



INSIDE MACINTOSH

Providing User Assistance With Apple Help

For MacOS 8.6



October 19, 1999
Technical Publications
© 1999 Apple Computer, Inc.

 Apple Computer, Inc.
© 1999 Apple Computer, 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 Computer, Inc., except to make a backup copy of any documentation provided on CD-ROM.

The Apple logo is a trademark of Apple Computer, 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 book. Apple retains all intellectual property rights associated with the technology described in this book. This book is intended to assist application developers to develop applications only for Apple-labeled or Apple-licensed computers.

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

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

Apple, Apple Guide, Apple Help, the Apple logo, Mac OS, Macintosh, and Sherlock are trademarks of Apple Computer, Inc., registered in the United States and other countries.

Adobe, Acrobat, and PostScript are trademarks of Adobe Systems Incorporated or its subsidiaries and may be registered in certain jurisdictions.

Helvetica and Palatino are registered trademarks of Heidelberger Druckmaschinen AG, available from Linotype Library GmbH.

ITC Zapf Dingbats is a registered trademark of International Typeface Corporation.

Simultaneously published in the United States and Canada.

Even though Apple has reviewed this manual, APPLE MAKES NO WARRANTY OR REPRESENTATION, EITHER EXPRESS OR IMPLIED, WITH RESPECT TO THIS MANUAL, ITS QUALITY, ACCURACY, MERCHANTABILITY, OR FITNESS FOR A PARTICULAR PURPOSE. AS A RESULT, THIS MANUAL IS SOLD “AS IS,” AND YOU, THE PURCHASER, 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 MANUAL, 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.

Table of Contents

Introduction	5
Using the Help Viewer	5
Benefits of Adopting Apple Help	5
Using Guide Files With Apple Help	6
Using HTML Content in Apple Help	7
Creating HTML Content	7
Indexing HTML Help Content	7
Segmenting Help Content	8
Controlling How Content is Indexed	8
Adding Descriptions	9
Adding Search Keywords	9
Using the Apple Help Indexing Tool	10
Preparing to Index Your Help Content	10
Indexing Non-English Help	10
Creating an Index	11
Indexing Tips	11
Creating a Title Page	12
Specifying a Title	12
Specifying a Title Font	13
Specifying an Icon	13
Specifying a URL	14
Specifying a Search Results Font	15
Creating Chapters	15
Specifying a Chapter Title	17
Specifying Chapter Order	17
Specifying a URL	18
Specifying a Target Frame	18
Creating Items in the Help Menu	19
Using Guide Maker to Create Guide Files	20
Importing Guide Script Source Files	21
Saving a Guide Maker File	21
Compiling a Guide	21
Troubleshooting Guide Files	22
Calling AppleScripts from HTML	22
Integrating Apple Help With Other Content	23
Using QuickTime Movies With Apple Help	23

Using Modified Apple Guide Functions	24
Using The Apple Help Console	24
Sending Messages to the Help Console	25
Glossary	26
Colophon	26

Providing User Assistance With Apple Help

Introduction

Apple Help is an update and enhancement to previous versions of the Mac OS help system. It supports all features of Apple Guide. Apple Help is available on Mac OS 8.6 and later. This document describes Apple Help for Mac OS 8.6.

This document describes the process of creating new content and adapting existing Apple Guide content for use with Apple Help. The material in this document supplements the content of *Apple Guide Complete*, which is available at <http://developer.apple.com/techpubs/mac/AppleGuide/AppleGuide-2.html>.

You should read this document if you are interested in using Apple Help to provide HTML-based help content in your Mac OS program.

Using the Help Viewer

The Help Viewer is a lightweight browser that displays standard HTML 3.2 content. The Help Viewer, which is launched automatically when required, is optimized to take advantage of AppleScript and the Sherlock search engine through the use of special HTML tags. Apple Help can also launch Apple Guide sequences from HTML links. You can take advantage of these special features in Apple Help by authoring your help content in HTML, as described in [Using HTML Content in Apple Help](#).

Benefits of Adopting Apple Help

- Apple Help uses HTML files, so you can take advantage of a wide range of HTML authoring tools to produce help content. Using HTML allows you to develop, test, and localize help content more quickly.

