Creating Topics

Chapter 2 showed you how to create a simple Help file with just two topics. This chapter goes into more depth about how to create and edit Help topic files. It also describes how to insert Help-specific codes that communicate information to the Help compiler about the particular structure and organization of the Help file. For example, topic codes are used to identify topics, to create keyword indexes and

What Are Topics?

browse orders, and to run Help macros.

The *topic* is the basic unit of information in a Help file; all information is displayed in topics. A topic can include text, graphics, hypertext links, and custom controls. Because their use varies, topics can range in size from short examples containing a single picture or word to lengthy explanations involving several graphics and hundreds of words. Figure 6.1 shows a topic from the sample Help file.

Graphic

Topics are stored in *topic files*, which contain one or more topics. How you organize the topics within topic files depends on your own work habits and the type of information contained in the topics. At the extreme, you can place all the topics in one large topic file, or you can create a separate topic file for each topic in the Help project. For most projects, it is probably best to avoid either extreme and organize the topics so that they reflect the overall organization of your Help file. For example, if you create a simple Help system with three kinds of information, you might want to create three topic files, one for each information category.

Just remember that the way you organize topics within the topic file does not affect how a user views the topics. The organization that the user sees when using the Help file is something you build into the Help file using authoring features. For example, you can use any of the following Help features to organize the presentation of topics in your Help file.

Help feature	Description
Hypertext links	Let you connect any two topics together so that users can "jump" from topic to topic following a particular line of interest. This is the primary method you use to organize the Help topics in a Help file.
Browse sequences	Let you organize a number of topics into a linear sequence that users can "browse" the same way they flip through pages in a printed book.
Keyword search	Lets you assign keywords to topics so that users can search for a particular item and access information directly without following a hierarchical path.

Getting Started

Before you start creating topics for the Help file, you need to make at least two preliminary decisions. You must:

- Choose a text editor that you will use for writing and saving the topic information.
- Make a project directory on the hard disk drive where you can store the topic files that the compiler will use to build the Help file.

Choosing a Text Editor

The first thing you have to decide before you can create any Help topics is which word processor or text editor you are going to use. The primary requirement is that it must support *rich-text format* (RTF). RTF is the standard Word document exchange format that Microsoft Word applications use to convert formatted documents into a set of instructions that other applications can read and interpret.

Most commonly, RTF is used to transfer Word documents between Macintosh and PC platforms or across communications networks. RTF file format consists of text intermixed with RTF statements that control document, paragraph, and

character formatting. For a full description of the RTF file format accepted by th Help computer, see Appendix B, "Help RTF Statements."

However, because the Windows Help compiler only supports topic files that have been saved in RTF, choosing the right word processor is critical to your online Help project. Although you can use any word processor or text editor that has an option to save files as RTF or that has some other means for converting its native format to RTF, the files it produces may not be fully compatible with the Help compiler. That is because some word processors and conversion utilities produce RTF that can be interpreted correctly by other word processors but cannot be interpreted correctly by the Help compiler because they use nonstandard techniques. For that reason, we strongly recommend that you use Word for Windows version 1.1 or 2.0.

Note

If you use Microsoft Word for the Macintosh to create topic files, we recommend using version 4.0b or later.

Making a Help Directory

After you choose an RTF editor, you should make a project directory and decide how to organize the directory structure that will hold all the project files. How you organize your specific Help project depends on the type of Help file you are creating. If the Help project is small or uncomplicated, you may want to place all the project files in one directory. If the Help project is large or complex, you may want to make a directory tree that reflects the organization of the Help project. For example, one method is to make a directory for each subcategory within the Help file and then create an \ART and \RTF subdirectory for the pictures and text used in each subcategory. Figure 6.2 shows an example of this kind of directory structure.

Graphic

Creating New Topics

The first step in creating a Help file is to prepare the text. There are several ways to create text for the new topics. You can:

◻ Type the topic text in Word for Windows (or other compatible editor).

While you are creating new material, you may want to keep the file in normal .DOC format until you are ready to compile the topics. Working in Word's normal format is easier and saves time. When you are ready to build the Help file, you can save the topic text as RTF.

◻ Import one or more topics from an existing RTF topic file into a new topic file.

Word for Windows includes an RTF filter that can import or export RTF text. If you have files that are already in RTF or Word format, you can import them directly into Word and save them as new RTF topic files. To import just a portion of an RTF file, you can open the RTF file and copy the text you want to the Windows Clipboard and then paste it into the new topic file.

