
Providing User Assistance With Apple Help (Legacy)

[User Experience](#) > [Help Technologies](#)



2007-10-31



Apple Inc.
© 2003, 2007 Apple Inc.
All rights reserved.

No part of this publication may be reproduced, stored in a retrieval system, or transmitted, in any form or by any means, mechanical, electronic, photocopying, recording, or otherwise, without prior written permission of Apple Inc., with the following exceptions: Any person is hereby authorized to store documentation on a single computer for personal use only and to print copies of documentation for personal use provided that the documentation contains Apple's copyright notice.

The Apple logo is a trademark of Apple Inc.

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

No licenses, express or implied, are granted with respect to any of the technology described in this document. Apple retains all intellectual property rights associated with the technology described in this document. This document is intended to assist application developers to develop applications only for Apple-labeled computers.

Every effort has been made to ensure that the information in this document is accurate. Apple is not responsible for typographical errors.

Apple Inc.
1 Infinite Loop
Cupertino, CA 95014
408-996-1010

Apple, the Apple logo, AppleScript, AppleWorks, Carbon, Cocoa, iTunes, Mac, Mac OS, Pages, QuickTime, and Sherlock are trademarks of Apple Inc., registered in the United States and other countries.

Finder is a trademark of Apple Inc.

Helvetica is a registered trademark of Heidelberger Druckmaschinen AG, available from Linotype Library GmbH.

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

Simultaneously published in the United States and Canada.

Even though Apple has reviewed this document, APPLE MAKES NO WARRANTY OR REPRESENTATION, EITHER EXPRESS OR IMPLIED, WITH RESPECT TO THIS DOCUMENT, ITS QUALITY, ACCURACY, MERCHANTABILITY, OR FITNESS FOR A PARTICULAR PURPOSE. AS A RESULT, THIS DOCUMENT IS PROVIDED "AS IS," AND YOU, THE READER, ARE ASSUMING THE ENTIRE RISK AS TO ITS QUALITY AND ACCURACY.

IN NO EVENT WILL APPLE BE LIABLE FOR DIRECT, INDIRECT, SPECIAL, INCIDENTAL, OR CONSEQUENTIAL DAMAGES RESULTING FROM ANY DEFECT OR INACCURACY IN THIS DOCUMENT, even if advised of the possibility of such damages.

THE WARRANTY AND REMEDIES SET FORTH ABOVE ARE EXCLUSIVE AND IN LIEU OF ALL OTHERS, ORAL OR WRITTEN, EXPRESS OR IMPLIED. No Apple dealer, agent, or employee is authorized to make any modification, extension, or addition to this warranty.

Some states do not allow the exclusion or limitation of implied warranties or liability for incidental or consequential damages, so the above limitation or exclusion may not apply to you. This warranty gives you specific legal rights, and you may also have other rights which vary from state to state.

Contents

Introduction **Introduction to Providing User Assistance With Apple Help** 7

- System Requirements 8
- Organization of This Document 8
- See Also 8

Chapter 1 **Apple Help Concepts** 11

- Help Viewer 11
 - The Help Viewer Window 11
 - Searching in Help Viewer 12
 - The Help Center 15
- Help Books 16
- Internet-Based Help Book Content 17
- How Users Access Your Help 17
 - The Help Menu 18
 - Help Buttons 19
 - Help in Contextual Menus 19
- Help Viewer HTML Extensions 20
 - Help-Specific Meta Tags 20
 - Help URLs 21
 - Apple Help Segments 21
- The Apple Help Application Programming Interface 21

Chapter 2 **Authoring User Help** 23

- Designing a Help Book 23
- Authoring Help Pages 24
 - Authoring Tools 24
 - Creating Topic Pages 24
 - Creating Navigation Pages 27
 - Guidelines for Help 29
- Creating a Basic Help Book 29
 - Organizing the Help Book Folder 30
 - Creating a Title Page 30
 - Specifying a Help Book Icon 32
- Creating a Chapter-Based Help Book 33
 - Organizing a Chapter-Based Help Book Folder 33
 - Creating a Dynamic Table of Contents 34
 - Specifying Chapter Order 36
- Indexing Your Help Book 36
 - Controlling Indexing of Your Help 37

- Using the Apple Help Indexing Tool 40
- Adding Specialized Content to Your Help Book 44
 - Adding QuickTime Movies to Your Help Book 45
 - Running Other Applications From Your Help Book 45
 - Using Help URLs in Your Help Book 45
- Localizing Your Help Book 47
 - Specifying Character Encoding 48
 - Specifying Fonts For Display in Help Viewer 48
 - Indexing a Non-English Help Book 48

Chapter 3 Registering Your Help Book 51

- Where to Place Your Help Book Folder 51
- How to Register Your Help Book 52
 - Editing the Information Property List File 52
 - Using the Apple Help Registration Function 56

Chapter 4 Opening Your Help Book in Help Viewer 57

- Displaying an Anchor Location 57
- Searching Your Help Book 58
- Loading a Help Book Page 59

Appendix A Apple Help Meta Tag Properties 63

Appendix B Apple Help URLs 65

Appendix C Apple Help Segments 69

Document Revision History 71

Figures, Tables, and Listings

Chapter 1 **Apple Help Concepts 11**

Figure 1-1	The Help Viewer window	12
Figure 1-2	A question entered in the search field of Help Viewer	13
Figure 1-3	Search results displayed in Help Viewer	14
Figure 1-4	A topic summary displayed for a search result in Help Viewer	15
Figure 1-5	The Help Center	16
Figure 1-6	The Help menu	18
Figure 1-7	A help button	19
Figure 1-8	Help in a contextual menu	20
Table 1-1	Apple Help functions for accessing your help book	21

Chapter 2 **Authoring User Help 23**

Figure 2-1	A help book page containing overview information	25
Figure 2-2	A task-oriented help book page	26
Figure 2-3	The top-level table of contents for AppleWorks Help	28
Figure 2-4	A topic level table of contents in iTunes Help	29
Figure 2-5	An example of a simple help book folder structure	30
Figure 2-6	The SurfWriter Help title page	31
Figure 2-7	The SurfWriter help book in the Help Center	32
Figure 2-8	SurfWriter help book with main folder installed	33
Figure 2-9	SurfWriter help book with optional chapter installed	34
Figure 2-10	The generated table of contents for SurfWriter Help	35
Figure 2-11	Example of a search result showing an abstract	38
Figure 2-12	Turning on anchor indexing in the Apple Help Indexing Tool	41
Figure 2-13	Specifying a remote server in the Apple Help Indexing Tool	42
Figure 2-14	A link to an AppleScript script in a help page	46
Figure 2-15	The Preferences dialog of the Apple Help Indexing Tool	49
Table 2-1	Values of the ROBOTS meta tag	40

Chapter 3 **Registering Your Help Book 51**

Figure 3-1	The location of an English-language help book in the application bundle.	51
Figure 3-2	The targets view in Project Builder	53
Figure 3-3	The Expert settings in the Targets pane of Project Builder	54
Figure 3-4	The <code>CFBundleHelpBookFolder</code> key-value pair added to the <code>Info.plist</code> file in Project Builder	55
Listing 3-1	Registering a help book with <code>AHRegisterHelpBook</code>	56

Chapter 4 **Opening Your Help Book in Help Viewer 57**

- Table 4-1 Arguments to AHGotoPage 59
- Listing 4-1 Displaying an anchor location 57
- Listing 4-2 A function that searches your help book 58
- Listing 4-3 A function that loads a help book page 60

Appendix A **Apple Help Meta Tag Properties 63**

- Table A-1 Apple Help meta tags 63

Appendix B **Apple Help URLs 65**

- Table B-1 Help URLs 65

Appendix C **Apple Help Segments 69**

- Table C-1 Commands for Apple Help segments 69

Introduction to Providing User Assistance With Apple Help

Important: The information in this document is relevant to Apple Help development for Mac OS X v10.3 and earlier. For Mac OS X v10.4, the information in this document is superseded by the information in *Apple Help Programming Guide*.

This document describes Apple Help, the HTML-based system for providing online user assistance in Mac OS X. Apple Help is the primary help system for the Mac OS 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. If you are creating an application, plug-in, or other software product with a user interface for Mac OS X v10.3 or earlier, you should read this document to learn how to create an Apple Help help book and display it in Help Viewer.

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.

Apple Help offers significant advantages over static help documents, such as Read Me files or manuals in PDF, to teach your users how to use your products more effectively. The benefits of adopting Apple Help for user assistance include these:

- **Searchability.** Apple Help takes advantage of Sherlock technology to offer users sophisticated search capabilities, including full-text searching and the ability to search on synonyms and common misspellings.
- **Full support for QuickTime media.** 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.

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. 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:

INTRODUCTION

Introduction to Providing User Assistance With Apple Help

- The Help Viewer application. This is the default application for viewing user assistance in the Mac OS. 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 menu help), you need to call the Apple Help functions. The Apple Help API, although a Carbon API, is available to all Carbon, Cocoa, and Java developers.
- The Apple Help Indexing Tool. This is a developer tool provided by Apple for indexing your help book. When you run the Apple Help Indexing Tool on your help book, the tool generates an index file that Help Viewer uses to make your help searchable.

System Requirements

Help Viewer and the Apple Help API are available in Mac OS X version 10.0 and later. They are also available in Mac OS 8.6 and later for Carbon applications. The Apple Help Indexing Tool is available in Mac OS X in `/Developer/Applications/Utilities` when the Developer package is installed.