- Apple Help can call AppleScripts from within HTML content. This gives you a way to specify and control a wide range of actions, including launching and interacting with other applications or system components. [Calling AppleScripts from HTML](#) describes this process in more detail.
- Apple Help can launch applications from within HTML content. For an example how to specify an application to launch, see [Integrating Apple Help With Other Content](#).
- Apple Help can launch QuickTime movies if QuickTime software is present on the user's system. For an example how to specify a movie to launch, see [Using QuickTime Movies With Apple Help](#).
- Apple Help takes advantage of the Sherlock search engine to help users find specific help content. You can enhance and control the search process by using special tags in your help content, as described in [Indexing HTML Help Content](#).

Note: This document refers to a complete set of help content as a **book** and a section in the book as a **chapter**.

- An Apple Help book can be dynamic, growing and shrinking as the user installs and removes help content. [Creating Chapters](#) describes this process in more detail.
- Apple Help lets you provide user access to your help book in two ways:
 1. Adding an item to the Help menu, as described in [Creating Items in the Help Menu](#).
 2. Providing “?” buttons in your application, as described in [Using Modified Apple Guide Functions](#).

Using Guide Files With Apple Help

If you have existing Apple Guide sequences that you prefer not to convert into HTML, Apple Help displays Apple Guide sequences without modification. Sequences designed to make extensive use of coachmarks to lead users through a step-by-step process may be better left in their original format.

Using HTML Content in Apple Help

Providing HTML-based content that works with Apple Help is a four-step process.

1. Creating and organizing the content, as described in [Creating HTML Content](#).
2. Creating a title page for the book, which ensures that the book is displayed in the Help Center. For a step-by-step discussion, see [Creating a Title Page](#).
3. Creating a Help menu item, as described in [Creating Items in the Help Menu](#).
4. Registering your help book with Apple Guide, which involves creating resources in your guide file. For details on this process, see [Integrating Apple Help With Other Content](#).

Creating HTML Content

Whether you are creating new content or adapting existing HTML files, you can save time and present a consistent user experience by basing your content on the templates supplied in the Apple Help SDK and following the guidelines presented in the *Apple Help Style Guide*. You can obtain the Apple Help SDK from <http://developer.apple.com/sdk>.

Note: The content, structure, folder names, and filenames used in Apple-provided help books (such as Mac OS Help) is subject to change. You should not create hard-coded links directly to these files and folders from within your help book. The best way to access Apple-provided content is to direct the user to the Help Viewer's Search function.

You can author your content in any application that generates valid HTML 3.2 files and view your content in any HTML 3.2-qualified browser. You should, however, preview your book in the Help Viewer to ensure your HTML content functions as intended.

Note: Apple Help 1.2 supports all HTML 3.2 features except forms, plug-ins, and Java.

Indexing HTML Help Content

Indexing your content (making your help content searchable by Apple Help) requires that you use the Indexing Tool to create an index file, as described in [Using](#)

[the Apple Help Indexing Tool](#). You can obtain the Apple Help Indexing Tool from the Apple Help SDK, which is available at <http://developer.apple.com/sdk>.

There are four HTML tag sets (described next) that you can use to control how your content is indexed by the Indexing Tool.

Segmenting Help Content

You may choose to divide a single file into multiple sections called **segments**. Segmenting allows one HTML file to contain several sections, each of which can be returned as a separate “hit” in a search. You specify the start of a segment by using the `AppleSegStart` comment. You must give the segment a title, which will be displayed if the segment is displayed as a search hit. The following example shows how to specify a segment titled “Locking a Disk”:

```
<!-- AppleSegStart = "Locking A Disk" -->
<A NAME="Locking A Disk"></A>
<!-- AppleSegDescription = "This section describes how to lock and unlock a floppy disk." -->
<<Content of Segment>>
<!-- AppleSegEnd -->
```

For each `AppleSegStart` comment there must be a matching `AppleSegEnd` comment.

If you want to specify an anchor tag, such as `<A NAME>`, for a particular segment, you must place the anchor tag between the `AppleSegStart` and `AppleSegEnd` comments, as shown in the example above. This ensures that users are taken to the correct segment when it is displayed in search results.