◻ Import text from any electronic document by converting its file format to a format readable by Word for Windows.

Word for Windows can import many types of word-processing files. If you have files that are not in RTF or Word format, you can import them into Word and save them as RTF files. Also, if your information has a consistent format, you can write automated conversion programs or Word macros to add the necessary RTF statements to the text document. Because the compiler expects valid RTF, if you import or generate invalid RTF, the topic files might fail to produce the results you expect.

◻ Use an optical character recognition (OCR) scanner to convert printed text to electronic form and import the resulting file into Word for Windows (or other compatible word processor).

Because printed manuals often serve as good source material for online documentation, you can scan the material if it does not exist in electronic form or if it cannot be converted to a compatible format. When saving the scanned file, ASCII (text-only format) is sufficient to

import the file into Word for Windows.

Dividing the Text into Topics

Whether you type the text from scratch or work with existing text documents, you must divide the text into discrete chunks of information, or topics. This is perhaps the most difficult aspect to writing online Help: deciding how to divide the body of information into separate topics. To do that you must decide how large to make the topics and where to make the breaks in the content. A general guideline is to keep topics short and present information in small, relatively equal-sized chunks that users can scan easily. However, longer, scrolling topics may be appropriate if the target audience is more expert or if the content suffers from fragmentation when divided into too many chunks.

Exactly how you divide the information will vary depending on the Help project and the specific content. In all cases, you must balance conflicting needs. Small topics provide more flexible ways to organize the information; longer topics simplify the design and help users understand relationships among ideas. In general, try to present one concept or idea per topic and limit each topic to roughly one screen of information. Using this method, most topics should be from one to seven paragraphs.

Identifying Topics

To identify a unit of text as a topic, you insert a hard page break in the topic file. The Help compiler treats all the information between page breaks as one topic.

To identify a new topic

n   Insert a hard page break (CTRL+ENTER in Word for Windows) at the end of the text block that you want to become a topic.

The "page" defines the topic boundaries.

Figure 6.3 shows two sample topics as they appear in Word for Windows: "Topic A" and "Topic B."

Graphic

In addition to the page break, each topic also includes a context string identifier,

and most topics have a title.

Topic footnote	Description
Context string	Unique identifier for the topic that the compiler uses as a reference. The context string is used to identify individual topics and to create links between topics in the Help file. The context string is required.
Title	Descriptive name for the topic. The topic title provides an identifying link between the name displayed in Help dialog boxes, such as Search and History, and the actual topic heading displayed in the Help window.

For complete information about these and other topic footnotes, see the

Editing Topics

"Understanding Topic Footnotes" section and following sections in this chapter.

While creating and editing the topic text, you work in Word for Windows so you can use Word's powerful editing features to perform the following tasks:

- Edit text, perform search and replace, or run the spell checker.
- Apply character, paragraph, and table formatting.
- Add Help features like hot spots and pictures.

Note

Some Word formatting attributes are ignored by the Help compiler. For a complete discussion of the formatting attributes supported by Help, see Chapter 7, "Formatting Topics." The list of unsupported features is small, so you have considerable freedom in how you format the topic text.

After creating and editing the topics, you save the file as RTF and exit Word or, if you want to leave Word running, just close the open document.

Understanding Topic Footnotes

Windows Help redefines a number of Word for Windows features so that they have a different meaning in Help topic files than they do in a normal word-processing document. For example, Windows Help uses the footnote feature in Word for Windows to pass important topic information to the Help compiler. Topic footnotes have very specific meanings in Help, as described in the following table.

Footnote	Name	Purpose
Asterisk (*)	Build tag	Defines a tag that specifies topics the compiler conditionally builds into the system. Build tags must be the first footnote in a topic when they are used.
At sign (@)	Comment	Includes an author-defined comment about the Help topic.
Dollar sign ($)	Title	Defines the title of a topic.
Exclamation Point (!)	Macro	Defines a macro that Help executes when the user enters the topic.

Letter K	Keyword	Defines a keyword the user uses to search for a topic.
Other letters	Multikey keyword	Defines an alternate keyword table that the user uses to search for a topic.
Plus sign (+)	Browse sequence	Defines a sequence that determines the order in which the user can browse through topics.
Pound sign (#)	Context string	Defines a context string that uniquely identifies a topic. Because hypertext relies on links provided by context strings, topics without context strings can only be accessed using keywords or browse sequences.