Organization of This Document

This document includes the following chapters and appendixes:

- [“Apple Help Concepts”](#) (page 11) describes the Help Viewer application and introduces the Apple Help API.
- [“Authoring User Help”](#) (page 23) shows how you can create a basic help book and describes how to use the Apple Help Indexing Tool to index your help book.
- [“Registering Your Help Book”](#) (page 51) describes how to register your help book with Help Viewer.
- [“Opening Your Help Book in Help Viewer”](#) (page 57) shows how to use the Apple Help functions to access and display help book content from your application..
- [“Apple Help Meta Tag Properties”](#) (page 63) lists the meta tags specific to Apple Help. You can use these tags to control how your help content is displayed.
- [“Apple Help URLs”](#) (page 65) lists Apple Help URLs that you can use to link to help pages and other resources.
- [“Apple Help Segments”](#) (page 69) lists the commands you can use to

See Also

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*.

INTRODUCTION

Introduction to Providing User Assistance With Apple Help

For guidelines on how to use help effectively within your application, see *Apple Human Interface Guidelines*.

INTRODUCTION

Introduction to Providing User Assistance With Apple Help

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

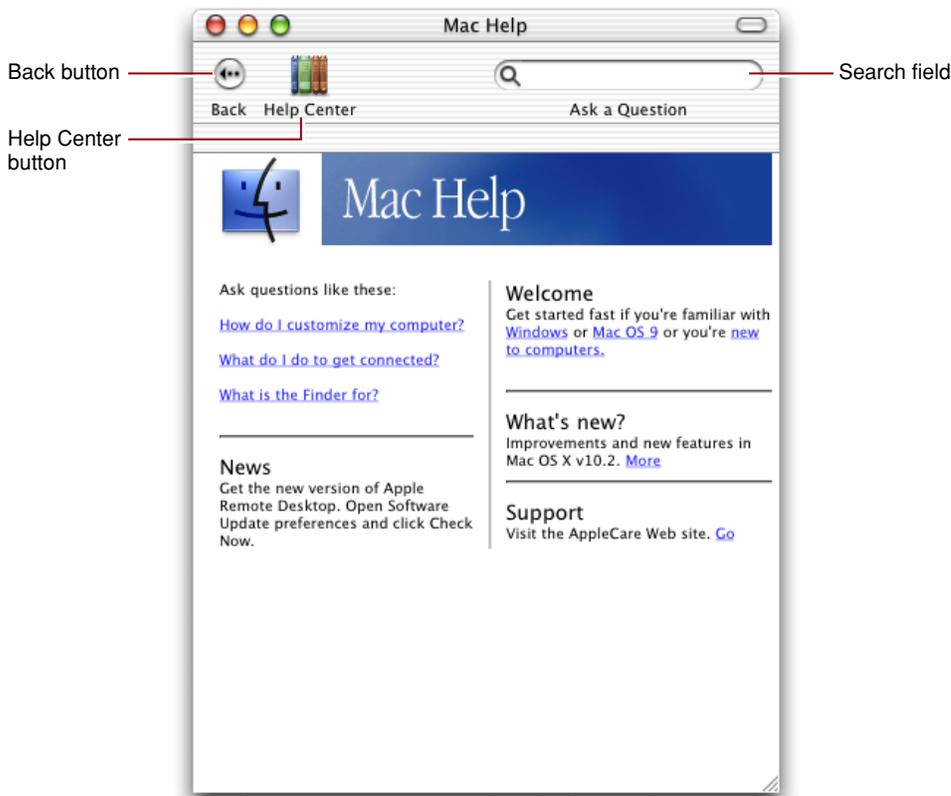
Users view online help in Help Viewer, a browser-like application designed to display HTML help content. Help Viewer displays files adhering to the HTML 3.2 specification. In addition, Help Viewer supports QuickTime media and AppleScript automation.

Note: Help Viewer does not currently support forms, JavaScript, or style sheets.

The Help Viewer Window

Users typically launch Help Viewer by selecting the application help item from the Help menu, described further in [“How Users Access Your Help”](#) (page 17). When Help Viewer launches, it brings up the window shown in Figure 1-1, 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



By default, there are three user interface elements in the toolbar at the top of the Help Viewer window: the Back button, the Help Center button, and the search field. The Back button on the left side of the toolbar allows users to navigate back to previously visited help pages; users can also view their navigation history and revisit pages using the Go menu. The Help Center button opens and closes the Help Center drawer; the Help Center is described further in “The Help Center” (page 15). The search field allows users to search all available help on the system.

Users can customize the Help Viewer toolbar by choosing the Customize Toolbar item from the View menu. Users can add additional controls, such as a Forward button, Print button, or font size buttons, or choose a different view for the toolbar.

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.

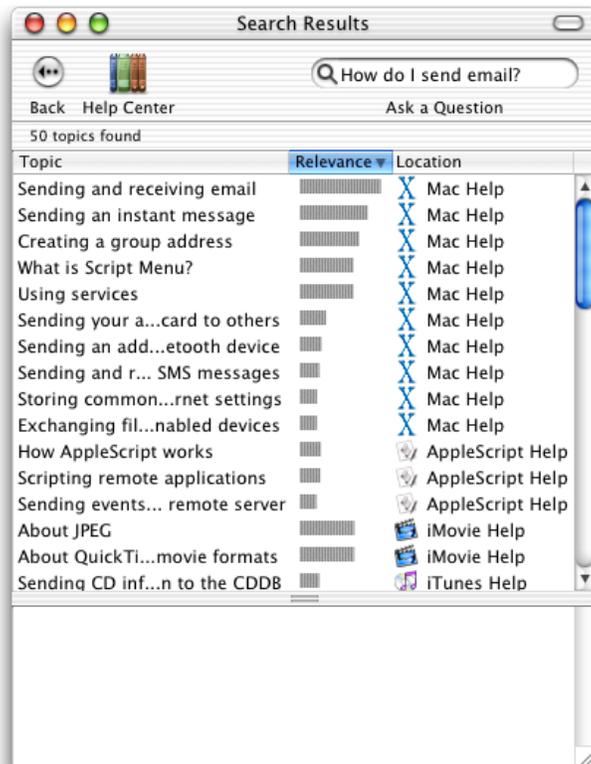
The text entry field at the right side of the Help Viewer window's toolbar allows users to search the available help content on the system. Users enter the term or concept for which they want to obtain help into the text field. They can also enter a question. Figure 1-2 shows a question entered in the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

Figure 1-2 A question entered in the search field of Help Viewer

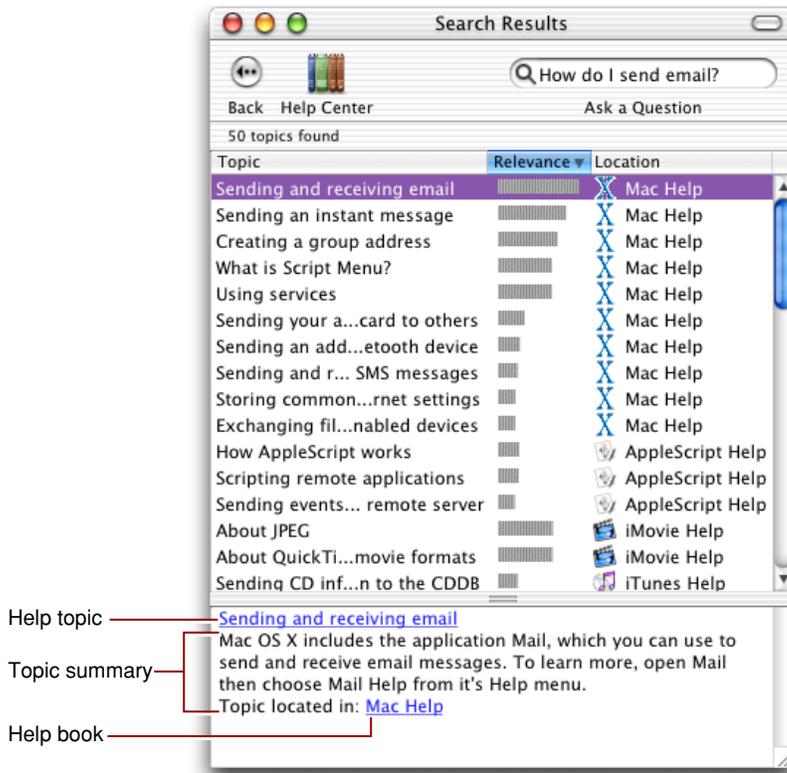


When the user presses Return, Help Viewer searches the help books installed on the system for the relevant term or terms. Figure 1-3 shows the search results returned by Help Viewer in response to the question entered in Figure 1-2.

Figure 1-3 Search results displayed in Help Viewer



Help Viewer displays the titles of each relevant help topic in a table of search results, along with the help book in which that topic is found. Search results are ranked by relevance, with the topmost hits having the highest relevance. When the user selects a topic from the search results, the bottom frame of the Help Viewer window displays a summary of the help topic, if available. The user can view the selected topic by double-clicking the topic in the search results table or by following the link in the bottom frame of the Help Viewer window. Figure 1-4 shows the summary for a help topic returned as a search result for the query entered in Figure 1-2 (page 13).

Figure 1-4 A topic summary displayed for a search result in Help Viewer

The Help Center

The Help Center, displayed in a drawer, lists all of the help books currently installed on the user's system. With the Help Center, users can easily access and browse all of the help available to them. Figure 1-5 shows the Help Center.

Figure 1-5 The Help Center



The Help Center is visible by default when Help Viewer is launched, but users can hide and show the Help Center using the button in the Help Viewer toolbar. Because the Help Center 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 selects a help book in the Help Center, Help Viewer loads that help book.

Help Books

To display help in Help Viewer, you must create and register a help book. As described in [“Introduction to Providing User Assistance With Apple Help”](#) (page 7), a help book is the collection of HTML files that constitute your help content. 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 should contain these two items:

- A default, or title, page. This is the HTML 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 20). 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 30).

- An index file. For a help book to be searchable, it must have an index file. You can create index files using the Apple Help Indexing Tool, described in [“Indexing Your Help Book”](#) (page 36).

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.

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 select your help book in the Help Center or choose the application help item from the Help menu, 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 14).

Important: You must register your help book for it to appear in the Help Center, be searchable in Help Viewer, and be accessible through the Apple Help API. For information on registering a help book, see [“Registering Your Help Book”](#) (page 51).

