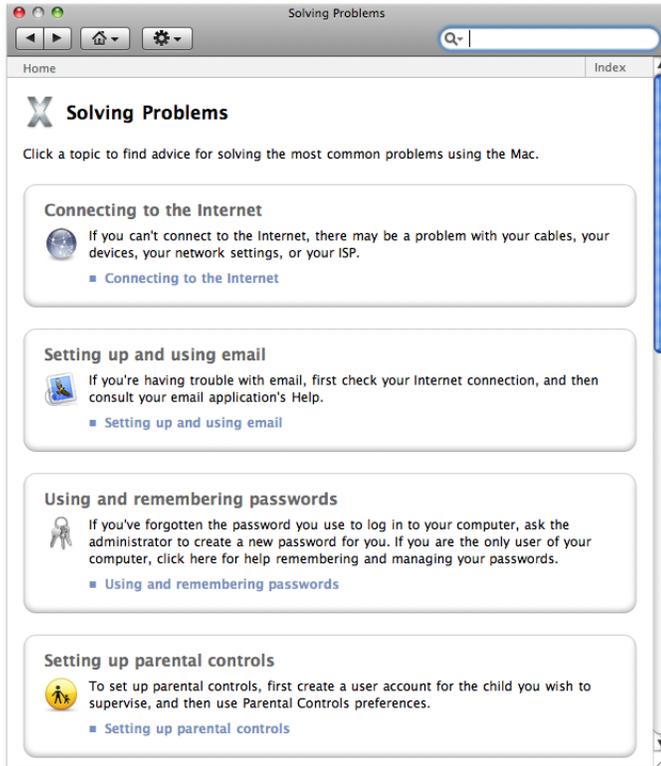
Figure 2-6   A subtopic-level table of contents in Mac help
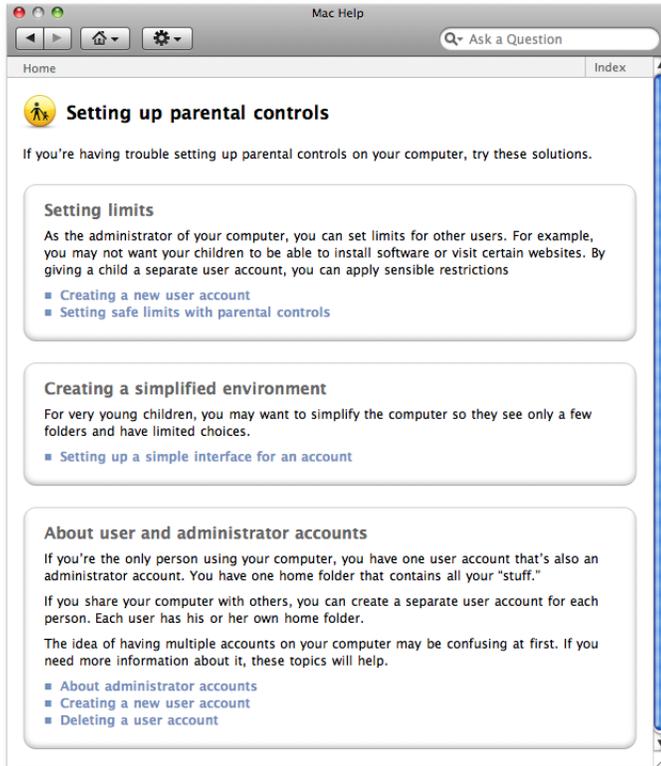
Figure 2-7   A task page for a subtopic in Mac help
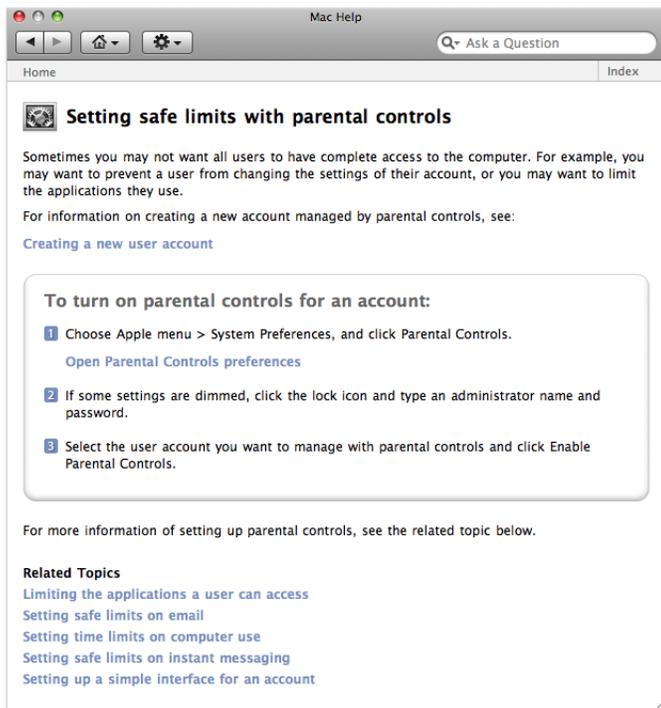
Creating a Basic Help Book

Once you create the HTML files containing your help content, you must organize them into a help book. To do this, create a help book folder and include the following items:

■ The topic and navigation pages. These are the HTML pages that you created for your help content, as described in "Authoring Help Pages" (page 24).

■ A title page (sometimes called a start page). This is the XHTML file that is displayed by default when the user first opens your help book.

■ A help book icon. This icon is displayed next to your help book in the Library menu.

For an example of a help book to use as a starting point, see the files for Mail Help in `/Applications/Mail.app/Contents/Resources/English.lproj/MailHelp/`. (Note that you have to select Show Package Contents from the contextual menu to see the contents of the Mail.app bundle.)

Organizing the Help Book Folder

Every help book must be enclosed in its own folder. For a simple help book, Apple recommends that you also create separate subfolders for your HTML help files and graphics. Figure 2-8 shows an example of a simple help folder for SurfWriter Help.

Figure 2-8   An example of a simple help book folder structure

At the top level of the SurfWriter Help folder are the index file and the title page. The book's icon and other graphics files are in the gfx folder. The pgs folder contains the HTML pages for all the help topics. The shrd folder contains AppleScript and JavaScript scripts. The sty folder contains style sheets. Note that this is an example of a typical help folder, not a prescription for how you need to organize your help book. Only the index file, the title page, and the exact match property list file (if any—see "Setting Up Exact Match Searching" (page 51)) should be located at the top level of the help folder. (For chapter-based help books, each chapter can have its own index file—see "Organizing a Chapter-Based Help Book Folder" (page 33)). For performance reasons, there should be only a single HTML file at the top level. What other files and folders you use are entirely up to you.

The title page is described in the next section, "Creating a Title Page." "Specifying a Help Book Icon" (page 32) describes how to add an icon to your help book. Index files are discussed in "Indexing Your Help Book" (page 35).

Creating a Title Page

The title page is your help book's default page, which appears when the user opens your help book, either by choosing the application help item from the Help menu in your application or by choosing your help book from the Library menu. The title page introduces your help book and serves as the entry point into the rest of your help content. All help books registered with the Help Viewer must have a title page.

There are many ways you can approach designing the title page for your help book. For example, the title page from Mac Help, the system help book, has a link to a help page that describes the new features in the current version of Mac OS X, a link to a page that introduces the capabilities of the system, and offers a number of links to help pages answering common queries to get the reader started.

To indicate the title page of your help book, include the `AppleTitle` meta tag in the header section of the XHTML file you want to use as your book's default page. This XHTML file must reside at the top level of your help book folder, as shown in Figure 2-8 (page 31). Here is how you would use the `AppleTitle` meta tag in the title page for a help book called SurfWriter Help:

```
<meta name="AppleTitle" content="SurfWriter Help">
```

Figure 2-4 (page 28) shows the title page for Mail Help.

Specifying a Help Book Icon

When you register a help book, your help book becomes visible in the Library menu. If you provide an icon, Help Viewer displays this icon in the Library menu, next to the title of your help book. Adding an icon to your help book makes it easier for users to associate your book with your application.

To specify an icon for your help book, use the `AppleIcon` meta tag in the header section of your title page file. Here is how you would specify an icon file called `surfwritericon16.png` for SurfWriter Help if the icon were at the top level of the help folder.

```
<meta name="AppleIcon" content="SurfWriter%20Help/surfwritericon16.png">
```

Figure 2-9 shows the Library menu, with each application's icon next to the book title.

Figure 2-9   The Library menu
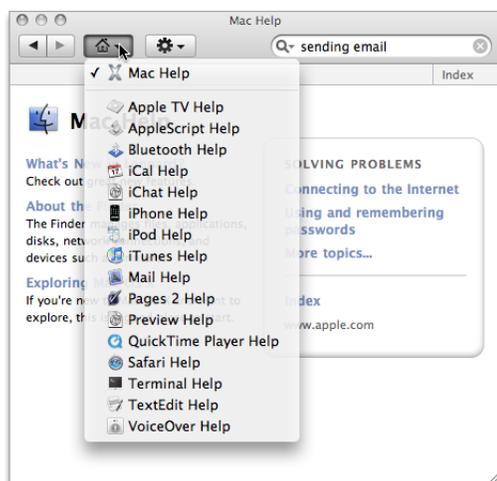
The icon file can be placed at the top level of the help folder or in any subfolder as long as you specify the path accordingly. Many applications place the icon file in the same graphics subfolder as all the illustrations for the help book. The icon should be a 16-by-16 pixel version of your application icon, with a transparent background, saved as a PNG file. Note that the icon cannot be updated from a remote server. Only the local icon file is used by Help Viewer.

Creating a Chapter-Based Help Book

If your application uses optional components such as plug-ins or expansion packs, you may find it useful to organize your help book into chapters. Chapters are subfolders within a help book. The advantage of creating a chapter-based help book is that you can choose to install only those chapters that support components the user has installed. If the user later adds or removes components, you can install or remove individual chapter folders as required.

Organizing a Chapter-Based Help Book Folder

Each chapter has its own title page and may contain its own index and subfolders. For example, SurfWriter features several optional components, so the help book is organized into a main folder and a subfolder for each component. Figure 2-10 shows the SurfWriter help book with only the main folder installed.