Controlling How Content is Indexed

Using the optional `ROBOTS` meta tag in your file's HTML header allows you to control how your content is indexed. You can specify one of four different values: `INDEX`, `NOINDEX`, `SEGMENTS`, and `KEYWORDS`. The following example shows how to specify a value of `NOINDEX`.

```
<META NAME = "ROBOTS" CONTENT = "NOINDEX">
```

Specifying a value of `INDEX` tells the Apple Help Indexing Tool to index the entire file (including all its segments and keywords) as a possible search result. Specifying a value of `NOINDEX` tells the Indexing Tool not to index the file. You may find it useful

to avoid indexing template or table-of-contents files. Specifying a value of `SEGMENTS` causes only segments to be indexed as possible search results. Specifying a value of `KEYWORDS` tells the Indexing Tool to index only those terms specified with the `KEYWORDS` meta tag (described in [Adding Search Keywords](#)), leaving the rest of the content unindexed.

If you do not use the `ROBOTS` meta tag, the Apple Help Indexing Tool will index the entire file as a possible search result.

Adding Descriptions

Using the optional description tags gives you two ways to add brief descriptive abstracts that are displayed in Apple Help search results. If you want to provide a description for an entire file, use the `DESCRIPTION` meta tag, as in the following example:

```
<META NAME = "DESCRIPTION" CONTENT = "This section describes how to use the
desktop Trash Can.">
```

If you want to describe a particular segment of a file, use the `AppleSegDescription` comment, as in the following example:

```
<!-- AppleSegDescription = "This section describes how to use the desktop
Trash Can." -->
```

Make sure you place the `AppleSegDescription` comment between the appropriate `AppleSegStart` and `AppleSegEnd` comments.

Adding Search Keywords

When you use the Indexing Tool on your help book, all content is automatically indexed (unless you specify otherwise by using the `ROBOTS` meta tag), but you may also find it useful to provide additional search synonyms called **keywords**. Keywords are a good way to help users obtain relevant search results when they don't know the exact name or spelling of the concept they are interested in. For example, a user who wants to learn how to use the Trash Can might search on terms like "delete" or "erase." If those words do not appear in the help content that describes using the Trash Can, you can specify those terms as keywords to ensure that the user is directed to the right place.

There are two ways to specify keywords. If you want to provide keywords that apply to an entire file, use the `KEYWORDS` meta tag, as in the following example:

```
<META NAME = "KEYWORDS" CONTENT = "delete, erase, throw away, discard">
```

If you want to provide keywords that apply only to a particular file segment, use the `AppleKeywords` comment, as in the following example:

```
<!-- AppleKeywords = "delete, erase, throw away, discard" -->
```

Make sure you place the `AppleKeywords` comment between the appropriate `AppleSegStart` and `AppleSegEnd` comments.

Using the Apple Help Indexing Tool

The Apple Help Viewer uses the Sherlock search engine to provide fast, full-text searching of Apple Help files. You use the Apple Help Index Tool to create a search index.

Note: Each help book must have at least one search index file. If this file is not present, users cannot search your Help book.

Preparing to Index Your Help Content

The Apple Help Indexing Tool creates an index that reflects your help content at the time the index was created. If you change the content after generating an index, the index may still work, but it will not reflect any changes and may have invalid page references if you have renamed or deleted pages since the index was created.

Before you create an index, make sure your help files are in their final folder locations. The index file stores relative path information, so moving content files after indexing invalidates the index file.

Indexing Non-English Help

To create an index for files that contain double-byte or non-English content, open the Apple Help Indexing Tool, select the Preferences menu item, and set the correct

language in the Preferences dialog box. The current version of Apple Help Indexing Tool supports Japanese and Korean language tokenizing.

Note: You must use the Indexing Tool on a Mac OS-J (Japanese) system to index Kanji text. You cannot simply install the Japanese Language Kit, because the Indexing Tool requires the Japanese language tokenizer contained in Mac OS-J 8.5 or later.

Creating an Index

To create an index file, drag-and-drop the folder containing your help content onto the Apple Help Indexing Tool. The Indexing Tool scans the folder and creates an Apple Help index file. The index file is saved inside the folder you dropped on the Tool. The file must remain in the folder for the index to work correctly.

If error or warning messages appear during indexing, you can select the message to display the contents of the file in the Index Tool window.

Indexing Tips

- During a search, each index file is analyzed separately, then the results are merged according to relevance. For best search results, create one index file for your book.
- To omit pages from the index, use the `ROBOTS` meta tag and specify the `NOINDEX` value. Apple recommends omitting title and TOC pages from the index. For a description of the `ROBOTS` tag, see [Controlling How Content is Indexed](#).
- To index portions of a file as if they were separate files, divide the file into segments, as described in [Segmenting Help Content](#).
- To include descriptive text in the search results page, use the `DESCRIPTION` meta tag, which is described in [Adding Descriptions](#).
- To influence search results, use the `KEYWORDS` meta tag. Apple recommends that you add keywords for common misspellings and synonyms, and that you do not repeat words contained in the content of the file. For more information, see [Adding Search Keywords](#).