Internet-Based Help Book Content

In Mac OS 9 and later, 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.

You specify the server from which Help Viewer should retrieve content when you index your help book. When the user opens your help book, Help Viewer loads your title page. When the user follows a link to another page, Help Viewer looks for the page in the local help book folder. If this page is not present in the local help folder, Help Viewer contacts the server specified in the index and downloads the page.

Beginning with Mac OS X version 10.2, you can specify Internet-based help as your primary help book content. When you specify Internet-primary help, Help Viewer first checks for content on your remote server when loading a help page, even if that page exists in the local help folder. Using Internet-primary help, you can provide a complete installation of your help book locally, yet still provide updated help content over the Internet. Also beginning in Mac OS X version 10.2, Help Viewer supports updating indexes over the Internet.

To learn more about supplying content remotely, see [“Storing Pages on Remote Servers”](#) (page 42).

How Users Access Your Help

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

1. Selecting an item from 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.
2. 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 your application’s user help.
3. Selecting 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 selects 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, 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.

Note: When the user has multiple keyboard layouts selected in System Preferences, the keyboard layout menu is the rightmost menu in the menu bar. In that case, the Help menu is just to the left of the keyboard layout menu.

Figure 1-6 shows the Help menu in iTunes.

Figure 1-6 The Help menu



The Help menu is supplied by the system. The first item in the Help menu should be titled *ApplicationName* Help, 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. If you have registered your help book, the system not only creates the Help menu and the application help menu item for you, it also handles the choice of the application help menu item by opening Help Viewer to your application’s registered help book.

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

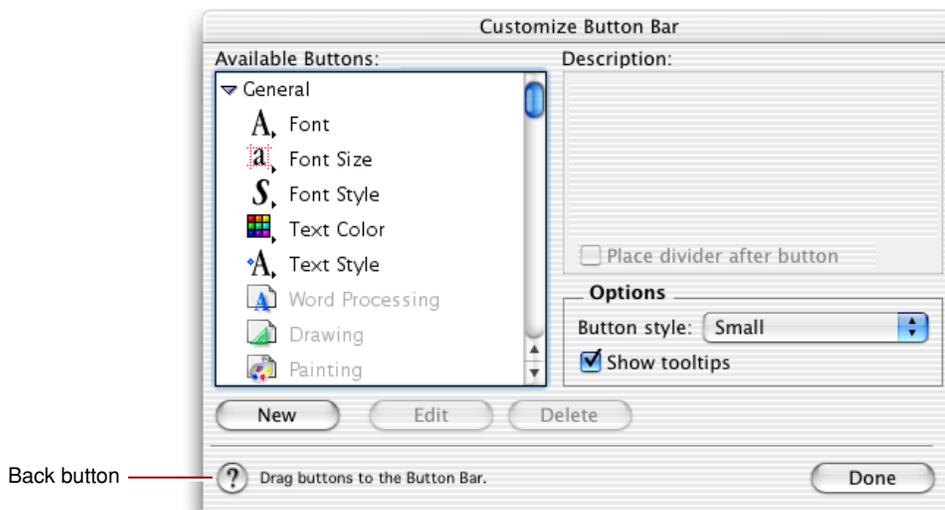
Help Buttons

A help button is a small round button displaying the standard help icon—a question mark.

When contextually relevant help is available, you can use help buttons in the lower-left corner of a window or dialog to take users directly to the pertinent information. For example, you can use a help button in a Preferences dialog to take users directly to information about setting preferences for your application.

Figure 1-7 shows a typical help button, in the lower-left corner of an AppleWorks dialog.

Figure 1-7 A help button

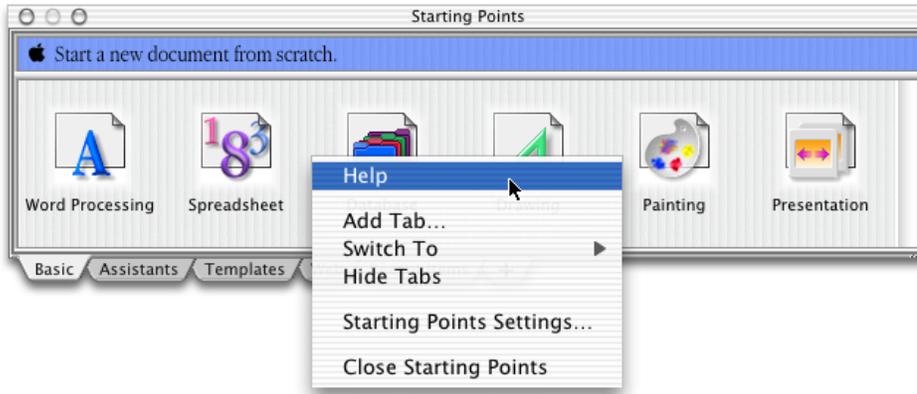


You should use help buttons only when contextually relevant help exists for the current window or dialog. Help buttons should allow users to obtain assistance for the task at hand without searching through your help book.

When the user clicks the help button, your application should load the relevant help topic in Help Viewer, using the Apple Help functions `AHLookupAnchor` or `AHGotoPage`. See [“The Apple Help Application Programming Interface”](#) (page 21) for more information on using these functions.

Help in Contextual Menus

You can also provide context-specific user assistance through contextual menu items. Contextual menus are displayed when the user Control-clicks an object in an application’s user interface. If help pertaining to that object exists—for example, a description of how to use that user interface element—the first item in the contextual menu should be a help item. Figure 1-8 shows a contextual menu containing a Help item.

Figure 1-8 Help in a contextual menu

If the user selects the help item from the menu, you should use the function `AHLookupAnchor` or `AHGoToPage` to load the relevant page of your help book in Help Viewer. To learn more about using Apple Help functions to access your help book, refer to “[Opening Your Help Book in Help Viewer](#)” (page 57).

Help Viewer HTML Extensions

Help Viewer recognizes a number of help-specific HTML extensions, which you can use in your help book to control how Help Viewer displays and accesses your help. These extensions to the HTML 3.2 specification fall into three categories:

- Meta information. Help Viewer uses this meta information 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 Help Center and in search results. You can also use these meta tags to control how your help is indexed. “[Authoring User Help](#)” (page 23) describes many of these meta tags and how you can use them. For a complete list of the help-specific meta information recognized by Help Viewer, see [Table A-1](#) (page 63).

Note: Most of the meta information recognized by Help Viewer is optional. However, you must include the `AppleTitle` meta tag in 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, their main advantage 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 described in more detail in the section “[Adding Specialized Content to Your Help Book](#)” (page 44). For a full listing of the supported help URLs, see [Table B-1](#) (page 65).

Apple Help Segments

Help Viewer also recognizes certain HTML comments that you can use to split an HTML file in 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 Pages](#)” (page 39) describes segments in more detail. See [Table C-1](#) (page 69) for a complete list of the comments denoting segments.

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. 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 you have registered your help book, the system opens the book in Help Viewer for you when the user selects 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 52).

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
<code>AHLookupAnchor</code>	Opens your help book to the section or page identified by the given anchor.

Function name	Action
AHSearch	Searches your help book for a given text string.
AHGoToPage	Opens a help book page at a known location.

If you know the exact location within your help book of the help topic you wish to display, you can use the `AHGoToPage` function to load the topic in Help Viewer. `AHGoToPage` requires that you know the full or partial path to the HTML file describing the desired help topic. `AHLookupAnchor`, on the other hand, allows you to access a help topic with only the anchor name; in most cases, this approach is more flexible than tracking the location of the file describing that 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*.

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 57).

Authoring User Help

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

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

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, and how can they solve problems doing it. Depending upon the complexity of the task, a well-designed help page may cover the first two questions in two sentences, the third question in one, 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 3.2 specification. For a comprehensive description of the HTML 3.2 specification, see

<http://www.w3.org/TR/REC-html32.html>

Authoring Tools

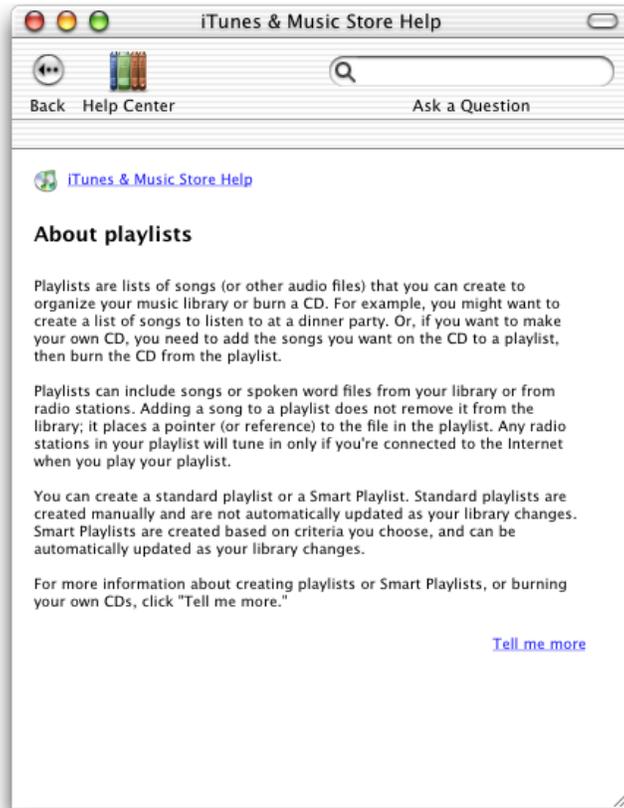
You can author your help book in any application that generates valid HTML 3.2 files. Likewise, you can view your help book in any HTML 3.2–qualified browser; however, you should always test your help book in Help Viewer to ensure that your help book displays properly. If you already have HTML help content, there are tools available to help you quickly and easily convert your files to Apple Help content. For a list of some of the tools you can use to create help content for your help book, see

<http://developer.apple.com/ue/help/>

Note: Apple Help and Help Viewer in Mac OS 9 and later support all HTML 3.2 features except forms, JavaScript, and style sheets. Help Viewer in Mac OS X version 10.3 and later adds support for JavaScript and style sheets.