Note

If you are using build tags, footnote them at the very beginning of the topic. Place other footnotes in any order you want. For information about using each of these specific footnotes, see the following sections in this chapter.

Assigning Context Strings

After creating individual topics by inserting page breaks, you then identify each topic by giving it a unique context string. *Context strings* identify each topic in the Help file. Each context string must be unique–it can be assigned to only one topic within the Help project.

Windows Help uses the context string to find a specific topic when users choose a hot spot that references that topic. Although not technically required, it is a good idea to assign a unique context-string identifier to all topics within the Help
system. Topics without context strings cannot be accessed by hot spots (jumps, pop-ups, or macros), although they can be accessed through browse sequences or keyword searches. If writers include topics without context strings, it is their responsibility to ensure that users can access that information.

See the "Assigning Keywords" and "Specifying Browse Sequences" sections in this chapter for more information. Also, see Chapter 8, "Creating Links and Hot Spots," for information about using context strings in hot spots.

Inserting Context-String Footnotes as Topic Identifiers

To enter a context string in the topic, you use a pound sign (#) footnote. A context-string footnote must be the first footnote in the topic unless you use build tags to specify individual topics for the build. If you use build tags, the build-tag footnote must precede the context string. (For more information about build tags, see "Assigning Build Tags," later in this chapter.)

To insert a context string

1. Position the insertion point at the beginning (extreme left) of the first line in the topic.
2. From the Insert menu, choose Footnote.
The Footnote dialog box appears.
3. Type the pound sign as the custom footnote mark, and then choose OK.

A superscript pound sign ($^{\#}$) appears in the text window, and the insertion point moves to the footnote window.
4. Type the context string to the right of the pound sign in the footnote window.
Use only a single space between the pound sign and the text string. (The context string itself shouldn't contain any spaces.) For example, you might type the following context string to identify the "Saving Files" topic in Cardfile Help:

   $^{\#}$ savefile_card

Figure 6.4 shows this footnote window.

Graphic

After inserting a footnote, you can close the footnote window or leave it open. If you are going to code additional footnotes, you may want to leave it open. In Word for Windows version 1.1, you close the footnote window by double-clicking the split bar (the black rectangle between the text window and the footnote window's vertical scroll bars). In Word for Windows version 2.0, you can just click the Close button.

Inserting Spot References

In addition to the context string that identifies the topic, you can also include additional context strings to identify particular locations within the topic. These secondary context strings are called *spot references* because they reference specific locations, or "spots," within a topic.

When executing a jump, Windows Help uses the location of the context string as a reference point. Windows Help displays topics with only one context string identifier from the top down, starting with the first line in the topic. If you want users to be able to jump to a specific location within the topic, you can insert a context string at that location. When executing a jump to the spot reference, Windows Help displays the topic as if that location were the "top" of the topic. For example, to create a jump to the middle of a topic, you insert a context string at that location.

Note

You cannot insert a title footnote in the middle of a topic, so any spot references that you define cannot have a "subtitle."

To insert a spot reference context string

1. Position the insertion point at the location where you want to insert a spot reference.
A context string inserted at the beginning of the topic is used to identify the topic. A context string inserted anywhere else in the topic is treated as a spot reference.
2. From the Insert menu, choose Footnote.

The Footnote dialog box appears.

3. Type the pound sign as the custom footnote mark, and then choose OK.

A superscript pound sign ($^{\#}$) appears in the text window, and the insertion point moves to the footnote window.

4. Type the context string to the right of the pound sign in the footnote window.

Use only a single space between the pound sign and the text string. (The context string itself shouldn't contain any spaces.) For example, you might type the following context string to identify the "Save As Option"topic within the "Saving Files" topic of Cardfile Help:

$^{\#}$ saveas_card

Figure 6.5 shows this footnote window.

Graphic

Context String Guidelines

When creating context strings, follow these guidelines:

- Every topic should have at least one context string, and that string must be unique within the Help file.

- Context strings can contain the alphabetic characters A—Z (uppercase or lowercase), the numeric characters 0—9, and the period (.) or the underscore (_) character. Even though Windows Help doesn't distinguish between uppercase and lowercase letters, you might want to mix the case of characters to make the string more readable, as in this example:

$^{\#}$ Saving_RTF_Files

- Context strings can have as many as 255 characters. However, shorter strings are easier to remember and easier to type when referencing them in jump, pop-up, and macro hot spots.