Figure 2-10   SurfWriter help book with main folder installed
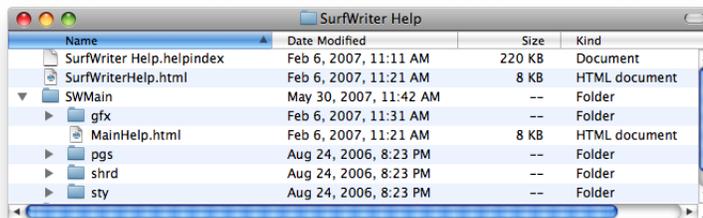
Figure 2-11 shows the SurfWriter help book after the user installs the optional spelling checker. Note that each subfolder has its own title page.

Figure 2-11   SurfWriter help book with optional chapter installed
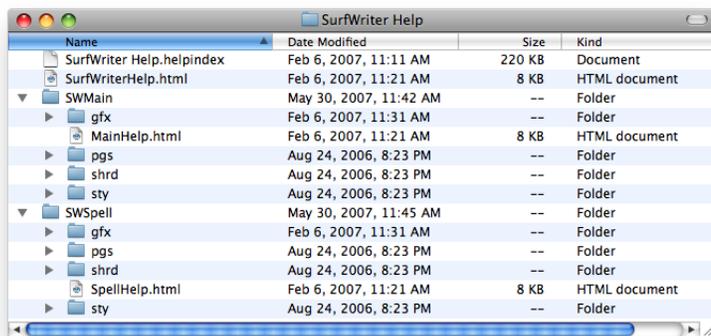
In addition, you should provide a separate table of contents for each chapter. This makes your help book truly modular, allowing chapters to be easily added and removed. You can use this chapter-level table of contents as your chapter title page.

> **Important:** Chapter-based books can have an index file for the whole book or separate index files for each chapter. If you are providing an index file for each chapter of your chapter-based help book, do not also provide a separate index file at the top-level of your help book.

Creating a Dynamic Table of Contents

If you have a chapter-based help book, you can create a dynamic table of contents for it. With a dynamic table of contents, you can install and remove chapters, and the top-level table of contents for your help book will always reflect the latest contents of the help book currently on the user's system.

To create a dynamic table of contents, each chapter in your help book must have a title page containing the `AppleTitle` meta tag. At the top level of your help book, include a template file for the table of contents. The template file must be named `toctmpl.htm`. In this template file, specify the format of the table, using HTML comments containing the strings `AppleTOCRowBegin` and `AppleTOCRowEnd`. For example, the following lines specify the format of a table row in the dynamically generated table of contents for SurfWriter Help:

```
<!-- AppleTOCRowBegin -->
<tr height="1">
    <td height="1" width="158"></td>
</tr>
<tr>
    <td width="158"><font face="Helvetica,Arial" size="2">
    <a href="AppleURL" target="AppleTarget"><b>AppleTitle</b></a></font>
    </td>
</tr>
<!-- AppleTOCRowEnd -->
```

The `AppleTOCRowBegin` and `AppleTOCRowEnd` HTML comments mark the beginning and end of the definition of a table row in the dynamically generated table of contents. Help Viewer locates the title page of each chapter in the book and, using the table row definition in your template file, inserts a row in the table

of contents linking to that page. Help Viewer uses the chapter title, defined by the `AppleTitle` meta tag in the chapter title page, as the text of the link. Any content that appears before the `AppleTOCRowBegin` comment in the template file is copied over to the dynamically generated table of contents without change.

You can use the generated table of contents as your help book's title page by including the `AppleTitle` tag at the top of your template file, above the `AppleTOCRow` comments. You should also include any other relevant title page information, such as your help book icon.

Specifying Chapter Order

You can control the ordering of the chapters in your help book's table of contents by using the `AppleOrder` meta tag in the header of each chapter's title page file. For example:

```
<meta name="AppleOrder" content="80">
```

When Help Viewer creates a dynamic table of contents for your help book, it lists each chapter in the order specified by the `AppleOrder` vaues, lowest to highest. Chapters with no `AppleOrder` value are listed last. If you do not specify chapter ordering, Help Viewer displays all chapter titles in alphabetical order. For example, given a help book with three chapters titled "About SurfWriter," "Using SurfWriter," and "SurfWriter Reference," the default chapter ordering is as follows:

1. About SurfWriter

2. SurfWriter Reference

3. Using SurfWriter

Using the `AppleOrder` tag, the chapter ordering for the same book can be changed. Given an `AppleOrder` value of 10 for "About SurfWriter," 20 for "Using SurfWriter," and 30 for "SurfWriter Reference," Help Viewer orders the table of contents as follows:

1. About SurfWriter

2. Using SurfWriter

3. SurfWriter Reference

If you specify the same `AppleOrder` value for more than one chapter, Help Viewer sorts those chapters alphabetically.

Indexing Your Help Book

To help users quickly find the information they need to accomplish their tasks, you should make your help book searchable by running the Help Indexer utility to create an index for the book. The Help Indexer utility is described in .

You should also include additional tags and metadata in your HTML help files to control how your help is indexed. Including this information improves the user experience of searching your help book by increasing the relevance of the results returned for searches of your help book and improving their appearance in Help Viewer. "Controlling Indexing of Your Help" describes how you can improve indexing and searching of your help book.

Compatibility Note: Starting with Mac OS X v10.4, index files support several advanced features such as automatic creation of help-page abstracts. These index files have the suffix `.helpindex`. Help Viewer in Mac OS X v10.3 and earlier, on the other hand, cannot read `*.helpindex` files. If you are writing help that must be compatible with Mac OS X v10.3 and earlier, you cannot use UTF-8 character encoding and you must generate index files with the suffix `.idx` in addition to the `*.helpindex` files. The `*.idx` index files do not support the new Help Viewer features. For more information on features and compatibility of these two types of indexes, see "Generating an Index Compatible with Different Versions of Mac OS X" (page 43). To learn how to generate either type of index file, see "Using the Help Indexer Utility" (page 40).

Controlling Indexing of Your Help

There are a number of tags that you can include in your HTML pages to control how your help content is indexed by the Help Indexer utility. In addition to indexing the text of your help topics, this tool indexes the following items:

- **Keywords**. Keywords allow you to specify synonyms or common misspellings for a help topic, ensuring that users who search on these alternate terms still get a hit on the relevant topic.

- **Abstracts**. An abstract is a brief summary of a help topic that is displayed when the user places the cursor over the topic in a list of search results. Users can determine whether they wish to learn more about the topic without actually loading the topic page.

- **Anchors**. Anchors allow you to uniquely identify a topic or section within a page. Anchors help you provide quick access to help content; you can specify an anchor as the destination of a link or use the Apple Help API to search for and display the content identified by an anchor.

- **Segments**. Segments divide a single HTML file into subsections, each of which can be indexed and returned as a separate hit in a search. Segments are particularly useful in help systems with lengthy topic pages.

Finally, you can specify which content in your help book should be indexed, as described in "Specifying What Is Indexed" (page 40). By using these various elements in your help book, you can greatly improve the search results for your help book and make your help book more easily accessible from your application.

Setting Keywords

Keywords are a set of additional search terms for an HTML help page. When a user searches your help book for a term that is designated as a keyword for a topic page, Help Viewer returns that page as a search result, even though the term may not appear in the body of the page. Using keywords, you can specify a set of synonyms and common misspellings for topics covered in your help book.

You can specify keywords for a help page using the `KEYWORDS` meta tag in the header of the help page's HTML file. The following example shows keywords that you could set for a help page that describes how to use the Trash:

```
<head>
```

```
<title>Importing Files</title>
<meta name="KEYWORDS" content="discard, dispose, delete, clear, erase">
</head>
```

Do not add keywords for terms that already occur on the page. Repeating a keyword that already appears on the page can cause the page to be given an incorrectly high relevance rating when Help Viewer returns it in response to a user's query.

Adding Abstracts

An abstract is a brief description of a help topic that appears when the user places the mouse cursor over that topic in a list of search results.

> **Note:** Beginning in Mac OS X v10.2, Help Viewer displays the default text "No summary provided" for help topics without abstracts. Beginning in Mac OS X v10.4, you have the option of having the Help Indexer utility select a sentence from the help page to use as an abstract when no abstract has been provided (see "Generating an Index Compatible with Different Versions of Mac OS X" (page 43)).

Well-written abstracts help users ascertain whether a given page, returned as a search result, in fact contains the information they were searching for. For example, SurfWriter Help contains a page describing how to import files from other formats. The page's title, "Importing Files," gives the user some idea of the page's contents, but you can provide a fuller description by using an abstract phrase such as, "SurfWriter can import files of the following formats: txt, rtf, pages, and doc."

To add an abstract to a help page, use the `description` meta tag in the header section of the page's HTML file. Here is an example of how to create such an abstract:

```
<head>
<title>Sorting your messages</title>
<meta name="description" content="Permanently remove messages that  you've
deleted to save space.">
</head>
```

"`Example of a search result showing an abstract`" shows how such a page and its abstract show up as a search result.

Figure 2-12 Example of a search result showing an abstract
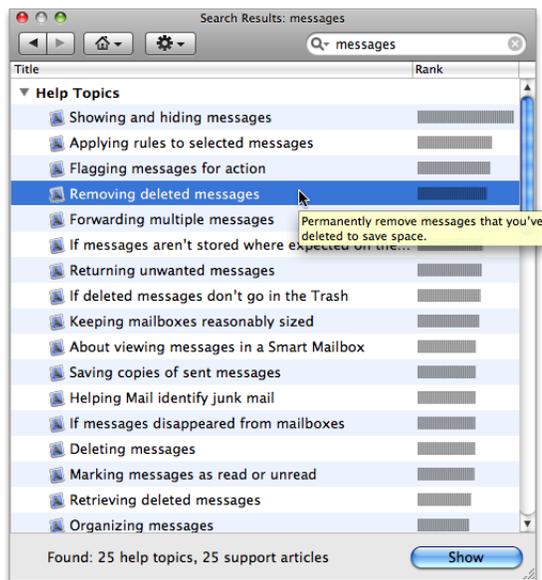
Setting Anchors

Anchors allow you to uniquely identify topics in your help book. When a user follows a link to an anchor, Help Viewer loads the page containing the anchor. If your link includes the specific file containing the anchor (such as `SurfScript.html#anchor1`), Help Viewer scrolls to the anchor location (if it is not at the top of the page). For example, assume that SurfWriter has a simple scripting language called SurfScript. In the help page for SurfScript, you could specify a unique anchor for each SurfScript command, enabling Help Viewer to scroll directly to the desired text when the page loads.