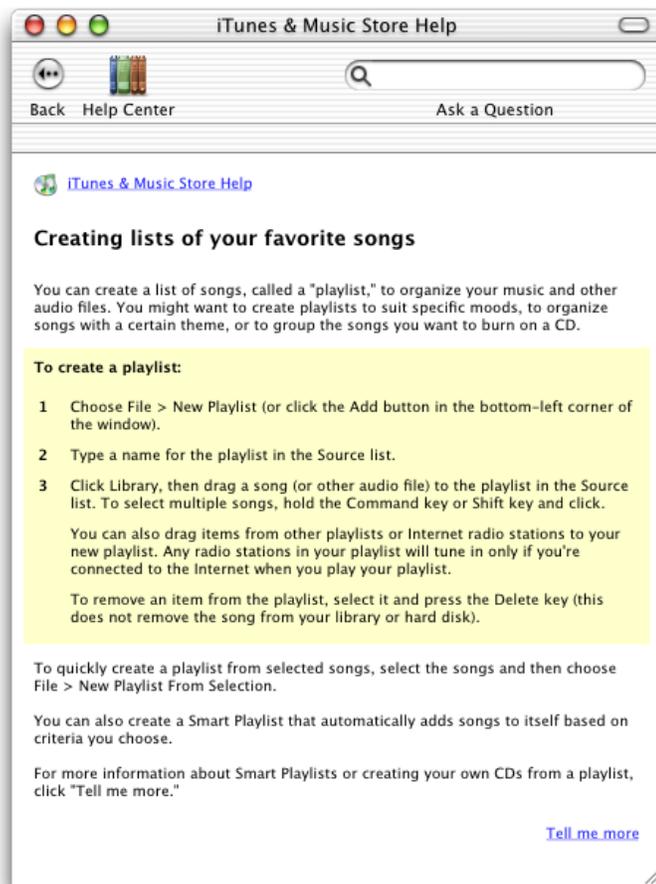
Creating Topic Pages

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 shown in Figure 2-1 defines a playlist, a concept fundamental to the iTunes application.

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 add a song from a CD to the user's playlist.

Figure 2-2 A task-oriented help book page



Topic pages typically include these elements:

- A title identifying the topic. In Figure 2-1, “About playlists” identifies “playlists” as the topic; in Figure 2-2, “Creating lists of your favorite songs” identifies the creation of playlists as the topic.
- The topic introduction. For an overview page, this section defines the relevant terms or concepts. For example, the first two paragraphs in Figure 2-1 define a “playlist” and why it is useful. For a task-oriented page, the introduction provides context for the task description that follows and explains why or when the user would perform the task. In Figure 2-2, the first paragraph briefly summarizes the definition of a playlist and gives a few reasons why you might want to create one.
- 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. [Figure 2-2](#) (page 26) shows the three steps necessary to create a new playlist. 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) links to related information on creating a playlists using Smart Playlists.

For an example of a basic topic page, see the template file `TopicTemplate.htm` located in `/Developer/Documentation/AppleHelp/Templates/`.

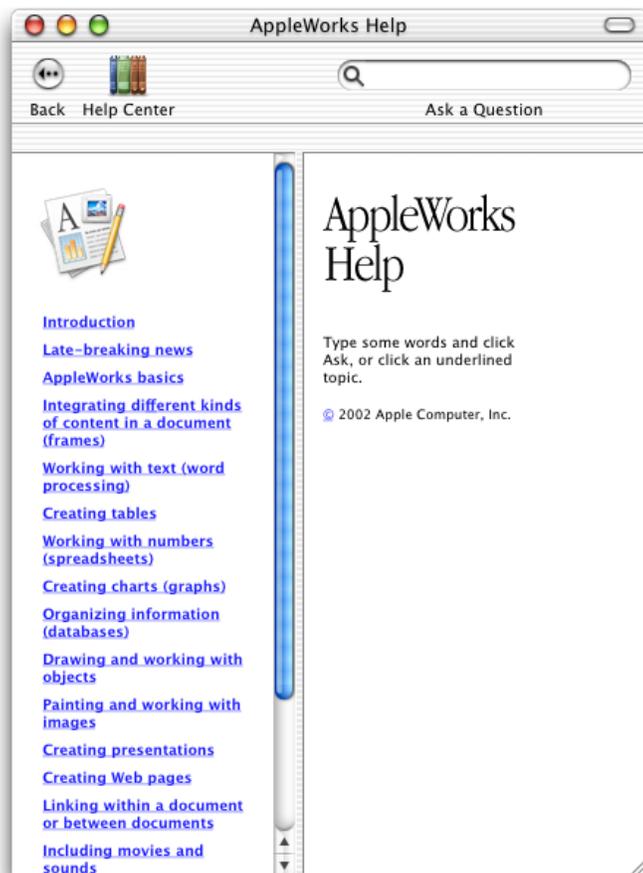
Creating Navigation Pages

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

The top-level table of contents page in your help book allows the user to select a starting point within your help book. A top-level table of contents gets the user started in finding help, even if they do not quite know what they are looking for. [Figure 2-3](#) shows the top-level table of contents for the AppleWorks application.

Figure 2-3 The top-level table of contents for AppleWorks Help

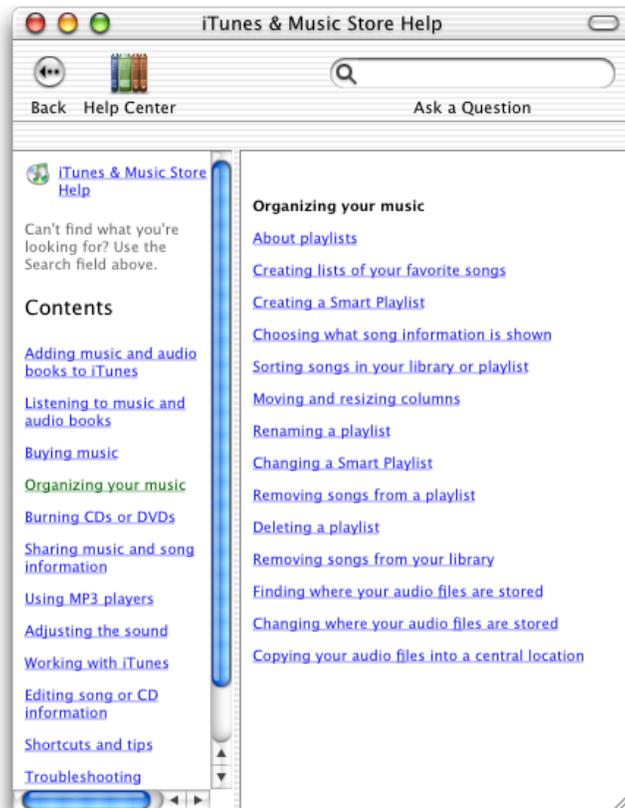


If your help book is chapter based, you should provide a separate table of contents for each chapter. Chapter-based help books are described further in [“Creating a Chapter-Based Help Book”](#) (page 33).

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.

Even if your help book is not chapter-based, you can create navigation pages to group related subtopics. This allows users to see and navigate to each of these table of contents subtopics. Figure 2-4 shows a page from iTunes Help. On the left side is the top-level table of contents for iTunes Help; on the right side is a table of contents listing the subtopics related to editing information in iTunes.

Figure 2-4 A topic level table of contents in iTunes Help



Guidelines for Help

Apple Human Interface Guidelines offers information on effectively incorporating help into your application.

Creating a Basic Help Book

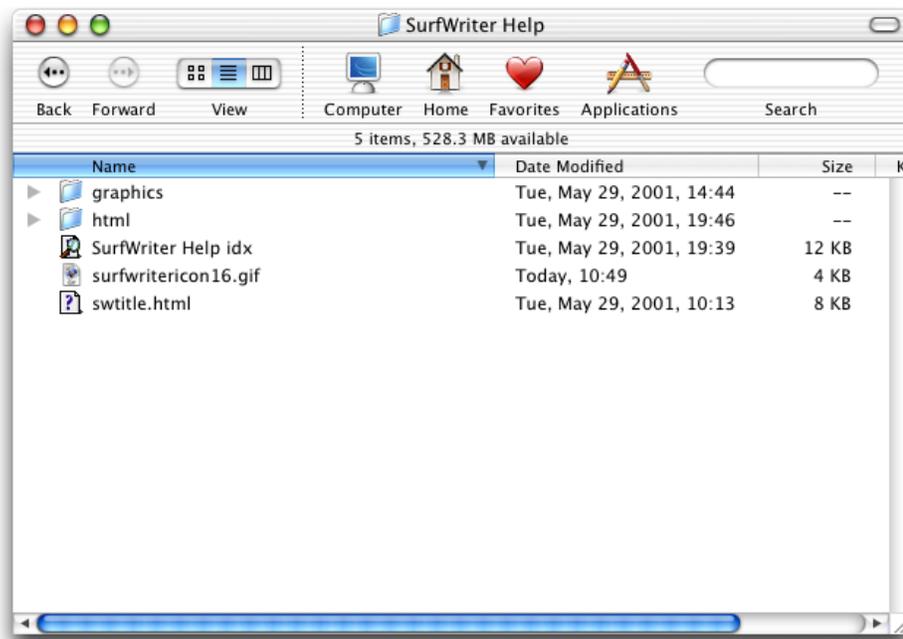
Once you create the HTML files containing your help content, you must organize them into a help book. To do this, create a help book folder and include the following items:

- 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. This is the HTML 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 Help Center.

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 one subfolder for your HTML help files and one subfolder for graphics. Figure 2-5 shows an example of a simple help folder for SurfWriter Help.

