Getting Started with Help Authoring

This chapter introduces you to the basic process of authoring and building Help files (.HLP). The "Building a Simple Help File" section presents two short walk-throughs that guide you through the authoring process by having you create and build a simple Help file yourself. Because each walk-through uses a different process, you can become familiar with different ways to create Help files and

Windows Help Files

choose the authoring method that works best for you.

When developing a Help system, the objective is to create one or more Help files. A *Help file* is the binary file that is actually displayed by the Windows Help application. In essence, the Help author's work parallels the programmer's development of an application. All the work and coding are produced for the express purpose of creating an application that a user can use. Similarly, all the writing and developing of Help files is done to produce a Help file that the user can view in Help.

That describes the purpose of a Help file, but what exactly is a Help file? In simple terms, it is a document. Like word-processing documents or spreadsheet documents, Help documents are defined in part by the information they contain. Spreadsheet documents are electronic ledger sheets that can perform calculations and other mathematical operations. Word-processing documents are electronic pages that can manipulate text and graphics. Help files are like word-processing documents in that they too can manipulate text and graphics. But Help documents also can include hypertext elements, such as links and hot spots, and multimedia elements, such as sound and video. And, unlike the other two kinds of documents, Help documents have no real-world equivalent. They exist only in electronic form on the computer. You can create a spreadsheet or a letter without a computer, but you can't create a Help document without a computer.

Help Topics

A Help file is made up of distinct units of information called *topics*. A topic in a

Help file is similar to a page in a word-processing document. Each topic focuses on a specific piece of information such as a concept, a step-by-step procedure, a command summary, an illustrative graphic, a keyboard table, an example, or any other unit of information that you define. A topic can include text or graphics or both text and graphics. By definition topics are an arbitrary length. Either the content itself or the Help author determines the length of a particular topic. The same is true of Help files. Since a Help file is made up of all the topics it contains, its size is determined wholly by the number of topics in it.

The first topic that appears when the user opens a Help file is the Contents topic, which acts as a table of contents for the Help file. It may also serve as the home screen from which all other topics or topic categories are accessed. The user can return to the Contents topic by choosing the Contents button (the first button on the Windows Help button bar). Figure 2.1 shows the Contents topic for the sample Help file–PLACEHOLDER.

Graphic

Note

You can view the sample application by clicking the PLACEHOLDER icon or by running Windows Help and opening the PLACEHOLDER.HLP file.

If the information in a Help topic doesn't fit in the current window, you can use the scroll bars or use the UP ARROW, DOWN ARROW, PAGE UP, or PAGE DOWN key to see the rest of the information.

Hypertext Jumps

Help topics are linked by *hypertext jumps*. When you choose text or a bitmap that activates a jump (called a *hot spot*), Windows Help clears the current topic and displays the topic indicated by the jump. Jumps are indicated in a Help file as follows:

- The mouse pointer changes from an arrow to a hand when it is over a hot spot.
- Text jumps appear in green with a solid underline.
- Bitmaps coded as jumps appear the same as regular bitmaps;

however, the pointer changes to a hand when it is over the bitmap hot spot.

Figure 2.2 shows a topic that contains a jump at the underlined text "PLACEHOLDER" (this text is green on a color monitor).

Graphic

To make the jump

- n   Position the mouse pointer over the hot spot and click the left mouse button.

Windows Help finds and displays the topic referenced by the jump, as in Figure 2.3.

Graphic

After jumping to a topic, you can return to the original topic by choosing the Back button on the Windows Help button bar.

Pop-Up Windows

Within topics, certain terms or concepts often require further explanation. Instead of having Windows Help jump away from the current topic to provide the additional information, you can have it display the information in a *pop-up window* on top of the current topic. Pop-up windows are stripped-down windows that have only a window border. They can display text and graphics, and they can include hot spots. Pop-up window hot spots are indicated in the Help file as follows:

- n   The mouse pointer changes from an arrow to a hand when it is over the hot spot.
- n   Text hot spots appear in green with a dotted underline.
- n   Bitmaps coded as pop-up hot spots appear the same as regular bitmaps; however, the pointer changes when it is over the bitmap hot spot.

In Figure 2.4, the first paragraph includes a hot spot that, when chosen, displays a pop-up window.

Graphic

When you click the hot spot, the pop-up window appears (Figure 2.5).