If you use multiple files for your help book, you can put an anchor at the beginning of each file and direct links to the anchors so that you don't have to know the final locations of the help files when you code your HTML. To do so, you use the `help:anchor` URL provided by Apple Help in your link (see "Creating a Link to an Anchor Location" (page 48)).

You can also use anchors to load an anchored page from within your application by calling the Apple Help function `AHLookupAnchor`. To continue the example, SurfWriter could provide an online lookup function that loads the help page for a SurfScript command by calling the `AHLookupAnchor` function and passing the appropriate anchor name when a user Option-clicks a command name in a SurfScript document.

If you need to access your help content programmatically—as you would, for example, if you provide contextually sensitive help—you should consider using anchors to make your help easily accessible from your application. Because you can change the location of anchors within your help book without affecting your product's code, anchors provide a simple and maintainable way for your application to access specific topics within your help book.

You can create multiple anchors in a single file. This is especially useful if you split a file into segments, as you can use the anchors to scroll directly to the beginning of each segment. Segments are described in the next section, "Creating Segments in Help Files."

You specify an anchor using the standard HTML 4 anchor element, as shown in the following example, which creates an anchor called `SurfScriptCommand_OPEN` in a help page describing SurfScript's `OPEN` command:

```
<a name="SurfScriptCommand_OPEN"></a>
<!-- Here is the description of SurfScript's OPEN command -->
```

You link to an anchor by using an HTML anchor element and a `help:anchor` URL:

```
<a href="help:anchor=SurfCmnd_OPEN bookID=SurfWriter Help">Open command</a>
```

The `NSAlert`, `SFChooseIdentityPanel`, `SFCertificatePanel` classes provide help buttons for dialogs. To display such a help button and link it to an anchor in your help book, use the methods `setShowsHelp:` and `setHelpAnchor:` in those classes. See *NSAlert Class Reference*, *SFChooseIdentityPanel Class Reference*, and *SFCertificatePanel Class Reference* for details.

> **Important:** If you use anchors in your help book, you must turn anchor indexing on when you index your help book with the Help Indexer utility. For more information, see "Anchor Indexing" (page 42).

Setting Anchors for Generated Lists

In addition to using anchors to jump to a location in a help book, anchors are used to implement generated lists ("Generating Lists from Anchors" (page 49)) and exact match searches ("Setting Up Exact Match Searching" (page 51)). Unlike anchors used to jump to a specific page, each of which must be used only once, the anchors used for lists can be placed in any number of places. When you specify an anchor to generate a list, every page containing that anchor is included in the list. For example, the anchor corresponding to the "Accounts preferences" generated list in Mac Help is located in the pages for "About administrator accounts," "About logging in and out," "About passwords in Mac OS X," and so forth. On the other hand, the "About administrator accounts" page (for example) includes a unique anchor used only in that file. The "About administrator accounts" page, therefore, contains one anchor used to jump to that page (`mchlp1746`) plus several other anchors used in generated lists, including the anchor used for the "Accounts preferences" generated list (`almh10339`):

```
<a name="mchlp1746"></a><a name="almh10045"></a><a name="almh10090"></a><a
name="almh10339"></a><a name="almh10564"></a><a name="almh10334"></a>
```

Creating Segments in Help Files

A large help book may contain hundreds of individual pages. Rather than using a separate HTML file for each page, you can combine several pages into one file by dividing the file into segments. Help Viewer treats segments much like separate files; each segment in a file can have a different abstract, anchor, and keyword set. The following example shows how SurfWriter Help defines a segment and specifies its abstract and keywords:

```
<!-- AppleSegStart="Opening a File" -->
<!-- AppleSegDescription="How to open a SurfWriter file" -->
<!-- AppleSegKeywords="launch, read, look , examine" -->
<!-- Topic text goes here -->
<div id="pagetitle">
<h1>How to open a SurfWriter file</h1>
</div>
<p>To open a SurfWriter file, you ...
<p>
...
<p>
</p>
<!-- AppleSegEnd -->
```

The beginning of the segment is marked by the `AppleSegStart` command. If the segment has an abstract or keywords, they are specified using the `AppleSegDescription` and `AppleSegKeywords` commands, respectively. The content of the segment follows. The `AppleSegEnd` command marks the end of the segment.

Specifying What Is Indexed

By default, each file in your help book is fully indexed. You can use the `ROBOTS` meta tag in the HTML header of a particular file to control how that file is indexed. The `ROBOTS` meta tag supports the values shown in Table 2-1.

Table 2-1   Values of the `ROBOTS` meta tag

Value	Meaning
`NOINDEX`	Specifies that the HTML file should not be indexed.
`KEYWORDS`	Specifies that the HTML file should be indexed for keywords only.
`SEGMENTS`	Specifies that the HTML file should be indexed for the content of the segments in that file.
`ANCHORS`	Specifes that the HTML file should be indexed for anchors only.

Apple recommends that you do not index a page that contains only links or graphics, or a page on which you don't want the user to land as a result of a search. For example, if you have a sequence of pages that are linked in a series, you might only want to index the first page in the sequence. To specify that a file should not be indexed, use the `ROBOTS` meta tag with the value `NOINDEX` as shown in the following example:

```
<meta name="ROBOTS" content="NOINDEX">
```

Specify `KEYWORDS` as the value of the `ROBOTS` meta tag if you want the indexer to pick up only your specified keywords for use as search results. If you define segments in a file, as described in "Creating Segments in Help Files" (page 39), you can supply a value of `SEGMENTS` in the file's header to specify that only the content of the segments is indexed. The `ANCHORS` value is useful for pages which you want to be able to retrieve using anchor lookup, but which are not useful as search results, such as your title page.

Using the Help Indexer Utility

Once you have finished adding abstracts, keywords, and any other indexing information to your help book files, you must run the Help Indexer utility on your help book. The Help Indexer utility creates an index file for your help content and stores it as a separate file at the top level of the help book folder. The index file contains links to all the words (or tokens) in your help files that are equal to or greater than the minimum term length, excluding any stopwords. You use Help Indexer preferences to specify the minimum term length and the list of stopwords.

Because the index file may be invalidated if you change or rearrange the content of the help folder, be sure to update the index file after the help book content is complete and all files are in their final location.

> **Note:** Some help books include a page labeled *Index*, containing a list of links to topics and to lists of topics. This Index is an HTML page created by the help book author, and is not related to the index file created by the Help Indexer utility. For information on how to generate an index list for your help book, see "Generating Lists from Anchors" (page 49)

You can create an index for your help book by opening Help Indexer (`/Developer/Applications/Utilities/Help Indexer.app`) and entering your help book folder in the dialog that appears. You can also drag the folder containing the help book onto the Help Indexer utility, or execute the utility from a command line. The command-line interface and Help Indexer preferences are documented in the Help Indexer Help. Figure 2-13 shows the Help Indexer preferences window.

Figure 2-13   Help Indexer preferences window
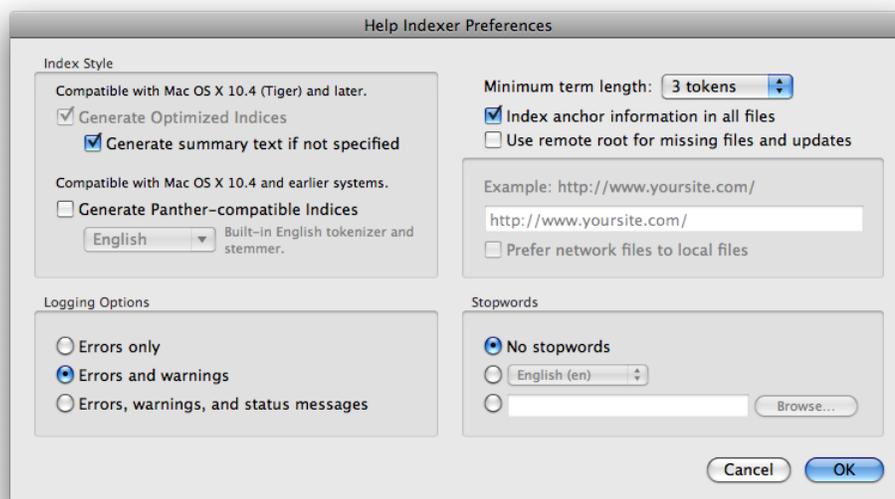
You must modify the default preference settings of the Help Indexer utility if you wish to do any of the following:

- Index a help book for a language other than English.
- Turn off the indexing of anchor information.
- Supply help content from a remote server.
- Provide an index compatible with versions of Mac OS X earlier than Mac OS X v10.4.
- Specify a minimum term length other than three tokens (note that three is the recommended setting).
- Specify a list of stopwords.
- Have warnings or status messages returned in addition to errors.

Indexing for Languages Other Than English

Read "Localizing Your Help Book" (page 53) for more information on indexing non-English help books.

Anchor Indexing

If you add anchors to your help book, you must index your help book with anchor indexing selected in the Help Indexer utility's preferences. When anchor indexing is on, the Help Indexer puts information about the anchors in the index file and Help Viewer uses that information to follow the links to `help:anchor` URLs in your files (see "Setting Anchors" (page 38)). Anchor indexing is on by default. If you don't use anchors, or don't want to use `help:anchor` URLs to link to your files, you can turn off anchor indexing. To toggle anchor indexing off or on, perform the following steps:

1. Open Help Indexer preferences by choosing Preferences from the application menu.

2. Click the checkbox labeled "Index anchor information in all files" (Figure 2-13 (page 41)).

Storing Pages on Remote Servers

As described in "Internet-Based Help Book Content" (page 16), Help Viewer supports Internet-based help. To provide updates to your help content via the Internet, you must specify a remote server from which to download updated help content. To specify a remote server for your help content, do the following:

1. Open Help Indexer preferences.

2. Click the checkbox labeled "Use remote root for missing files and updates."

3. In the text field under this checkbox, enter the server address where your remote help content is available. Figure 2-14 shows a remote root specified in Help Indexer preferences.