Creating a Title Page

You should create a **title page** for your help book. The title page specifies how your book is listed in the Help Center. The title page is an HTML file with a creator type of 'hbwr', placed at the root level of your Help book folder. In the header of the title page file, use meta tags to specify the title, icon and URL of your book as it is displayed in the Help Center window.

Specifying a Title

You must specify a title for your book by using the `AppleTitle` meta tag in the HTML header of your title page file. This title is displayed in the Help Center listing. The following example specifies the title "SurfWriter Help."

```
<META NAME = "AppleTitle" CONTENT = "SurfWriter Help">
```

[Figure 1](#) shows an example of the Help Center window as it appears when you specify the title as shown in the example.

Figure 1 Help Center listing



Specifying a Title Font

You can specify a font to display the title of your help book by using the optional `AppleFont` meta tag in the header of your main TOC file. This is useful when your book uses a script system which may not be directly supported by a non-localized version of the Help Viewer. Here is an example:

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

You should not specify an alternate font for your Help Center listing unless your book uses a non-Roman font.

Note: The Help Center will apply the Bold style to any font you specify with the `AppleFont` tag.

Specifying an Icon

You can use the optional `AppleIcon` meta tag in your title page file's HTML header to specify an icon in your book's Help Center listing. Here is an example that

specifies an icon called `swicon.gif` in the SurfWriter Help folder, as shown in [Figure 2](#):

```
<META NAME = "AppleIcon" CONTENT = "SurfWriter%20Help/swicon.gif">
```

Figure 2 Help Center listing with icon



The icon should be 16 x 16 pixels in size with a transparent background. Save the icon as a Graphics Interchange Format (GIF) file. Specify the path to the icon file relative to the Help folder.

If you do not specify an icon, Apple Help provides a default icon.

Specifying a URL

You can specify a Universal Resource Locator (URL) for your help book by using the optional `AppleURL` meta tag in the HTML header of your title page file. Specifying a URL might be useful if you want to direct the user to a help area on a CD or on your website, for example. The URL you specify is displayed by the user's default web browser. This URL may reference a local file or a file on the Internet.

The following example shows how to specify a link to a page called SurfWriterUpdates.html:

```
<META NAME = "AppleURL" CONTENT = "http://support.apple.com/SurfWriterUpdates.html">
```

In most cases, you will not need to specify a URL for your book. If you do not specify the `AppleURL` meta tag, the user is directed to the book's title page (the file in which you placed the `AppleTitle` meta tag).

Note: If you specify a URL to a volume (such as a CD-ROM) that is not mounted when the Help Viewer is launched, the Help Center does not list your book. Therefore, you should always add an item to the Help menu to ensure access to your help book.

Specifying a Search Results Font

You can specify a font to display any search results from your book by using the `AppleSearchResultsFont` meta tag in the header of your title page file. This is useful when your book uses a non-Roman script system which may not be directly supported by a non-localized version of the Help Viewer. Here is an example:

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

You should not specify an alternate font for search results unless your book uses a non-Roman font.

Creating Chapters

Most help books are relatively simple and compact, so you should keep the content files in a single folder. If your help book is large and complex, or if you decide to use dynamic help content (for example, basing your help content on which plug-ins a user has installed), you should subdivide the content into **chapters** by using subfolders.

[Figure 3](#) shows the layout of a sample help folder. In this example, SurfWriter has installed only the basic help book files.

Figure 3 Help folder structure (basic installation)



If the user installs SurfWriter's optional spell-check module, for instance, you should create a subfolder for the spell-check help chapter, as shown in [Figure 4](#).

Figure 4 Help folder structure (additional chapter installed)



If you decide to use chapters, you should create a title page and index for each chapter subfolder.