Graphic

Pop-up windows remain displayed until you click the mouse button again or press

Developing Help Files

any key.

You create Help files using the Help tools. At a minimum, you will use Word for Windows or another text editor, the Windows Help compiler, and the Windows Help application. (For a complete list of Help tools, see the Introduction to theis authoring guide.)

Developing a Help file involves several processes:

1. Develop a documentation plan for the Help project.
2. Program the application to access Help.
3. Create the topic files.
4. Create the Help project file.
5. Compile the topic files into a binary .HLP file.
6. Test and debug the Help file.

Developing a Documentation Plan

The documentation plan provides a way for Help authors to organize the Help project and to begin thinking about design issues. The document defines how the Help system will teach users about the product. Usually, one or more writers or instructional designers create the plan with input and suggestions from other team members, writing managers, developmental editors, product marketing managers, and program managers.

The documentation plan should describe:

- The product definition

- The documentation objectives and instructional model

- The components in the documentation package

- A content outline of each component

- Technical considerations and assumptions

- Resources and tools

- The delivery schedule and intermediate milestones

- Project issues and concerns

Programming the Application

In order for users to access your Help file, the application developers must program the application to access Help. Ideally, the application should provide context-sensitive Help. When users ask for Help, they expect the application to know where they are and what they have been doing. They become impatient when they ask for Help and are given irrelevant information or a long list of topics (from which they must choose the correct information). Context-sensitive Help is convenient, saves users time, and prevents users from overlooking something important.

Context-sensitive Help must be programmed into the application using the **WinHelp** API function (explained in Chapter 19). When creating context-sensitive Help, a software engineer assigns special ID numbers (often called hooks) to each context. After these context IDs have been assigned, the Help author can create the Help topics that the user will see when requesting Help on a particular interface component.

The application should be programmed early in the development cycle so that Help authors have adequate time to create the topics. Otherwise, writers will have to postpone their writing or reorganize the Help topics later in the process.

Creating the Topic Files

Windows Help files usually consist of several rich-text format (RTF) files, with each file divided into one or more topics. A topic is any distinct unit of

information that is separated by a page break, such as a Contents screen, a conceptual description, a set of instructions, a keyboard table, a glossary definition, a list of jumps, a picture, and so on.

You create *topic files* using a word processor or a text editor that can generate RTF files. (To ensure the highest compatibility, we recommend that you use Microsoft Word for Windows.) Creating Help topics involves roughly the same phases as creating printed books: gathering information, writing, and editing.

Generally, when gathering information for the Help topics, a Help author can rely on these sources:

n   The product specification

n   The application's printed documentation

n   Help files from a previous release of the application

n   Hands-on experience using the alpha and beta software

n   The software engineers creating the application

Topic files also contain special Help codes that pass organizational and build information to the Windows Help compiler. Within the topic text, you create cross-reference links, pop-up window hot spots, macro commands, references to graphics files, and other codes particular to your Help file. The finished topic files contain the content and build information for the Help file.

Creating the Help Project File

After you create the topic files, you build the Help file with the Windows Help compiler (HC31.EXE), which uses the topic files and the *Help project file* (a special build file with an .HPJ extension). The Help project file tells the compiler exactly how to build the Help file.

Building the Help File

After you create the topic files and you have a Help project file, you use the Help compiler to build the Help file. The Help compiler uses the Help project file and topic files to create a single Help file (with an .HLP extension).

During the build, the Help compiler displays error messages when it encounters a

problem. Help authors can use these build errors to debug the Help file.

Displaying the Built Help File

You can display the compiled Help file using the Windows Help application (WINHELP.EXE). Windows Help interprets and displays the elements of the Help file–text, pictures, hypertext links, and keyword search index–and lets the user interactively explore the information presented. In a built file, all the features designed into the Help file come into use; for example, users can browse topics as they would in a printed book or use hypertext links to jump around.

Note

   With the exception of being able to add notes and bookmarks, users can only view a compiled Help file; they can't change the actual content or structure of the file.

Because certain types of errors and authoring mistakes only show up when viewing the built file, the Help author should perform additional testing and debugging at this time. If errors do appear in the built version, the Help author

Building a Simple Help File

should correct the problem and rebuild the file.