Figure 2-14   Specifying a remote server in Help Indexer
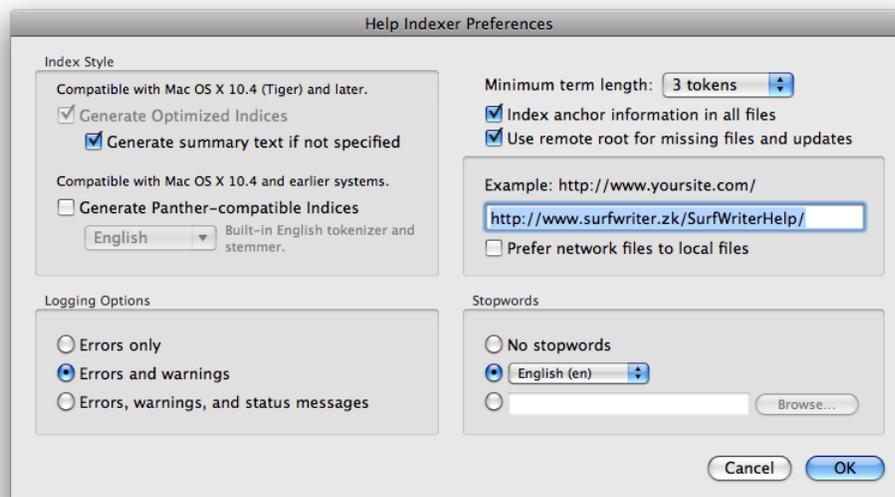
> **Important:** The web server you use to provide remote content for your help files must support HTTP/1.1 Conditional Get requests. Mac OS X Server does support such requests.

You can specify "Internet-primary" help by clicking the checkbox labeled "Prefer network files to local files." When you choose Internet-primary help, before loading a page Help Viewer checks the specified server for a newer version of the page. If the local page is the same as the remote page, Help Viewer uses the local page. If the remote page is newer, Help Viewer downloads the remote page in preference to the local page. Help Viewer caches remotely accessed pages on the user's system.

> **Note:** If you select "Use remote root for missing files and updates" and specify a remote server, but do not choose Internet-primary help, you can provide "Internet-only" help by installing a minimal help folder on the user's system that contains only a title page, index, and any other pages that should be quickly accessible and that don't have to be updated. Any files that you want to provide exclusively online should be stored on the remote server, but not installed locally.
>
> Whether you are installing a minimal help book on the local system or are placing the full contents of your help book locally, you must index your help folder with all pages installed and organized into the correct subfolders, if any.

Help Viewer can also download updated index files from a remote server. When Help Viewer launches, it looks in your help book's index file for a remote root. If you specified a remote root in Help Indexer preferences, Help Viewer contacts the server and checks for an index file at that location. If an index file is present, and if it is newer than the locally stored index, then Help Viewer downloads the file.

> **Important:** Because the index file for your help book records the remote root at which Help Viewer can access your Internet-based help, you must include an index file in the help book folder that you install locally.

You can control how long your help pages are cached, using the two meta tags shown below:

```
<meta http-equiv="Expires" content="Tue, 01 Jan 1980 1:00:00 GMT">
<meta http-equiv="Pragma" content="no-cache">
```

The first example sets a specific expiration date for the cached page. A 24-hour clock is used, and the time zone is included when specifying a particular expiration date. The second example tells Help Viewer that the file should never be cached.

Generating an Index Compatible with Different Versions of Mac OS X

Starting with Mac OS X v10.4, the Help Indexer utility can create index files that take advantage of the Search Kit features introduced with that version of the operating system. Improvements starting with Mac OS X v10.4 include the ability to:

- Extract a summary sentence from a help file to use when no `DESCRIPTION` meta tag is provided (see "Adding Abstracts" (page 37))

- Specify a list of stopwords (see "Specifying a Minimum Term Length and Stopwords" (page 44))

- Specify a minimum number of tokens for indexed terms (see "Specifying a Minimum Term Length and Stopwords" (page 44))

Help Viewer can use indexes created for Mac OS X v10.4 or later (file suffix `.helpindex`) or those created for Mac OS X v10.3 or earlier (file suffix `.idx`). However, to take advantage of the newer features, a `*.helpindex` file must be present. Starting with Mac OS X v10.5, the Help Indexer utility always creates a `*.helpindex` file. The Help Indexer Utility for Mac OS X versions 10.4 and 10.5 can optionally also create a `*.idx` index file.

If your help book is used only on Mac OS X v10.4 or later, use HTML 4.01 and use UTF-8 as the character set.

If your help book is also used on Mac OS X v10.3 or earlier, use HTML 3.2 and do not use UTF-8. If you use UTF-8, you will not be able to reliably create an index compatible with those operating systems. In this case, generate an `*.idx` file in addition to the `*.helpindex` index file by selecting the "Generate Panther-Compatible Indices" option.

If you choose to create an index for Mac OS X v10.3 and earlier systems, you must specify the language of the tokenizer to use. For Mac OS X v10.4 and later indexes, no language choice is necessary because UTF-8 supports all languages. However, you can specify a language-specific list of stopwords, as described in the following section, "Specifying a Minimum Term Length and Stopwords." See "Indexing a Non-English Help Book" (page 54) for more information.

Specifying a Minimum Term Length and Stopwords

Staring with Mac OS X v10.4, the Help Indexer utility enables you to specify the minimum length of terms to be indexed. For example, if the minimum term length is 3 tokens (the default), then the words "I," "to," and "at" are not indexed, but "and" and "but" are included in the index. Apple recommends using the default value.

To prevent commonly used or irrelevant words from being included in the index, you can also include a list of words—referred to as stopwords—to be excluded from the index. In Help Indexer preferences (Figure 2-13 (page 41)), you can specify whether to use no list of stopwords, one of the language-specific stopword lists provided with the Help Indexer, or your own stopword list. The provided stopword lists are the ones used by Apple, and should be adequate for most purposes.

The provided stopword lists are located in a property list in the Help Indexer application bundle at `/Developer/Applications/Utilities/Help Indexer.app/Contents/Resources/Stopwords.plist`.

Instead of one of the standard lists of stopwords, you can specify your own list. A stopword list must be a newline-deliminated list of words in a text file.

Setting Error Verbosity

There are three levels of information you can log during a run of the Help Indexer utility: errors, warnings, and status messages. Errors are problems that prevent Help Indexer from creating an index. . Warnings are problems that don't prevent the creation of an index, but that might cause the results to be different from what you are expecting. . Status messages indicate the progress of the utility so that you can check for expected results, such as anchors being indexed or pages being skipped due to the `ROBOTS NOINDEX` meta tag (see "Specifying What Is Indexed" (page 40)). You use the logging options section of Help Indexer preferences (Figure 2-13 (page 41)) to set the level of messages you wish to have logged. If any messages are generated, the log window opens after the utility completes execution (Figure 2-15).

Figure 2-15 Help Indexer log window
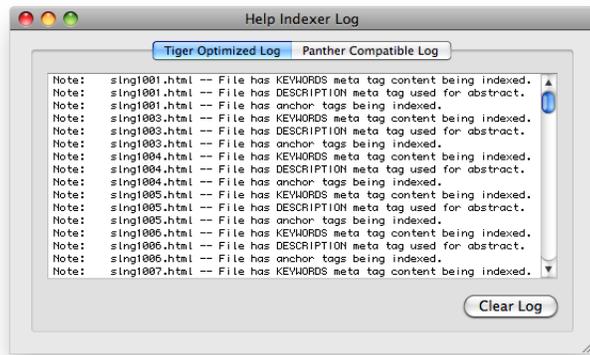
Running Help Indexer from the Command Line

You can run the Help Indexer utility from the command line. The command-line syntax allows you to specify any of the preferences available in the preferences window; if you don't specify any preferences on the command line, then the utility uses the preferences last set in the preferences window, or a default set of preferences if there is no preferences file.

The command-line interface is fully documented in the Help Indexer Help text file available from the Help menu.

Adding Specialized Content to Your Help Book

You can enrich the user experience of your help book by including additional resources such as animated tutorials, scripted tasks, and other multimedia content supported by Help Viewer. This section shows you how to:

■ Embed QuickTime movies in your help book

■ Automate tasks using AppleScript scripts in your help book

■ Initiate Help Viewer searches from a link in your help book

■ Link to anchors in your help book

Adding QuickTime Movies to Your Help Book

You can use QuickTime to create animated tutorials or demos for your help book. To link to a QuickTime movie, use the standard HTML `embed` tag. The following example shows how to call a movie file called `SurfScriptDemo.mov` located in the movies subfolder of the SurfWriter Help folder:

```
<embed src="SurfWriter%20Help/movies/SurfScriptDemo.mov">
```

Use only legal URL characters to specify the URL for the path; for example, the folder name "SurfWriter Help" is specified as `SurfWriter%20Help`.

Running Other Applications from Your Help Book

You can launch other applications, such as media players or custom help applications, from your help book by using the standard HTML anchor element. Here is an example of how you would specify a link that opens an application called SurfScriptPlayer located in the SurfWriter Help folder:

```
<a href="SurfWriter%20Help/SurfScriptPlayer">
```

Use only legal URL characters to specify the URL for the path; for example, the folder name "SurfWriter Help" is specified as `SurfWriter%20Help`.

A better strategy than directly executing an application is to execute an AppleScript script that can handle errors such as a missing program, incorrect permissions, and so on. See "Automating Help Tasks with AppleScript" (page 47).

Opening an External Web Page in Help Viewer

When you include an `http://` link in your help book HTML, Help Viewer ordinarily opens the link in the user's default web browser. You can use the `target-"_helpViewer"` argument to cause the link to open in the Help Viewer window instead. For example, you can use a link of this sort to open a page on your customer service website in Help Viewer, making it appear to the user as if it's part of your help book. The following line would cause the support page for iCal on the Apple, Inc. website to open in the Help Viewer window:

```
<a href="http://www.apple.com/support/ical" target="_helpViewer">Support
Website</a>
```

Using Help URLs in Your Help Book

The URLs using the Help Viewer `help:` protocol, introduced in "Help URLs" (page 19), allow you to access other help content, including:

■ Executing AppleScript scripts

■ Initiating Help Viewer searches

■ Jumping to anchor locations

■ Generating lists from anchors

■ Opening other help books

> **Important:** The syntax used for these URLs is specific to Apple Help. You need to follow the guidelines given here exactly or the links won't work as desired.

Automating Help Tasks with AppleScript

You can take advantage of AppleScript's ability to automate Mac OS events by linking to scripts in your help book. For example, Figure 2-16 shows a page from Mac Help that uses an AppleScript script to open the Network Diagnostics utility when the user clicks the link at the bottom of the page.