Specifying a Chapter Title

If you use a chapter-based structure for your help book, you must specify a title for each chapter by using the `AppleTitle` meta tag in the header of each chapter's title page file. This title is displayed on your book's main title page. The following example specifies the title "Chapter 1: Mac OS Introduction."

```
<META NAME = "AppleTitle" CONTENT = "Chapter 1: Mac OS Introduction">
```

Specifying Chapter Order

You can use the optional `AppleOrder` meta tag in the HTML header of a chapter's title page file to specify the order in which chapters are displayed on the book's title page. Here is an example:

```
<META NAME = "AppleOrder" CONTENT = "80">
```

The Help Viewer lists each chapter incrementally by the `AppleOrder` value you specify; chapters with no `AppleOrder` value are listed last. If you decide not to specify chapter ordering, Apple Help will display all chapter titles in alphabetical order. For example, if your book has three chapters titled "About," "Using," and "Reference," the default display ordering is:

1. About
2. Reference
3. Using

If you use `AppleOrder` tags to specify values of 10 for "About," 20 for "Using," and 30 for "Reference," you obtain the following display order:

1. About
2. Using
3. Reference

Note: If you specify the same `AppleOrder` value for more than one chapter, those chapters are sorted alphabetically.

Specifying a URL

You can specify an URL for a title page entry by using the `AppleURL` meta tag in the header of the chapter's title page file. This URL may reference a local file or a file on the Internet. This might be useful if you want to direct the user to a user help area on a CD or on your website, for example. The following example shows how to direct the user to a page called `SurfWriterUpdates.html`:

```
<META NAME = "AppleURL" CONTENT = "http://support.apple.com/  
SurfWriterUpdates.html">
```

If you specify the `AppleURL` meta tag for a URL that is not available, the user is directed to the file in which the meta tag appears.

Specifying a Target Frame

You can specify a **target frame** (an area of the Help Viewer in which your content is displayed) by using the `AppleTarget` tag in the header of the chapter's title page file.

You may want to display your book with a table of contents frame on the left and a content frame on the right, for example. If you do not specify a target frame, the Help Viewer displays content without frames (that is, it uses the entire window). The following example assumes you have defined a frameset for your book with frames called `_left` and `_right`:

```
<META NAME = "AppleTarget" CONTENT = "_right">
```

Note: The Help Viewer cannot display more than two frames.

Creating Items in the Help Menu

Your application should make its help book available from the Help menu whenever the application is the active process. To install a Help menu item, you create a simple Apple Guide file using the Guide Maker tool provided in the Apple Help SDK, which is available at:

```
<http://developer.apple.com/sdk>
```

You can find a detailed tutorial on creating help menu items in the SDK. Here is a brief description of the process:

1. Create a Guide Script source file containing the `AppCreator` command. For more information on Guide Script commands, see *Apple Guide Complete*.
2. Specify a menu item or items in your source file with the `HelpMenu` command. You can specify all your application's Help menu items from within a single guide file.
3. Compile your source file with the Guide Maker tool, as described in [Using Guide Maker to Create Guide Files](#).

After you have compiled the source file, you must install two resources in your compiled guide file:

1. An empty resource of type `'html'` with ID 1000. This resource registers your help book with Apple Guide.
2. A resource of type `'TEXT'` for each Help menu item you want to install. The name of the `'TEXT'` resource must exactly match the corresponding Help menu item you added with the `HelpMenu` command. The `'TEXT'` resource contains a string specifying the path to the appropriate help file.

You can create and install resources with a resource editor such as ResEdit, which is available at:

<http://asu.info.apple.com/swupdates.nsf/artnum/n10964>

Place the compiled guide file in the same folder as your application.

If the user selects one of the Help menu items installed by your program, Apple Guide searches your guide file for the corresponding 'TEXT' resource and uses the URL specified by the string in the 'TEXT' resource. This string can specify a partial path relative to the Help folder or a full path. You might find it useful to specify a full path if you want to direct the user to a help file on a CD, for example. If the specified file is a text file (such as an HTML file), Apple Guide launches the Help Viewer and loads the text file.

For example, SurfWriter installs a Help menu item called "SurfWriter Help" that displays the HTML file `swmain.html`, located in the SurfWriter Help folder. SurfWriter's guide file must contain an empty 'html' resource and a 'TEXT' resource called SurfWriter Help. The 'TEXT' resource contains the string `SurfWriter%20Help/swmain.html`.

If the file specified in the 'TEXT' resource is an AppleScript, Apple Help will launch the script if you provide the `runscript` command. For example, if SurfWriter uses an AppleScript called `LaunchSurfHelp`, located in the SurfWriter Help folder, to provide user help, the 'TEXT' resource contains the string `runscript:SurfWriter%20Help/LaunchSurfHelp`.