Inserting Topic Titles

Enter a *topic title* after the context string. Topic titles help users identify individual topics. For example, Windows Help displays the topic title you enter in the Bookmark dialog box, in the History window, and in the Go To list of the Search dialog box. You use a dollar sign ($) footnote to identify the topic title.

To insert a topic title

1. Position the insertion point at the beginning of the topic (to the right of the pound-sign footnote).
2. From the Insert menu, choose Footnote.
The Footnote dialog box appears.
3. Type the dollar sign as the custom footnote mark, and then choose OK.

A superscript dollar sign ($^{\$}$) appears in the text window, and the insertion point moves to the footnote window.
4. Type the topic title to the right of the dollar sign in the footnote window.
Use only a single space between the dollar sign and the first word of the title. For example, you might type the following title for the "Saving Files" topic in Cardfile Help:

$^{\$}$Saving Files

Figure 4.6 shows this footnote window.

Graphic

Note that you *can* include spaces in topic titles.

Topic Title Guidelines

When creating topic titles, follow these guidelines:

- You can define only one title footnote per topic, and it must be at the beginning of the topic. You cannot place title footnotes at spot references.

- Since you only need topic titles to identify topics that users can jump to or that can be displayed in Help dialog boxes, you might consider

not assigning topic titles to topics that are displayed only in pop-up windows.

- Enter the title string exactly as it appears in the title heading of the topic text because it is the string Windows Help displays in dialog boxes and windows for the user. In the previous example, the topic title "Saving Files" exactly matches the bold heading "Saving Files" in the topic text.

- Titles can have as many as 127 standard characters, including accented characters. Any embedded spaces in the title count as characters. Windows Help truncates titles longer than 127 characters.

- Windows Help follows these rules when sorting titles alphabetically for the Search facility:

  - Short titles come before long titles.

  - Punctuation characters come before numeric characters.

Assigning Keywords

  - Numeric characters come before alphabetic characters.

In printed books, a good index provides an outline of the book's contents and helps readers quickly find specific information. Indexes also associate related information in a book; when several pages are listed under a common index entry, readers know that the information on those pages is related in some way.

In Windows Help, the Search feature provides a type of "electronic index," similar to a book index, in which users look up words in the index to find specific information quickly. By typing a keyword or choosing one from the keyword list in the Search dialog box, users can see all the topics associated with a keyword and go directly to one of the associated topics. Because keyword search is a fast way for users to find information, you should consider assigning keywords to most topics in your Help file. Likewise, if you do not want users to access certain topics through the Search facility, you should not assign any keywords to those topics.

Help makes it easy to add keywords to topics. However, much of the work in

creating a good keyword index involves analysis and planning. When you create an index for a book, you need to plan the index entries carefully, eliminate duplicate or ambiguous entries, and apply the entries consistently to all pages of the book. The same issues apply to Help keyword indexes.

How Keywords Work in Windows Help

Keyword indexes are similar to book indexes–the keywords are analogous to first-level entries, and the topics containing the keywords are analogous to second-level index entries. A keyword index consists of a list of index entries, or *keywords*. In a book index, each entry includes a page reference, which tells the reader where to find the indexed information. In a keyword index, each keyword includes a list of topics (the electronic equivalent of "page references"), which tells users where to find the information associated with that particular keyword. Just as you can have more than one reference following a single index entry in a book, you can assign the same keyword to more than one topic in the Help file.

When a user chooses the Search button, a dialog box appears that lists all the keywords associated with each topic in the upper half of the dialog box. The user types a word or selects one from the list and then chooses the Show Topics button. Help lists all the topics containing the selected keyword in the lower half of the dialog box. (Remember, topics are identified by the $ footnote.) Figure 6.7 shows how the sample application has the keyword "open" in two topics: "PLACEHOLDER" and "PLACEHOLDER."

Graphic

The user can double-click a topic in the list box to display that topic in the main Help window.

Defining Keywords for a Topic

The keywords that appear in the Search dialog box come from the keywords you add to individual topics in your Help file. A topic can have many keywords, and the keywords can be from a single index or from multiple indexes (see "Creating Multiple Keyword Indexes," later in this section).