Figure 2-16   A link to an AppleScript script in a help page
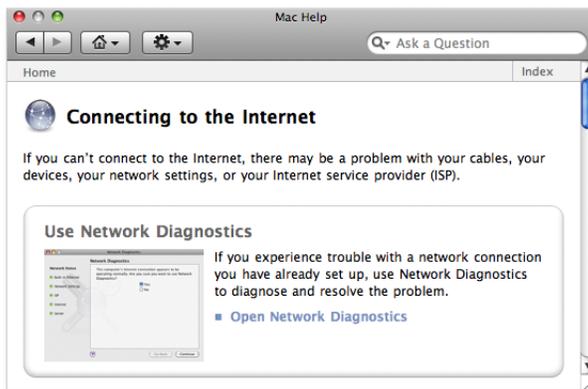
The syntax for calling a script is as follows:

```
<a
href="help:runscript=help_folder_name/subfolder/scriptnamestring='optional_string_parameter'">
```

The optional string parameter is a string value that you can pass to the script. The script is responsible for parsing the value passed in the string parameter. If the script accepts multiple parameters, use commas to separate the values. The following example shows how to specify a script called "MakeNewSaveFolder" in the scripts subfolder of the SurfWriter Help folder and pass it a folder name value in the string parameter:

```
<a href="help:runscript=SurfWriter%20Help/scripts/MakeNewSaveFolderstring='My
Save Folder'">
```

Make sure that you enclose the entire tag value in double quotes.

If you pass the script any parameters with the `help:runscript` command, then you have to have a `helphdhp` routine inside the script to receive the parameters. For example, Mac Help has an AppleScript handler used to open an application from a Help Viewer page. The HTML command to open the Mail application looks like this:

```
<a
href="help:runscript=MacHelp.help/Contents/Resources/English.lproj/shrd/opnbndsig.scpt
 string='com.apple.mail'"> Open Mail</a>
```

The AppleScript handler that receives the parameter (in this case, an application bundle ID) is as follows:

```
on «event helphdhp» (completeParam)
    -- localizable text
```

```
    set cancelBtn to "Cancel"
    set errorText to "The item cannot be opened. It may be disabled or not
installed."
    --end localizable text

    try
        tell application "Finder"
            open application file id completeParam
        end tell

    on error errMsg number errNum
        display dialog errorText buttons {cancelBtn} default button 1 with icon
 0
        return
    end try
end «event helphdhp»
```

For examples of AppleScript scripts in a help book to use as a starting point, see the scripts included in Mac Help at `/Library/Documentation/Help/MacHelp.help/Contents/Resources/`. Do not link to any of these scripts; however, you are welcome to make copies of them to put in your own help book.

Initiating a Search from Your Help Book

The `help:search` URL allows you to create a link in your help book that, when clicked by the user, initiates a search for a particular term or phrase. This is particularly useful for linking to further information about subjects that appear in multiple help pages. Rather than link to each topic page, you can simply set up a search that will find all pages in your help book in which the subject appears. The syntax for initiating a Help Viewer search is as follows:

```
<a href="help:search='search_term' bookID=help_book_name">
```

The `bookID` parameter is a string value specifying which help book Help Viewer should search. If you do not specify a book, Help Viewer searches all help books currently installed on the system. Your book must be indexed for the search to return results.

The following example creates a link to a search for topics related to importing files in the SurfWriter help book:

```
<a href="help:search='importing files' bookID=SurfWriter%20Help">
```

The value for the book ID should be the help book title, as defined by the `AppleTitle` tag in the title page of the book.

When the user clicks the resulting link, Help Viewer searches SurfWriter Help for all topics pertaining to importing files, just as if the user had typed the query string "importing files" into Help Viewer's search field.

Creating a Link to an Anchor Location

Using the `help:anchor` URL, you can create a link to any help book location identified by an anchor. It is often simpler to create links using anchors than to hardcode the path to the destination in the link, and it allows a link to be moved without having to update all the pages that point to it. The syntax for linking to an anchor location is as follows:

```
<a href="help:anchor=anchor_name bookID=help_book_name">
```

The `bookID` parameter is a string value identifying the help book in which Help Viewer should search for the anchor. If no help book is specified, Help Viewer searches all of the registered help books currently on the system. The following example creates a link to the topic on opening files in SurfWriter Help:

```
<a href="help:anchor=openfile bookID=SurfWriter%20Help">
```

When the user clicks the link, Help Viewer takes the user to the location identified by the anchor named "openfile." If more than one anchor is found matching the anchor name, Help Viewer displays all of the matching anchor locations in a search results page. To link to anchor locations in your help book, you must index your help book with anchor indexing turned on, as described in

Generating Lists from Anchors

The `help:topic_list` URL allows you to generate lists of pages to which the user can jump without having to create and maintain such lists manually.

You must provide an anchor ID, a path to the book ID, a path to a template file (used by a Help Viewer XQuery script), a path to a CSS file that specifies the styling for the list, and the name of the list item. Here is the HTML in Mac Help used to generate the Accounts preferences index list:

```
<p><a href="help:topic_list=almh10090 bookID=Mac Help template=sty/genlist.html
 stylesheet=sty/genlist_style.css Other=Accounts preferences">Accounts
preferences</a></p>
```

Here is the XQuery template from Mac Help that is referenced by this URL:

```
<html>

    <head>
        <meta http-equiv="content-type" content="text/html;charset=utf-8">
        <title>AppleTopicListOther</title>
        <meta name="robots" content="noindex">
        <link href="AppleTopicListCSS" rel="stylesheet" media="all">
    </head>

    <body>
        <div id="list">
            <h1>AppleTopicListOther</h1>
            <p>Click a topic below.</p>
            <!-- AppleTopicListRowBegin -->
            <p><a href="AppleTopicListURL">AppleTopicListItemTitle</a></p>
            <!-- AppleTopicListRowEnd --></div>
        <div id="banner">
            <div id="machelp">
                <a class="bread" href="help:anchor='access' bookID=Mac Help">Mac
 Help</a></div>
            <div id="index">
                <a class="leftborder" href="help:anchor='x1' bookID=Mac
Help">Index</a></div>
        </div>
    </body>

</html>
```

This template replaces the variable `AppleTopicListOther` with the value passed as the `Other` parameter. The variable `AppleTopicListURL` is used to look up all instances of the anchor passed as the `topic_list` parameter, and the title of the page identified by that anchor is substituted for the `AppleTopicListItemTitle` variable. The resulting list generated for the "Accounts preferences" index item is shown in Figure 2-17. Each page in this list contains the anchor specified in the `help:topic_list` URL (see "Setting Anchors for Generated Lists" (page 39)). To add or remove items from a generated list, you add or remove that list's anchor ID on the relevant HTML pages in the help book and generate a new index file. Because the list is generated each time it's requested, it always reflects the current set of relevant help pages.

For other examples of code used to generate lists, look in the help folders of Mac Help and other Apple help books. Generated lists are used to create the help book index and the See-also lists at the end of topic pages.

Figure 2-17   Accounts preferences generated list
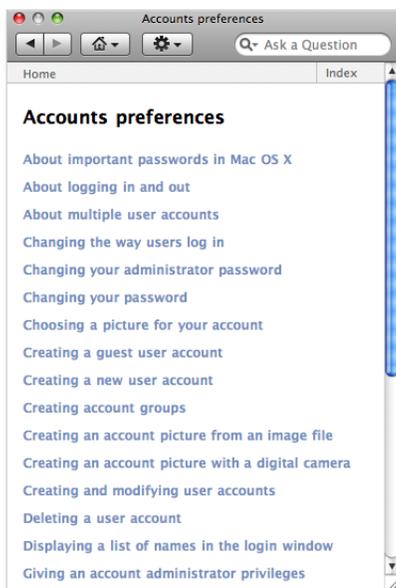
Opening Other Help Books

You can use the `help:openbook` URL to open a specified help book in Help Viewer. You can use this URL, for example, to open the help book for a related application in an application suite. The syntax for opening a help book is as follows:

```
<A HREF="help:openbook=help_book_name">
```

When the user clicks the link, Help Viewer opens the title page of the specified help book. For example, to create a link that opens the SurfWriter Help book to the title page, you would include the following code in your help book:

```
<A HREF="help:openbook=SurfWriter%20Help">
```

To create a link that jumps to a particular location in a help book, see "Creating a Link to an Anchor Location" (page 48).

Setting Up Exact Match Searching

You can provide a list of search terms and corresponding search results. When the user enters a search term that exactly matches a term in your list, Help Viewer takes the corresponding search result from your list, gives it a relevance rating of 100%, and displays it along with other search results. You can use an exact match search list, for example, to provide responses to search terms too short to normally be used in a seach (such as "CD") or to make sure that the most relevant results receive the highest relevancy rating.

To set up exact match searching, you create a property list containing the search terms and search results and place the `.plist` file at the top level of the help book, along with the title page and the help book index (see "Organizing the Help Book Folder" (page 31)). The property list contains a set of key-value pairs. All keys and values are strings. Each key is a search term, specified as all lowercase with all spaces removed. Hyphens and other punctuation marks are not allowed. Each corresponding value is an anchor ID. As with generated lists, each anchor can be used in any number of help book pages. When a user enters a search term, Help Viewer takes a lowercase, spaces-removed version of the search string and compares it to the keys in the exact match property list. If it finds an exact match for the entire string, Help Viewer returns every page containing the referenced anchor string, assigns it a relevancy rank of 100%, adds it to the list of search results, and displays the results. Figure 2-18 shows the exact match property list from Mac Help. Figure 2-19 (page 52) shows the search results displayed when the user searches for one of the terms in that property list. The top items in the search results, with rankings of 100%, contain the anchor ID in the property list for that search term. It's important to note that exact match searching does not return results from stems or partial matches. If you want exact match search results for "CD", "CDs", "VCD", and "CD or DVD", for example, you need four entries in the property list: `cd`, `cds`, `vcd`, and `cdordvd`.