If the specified file is an application, Apple Guide launches the application. For example, if SurfWriter uses a help application called `SurfHelpViewer` in the Docs folder on a CD called `SurfSampler`, the 'TEXT' resource contains the string `SurfSampler/Docs/SurfHelpViewer`.

Using Guide Maker to Create Guide Files

In order to create a guide file that works with Apple Help, you must use Guide Maker to compile Guide Script source code. For complete information on the Guide Script command language, see *Apple Guide Complete* or visit the Apple Guide Complete web page at:

<http://developer.apple.com/techpubs/mac/AppleGuide/AppleGuide-2.html>

The basic process of creating guide files with Guide Maker 2.0 is as follows:

1. Import the Guide Script source files, as described in [Importing Guide Script Source Files](#).
2. Save a Guide Maker file, as detailed in [Saving a Guide Maker File](#).
3. Compile the guide with Guide Maker 2.0. This process is described in [Compiling a Guide](#).
4. Troubleshoot the guide, if needed. You can read more about this in [Troubleshooting Guide Files](#).

Importing Guide Script Source Files

To import Guide Script source files, open Guide Maker 2.0, open the File menu and choose Import Source. Select the main Guide Script source file in the dialog box and click Open.

Guide Maker displays a window that shows individual Guide Script commands as they are processed. If syntax errors occur during the file loading process, double-click the error report to see the error listing in a separate window.

Saving a Guide Maker File

After you import the source files, save the source as a Guide Maker 2.0 document by opening the File menu, choosing Save, then selecting a location for the Guide Maker file. Enter a name for your Guide Maker file and click Save.

Note: If your Guide Script source code involves more than one file or calls on resources, AppleScripts, or other external elements, you must save your Guide Maker file in the same folder as the main source code file. This ensures that Guide Maker can find all the files at compile time.

Compiling a Guide

To compile your guide file, open the File menu and choose Compile, then select a location to save the Guide file. Enter a name for the Guide file and click Save. Guide Maker 2.0 displays a window named “Error Messages”. If Guide Maker displays the message `Database successfully created` in the top right side of the window, your guide was successfully compiled.

Troubleshooting Guide Files

If Guide Maker 2.0 displays the message `Database not created: Fatal errors`, serious errors were discovered in the Guide Script syntax while the source document was being compiled. You can double-click on an error line to see the error in a separate window. Searching on unique portions of the Guide Script copied from the window should help you to find the error in the source file.

Note: For certain types of errors, you may need to look above the line that is indicated as an error.

There are some errors that may cause Apple Guide to stop reading the source document before all components are created. You can find these errors by tracking which components failed to be created in the “guideName.gtb” window when you import the source document.

If you cannot find a syntax error, you may find it helpful to attempt to compile the guide with Guide Maker 1.2. Guide Maker 1.2 recognizes and reports some errors that Guide Maker 2.0 does not recognize.

Calling AppleScripts from HTML

You can specify an AppleScript to call from an HTML link. In this example, the link calls the script `OpenURL` (in the `shared` folder in the `Mac OS Help` folder) and passes a URL in the `string` parameter.

```
<A HREF='help:runscript="Mac%20S%20Help/shrd/OpenURL"
string="http://www.apple.com/quicktime"'>
```

Note: The content, structure, folder names, and filenames used in Apple-provided help books (such as `Mac OS Help`) is subject to change. You should not create hard-coded links directly to Apple-provided scripts. Always install the scripts you need within your own help book folder.

Specify the path to the script relative to the Help folder (which is in the System folder). If you specify one of your own AppleScripts, make sure the script is located within your own help book folder.

The optional `string` parameter is a single string that is passed to the script. The script is responsible for parsing the string. Some scripts do not require parameters.

If the script accepts multiple parameters, use commas to separate the parameters within the string.

Note: Make sure the path you specify does not contain any commas. The Help Viewer may interpret commas as parameter separators.

Make sure you enclose the entire command in single quotes and enclose the path and string parameters in double quotes. Note that you must specify only legal URL characters in the path. For example, each space character in the path should be converted to the HTML escape character `<%20>`. In the example, “Mac OS Help” is specified as `Mac%20OS%20Help`.