Now that you've seen what a Help file looks like to the user and are familiar with the basic features of the Windows Help application, you're ready to learn how to build a Help file. You have two options to create topic files for Windows Help: you can use Word for Windows and enter all the Help-specific coding and information manually, or you can use Microsoft Help Author and enter the information in dialog boxes. Although the first process certainly works, using Help Author as your primary authoring tool makes creating Help files simpler and easier.

The following sections show you how to build a simple Help file using both of these methods. The Help file you build in each case is exactly the same. If you are new to Help authoring, it is a good idea to be familiar with both authoring methods, but if you want to save time, and you know which method you want to use, you can complete just that section.

Using Help Author to Build a Simple Help File

This section explains how to create and build a simple Help file using Microsoft Help Author. The simple Help file consists of two topics, and one of the two topics includes a jump to the other topic.

To build this simple Help file, you will:

▫ Create a project subdirectory for the source files.

▫ Define a new Help project using the Help Project Editor.

▫ Create two topics using Word for Windows: enter context strings for the two topics, code a jump from one topic to the other, add a bitmap to one topic, and save the topic file as RTF.

▫ Build the Help file.

▫ Test the Help file by displaying it in Windows Help.

Note

This section assumes you have already installed Help Author. If you have not, please go to Chapter 5, "Using Help Author," and read the installation instructions before going any further.

Creating a Project Subdirectory

When you create a Help file using Help Author, you may want to create a separate project subdirectory for all the source files. (You might also create separate subdirectories for the bitmaps in your Help file.) For this example, create a temporary project subdirectory called SAMPLE in the Help Author directory.

To set up the SAMPLE subdirectory

▫ Enter the following commands at the MS-DOS prompt:

cd \helpauth (or the name of the directory you specified during setup)
md sample

Starting the Help Project Editor

Help Author uses the Help Project Editor to control the Help creation process. So, to create a Help file, you first start the Help Project Editor. After starting the Help Project Editor, you can define the new project, add topic files, start builds, and view the completed Help file.

To open the Help Project Editor

1. Start Windows if it is not already running.
2. Open the Help Author group window in Program Manager, and double-click the Help Project Editor icon.
When you first start the Help Project Editor, a blank, untitled Help Project Editor window opens (Figure 2.6).

Graphic

Creating a New Help Project

If you have never used the Help Project Editor before, it will automatically start with a new, undefined project. But just to make sure, we'll create a new project for this sample walk-through.

To create a new Help project file

n From the File menu, choose New Project.

Defining the New Project

After you create a new project, you define project-level information, including the title of the Help project, the context ID for the Contents topic, and the version of the Help compiler to use for the build.

To define the new project

1. From the Edit menu, choose Project.
The Project dialog box appears (Figure 2.7).

Graphic

2. Type the following text in the Title box:
Sample Help File

This title will appear in the title bar of the built Help file.

3. Type the following text in the Contents box:

topic1_ID

We will use this context string to define both the Contents and the first topic in the Help file.

4. In the Help Version box, select the Help 3.1 option button.

This indicates that you want to use Windows Help version 3.1.

5. Choose OK.

Adding a Topic File to the Help Project

When you create a Help project file, you add all the topic files that you want to include in the Help file. For this sample Help file, we will create just one topic file.

To add a new topic file

1. From the Edit menu, choose Add New Or Existing File.

The Add New Or Existing File dialog box appears.

2. In the Directories box, double-click the Sample subdirectory.

Or press the UP ARROW or DOWN ARROW key to select the directory, and then press ENTER.

3. Type the following filename:

sample.rtf

4. Choose OK.

Since this file is new, the Help Project Editor asks if you want to create it.

5. Choose OK.

The Help Project Editor displays SAMPLE.RTF in the project window. Your screen should look like the one in Figure 2.8.

Graphic

Creating the Topic File

To have a Help file, you must create topics. Topics contain the text and other information that appear in your Help file. You create topics using Word for Windows and the appropriate Help Authoring Template and then save the topics in a topic file. Once you add a topic file to the project, you can edit it without

quitting the Help Project Editor.

To edit the sample topic file

n Double-click SAMPLE.RTF in the Project Editor window.

Or select the file, and choose Edit File from the Edit menu.
The Help Project Editor starts Word for Windows and opens the sample
file. Your screen should look like the one in Figure 2.9.

Graphic

Note
All Word for Windows screens in this guide are shown as they
appear with Hidden Text and Paragraph Marks selected from the
View Preferences (version 1.1) or the Tools Options (version 2.0)
dialog box.