Figure 2-18   Exact match property list
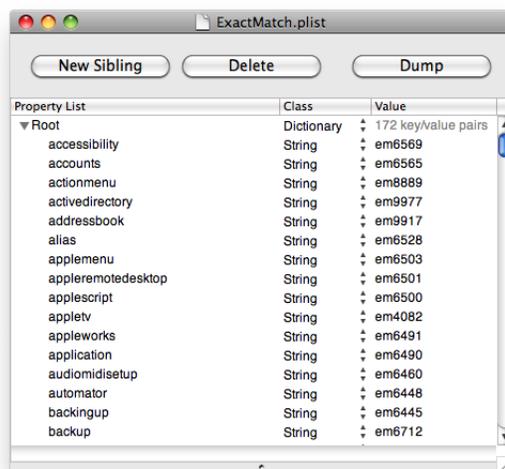
Figure 2-19   Search results corresponding to an exact match

Providing Your Own Online Support Articles

The default behavior for the Help Viewer application is to provide a listing of online support articles in addition to help book topics in response to a query. Help Viewer sends the query to Apple's Knowledge Base support website. Instead, you can provide a URL to your own support site. To do so, provide a URL in your search script on the main page of your help book, as follows:

```
<meta name="AppleKnowledgeBaseURL"
content="http://www.yourcompany.com/search.py?lang=en&amp;product='product'&amp;query='query'"/>
```

Help Viewer replaces the `'query'` argument of the URL with the user's query and the optional `'product'` argument with the product name. It takes the product name from the `AppleKnowledgeBaseProduct` meta tag, or if there is none, from the name of the help book minus the word "Help". For example, iChat uses the following lines of HTML code to generate the URL used to communicate with the Knowledge Base support site:

```
<meta name="AppleKnowledgeBaseProduct" content="iChat" />
<meta name="AppleKnowledgeBaseURL"
content="http://www.info.apple.com/searchredir.html?bucket=helpviewer&amp;lang=en&amp;as_oq='product'&amp;val='query'"/>
```

Help Viewer expects the returned results to be in XML, in the form of an array of dictionaries sorted in order of relevance, as follows:

```
<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE plist PUBLIC "-//Apple Computer//DTD PLIST 1.0//EN"
"http://www.apple.com/DTDs/PropertyList-1.0.dtd">
<plist version="1.0">
<array>
 <dict>
  <key>title</key>
  <string>Article title</string>
  <key>url</key>
  <string>http://kb.yourcompany.com/article.py?num=23</string>
  <key>abstract</key>
  <string>Article abstract ...</string>
 </dict>
</array>
</plist>
```

Return no more than 25 articles, the maximum that Help Viewer will display.

If there are no results for the given query, return the following XML:

```
<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE plist PUBLIC "-//Apple Computer//DTD PLIST 1.0//EN"
"http://www.apple.com/DTDs/PropertyList-1.0.dtd">
<plist version="1.0">
<array>
 <dict>
   <key>NoResultsFound</key>
   <string>No results were found.</string>
 </dict>
</array> </plist>
```

Localizing Your Help Book

If your application will be used in more than one part of the world, your help book should be localized for every relevant language, country, or cultural region where it will be used. Localizing your help book involves translating the text of your help content and customizing graphics and other resources used in your help book.

This section shows you how you can ensure that your localized help content appears correctly in Help Viewer.

For more information on internationalization and HTML, see http://www.w3.org/International/

Language-Specific Resource Directories

For each language you support, you must provide a complete, localized help book in its own resource subdirectory within the `Resources` directory in the software bundle. Figure 2-20 shows a set of resource subdirectories (each with a prefix specifying the language and the `.lproj` extension) in an application bundle.

Figure 2-20   Resource subdirectories in an application bundle
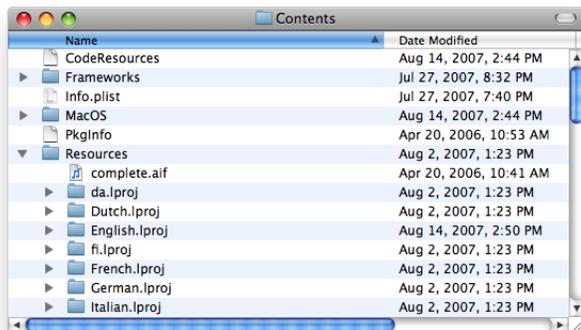
Each localized version of the help book should have a localized book name and localized meta tags, including `AppleTitle`, `AppleKnowledgeBaseProduct/URL`, `Keywords`, **and so forth**) as appropriate.

Specifying Character Encoding

To specify the character encoding used by your help book, use the standard HTML 4 syntax, as shown in the example below, which specifies the UTF-8 character encoding:

```
<meta http-equiv="content-type" content="text/html;charset=UTF-8">
```

Important: Starting with Mac OS X v10.4, UTF-8 is recommended for all your content. See "Generating an Index Compatible with Different Versions of Mac OS X" (page 43).

Indexing a Non-English Help Book

Once you have created your localized help book, you must run the Help Indexer utility on the book. For indexes compatible with Mac OS X v10.3 and earlier, you must specify a tokenizer compatible with the language of the help book. To do so, open the Preferences dialog in the Help Indexer utility, check the "Generate Panther-compatible indices" checkbox, and select a tokenizer from the pop-up menu).

The use of UTF-8 character encoding enables Help Indexer to support all languages for indexers compatible with Mac OS X v 10.4 and later. See "Generating an Index Compatible with Different Versions of Mac OS X" (page 43) for information about creating indexes compatible with different versions of Mac OS X.

Help Book Registration

This chapter describes how to make sure your completed help book is registered so that users—and your application—can access it. When a help book has been registered, it becomes available from the Library menu and from the application Help menu. If the help book is listed in the application's `Info.plist` file, it is registered automatically when the user chooses application help from the Help menu or from a help button. You can also register your help book by calling the `AHRegisterHelpBook` function during application startup. If you are creating a tool or application for Mac OS X, you should read this chapter to learn how to add your help book to your software product.

Where to Place Your Help Book Folder

Help books, like other resources that are language or region-specific, are kept in localized resource directories, named for their language or region and identified by the `.lproj` extension. These localized resource directories are placed in the `Resources` directory of your software bundle.

For example, if your application is localized for English, French, and Japanese audiences, your application bundle should contain a localized resource directory for each language. You should place a help book folder with your application's help content, localized for the target region, in each directory.

Figure 3-1 shows the contents of an application bundle. The folder containing the English-language version of the application help book is located in the `English.lproj` subdirectory of the `Resources` directory. For more information on localization of resources, naming conventions for `.lproj` directories, and bundle structure, see *Bundle Programming Guide*.

Figure 3-1   The location of an English-language help book in the application bundle.
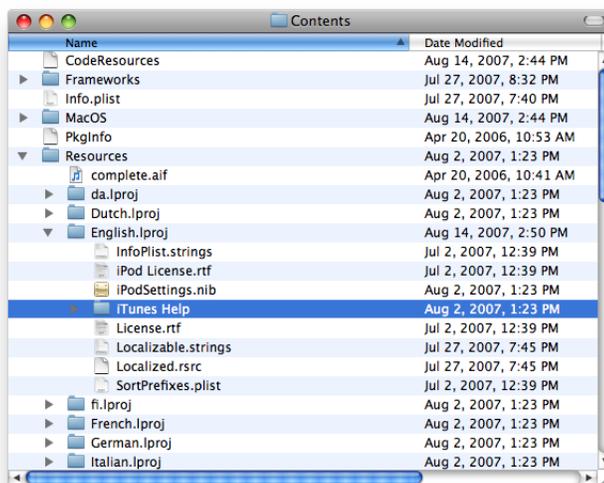
Apple strongly recommends that you place your help book in your software bundle. When your help book is bundled with your software, it is installed and moved along with your software.

> **Note:** It is not necessary to place the help book in an application bundle (`*.app`); any bundle will do. For example, printer drivers sometimes include help books. As long as you register the help book properly (see "How to Register Your Help Book" (page 57)), Help Viewer can find it.

> **Important:** The `.help` bundles and the directories `~/Library/Documentation/Help` and `/Library/Documentation/Help` are reserved for use by Apple; use by third-party developers is not supported.

You can use the following procedure to install the help folder in the `.lproj` directory using Xcode:

1. In the main project window, select Resources in the Groups & Files pane.

2. From the Action menu, choose Add > Existing files (Figure 3-2).

Figure 3-2   Adding help files in Xcode
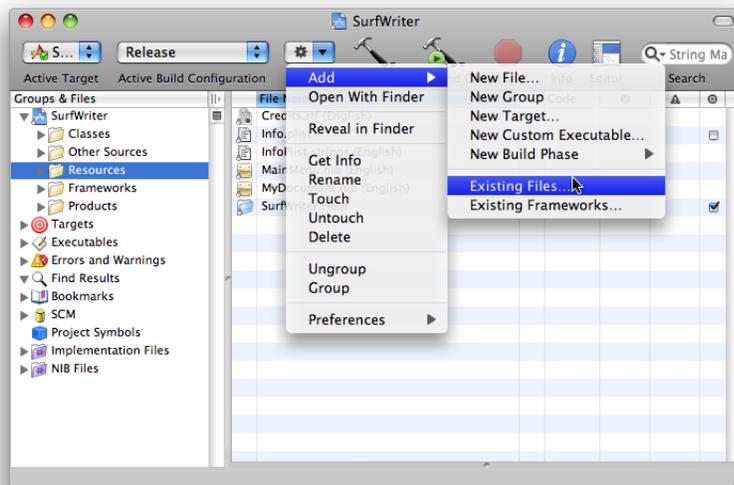
3. Select the help folder in the Add Files dialog and click Add.

4. Select the "Create Folder References for any added folders" radio button and click Add (Figure 3-3).

> **Important:** If you neglect to perform this step, your help book will work as expected on your development system, but will not work when you transfer the help book to another system.

Figure 3-3 Create folder references in Xcode
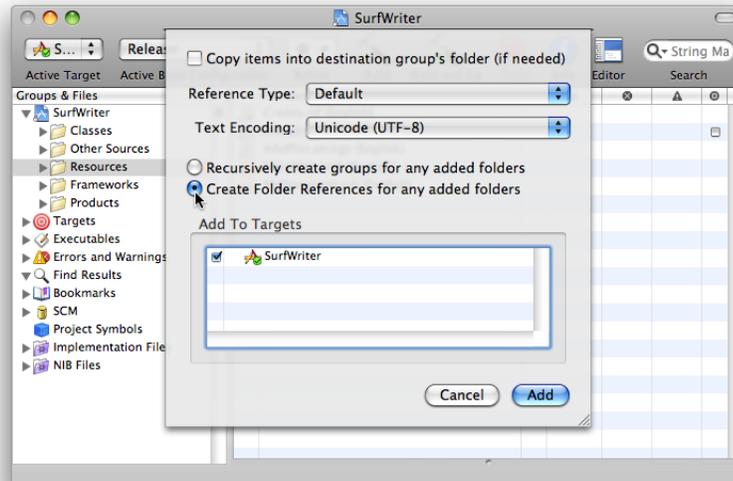
How to Register Your Help Book

For Cocoa and Java applications, you can take advantage of automatic help book registration by adding the help book name and location to your information property list (`Info.plist`) file. For Carbon applications, you must take the additional step of calling the Apple Help registration function, `AHRegisterHelpBook`.