Figure 2-5 An example of a simple help book folder structure



The three files at the top level of the SurfWriter Help folder are the index file, the title page, and the book's icon. The title page is described in the next section, "[Creating a Title Page](#)" (page 30). "[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 36).

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 selecting the application help item from the Help menu or by selecting your help book in the Help Center. 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, welcomes the reader and offers a number of links to help pages answering common queries to get the reader started.

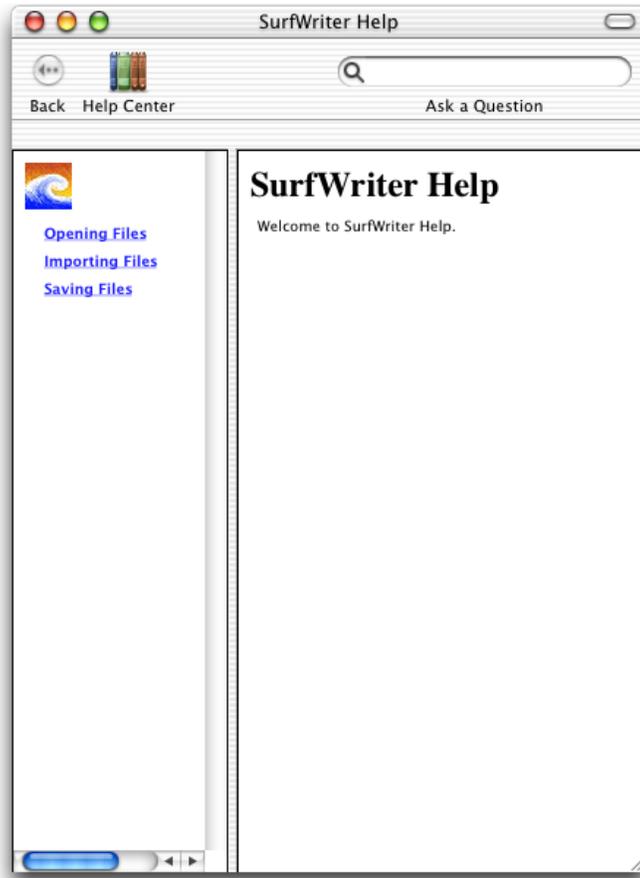
You can also specify a frameset as the title page, as AppleWorks Help does. AppleWorks Help uses frames to combine an introductory page with a table of contents that allows readers to see what topics are covered by the help book. [Figure 2-3](#) (page 28) shows the AppleWorks Help title page.

To specify the title page of your help book, include the `AppleTitle` meta tag in the header section of the HTML file you want to use as your book's default page. This HTML file must reside at the top level of your help book folder, as shown in [Figure 2-5](#) (page 30). 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 2-6 shows the title page for SurfWriter Help.

Figure 2-6 The SurfWriter Help title page



For a good example of a basic title page, see the template file `TOCTemplate.htm` installed by the Mac OS X Developer Tools CD in the directory `Developer/Documentation/AppleHelp/Templates/`.

Note: In Mac OS 8 and 9, the title page file must have a creator code of 'hbwr' in order to associate the file with the Help Viewer.

Specifying a Help Book Icon

When you register a help book, your help book becomes visible in the Help Center. If you provide an icon, Help Viewer displays this icon in the Help Center drawer, 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.

```
<META NAME="AppleIcon" CONTENT="SurfWriter%20Help/surfwritericon16.png">
```

Figure 2-7 shows SurfWriter Help in the Help Center, with the SurfWriter icon next to the book title.

Figure 2-7 The SurfWriter help book in the Help Center



In the example help book folder shown in [Figure 2-5](#) (page 30), the icon file is placed at the top level of the SurfWriter Help folder, but you can place the icon file in a graphics subfolder and specify the path accordingly. The icon should be a 16-by-16 pixel version of your application icon, saved as a .PNG file.

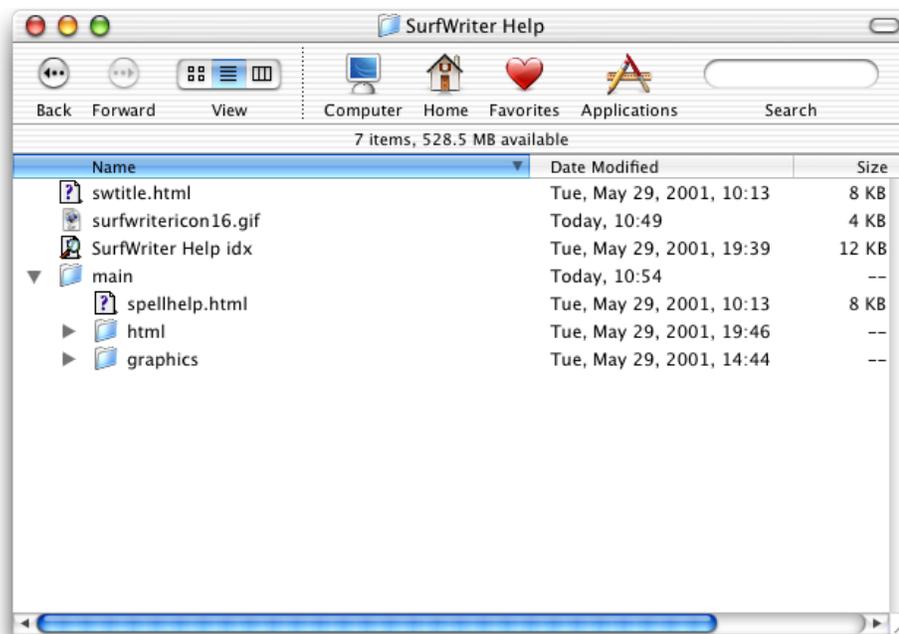
Creating a Chapter-Based Help Book

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

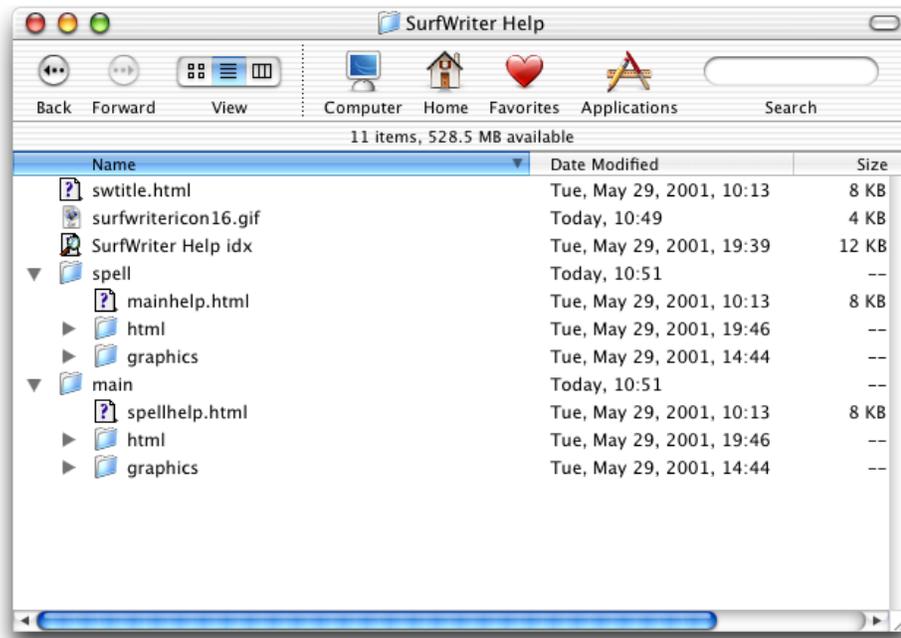
Each chapter has its own title page and may contain its own index and subfolders. For example, SurfWriter features several optional components, so the help book is organized into a main folder and a subfolder for each component. “[SurfWriter help book with main folder installed](#)” (page 33) shows the SurfWriter help book with only the main folder installed.

Figure 2-8 SurfWriter help book with main folder installed



“[SurfWriter help book with optional chapter installed](#)” (page 34) shows the SurfWriter help book after the user installs the optional spelling checker. Note that each subfolder has its own title page.

Figure 2-9 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. 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>
```

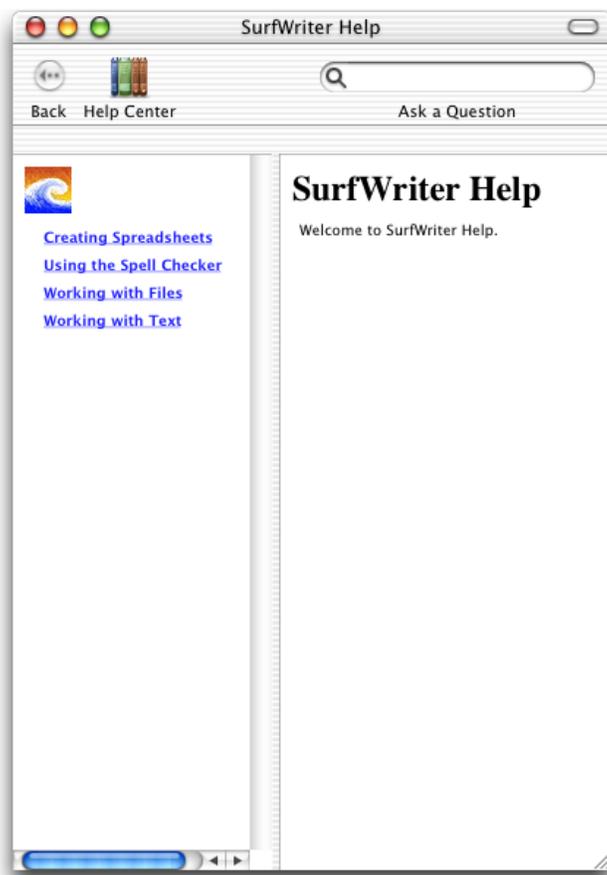
```

</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` command in the template file is copied over to the dynamically generated table of contents without change. Figure 2-10 shows the dynamically generated table of contents for SurfWriter Help in the left frame.