When you create a new topic, you can define the keywords you want to associate with that topic. The topic must have at least one keyword to be included in the Search dialog box (if you don't want an index entry for the topic, don't define any keywords). You use an uppercase K footnote to create keyword entries for a topic.

To assign a keyword to a topic

1. Position the cursor at the beginning of the topic text.

2. From the Insert menu, choose Footnote.

3. Type an uppercase K as the custom footnote mark, and then choose OK.

A superscript K (^{K}) appears next to the heading, and the insertion point moves to the footnote window.

4. Type the topic keyword(s) to the right of the K in the footnote window.

Use only a single space between the K and the first keyword. Separate multiple keywords with a semicolon (;). For example, you might type the following keywords for the "Cardfile" topic:

^{K} Cardfile keyword go here PLACEHOLDER

Figure 6.8 shows this footnote window.

Graphic

Placing Keywords in the Topic

When the user goes to a topic by choosing a keyword from the Search dialog box, Help displays the selected topic in the main window, starting from the beginning of the topic. If the information related to the keyword is located in the middle or toward the end of the topic, the user may not be able to see the relevant information without scrolling the topic. This result may not be what you want.

If you want users to be able to go directly to relevant information within a topic (and see it without scrolling), you can place additional keywords with the information you want users to find. Keywords placed within a topic function as *spot references* (similar to context-string spot references) because they index specific locations, or "spots," within the topic. To access the spot-referenced material, users choose the keyword from the Search dialog box.

In the Search dialog box, all keywords appear the same. The user cannot tell the difference between keywords placed at the beginning of the topic and those placed elsewhere in the topic. However, when the user goes to the topic, Windows Help uses the location of the keyword footnote as a reference point. If the keyword footnote is located in the middle of the topic, Help displays the topic as if the middle location were the "top" of the topic.

Because you cannot insert a title footnote in the middle of a topic, any keywords that you place in the middle of the topic use the main topic title in the Search dialog box.

To define a keyword in the middle of the topic

1. Place the insertion point where you want to define the keyword.
A keyword inserted anywhere except the beginning of the topic is treated as a spot reference.
2. From the Insert menu, choose Footnote.
The Footnote dialog box appears.
3. Type an uppercase K as the custom footnote mark, and then choose OK.

A superscript K (^{K}) appears in the text window, and the insertion point moves to the footnote window.
4. Type the keyword(s) to the right of the K in the footnote window.
Use only a single space between the K and the first keyword. Separate multiple keywords with a semicolon (;). For example, you might type the following keywords to index a procedure within the "Cardfile" topic:

^{K} switching to List view

Figure 6.9 shows this footnote window.

Graphic

Creating Multiple Keyword Indexes

Just as some books include multiple indexes, Help supports multiple keyword indexes. The additional keyword indexes enable an application to look up information that is defined in alternate, custom keyword indexes. For example, you can create a general index for all subjects in the Help file and create specialized indexes for more narrowly defined areas. If your application includes embedded functions, you can use a second keyword index to enable users to get context-sensitive Help on the function calls included in the parent application. Or, if your application competes in the marketplace with another popular program, you can use an additional keyword index for the competitor's product. Users

familiar with keywords from the competitor's application can find matching keywords in your application's Help file.

Creating additional keyword indexes is a two-part process. First, the **MULTIKEY** option must be placed in the [OPTIONS] section of the Help project file, as in this example:

[OPTIONS]
MULTIKEY=L

Note

Be sure to limit your multikey index footnotes to one case, usually uppercase. In this example, topics with the footnote L would have their keywords incorporated into the additional keyword index, whereas those assigned the lowercase letter l would not. We recommend using uppercase letters over lowercase letters. (See the **MULTIKEY** option in Chapter 16, "The Help Project File.")

Second, the topics to be associated with the additional keyword index must be written and assigned footnotes using the letter specified in the **MULTIKEY** option. Multikey footnotes can be placed either at the beginning of the topic or elsewhere if you want to create multikey spot references.

To assign a multikey keyword to a topic

1. Place the insertion point at the beginning of the topic text or wherever you want to define the keyword.
A keyword inserted anywhere except the beginning of the topic is treated as a spot reference.
2. From the Insert menu, choose Footnote.
3. Type the uppercase letter used in the **MULTIKEY** option as the custom footnote mark, and then choose OK.
A superscript letter appears in the topic, and the insertion point moves to the footnote window.
4. Type the keyword(s) to the right of the letter in the footnote window. Use only a single space between the footnote character and the first keyword. Separate multiple keywords with a semicolon (;). Figure 6.10 shows a sample multikey footnote entry.