Integrating Apple Help With Other Content

If you wish to take advantage of the features of Apple Help, you must convert help content to HTML. Note that you do not have to convert all help content for a particular program to HTML. You can use a mix of HTML and guide file content to take advantage of the strengths of each format. The following example shows how your Apple Help HTML file can call a guide file sequence called “Printing Your Document” contained in a guide file called `surfhelp.gtb`:

```
<A HREF='help:openguide="SurfWriterHelp/surfhelp.gtb"
sequencename="Printing Your Document"
frontprocess = "MACS"'>
```

This example switches to the `MACS` process (which is the Finder) and opens the sequence. When the user closes this sequence, the Help Viewer becomes the active process again.

You can specify an application for Apple Help to launch. Specify the path to the movie relative to the Help folder. For example, if SurfWriter installs an application called `SurfHelpPlayer` in the `Surf Help` folder (which is in the Help folder), you would use the following syntax:

```
<A HREF="Surf%20Help/SurfHelpPlayer">
```

Using QuickTime Movies With Apple Help

You can specify a QuickTime movie to play by using the `EMBED` tag. Specify the path to the movie relative to the Help folder. Here is an example in which the Help

Viewer plays a movie called `SurfHelpDemo` in the Surf Help folder (which is in the Help folder):

```
<EMBED SRC="Surf%20Help/SurfHelpDemo">
```

Using Modified Apple Guide Functions

Apple Help modifies the Apple Guide functions `AGOpen` and `AGOpenWithSearch` so that you can specify a path to an HTML file you want the Help Viewer to open. To specify a path, you must do the following:

1. Recompile your application's guide file with Guide Maker 2.0, as described in [Using Guide Maker to Create Guide Files](#). Recompiling adds the new resources that allow the Help Viewer to interact with Apple Guide functions.
2. Add a resource of type 'TEXT' to the compiled guide file. The name of the resource must match the string you pass in the `searchString` parameter of the `AGOpen` or `AGOpenWithSearch` function. The 'TEXT' resource contains a string specifying the path to the file that you want Help Viewer to open. You express the path as a URL, either relative to the Help folder or as a full path.

Using The Apple Help Console

The Apple Help Console is a developer tool included with the Apple Help SDK. The Help Console displays diagnostic information about an Apple Help project, including AppleScript and AppleGuide errors, as well as any messages you send to the Help Console from AppleScripts. This diagnostic information can help you analyze problems with the project and the scripts and guides associated with it. To see diagnostic information for your Apple Help project, launch the Apple Help Console. The Help Console opens a message window and creates a new log file in the Console folder. The log file, which is named with the current date and time, is used to save Help Console messages.

Once you have launched the Apple Help Console, you can open your help project and perform tasks, such as opening pages and executing scripts. The Help Console displays status messages in the active console window.

Note: The active console window has a flashing icon in the bottom left corner of the window.

Sending Messages to the Help Console

You can send messages to the Help Console from AppleScripts. You might do this to mark a location in a script to help you isolate a problem, for example. The following example shows how to send the message `Starting to Execute` to the Help Console from a script:

```
tell application "Apple Help Console"  
    log string "Starting to Execute"  
end tell
```

Glossary

Apple Guide An integrated system for managing the Help menu and displaying sequenced help information on Mac OS computers.

Apple Help An AppleScript-enabled system for displaying HTML-based help content, in addition to Apple Guide sequences.

help book The content of a program's help files.

help chapter A major section of a help book.

hit A single item in a list of search results.

indexing The process of preparing a help book for content searches.

keyword A search synonym or alternate spelling for a frequently used term.

segment One part of an HTML file that has been subdivided for purposes of searching.

Sherlock A Mac OS application designed for sophisticated content searches.

title page An HTML file that contains an AppleTitle meta tag in the HTML header. In most cases, this file is the first one displayed when the user opens your help book.

Colophon

This Apple manual was written, edited, and composed on a desktop publishing system using Apple Macintosh computers and Adobe™ FrameMaker software. HTML processing was done with BBEdit™. Line art was created using Adobe Illustrator and Adobe Photoshop.

Text type is Palatino® and display type is Helvetica®. Bullets are ITC Zapf Dingbats®. Some elements, such as program listings, are set in Adobe Letter Gothic.