Figure 2-10 The generated table of contents for SurfWriter Help



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` commands. You should also include any other relevant title page information, such as your help book icon. If you specify a frameset for your top-level title page, you can include the generated table of contents by linking to the `toc.htm` file. The frameset for the title page for SurfWriter Help, shown in Figure 2-10 (page 35), incorporates the dynamically generated table of contents in this way.

For more examples of dynamically generated tables of content, see `/Developer/Documentation/Apple Help/Sample Books`.

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 the Help Viewer creates a dynamic table of contents for your help book, it lists each chapter in the order specified by the `AppleOrder` values, 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. To make your help book searchable, you must run the Apple Help Indexing Tool to create an index for your help book. The indexing tool is described in ["Using the Apple Help Indexing Tool"](#) (page 40).

You should also include additional tags and meta information 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"](#) (page 37) describes how you can improve indexing and searching of your help book.

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 Apple Help Indexing Tool. 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 previews 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 large help systems.

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, erase, scrap">
</HEAD>
```

Adding Abstracts

An abstract is a brief description of a help topic that appears in the bottom frame of the Help Viewer window when the user selects that topic from a list of search results.

Note: Beginning in Mac OS X version 10.2, Help Viewer displays the default text “No summary provided” for help topics without abstracts.

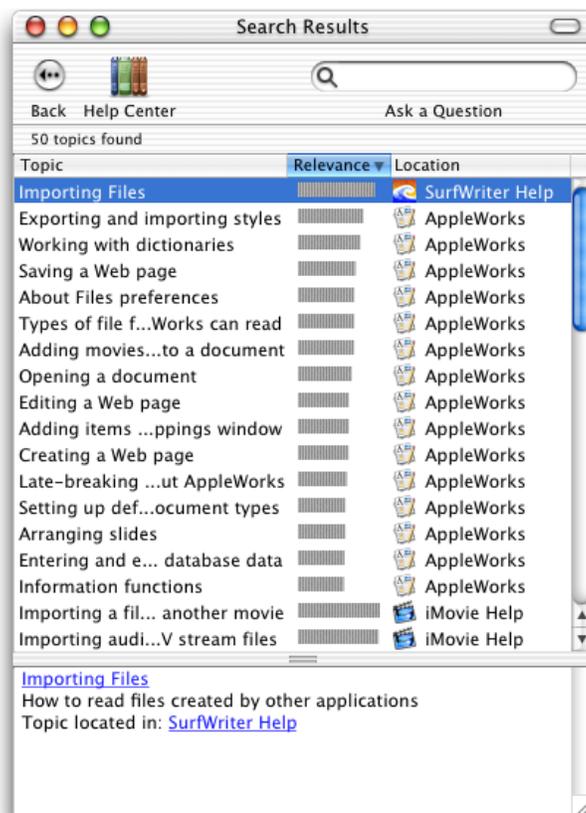
Well-written abstracts help users ascertain whether a given page, returned as a search result, in fact contains the information they were searching for. For example, SurfWriter Help contains a page describing how to import files from other formats. The page’s title, “Importing Files,” gives the user some idea of the page’s contents, but you can provide a fuller description by using an abstract phrase such as, “How to read files created by other applications.”

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>Importing Files</TITLE>
<META NAME="DESCRIPTION" CONTENT="How to read files created by other
applications">
</HEAD>
```

“Example of a search result showing an abstract” (page 38) shows how such a page shows up as a search result.

Figure 2-11 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 and scrolls to its 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 pages for SurfScript, you could specify a unique anchor for each SurfScript command. This allows Help Viewer to scroll directly to the desired text when the page loads.

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.

Note: The Apple Help function `AHGoToPage` also allows you to specify an anchor location as an argument.

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, described in the next section.

You specify an anchor using the standard HTML 3.2 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-->
```

If you use anchors in your help book, you must turn anchor indexing on when you index your help book with the Apple Help Indexing Tool. For more information, see [“Specifying Anchor Indexing”](#) (page 41).

Creating Segments in Help Pages

A large help book may contain hundreds of individual pages. If you need to reduce the number of files in your help book, you can combine several pages into one through the use of subsections called 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" -->
<!-- AppleKeywords="launch, read, look , examine" -->
<!-- Topic text goes here-->
<!-- 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
<code>NOINDEX</code>	Specifies that the HTML file should not be indexed.
<code>KEYWORDS</code>	Specifies that the HTML file should be indexed for keywords only.
<code>SEGMENTS</code>	Specifies that the HTML file should be indexed for the content of the segments in that file.
<code>ANCHORS</code>	Specifies that the HTML file should be indexed for anchors only.

Apple recommends that you do not index files that contain only links or graphics. 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">
```

Specifying `KEYWORDS` as the value of the `ROBOTS` meta tag may be useful if you have a keyword file that is separate from its associated topic pages, for example. If you define segments in a file, as described in [“Creating Segments in Help Pages”](#) (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 main entry page.

Using the Apple Help Indexing Tool

Once you have finished adding abstracts, keywords, and any other indexing information to your help book files, you must run the Apple Help Indexing Tool on your help book. The Apple Help Indexing Tool creates an index for your help content and stores it as a separate file at the top level of the help book folder. The index file may be invalidated if you change or rearrange the content of the help folder, so you should wait to create the index file until the help book content is complete and all files are in their final location.

You can create an index for your help book simply by dragging the folder containing the help book onto the Apple Help Indexing Tool. However, you must modify the default preference settings of the Apple Help Indexing Tool if you are doing any of the following:

1. Indexing a help book for a language other than English. Read [“Localizing Your Help Book”](#) (page 47) for more information on indexing non-English help books.
2. Using anchors.
3. Supplying help content from a remote server.

Specifying Anchor Indexing

If you add anchors to your help book, you must index your help book with anchor indexing selected in the Apple Help Indexing Tool's preferences so that Help Viewer can find your anchors. To turn on anchor indexing, perform the following steps:

1. Open the Apple Help Indexing Tool Preferences dialog, by selecting Preferences from the application menu.
2. Click the check-box labeled "Index anchor information in all files," which is near the bottom of the dialog shown in Figure 2-12.

Figure 2-12 Turning on anchor indexing in the Apple Help Indexing Tool

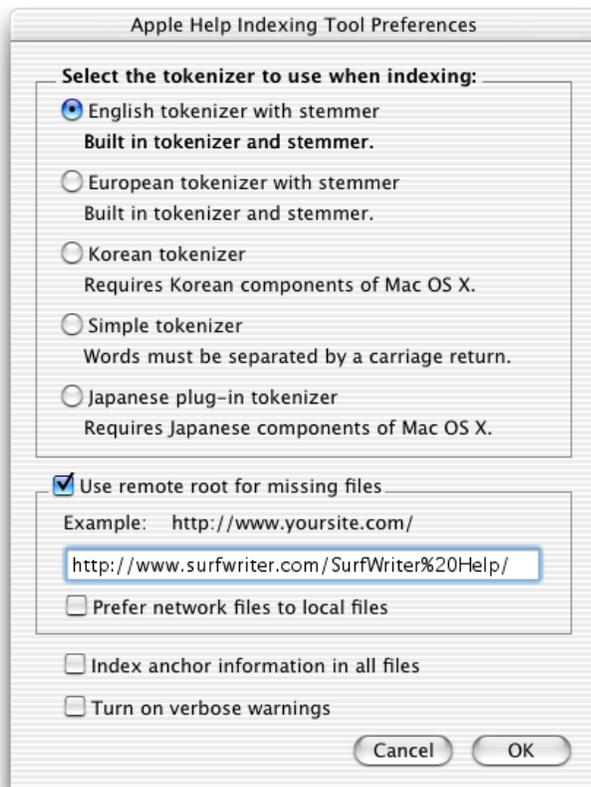


Storing Pages on Remote Servers

As described in “[Internet-Based Help Book Content](#)” (page 17), Help Viewer supports Internet-based help in Mac OS X version 10.1 and later. To provide updates to your help content via the Internet, you must specify a remote server from which to download updated help content before you index your help book. To specify a remote server for your help content you must do the following:

1. Open the Preferences dialog for the Apple Help Indexing Tool.
2. Click the check-box labeled “Use remote root for missing files.”
3. In the text field under this checkbox, enter the server address where your remote help content is available. Figure 2-13 shows a remote root specified in the indexing tool preferences.

Figure 2-13 Specifying a remote server in the Apple Help Indexing Tool



By default, Help Viewer downloads only content that is missing from your locally installed help book folder. However, in Mac OS X version 10.2 and later, you can specify “Internet-primary” help by clicking the check-box labeled “Prefer network files to local files” in the Apple Help Indexing Tool preferences. When you choose Internet-primary help, Help Viewer checks the server specified in your index file for help content, regardless of the content currently installed on the local system.

Note: If you do not choose Internet-primary help, or if you are running on a version of the Mac OS prior to Mac OS X version 10.2, you can provide Internet-based help by installing a minimal help folder on the user's system that contains a title page, index, and any other pages that should be quickly accessible. Any files that you want to update 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.

Remember that remote-page access is not available to users running systems prior to Mac OS 9, so you must install all content locally for such users.

When the user loads a page from your help book, Help Viewer looks for the page in your local help folder. If the page is not found in the help folder, or if you have specified Internet-primary help, Help Viewer contacts the server specified in the index and obtains the page. Help Viewer caches remotely accessed pages on the user's system.

Beginning with Mac OS X version 10.2, 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 have specified a remote root, Help Viewer contacts the server and checks for an index file at that location. If an index file is present, Help Viewer downloads the file. Help Viewer also purges the cache of any HTML or graphics files downloaded using the remote root from the previous index file; this ensures that users see only content relevant to the latest and most up-to-date index 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.