Graphic

When assigning multikey keywords to topics, be sure to associate only one topic with a multikey keyword. Help does not display the standard Search dialog box for a multikey keyword search. Instead, it displays the first topic found with the specified keyword. If you want the topics in your additional keyword index to appear in the Search dialog box, you must specify a K footnote for the topics.

If Help cannot find the multikey keyword in the alternate keyword index, Help displays a default topic that contains the keyword "Default Topic," or it displays the "Help topic not found" error message if it cannot find the default topic.

An application can display a Help topic that contains a multikey footnote. For information on the parameters passed by the application to Help, see the "Searching for Help with Keywords" section in Chapter 19, "The WinHelp API."

Keyword Guidelines

When adding keywords, follow these guidelines:

n A topic should have a title footnote assigned before you assign it keywords.

n Index keywords can include any ANSI character, including punctuation (except semicolons) and accented characters. You can use spaces between words to create a keyword phrase. For example, you could make "PLACEHOLDER" a single index keyword as follows:

K keyword phrase; keyword; keyword

n Help ignores any formatting assigned to keywords (such as bold or underline).

n The keyword search handles uppercase and lowercase letters the same–for example, "PLACEHOLDER" is the same as "placeholder."

n A keyword footnote can have as many as 1023 characters. If your keyword list exceeds that limit, you can use additional K footnotes within the same topic. For example, you might have three keyword footnotes in a topic:

K keyword phrase; keyword;keyword

Figure 6.11 shows this footnote window.

Specifying Topic-Entry Macros

Graphic

To customize the way Windows Help works with your Help file, you may want to execute a macro when the user enters a topic. Windows Help includes a variety of macros, any of which can be included as a footnote macro. In general, Help macros can be used to:

▫ Modify Help menus and menu functionality.

▫ Modify Help buttons and button functionality.

▫ Modify Help window behavior.

▫ Create hypertext links.

▫ Create and manipulate text markers.

▫ Assign a Help macro to a keyboard access key (or key combination).

▫ Start an application.

▫ Register a function within a DLL as a Help macro.

The complete list of macros that Windows Help version 3.1 supports is found in Chapter 15, "Help Macro Reference." Refer to that chapter for full details about each macro.

How Entry Macros Work

If a Help macro is included in a topic footnote, Windows Help executes that macro when the user first enters that topic. Users can display topics by:

▫ Clicking on a jump to that topic.

n Clicking the Back button or a browse button.

Selecting a topic from the history list or keyword search list.

Windows Help does not execute the macro a second time if the user performs any operation within the original topic, such as displaying a pop-up window or jumping to a spot reference.

Note

Windows Help executes any entry macros after it has laid out the text and graphics in the topic.

Inserting Macro Footnotes

You use an exclamation point (!) as the footnote character to execute topic-entry macros.

To insert a topic-entry macro

1. Position the insertion point at the beginning of the topic (to the right of the other footnotes).
2. From the Insert menu, choose Footnote.
The Footnote dialog box appears.
3. Type the exclamation point as the custom footnote mark, and then choose OK.

A superscript exclamation point ($^!$) appears in the text window, and the insertion point moves to the footnote window.
4. Type the macro to the right of the exclamation point in the footnote window.
Use only a single space between the exclamation point and the first word of the title. For example, you might type the following macro:

$^!$ ChangeButttonBinding("btn_contents", "JumpID(`hgcd.hlp', `acc_idx_hg')")

Figure 6.12 shows this footnote window.

Graphic

Note that you *can* include spaces in macros.

Entry Macro Guidelines

When creating topic entry macros, follow these guidelines:

- n   You can execute any macro listed in Chapter 15, "Help Macro Reference," within a topic footnote; however, if the topic is to be displayed in a secondary window, the macro may be ignored or cause undesirable behavior.

- n   The macro text can have as many as 512 characters, including any embedded spaces.