You must register your help book to get the automatic help book support provided by the system. When you register your help book, the system creates a Help menu for your application, populates it with an application help item, and opens your help book when a user chooses this item. In addition, your help book must be registered for it to appear in the Library menu.

Cocoa and Java applications that use the Apple Help API to access their help book content must also call the `AHRegisterHelpBook` function before making any calls to other Apple Help functions.

> **Note:** Apple strongly recommends that you add your help book to your `Info.plist` file or call `AHRegisterHelpBook` to register your help book. If you do not do so, you are responsible for handling all access to your help book yourself, as described in the *Apple Help Reference*.

Editing the Information Property List File

To register a help book, you need to include the `CFBundleHelpBookFolder` and `CFBundleHelpBookName` keys in your `Info.plist` file. The `CFBundleHelpBookFolder` key identifies the help book folder; the value associated with this key should be a string specifying the folder name. For example, here is how you would enter the name of the SurfWriter help book folder:

```
<key>CFBundleHelpBookFolder</key>
<string>SurfWriter Help</string>
```

The `CFBundleHelpBookName` key identifies the help book. The value associated with this key should be a string specifying the help book title, as defined by the `AppleTitle` tag in the title page of the book. For example, here is how you would enter the title of the SurfWriter Help book:

```
<key>CFBundleHelpBookName</key>
<string>SurfWriter Help</string>
```

If you are using Xcode to develop your software product, you can add these keys to your `Info.plist` file with the following steps:

1. From the main project window, select the name of the project in the Groups & Files pane.

2. Select the `Info.plist` file in the right pane

3. Add the help book folder and help book name keys to the `Info.plist` file, as shown in Figure 3-4.

Figure 3-4   Editing the info.plist file in Xcode
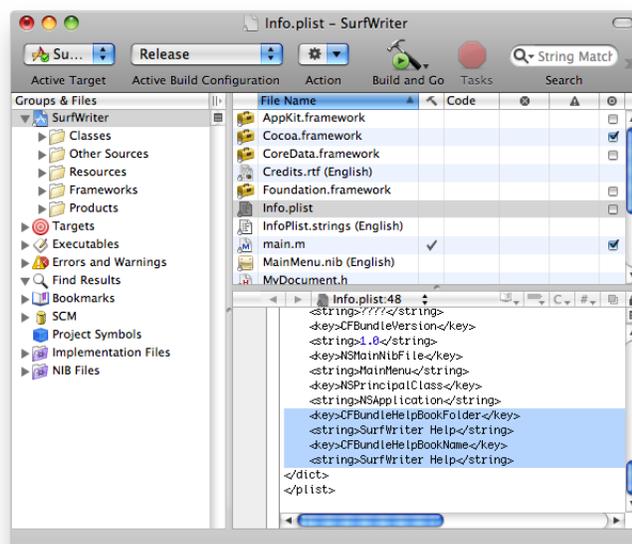
If you have more than one help book, you can set the value of each of the `CFBundleHelpBookName` and `CFBundleHelpBookFolder` keys to an array of strings. However, if you use an array for the value of these two keys, you do not get automatic support for your application help item in the Help menu.

> **Note:** If you are localizing your help book, you should provide localized values for the `CFBundleHelpBookName` key. For each language or region you are targeting, add an entry for the `CFBundleHelpBookName` key to the `InfoPlist.strings` file in the appropriate `.lproj` directory. The value associated with the key should be a string specifying the localized help book title for the target language. For example, here is how you would specify the German localized help book title for SurfWriter Help in the `InfoPlist.strings` file in the `German.lproj` directory:
>
> `CFBundleHelpBookName = "SurfWriter Hilfe";`
>
> For more information on the `InfoPlist.strings` file and other localized strings files, see *Internationalization Programming Topics*.

Note that your `Info.plist` file must also contain a valid `CFBundleIdentifier` entry. For more information on application packaging and property lists, see *Bundle Programming Guide*.

Using the Apple Help Registration Function

Applications that call Apple Help functions must first call the Apple Help function `AHRegisterHelpBook` to register their help book. You would typically do this during application initialization. Once your application has called `AHRegisterHelpBook`, your help content is accessible through the use of the Apple Help functions.

> **Note:** Carbon applications must call `AHRegisterHelpBook` even if they do not use any other Apple Help. If a Carbon application does not call `AHRegisterHelpBook`, the application's help book does not open when the user chooses the application help item from the Help menu. In addition, Help Viewer does not include the help book in the Library menu. (Cocoa applications that call the `NSHelpManager` equivalents to `AHLookupAnchor` and `AHSearch` (see Table 1-1 (page 20)) do not need to call the `AHRegisterHelpBook` function, as those methods register the help book if necessary.)

Listing 3-1 shows an example of how to register a help book using `AHRegisterHelpBook`.

Listing 3-1   Registering a help book with `AHRegisterHelpBook`

```
OSStatus RegisterMyHelpBook(void)
{
    CFBundleRef myApplicationBundle;
    CFURLRef myBundleURL;
    FSRef myBundleRef;
    OSStatus err = noErr;

    myApplicationBundle = NULL;
    myBundleURL = NULL;

    myApplicationBundle = CFBundleGetMainBundle();                      // 1
    if (myApplicationBundle == NULL) {err = fnfErr; goto bail;}

    myBundleURL = CFBundleCopyBundleURL(myApplicationBundle);           // 2
    if (myBundleURL == NULL) {err = fnfErr; goto bail;}
```

```
if (!CFURLGetFSRef(myBundleURL, &myBundleRef)) err = fnfErr;          // 3

if (err == noErr) err = AHRegisterHelpBook(&myBundleRef);            // 4
return err;

}
```

Here is what the code in Listing 3-1 does:

1. Calls the Core Foundation function `CFBundleGetMainBundle` to retrieve a reference to the application's main bundle.

2. Calls the Core Foundation function `CFBundleCopyBundleURL` to get the path to the application bundle.

3. Calls the Core Foundation function `CFURLGetFSRef` to convert the path obtained in Step 2 into a file system reference (an `FSRef` structure).

4. Calls `AHRegisterHelpBook`, passing the file system reference obtained in the last step. Apple Help finds the help book located in the bundle and caches the name and location of the help book. Apple Help chooses which localized version of the help book to use based upon the current language of the system.

Opening Your Help Book in Help Viewer

This chapter describes how to use Apple Help functions to load content from your help book in Help Viewer. If you are providing contextually sensitive help, or if you have help books in addition to your primary application help book, you need to know how to access your help book using the Apple Help API.

When users choose an item from the Help menu, click a help button, or choose help from a contextual menu, your application must display the pertinent help book content in Help Viewer. To open your help book in Help Viewer, use one of the following Apple Help functions:

■ `AHLookupAnchor` opens a location in your help book identified by an anchor.

■ `AHSearch` searches your help book for a term or phrase.

Help functions are fully documented in *Apple Help Reference*.

Displaying an Anchor Location

If you specify anchor locations in your help book, as described in "Indexing Your Help Book" (page 35), you can use the Apple Help function `AHLookupAnchor` to find and display help content by anchor name. `AHLookupAnchor` allows you to search for a particular help topic without knowing the path to the page that it is on. If you are implementing contextually sensitive help, you can load it by anchor, without having to track the path to every help page you may access.

If an anchor name appears more than once in your help book, Help Viewer displays all of the content associated with that anchor in your help book in a search results table. To use `AHLookupAnchor`, you must index your help book with anchor indexing turned on.

Listing 4-1 shows a function that uses `AHLookupAnchor` to find and display the text associated with a help book anchor.

Listing 4-1 Displaying an anchor location

```
OSStatus MyGotoHelpAnchor( CFStringRef anchorName)
{
    CFBundleRef myApplicationBundle = NULL;
    CFTypeRef myBookName = NULL;
    OSStatus err = noErr;

    myApplicationBundle = CFBundleGetMainBundle();                          // 1
    if (myApplicationBundle == NULL) {err = fnfErr; goto bail;}

    myBookName = CFBundleGetValueForInfoDictionaryKey(                      // 2
                    myApplicationBundle,
                    CFSTR("CFBundleHelpBookName"));
    if (myBookName == NULL) {err = fnfErr; goto bail;}
```

```
    if (CFGetTypeID(myBookName) != CFStringGetTypeID()) {                    // 3
        err = paramErr;
    }

    if (err == noErr) err = AHLookupAnchor (myBookName, anchorName);         // 4
    return err;

Bail:
    return err;
}
```

Here is what the function in Listing 4-1 does:

1. Calls the Core Foundation function `CFBundleGetMainBundle` to retrieve a reference to the application's main bundle.

2. Calls the Core Foundation function `CFBundleGetValueForInfoDictionaryKey` to find the name of the application's help book. When you register your help book, you store your help book's name in the `Info.plist` file with the key `CFBundleHelpBookName`. Rather than hard code your help book name—which can change as the help book content is updated—in your application, use Core Foundation functions to retrieve the help book name from the property list file.

3. Checks that the value returned in step 2 was of type CFString.

4. Calls the Apple Help function `AHLookupAnchor` to look up the anchor in the application's help book.

Here is an example of how you could call the `MyGotoHelpAnchor` function described in Listing 4-1 (page 61):

```
err = MyGotoHelpAnchor(CFSTR("surfing"));
```

Searching Your Help Book

Apple Help also offers a way for you to send Help Viewer a search query to execute on your help book. Using the `AHSearch` function, you can search your help book for a term or phrase. For example, if you are implementing contextually sensitive help for a user interface element that is referenced in numerous help pages, you can call `AHSearch` to find and display those pages in a search results table. Listing 4-2 shows a function that searches your help book for a search term or query using the `AHSearch` function.

> **Cocoa note:** The `NSHelpManager` method `findString:inBook:` is a wrapper for `AHRegisterHelpBook` and `AHSearch`.