Creating Individual Topics

First you add all the topics that you want included in the Help file.

To create a new topic

1. From the Insert menu, choose Topic.
The Insert Topic dialog box appears (Figure 2.10).

Graphic

Note   The text boxes in the dialog box represent the topic footnotes
supported by the Help compiler.
2. In the Title box, type the following text:
Topic 1
Each topic usually has a *title.* The title identifies the topic in the history
list and in keyword searches the user performs.
3. In the Context String box, type the following text:
topic1_ID
Each topic usually has a *context string.* The context string identifies the
topic within the Help file. For example, to jump to a topic you must
provide the context string of the topic string to which you want to jump.
4. Skip Keywords, Browse Sequence, Build Tag, Entry Macro, and
Comments.

5. Choose OK.

Your screen should look like the one in Figure 2.11.

Graphic

6. Repeat steps 1 through 5 for Topic 2. However, substitute the number 2 for the number 1 (Topic 2, topic2_ID).

Your screen should now look like the one in Figure 2.12.

Graphic

Inserting a Jump Hot Spot

Now that you've identified your two topics, you can insert a jump from Topic 1 to Topic 2.

To insert a jump

1. Create a new paragraph after the paragraph containing the words "Topic 1 Title," and then type:

Jump to Topic 2

2. Select the text you typed in step 1.

3. From the Insert menu, choose Jump or Pop-Up Hot Spot.

The Insert Jump or Pop-Up Hot Spot dialog box appears (Figure 2.13).

Graphic

4. Press TAB.

5. Type the following context string:

topic2_ID

6. Choose OK.

Your screen should look like the one in Figure 2.14.

Graphic

Adding a Picture

The next step is to add a picture to one of your topics. Although you can add bitmaps directly to a topic, it is usually better to enter a bitmap reference that tells Windows Help the name of the bitmap file to display in the topic.

To insert a bitmap in the topic

1. Position the insertion point in the empty paragraph after the paragraph containing the words "Topic 2 Title."

2. From the Insert menu, choose Graphic.

The Insert Graphic dialog box appears (Figure 2.15).

Graphic

3. In the File Name box, type the following text:

sample.bmp

Note   SAMPLE.BMP is one of the sample bitmaps provided with the Windows Help software.

4. Choose OK.

Your screen should look like the one in Figure 2.16.

Graphic

Saving the Topic File

Now that you've created the topic file, save it as RTF.

To save your topic file and close Word for Windows

1. From the File menu, choose Save As.
2. In the File Name box, type this filename:

sample.rtf

3. Choose the options button.
4. From File Format, select RFT.
5. Choose OK to save the file.
6. From the File menu, choose Exit.

If Word for Windows asks if you want to save changes to SAMPLE.RTF, choose No. You already saved your changes in RTF format in steps 4 and 5.

Saving the Project File

Now you're ready to save the Help project file.

To save the project file

1. From the File menu, choose Save Project As.

2. In the File Name box, type this filename:
sample.hpj
3. Choose OK to save the file.

Building the Help File

After you have created topics for your Help file, you use the Help Project Editor to define the build options. Depending on the complexity of the Help file you are building, the Help project file may have very little information or quite a lot. For this sample Help file, you won't need to define any additional options. You are ready to start the build.

To start the build

- From the Compile menu, choose Start.

The Compilation in Progress window appears, showing the progress of the build and any errors that occur.

Displaying the Help File

After you build a Help file, you use the Help Project Editor to display it in Windows Help to test it and to make sure the features you added work correctly.

To display the sample Help file

- From the File menu, choose Run Help On SAMPLE.HLP.

The Help Project Editor displays the Help file in Windows Help (Figure 2.17).

Graphic

As you can see, "Jump to Topic 2" appears in green, underlined text, indicating that it is a jump hot spot.

To test the jump

1. Move the mouse pointer over the green text.
Notice that the pointer changes to a hand shape with a pointing finger.
2. Click the left mouse button.

Your second topic appears, as in Figure 2.18.

The SAMPLE.BMP bitmap file appears below the "Topic 2" text.

Now that you have your Help file working, you can quit Windows Help.

To quit Windows Help

n   From the File menu, choose Exit.

Quitting Help Author

Now that you have successfully created a sample Help file using Help Author, you can close the program.

To quit Help Author

n   From the File menu, choose Exit.

Or double-click the Control-menu box.
If you haven't saved your changes, the Help Project Editor prompts you to save the project file.