- n   You can specify more than one macro to be executed by separating them with a semicolon or colon, as in the following example:

! ChangeButttonBinding("btn_contents", "JumpID(`hgcd.hlp',
`acc_idx_hg')");EnableButton("btn_up");ChangeButttonBinding("btn_up",

Specifying Browse Sequences

"JumpID(`cdcd.hlp', `hlpidx_idx_card')")

Invariably, certain topics relate to other topics. For the information in your Help file to be clear and understandable, you might need to display certain groups of topics in a specific sequence. Windows Help lets you group specific topics and structure them in units called *browse sequences*. A browse sequence is simply a series of topics that appear in a sequence when the user chooses the << and >> buttons–like turning pages in a book. These buttons are optional in Windows Help version 3.1. To use browse buttons in a Help file, you must include the **BrowseButtons** macro in your Help project file. For more information, see the [CONFIG] section in Chapter 16, "The Help Project File," and the "BrowseButtons" section in Chapter 15, "Help Macro Reference."

When you design your Help file, you must first decide if you want to use browse sequences. If you do, you must decide which topics to group together into browse sequences. You might decide to place the topics within a browse sequence in separate files, with the first topic in the sequence at the beginning of the file and the last topic in the sequence at the end of the file. In the sample Help file, each category of information is grouped in a separate browse sequence.

Once these sequences become part of a Help file, the << and >> buttons on the Windows Help button bar let users move back and forth between topics within the browse sequence. The browse buttons become active whenever Windows Help displays topics coded with a browse sequence. At the beginning of the sequence, only the >> button is active. At the end of the sequence, only the << button is active. In the middle of the sequence, both buttons are active.

How Browse Sequences Work

You define browse sequences for your topics by assigning a plus sign (+) footnote to the topics in each browse sequence. A browse sequence footnote has a *sequence name*, followed by a colon, followed by a *sequence number*. A sequence name identifies the sequence to which the topic belongs. The sequence number specifies the location in the sequence where the topic is displayed.

A Single Browse Sequence

If your Help file has only one browse sequence, you don't need to assign a sequence name to browse sequence footnotes. You simply specify the sequence number in each footnote. Windows Help then displays the topics in the order of these sequence numbers.

You might want to skip one or more numbers in a sequence so you can add new topics later. For example, use increments of five or ten. The Windows Help compiler ignores skipped numbers; only their order is significant. (See the "Sorting Order" section, later in this chapter.)

You aren't required to use sequence numbers in footnotes. If you don't, Windows Help displays topics in their order of appearance in the topic file.

Multiple Browse Sequences

To set up more than one browse sequence in a Help file, use different sequence names to identify the different browse sequences. In the browse sequence footnote for each topic, assign a sequence name and a sequence number separated by a colon, as shown below:

*sequence-name***:***sequence-number*

For example, here's how the command topics within the FILENAME.RTF file, and the procedure topics within the FILENAME.RTF file, are arranged in a browse sequence:

cmd:0005	how:0005
cmd:0010	how:0010
cmd:0015	how:0015
cmd:0020	how:0020
cmd:0025	how:0025
.	.
.	.
.	.

This creates two different browse sequences: one for PLACEHOLDER and one for PLACEHOLDER.

If you don't want to keep track of sequence numbers for each sequence and you want to display topics in their order of appearance, use a constant value for the browse-sequence number in each footnote. For example, to display the topics in the PLACEHOLDER and PLACEHOLDER sequences in their order of appearance within the file, assign the sequence names and a constant number as follows:

FILENAME.RTF   FILENAME.RTF

cmd:0000	how:0000
cmd:0000	how:0000
cmd:0000	how:0000
cmd:0000	how:0000
cmd:0000	how:0000
.	.
.	.
.	.

This still creates the two browse sequences, but the topics in each sequence are displayed in their order of appearance in the topic file.

Sorting Order

Sequence numbers can include both numbers and alphabetic characters. During the compiling process, they're sorted using an alphabetic sort, not a numeric sort. Always use the same number of characters for each number in a sequence (don't use a two-character number for one topic and a three-character number for another). Otherwise, in certain cases, a higher sequence number could appear before a lower one.

For example, even though the number 100 is numerically higher than 99, the topic with sequence number 100 will appear before topic 99 because Windows Help is comparing a string beginning with 1 to a string beginning with 9. To keep the topics in their correct numeric order, you must make 99 a three-digit string (099).

Assigning Browse Sequence Footnotes

Once you determine the number and names of browse sequences and the order of topics in each sequence, you can assign sequence names and numbers to the topics.