Listing 4-2   A function that searches your help book

```
OSStatus MySearchHelpBook(CFStringRef theQuery)
{
    CFBundleRef myApplicationBundle = NULL;
    CFStringRef myBookName = NULL;
    OSStatus err = noErr;

    myApplicationBundle = CFBundleGetMainBundle();                           // 1
```

```
    if (myApplicationBundle != NULL) {
        myBookName = CFBundleGetValueForInfoDictionaryKey(                    // 2
                        myApplicationBundle,
                        CFSTR("CFBundleHelpBookName"));
    } else err = fnfErr;

    if (myBookName != NULL) {
            err = AHSearch(myBookName, theQuery);                            // 3
    } else err = fnfErr;

    return err;
}
```

Here is what the function in Listing 4-2 does:

1. Calls `CFBundleGetMainBundle` to retrieve a reference to the application's main bundle.

2. Calls `CFBundleGetValueForInfoDictionaryKey` to retrieve the help book name associated with the application bundle.

3. Calls `AHSearch` to search the help book for the string passed to `MySearchHelpBook` in the `theQuery` parameter.

Here is an example of how you could call the function shown in Listing 4-2 (page 62) to search your help book for information on printing. You can use a phrase for your query, such as "print a document" or you can search for a term, such as "print".

```
err = SearchHelpBook(CFSTR("print a document"));
err = SearchHelpBook(CFSTR("print"));
```

Loading a Help Book Page

The Apple Help function `AHGotoPage` allows you to open a help book page at a known location and display it in Help Viewer. If you know the path to the information you want to display, or if you simply wish to open your help book to its title page, you can use `AHGotoPage`.

> **Note:** Whereas the `AHGotoPage` function requires that you know the full or partial path to the HTML file describing the desired help topic, `AHLookupAnchor` allows you to access a help topic with only the anchor name. In most cases, using an anchor is easier and more flexible than tracking the location of the file describing the topic.

You can specify the location of the page using either a full `file://` URL or a combination of a relative path and the help book name. Relative paths should be specified relative to the help book's folder. In addition, you can specify an anchor within the given help page; when you specify an anchor, Help Viewer scrolls directly to the location of that anchor on the help page before displaying the page.

> **Note:** Your help book must be registered to access its contents using a relative path or book name. If your help book is not registered, you must call `AHGotoPage` with a `file://` URL.

Table 4-1 shows the arguments you can pass to `AHGotoPage` and what Help Viewer displays in response.

Table 4-1   Arguments to `AHGotoPage`

Arguments provided to AHGotoPage	Results
Help book name	Help Viewer opens the help book to its title page
Help book name, relative path	Help Viewer opens the page at the given path in the help book
Help book name, relative path, anchor name	Help Viewer opens the page at the path and scrolls to the section identified by the anchor
`file://` URL	Help Viewer opens the page at that path

The function shown in Listing 4-3 takes a path and an anchor name as arguments and calls `AHGotoPage` to open a help book page in Help Viewer.

Listing 4-3   A function that loads a help book page

```
OSStatus MyGotoHelpPage (CFStringRef pagePath, CFStringRef anchorName)
{
    CFBundleRef myApplicationBundle = NULL;
    CFStringRef myBookName = NULL;
    OSStatus err = noErr;

    myApplicationBundle = CFBundleGetMainBundle();                          // 1
    if (myApplicationBundle == NULL) {err = fnfErr; goto bail;}             // 2

    myBookName = CFBundleGetValueForInfoDictionaryKey(                      // 3
                    myApplicationBundle,
                    CFSTR("CFBundleHelpBookName"));
    if (myBookName == NULL) {err = fnfErr; goto bail;}

    if (CFGetTypeID(myBookName) != CFStringGetTypeID()) {                   // 4
        err = paramErr;
    }

    if (err == noErr) err = AHGotoPage (myBookName, pagePath, anchorName);  // 5
    return err;

}
```

Here is what the code does:

1. Calls the Core Foundation function `CFBundleGetMainBundle` to retrieve the application's bundle.

2. If `CFBundleGetMainBundle` cannot find the application's main bundle, returns an error specifying that the file was not found.

3. Calls the Core Foundation function `CFBundleGetValueForInfoDictionaryKey` to retrieve the name of the help book associated with the application's main bundle.

4. Checks that the value returned in step 3 is of type CFString. The Core Foundation function `CFGetTypeID` returns the type ID of the value returned in step 3; the function `CFStringGetTypeID` returns the type ID of a CFString. If the type IDs do not match, `MyGotoHelpPage` returns a parameter error.

5. Calls the Apple Help function `AHGotoPage` to open the application's help book to the page and anchor passed in as arguments to the `MyGotoHelpPage` function. If the `pagePath` and `anchorName` arguments are both `NULL`, `AHGotoPage` opens the application's help book to its title page.

Here are three examples of how you could call the `MyGotoHelpPage` function described in :

```
err = MyGotoHelpPage(CFSTR("pages/howto.html"), CFSTR("surfing"));
err = MyGotoHelpPage(CFSTR("pages/howto.html"), NULL);
err = MyGotoHelpPage(NULL, NULL);
```

Apple Help Meta Tag Properties

Table A-1 lists the properties defined by Apple Help for use with the `meta` element. The Apple Help meta tag properties control how your help book is identified and displayed by Help Viewer.

Table A-1 Apple Help meta tags

Property name	Specifies	Example
`AppleTitle`	The help book title. See "Creating a Title Page" (page 32).	`<meta name="AppleTitle" content="My Application Help">`
`AppleIcon`	The help book icon file. See "Specifying a Help Book Icon" (page 32).	`<meta name="AppleIcon" content="My%20Application%20Help/ myhelpicon.gif">`
`AppleKnowledge-BaseProduct`	The product name. See "Providing Your Own Online Support Articles" (page 52).	`<meta name="AppleKnowledgeBaseProduct" content="MyAppName" />`
`AppleKnowledge-BaseURL`	A query to send to your company's online support website. See "Providing Your Own Online Support Articles" (page 52).	`<meta name="AppleKnowledgeBaseURL" content="http: //www.yourcompany.com/ search.py?lang=en&query='query'" />`
`AppleOrder`	The order in which chapters should appear in the table of contents for a chapter-based book. See "Specifying Chapter Order" (page 35).	`<meta name="AppleOrder" content="20">`
`KEYWORDS`	Additional search terms for an HTML help page. See "Setting Keywords" (page 36).	`<meta name="KEYWORDS" content="discard, dispose, delete, clear, erase">`
`ROBOTS`	Controls how a file is indexed. See "Specifying What Is Indexed" (page 40).	`<meta name="ROBOTS" content="NOINDEX">`

Apple Help URLs

Table B-1 lists the help-specific URLs supported by Help Viewer. Use these URLs in your help book to link to other help topics and additional help resources. Arguments to help-specific URLs can either be enclosed in single quotes or can use standard URL encoding; for example the book name "SurfWriter Help" would be specified as `SurfWriter%20Help`. See "Using Help URLs in Your Help Book" (page 46) for some examples of use of these URLs.

Table B-1   Help URLs

URL	Syntax	Action	Cross reference
`help:anchor`	`help:anchor=anchor_name` `bookID=help_book_name`	Opens Help Viewer to the location in a help book identified by the given anchor.	"Creating a Link to an Anchor Location" (page 48)
`help://full/path.html`	`help: //path/to/page.html`	Opens the specified file in Help Viewer.	"Providing Your Own Online Support Articles" (page 52)
`help:openbook`	`help: openbook=help_book_name`	Opens the specified help book in Help Viewer.	"Opening Other Help Books" (page 50)

`help:runscript`	`help: runscript=help_folder_name/`
	`subfolder/scriptname string=`
	`'optional_string_parameter'`
		Runs the specified script. The `string` argument is an optional argument that is passed to the script.	"Automating Help Tasks with AppleScript" (page 47)

`help:search`	`help:search='search_string'` `bookID='help_book_name'`	Initiates a search of a help book using the specified search criteria. Help Viewer then displays the search results.	"Initiating a Search from Your Help Book" (page 48)

URL	Syntax	Action	Cross reference

`help:topic_list`	`help:topic_list=anchor_ID`
	`bookID=book_ID`
	`template=path_to_XQuery_template`
	`stylesheet=path_to_CSS`
	`Other=item_name`
		Generates a list of the help pages that include `anchor_ID`.	"Generating Lists from Anchors" (page 49)

Apple Help Segments

Help Viewer recognizes certain commands in the form of HTML comments that you can use to split your help book into segments. Table C-1 lists the HTML comments used as segment commands by Apple Help. The syntax for using these commands is as follows:

```
<!-- AppleSegCommand="value">
```

Table C-1   Commands for Apple Help segments

Command	Specifies
AppleSegStart	The beginning of a segment.
AppleSegDescription	An abstract for the given segment. See "Creating Segments in Help Files" (page 39) for more information on segment abstracts.
AppleKeywords	Keywords for the given segment. See "Setting Keywords" (page 36) for more information on keywords.
AppleSegEnd	The end of a segment.

Document Revision History

This table describes the changes to *Apple Help Programming Guide*.

Date	Notes
2007-10-31	Added information about Knowledge Base searching, ExactMatch, and generated lists. Updated all screen shots.
	See "Providing Your Own Online Support Articles" (page 52), "Setting Up Exact Match Searching" (page 51), and "Generating Lists from Anchors" (page 49).
	Also, see "Printing A Page's URL" (page 27), "Opening an External Web Page in Help Viewer" (page 46), and "Automating Help Tasks with AppleScript" (page 47).
2007-05-30	Updated for Mac OS X v10.4 and Mac OS X v10.5. Changed the title from "Providing User Assistance With Apple Help"
	Updated HTML 3.2 references to HTML 4.01 and specified that the title page must be XHTML 1.0.
	Updated information about how to organize the help book folder ("Organizing the Help Book Folder" (page 31)).
	Changed Apple Help Indexing Tool references to Help Indexer utility and documented the use of the new utility to index a help book ("Indexing Your Help Book" (page 35)).
	Updated character encoding information to specify UTF-8 ("Specifying Character Encoding" (page 54)).
	Added some information about localizing help books ("Localizing Your Help Book" (page 53)).
	Revised procedures for registering your help book to use Xcode rather than Project Builder ("Help Book Registration" (page 55)).
2004-06-28	Added revision history. Updated with new location of Apple Help Indexing Tool. Updated cross-references.
2003-05-15	Revision of preliminary version.