In Mac OS X version 10.2 and later, you can control how your long 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. If you are using specific cache expiration dates for your help pages in conjunction with index files updated over the Internet, the relevant portion of Help Viewer's cache can be flushed completely whenever a new index file is downloaded, making your cache expiration dates irrelevant. The second example tells Help Viewer that the file should never expire.

Note: In earlier versions of the Mac OS, Help Viewer caches all pages downloaded from the Internet for three days.

Automating the Apple Help Indexing Tool

The Apple Help Indexing Tool is scriptable, allowing you to automate the indexing process with AppleScript. For example, the following script defines a droplet.

```
on open these_items
tell application "Finder"
    activate // 1
    set folderAlias to these_items as alias // 2
end tell
```

```

tell application "Apple Help Indexing Tool"
    activate // 3
    turn remote root "on" with root url "http://www.mycompany.com/myhelp/" // 4
    use tokenizer "English" // 5
    turn anchor indexing "on" // 6
    open folderAlias // 7
end tell
end open

```

When you drag your help book folder to the droplet, the Finder sends an Open command to the droplet, invoking the script. The identity of the help book folder is passed to the script in the `these_items` parameter. Here is what the script does:

1. Activates the Finder application.
2. Tells the Finder to create an alias, stored in the `folderAlias` variable, to the help book folder.
3. Activates the Apple Help Indexing Tool.
4. Specifies a remote server for help content.
5. Sets the tokenizer to use while indexing.
6. Turns on anchor indexing.
7. Runs the indexing tool on the help book folder.

See the Apple Help Indexing Tool's AppleScript dictionary for more information on the AppleScript commands supported by the indexing tool. You can view the indexing tool's AppleScript dictionary by selecting Open Dictionary from the File menu in the Script Editor application. Double-click the name of the Apple Help Indexing Tool in the dialog that appears.

For more information on AppleScript and scripting, see the AppleScript documentation in the ADC Reference Library.

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`.

Using Help URLs in Your Help Book

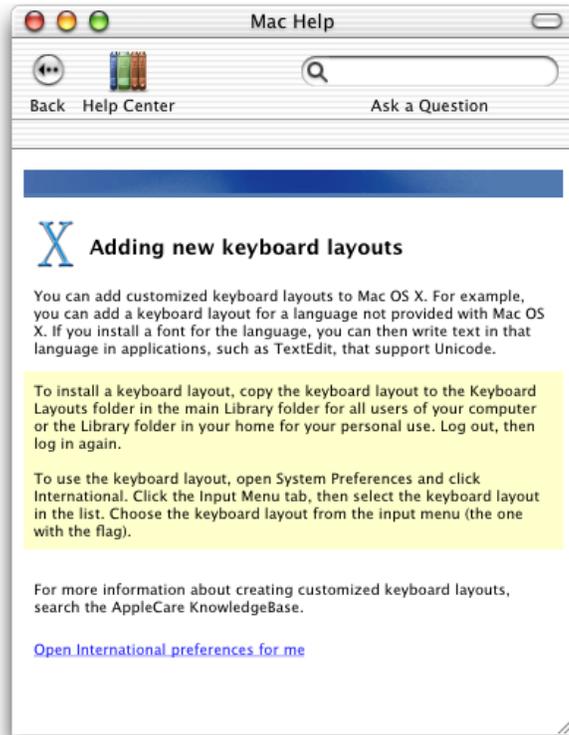
The URLs using the Help Viewer `help:` protocol, introduced in “[Help-Specific Meta Tags](#)” (page 20), allow you to create links to other help content in your help book, including

- AppleScript scripts
- Help Viewer searches
- anchor locations

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 2-14 shows a page from Mac Help that uses an AppleScript script to open the International preferences pane in System Preferences when the user clicks the link at the bottom of the page.

Figure 2-14 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. You can enclose arguments to help URLs in single quotes, as the string values is shown in the previous example, or you can use standard URL encoding; for example, the folder name “SurfWriter Help” is specified as `SurfWriter%20Help`. For more information on standard URL encoding, see

<http://www.w3.org/Addressing/>

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='How do I import files?' bookID=SurfWriter%20Help">
```

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 "How do I import 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. 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 "[Specifying Anchor Indexing](#)" (page 41).

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/>

Specifying Character Encoding

To specify the character encoding used by your help book, use the standard HTML 3.2 syntax, as shown in the example below, which specifies the Latin-1 character encoding:

```
<META HTTP-EQUIV="content-type" CONTENT="text/html; charset=iso-8859-1">
```

In versions of Mac OS prior to 10.1, Help Viewer does not respect the character set tag when rendering HTML. Furthermore, prior to Mac OS X v10.1, Help Viewer supports only a limited number of character encodings. In particular, Help Viewer supports the MacRoman, MacJapanese, MacKorean, MacChineseTrad, and MacChineseSimp encodings.

Note: Although Help Viewer in Mac OS X v10.1 and later renders Unicode content properly, it currently does not index or search Unicode content.

Specifying Fonts For Display in Help Viewer

If you have a localized help book that uses a non-Roman font, you can specify which font Help Viewer uses for your help book title when it displays your help book in the Help Center, as well as the font used by Help Viewer to display search results from your help book.

To specify a font for your book's Help Center listing, use the `AppleFont` property for the `<META>` element in your help book title page. For example, here is how you would specify the Osaka font for your Help Center listing:

```
<META NAME="AppleFont" CONTENT="Osaka">
```

If you need to specify a non-Roman font in which to display search results for your help book, you can use the `AppleSearchResultsFont` property in the header of your title page file. Here is an example of how to specify the Osaka font to display search results:

```
<META NAME="AppleSearchResultsFont" CONTENT="Osaka">
```

Indexing a Non-English Help Book

Once you have created your localized help book, you must run the Apple Help Indexing Tool on the book. To index a non-US English help book, open the Preferences dialog in the Apple Help Indexing Tool by choosing Preferences from the application menu. From the radio button group labeled "Select the tokenizer to use when indexing," choose the appropriate setting for your target language. Figure 2-15 shows the Preferences dialog of the Apple Help Indexing Tool.

Figure 2-15 The Preferences dialog of the Apple Help Indexing Tool

Apple Help Indexing Tool 1.3 supports U.S. English, Western European English, Japanese, Korean, and Chinese language indexing. To use Chinese language indexing, choose the Simple Tokenizer option in the Apple Help Indexing Tool Preferences; this tokenizer supports both the Traditional and Simplified character sets.

Note: In Mac OS 8 and 9, you must run the Indexing Tool in Mac OS-J (Japanese) 8.5 or newer to index Kanji text. The Japanese Language Kit does not contain the tokenizer required by the Indexing Tool.

Registering Your Help Book

This chapter describes how to register your completed help book so that users—and your application—can access it. When you register your help book, it becomes available in the Help Center and from the Help menu. All developers creating help books for the Mac should read this section to learn how to add their help books to their software products.

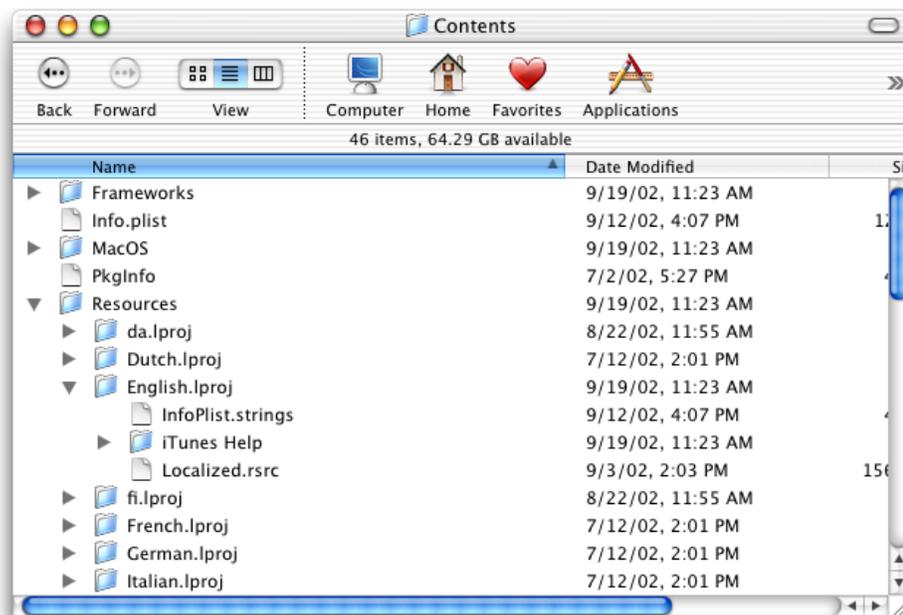
Where to Place Your Help Book Folder

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 application. You can also place your help book at one of the standard locations for third-party help, `/Library/Documentation/Help` or `Library/Documentation/Help` in the user's home directory. If you place your help at one of these locations, you do not need to register your help book.

Important: The `.help` bundles are reserved for use by Apple; use by third-party developers is not supported.

How to Register Your Help Book

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 selects this item. In addition, your help book must be registered for it to appear in the Help Center.

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`.

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 register your help book. If you do not register your help book or place it in one of the known help locations, you are responsible for handling all access to your help book yourself, using the `AHGotoPage` function, 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:

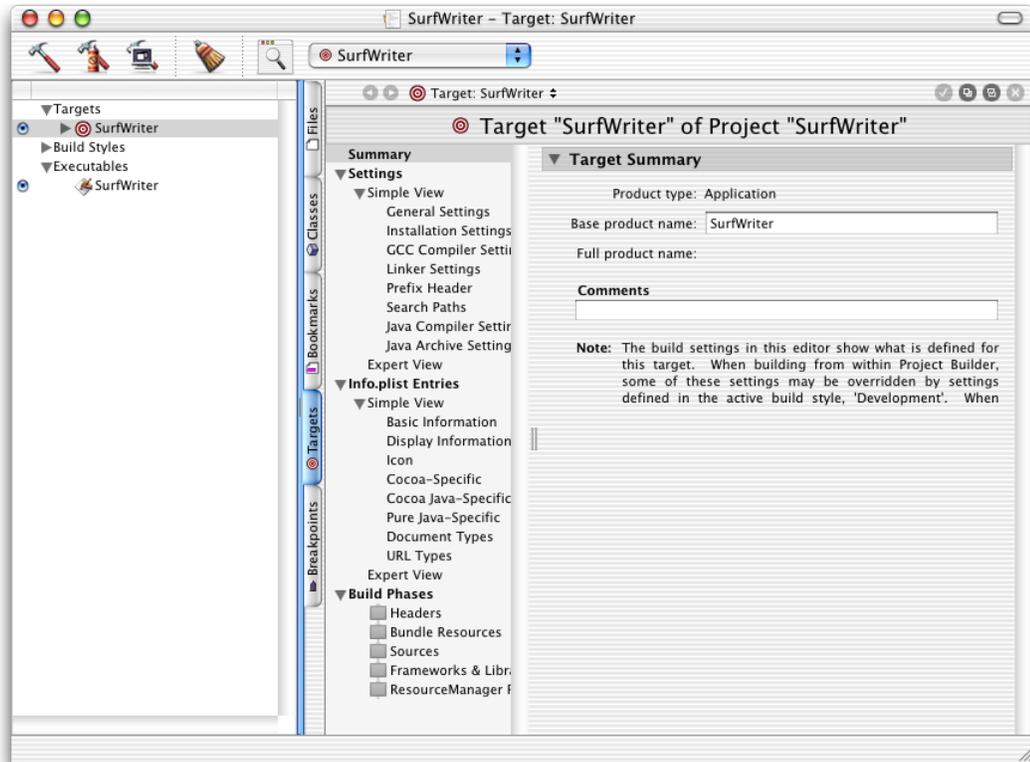
```
<key>CFBundleHelpBookName</key>
<string>SurfWriter Help</string>
```

If you are using Project Builder 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, click on the Targets tab. In the left column, your project's targets appear.

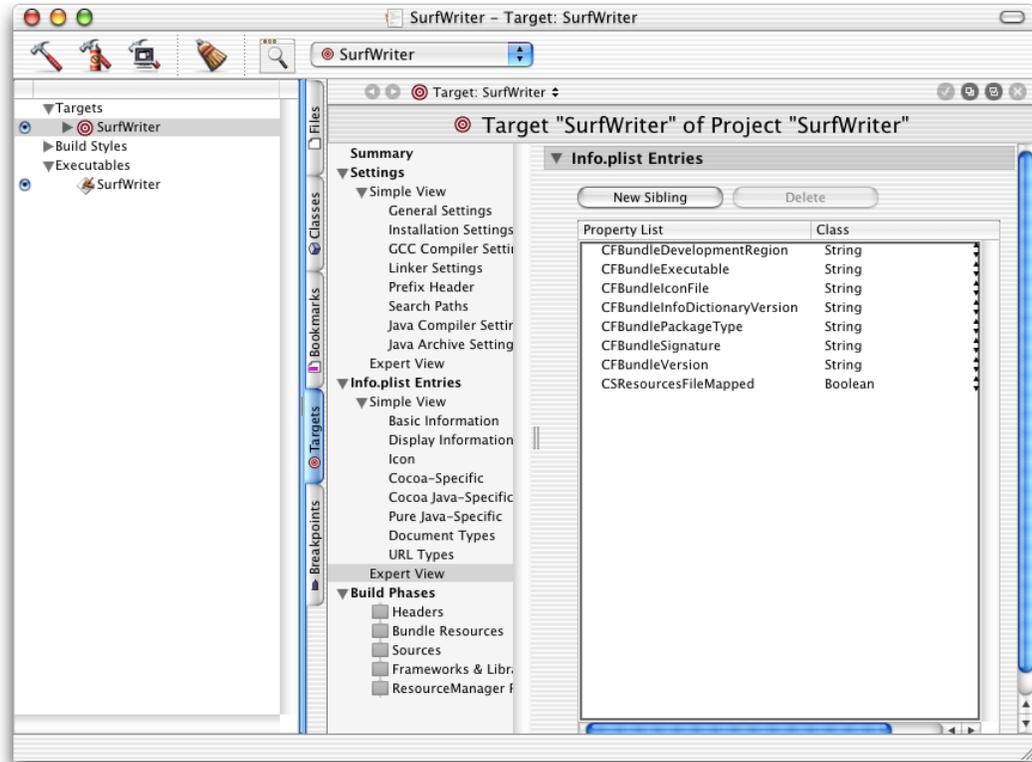
2. Select the target for which you want to edit the `Info.plist` file. You should see a window such as that shown in Figure 3-2.

Figure 3-2 The targets view in Project Builder



- Under Info.plist Entries, select the Expert View item. You should see a pane similar to that shown in Figure 3-3.

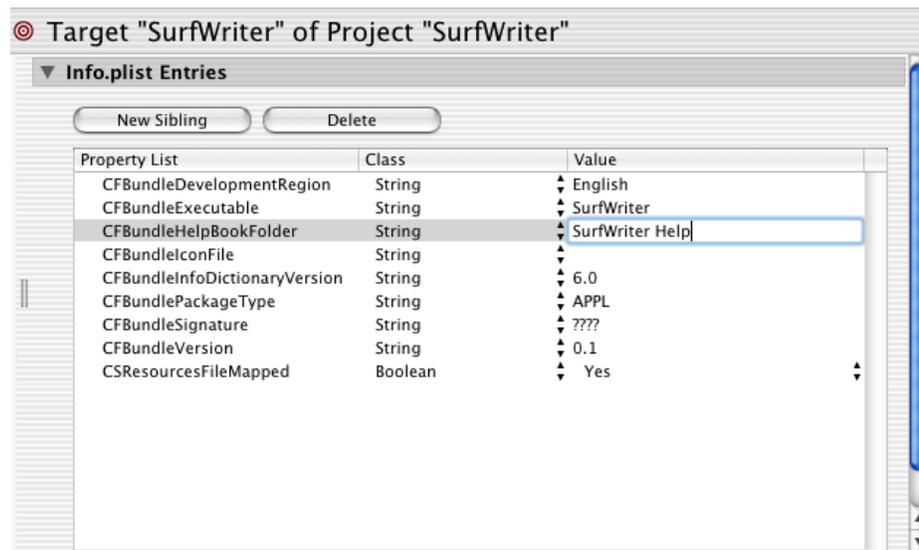
Figure 3-3 The Expert settings in the Targets pane of Project Builder



- Click the New Sibling button to create a new key-value pair. Type `CFBundleHelpBookFolder` for the new item's key.

- Double-click in the Value column to edit the value associated with the key. Type the name of the folder containing your application's main help book. For example, Figure 3-4 shows entry for the SurfWriter application's help book folder.

Figure 3-4 The `CFBundleHelpBookFolder` key-value pair added to the `Info.plist` file in Project Builder



- Click the New Sibling button to create another key-value pair. Type `CFBundleHelpBookName` for the key. Enter the name of your main help book, as it appears in the title page.

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

Carbon applications and Cocoa or Java 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 the 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 Help Center.

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.
- `AHGotoPage` opens a help book page in Help Viewer.

Displaying an Anchor Location

If you specify anchor locations in your help book, as described in “[Indexing Your Help Book](#)” (page 36), 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;
}

```

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 3 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 57):

```
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.

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);
    } 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 58) to search your help book for information on printing. You can use a full question for your query, such as "How do I print a document?" or you can search for a term, such as "print".

```

err = SearchHelpBook(CFSTR("How do I 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, use `AHGoToPage`.

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 or installed at one of the system help locations, 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 <code>AHGoToPage</code>	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

Arguments provided to AHGotoPage	Results
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 [Listing 4-3](#) (page 60):

```
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.	<META NAME="AppleTitle" CONTENT="My Application Help">
AppleIcon	The help book icon file.	<META NAME="AppleIcon" CONTENT="My%20Application%20Help/myhelpicon.gif">
AppleFont	The font used by Help Viewer for displaying the help book title in the Help Center.	<META NAME="AppleFont" CONTENT="Osaka">
AppleSearchResults-Font	The font used by Help Viewer to display search results from the book.	<META NAME="AppleSearchResultsFont" CONTENT="Osaka">
AppleOrder	The order in which chapters should appear in the table of contents for a chapter-based book.	<META NAME="AppleOrder" CONTENT="20">

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.

Table B-1 Help URLs

URL	Syntax	Action
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.
help: //full/path.html	help: //path/to/page.html	Opens the specified file in Help Viewer.

URL	Syntax	Action
help:goto_helpcenter	help:goto_helpcenter=developer	Loads the Help Center in Help Viewer. There are two possible arguments to this URL: <code>user</code> and <code>developer</code> . These specify the user and developer help centers, respectively. In Mac OS X version 10.2 and later, this URL toggles the position of the Help Center drawer.
help:openbook	help: openbook=help_book_name	Opens the specified help book in Help Viewer.
help:runscript	help:runscript='elp_folder_name/subfolder/scriptname string='optional_string_parameter'	Runs the specified script. The <code>string</code> argument is an optional argument that is passed to the script.

APPENDIX B

Apple Help URLs

URL	Syntax	Action
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.

APPENDIX B

Apple Help URLs

Apple Help Segments

Table C-1 lists the commands defined by Apple Help for dividing HTML help pages into multiple segments. 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 Pages” (page 39) for more information on segment abstracts.
AppleKeywords	Keywords for the given segment. See “Setting Keywords” (page 37) for more information on keywords.
AppleSegEnd	The end of a segment.

Document Revision History

This table describes the changes to *Providing User Assistance With Apple Help*.

Date	Notes
2007-10-31	Moved to Legacy Documents area of ADC Reference Library.
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.

REVISION HISTORY

Document Revision History