WRITER

Otto Schlosser

PRODUCTION EDITOR

Glen Frank

ILLUSTRATOR

Karin Stroud

ACKNOWLEDGMENTS

Tony Francis, Alex Hamdy, Bill Harris, Rick Hoiberg, Riley Howard, Donna Lee, Gordon Meyer, James Miyake, Dan Wolfe

Index

A

adding 9
adding file descriptions 9
adding items to Help menu 15
adding keywords 9
AGOpen function 24
AGOpenWithSearch function 24
AppleFont tag 13
Apple Guide Complete 5, 20
Apple Guide content, integrating with HTML 23
Apple Guide functions, using 24
Apple Guide sequences, displaying 6
Apple Help, benefits of adopting 5
Apple Help, special features 5
Apple Help availability 5
Apple Help Console 24
Apple Help Console, using 24
Apple Help Indexing Tool, using 10
AppleIcon tag 13
AppleKeywords tag 10
AppleOrder tag 17
AppleScript 6, 20
AppleScripts, calling from HTML 22
AppleSearchResultsFont tag 15
AppleSegDescription tag 9
AppleSegEnd tag 8, 9, 10
AppleSegStart 8, 9, 10
AppleTarget tag 18
AppleTitle tag 12, 15, 17
AppleURL tag 14, 18
applications, launching 20

B

benefits of adopting Apple Help 5

C

calling a guide file sequence 23
calling AppleScripts from HTML 22
chapter order, specifying 17
chapters 15
chapters, creating 15
chapter titles, specifying 17
coachmarks 6
compiling a guide 21
content, controlling how indexed 8
controlling how content is indexed 8
creating an index 11
creating a title page 12
creating chapters 15
creating HTML content 7
creating items in Help Menu 19

D

default icon 14
default web browser 14
DESCRIPTION tag 9, 11
displaying Apple Guide sequences 6
dynamic help content 6

E

EMBED tag 23
error messages during indexing 11

F

file descriptions, adding 9

font, specifying search results 15
 fonts, non-Roman 13
 fonts, specifying title 13

G

Graphics Interchange Format (GIF) 14
 guide, compiling a 21
 guide files, recompiling 24
 guide files, troubleshooting 22
 guide files, using with Apple Help 6
 guide file sequences, calling 23
 Guide Maker, using 20
 Guide Maker 1.2 22
 Guide Maker files, saving 21
 Guide Maker tool 19, 21
 Guide Script 22
 Guide Script source files 19
 Guide Script source files, importing 21

H

Help Center 12
 Help Console, sending messages to 25
 help content, dynamic 6
 help content, non-English 10
 help content, preparing to index 10
 help content, segmenting 8
 help folder, sample 15
 Help Menu, adding items 15
 Help Menu, creating items in 19
 Help Viewer 5, 20
 Help Viewer, using 5
 HTML 3.2 5
 HTML 3.2 feature support 7
 HTML content, creating 7
 HTML content, integrating with Apple Guide 23
 HTML content, providing 7
 HTML help content, indexing 7
 'html' resource 19

I

icons, specifying 13
 icon size 14
 importing Guide Script source files 21
 indexes, creating 11
 indexing, error messages 11
 indexing HTML help content 7
 indexing non-English help content 10
 indexing tips 11
 integrating Apple Guide and HTML content 23

J

Japanese Language Kit 11

K

Kanji text 11
 keywords 9
 KEYWORDS tag 9, 10, 11

L

launching applications 20

M

Mac OS-J 11
 'MACS' process 23

N

non-Roman fonts 13

P

preparing to index help content 10
 providing HTML content 7
 providing user access 6

Q

QuickTime 23

R

recompiling guide files 24
 ResEdit 20
 ROBOTS tag 8, 9

S

sample help folder 15
 saving Guide Maker files 21
 segmenting help content 8
 sending messages to Help Console 25
 Sherlock 5, 10
 source files, Guide Script 19
 special features in Apple Help 5
 specifying a target frame 18
 specifying a title 12
 specifying a title font 13
 specifying a URL 14, 18
 specifying chapter order 17
 specifying chapter titles 17
 specifying icons 13
 specifying search results font 15

T

target frame, specifying 18
 'TEXT' resource 19, 24

title pages, creating 12
 titles, specifying 12
 Trash Can 9
 troubleshooting guide files 22

U

URL, specifying a 14, 18
 user access, providing 6
 using Apple Guide functions 24
 using Apple Help Console 24
 using guide files with Apple Help 6
 using Guide Maker 20
 using Help Viewer 5
 using the Apple Help Indexing Tool 10

I N D E X