To assign a browse sequence

1. Position the cursor at the beginning of the topic text.
2. From the Insert menu, choose Footnote.
3. Type a plus sign as the custom footnote mark, and then choose OK.

A superscript plus sign ($^+$) appears next to the heading, and the cursor moves to the footnote window.

4. Type the sequence name, a colon, and the sequence number to the right of the plus sign.

Use only a single space between the plus sign and the sequence name. For example:

$^+$ how:000

Figure 6.13 shows this footnote window.

Graphic

Assigning Build Tags

Build tags are labels (strings) that you assign to a topic to exclude that topic from a build under certain conditions.

Build tags are optional and not necessary for most Help projects. However, they do provide a means of supporting different versions of a Help system without having to create different source files for each version. Topics without build tags are always included in a build, as are all topics not expressly excluded from the build in the **BUILD** expression of the Help project file.

How Build Tags Work

Including a build tag footnote with a topic is equivalent to setting the tag to True when compared to the value set in the Help project file. The Help compiler assumes all other build tags to be False for that topic. After setting the value of the buildtag footnotes to True, the Help compiler evaluates the **BUILD** expression in the [OPTIONS] section of the Help project file.

Note

All build tags must be declared in the Help project file, regardless of whether the tags are declared conditionally. If the evaluation results in a True state, the Help compiler includes the topic in the build. Otherwise, the Help compiler omits the topic.

The Help compiler includes topics that do not have a build tag footnote in all builds, regardless of the build tag expressions defined in the Help project file. For this reason, you may want to use build tags primarily to exclude specific topics from certain builds. If the Help compiler finds any build tags not declared in the Help project file, it displays an error message.

By allowing conditional exclusion of specific topics, you can create multiple builds using the same topic files. This saves the Help writing team time and also enables you to maintain a higher level of consistency across your product lines.

Assigning Build Tag Footnotes

You use the asterisk (*) to insert build tags as footnotes. When you assign a build tag footnote to a topic, the Help compiler includes or excludes the topic according to information specified in the **BUILD** option and [BUILDTAGS] section of the Help project file. For more information about the **BUILD** option and the [BUILDTAGS] section, see Chapter 16, "The Help Project File."

To assign a build tag to a topic

1. Position the cursor at the beginning of the topic text, so that it appears before all other footnotes for that topic.
2. From the Insert menu, choose Footnote.
3. Type an asterisk as the custom footnote mark, and then choose OK.

A superscript asterisk (*) appears next to the heading, and the cursor moves to the footnote window.

4. Type the build tag name to the right of the asterisk in the footnote window.

Use only a single space between the asterisk and the build tag name. For example:

* test_build

Figure 6.14 shows this footnote window.

Graphic

Build Tag Guidelines

When adding build tags, follow these guidelines:

- A build tag must be the first footnote in a topic.
- Build tags can be made up of any alphanumeric characters and are not case sensitive. For example, "TEST_BUILD" is the same as "test_build."
- A build tag can have as many as 32 characters.
- Build tags may not contain spaces. Instead of spaces use underscores to separate parts of the build tag, as in this example:

  * beta2_build
- You can specify multiple build tags by separating them with a semicolon, as in the following example:

  * procedure_topics; beta2_build; test_build

Inserting Topic Comments

You can include comments with any topic in your Help file. Comments are purely optional statements that you create for your own use. They have no meaning to users or to the Help compiler, but they may be useful in your process. For example, you could use a comment footnote that tells writers and editors the topic's status. Whether you use comments and how you use them is strictly up to you. You use an at sign (@) footnote to identify a comment.

To insert a comment into a topic

1. Position the insertion point at the beginning of the topic (to the right of the other footnotes).
2. From the Insert menu, choose Footnote.
The Footnote dialog box appears.
3. Type the at sign as the custom footnote mark, and then choose OK.

A superscript at sign ($^{@}$) appears in the text window, and the insertion point moves to the footnote window.

4. Type the comment to the right of the at sign in the footnote window. Use only a single space between the at sign and the first word of the comment. For example, you might type the following comment for the "Saving Files" topic in Cardfile Help:

$^{@}$ This topic has been edited but not proofed yet.

Figure 6.15 shows this footnote window.

Graphic

Comment Guidelines

Because the Help compiler ignores comment footnotes, comments can have as many standard characters as you want, including accented characters and spaces.