Building a Simple Help File Without Help Author

This section explains how to create and build a simple Help file using Microsoft Word for Windows. The simple Help file consists of two topics, and one of the two topics includes a jump to the other topic.

To build this simple Help file, you will:

n   Create a project directory for the source files.
n   Create two topics using Word for Windows: enter context strings for the two topics, code a jump from one topic to the other, add a bitmap to one topic, and save the topic file as RTF.
n   Create a Help project file with information for building the Help file.

n   Build the Help file with the Help compiler.

Microsoft Windows Help Authoring Guide Test the Help file by displaying it in Windows Help.

Creating a Help Project Directory

When you develop a Help file, you ordinarily create a separate project directory on your hard disk drive to hold and organize all the source files. (You might also create separate subdirectories for the text files and bitmaps in your Help file.) For this example, create a temporary project directory called SAMPLE and copy the Windows Help software to that directory.

To set up the SAMPLE subdirectory

n   Enter the following commands at the MS-DOS prompt:

**cd **
md sample
copy [*sourcepath*]\\hc31.exe c:\\sample\\hc.exe
copy [*sourcepath*]\\hc31.err c:\\sample\\hc.err
copy [*sourcepath*]\\winhelp.exe c:\\sample
   The *sourcepath* is the floppy drive letter or other drive letter or path that describes where the files are located.

Creating a Help Topic File

The first step in creating a Help file is to create the topic files. Topic files contain the text and other information that appears in your Help file.

Creating Individual Topics

The topic is the basic building block of a Help file. You create topics using Word for Windows (or other RTF editor) and then save them in a topic file.

To create a topic file

1. Start Word for Windows and open a new document.
2. Type the following text:
Topic 1
3. Press CTRL+ENTER to insert a hard page break.

Hard page breaks separate topics in a topic file.

4. Type the following text for the second topic:

Topic 2

Your screen should look like the one in Figure 2.19.

Graphic

Note

All Word for Windows screens in this guide are shown as they appear with Hidden Text and Paragraph Marks selected from the View Preferences (version 1.1) or the Tools Options (version 2.0) dialog box.

Entering Context Strings

Each topic usually has a *context string*. The context string identifies the topic within the Help file. For example, to jump to a topic you must provide the context string of the topic string to which you want to jump. You insert the context string using the pound-sign (#) footnote in Word for Windows.

To enter a context string

1. Position the cursor at the beginning of the line containing the words "Topic 1."
2. From the Insert menu, choose Footnote.
3. Type a pound sign (#) as the footnote character.
4. Type the following text as the footnote reference:

topic1_ID

5. Repeat steps 1 through 4 for Topic 2. However, type **topic2_ID** as the footnote text.

Your screen should look like the one in Figure 2.20.

Graphic

Inserting a Jump

Now that you've identified your two topics, you can insert a jump from Topic 1 to Topic 2. To insert a jump, you first type the hot-spot text–that is, the text the user will click to make the jump–and then you identify the destination topic for the jump.

To insert a jump

1. Position the insertion point at the end of the first paragraph in Topic 1 and press ENTER. Then type:

Jump to Topic 2

2. Select the text you typed in step 1.

3. From the Format menu, choose Character.

4. Select the Double Underline check box, and then choose OK.

Formatting text as double underline tells the compiler to make the text "Jump to Topic 2" into a hot spot.

5. Position the insertion point immediately after the "Jump to Topic 2" text and type:

topic2_ID

6. Select the text you typed in step 5.

Be sure not to select any of the double-underlined text or the paragraph mark at the end of the line.

7. From the Format menu, choose Character.

8. Clear the Double Underline check box, select the Hidden check box, and then choose OK.

Formatting text as hidden tells the compiler that "topic2_ID" is the context string of the destination topic for the jump.

Your screen should look like the one in Figure 2.21.

Graphic

Note

When you remove the selection highlight from the text, you will be able to see the hidden text formatting.

Adding a Picture

The next step is to add a picture to one of your topics. Although you can add bitmaps directly to a topic, it is usually better to enter a bitmap reference that tells Windows Help the name of the bitmap file to display in the topic.

To insert a bitmap in the topic

1. Create a new line after the line containing the words "Topic 2."

2. On the new line, type the following:

{bmc sample.bmp}

Your screen should look like the one in Figure 2.22.

Graphic

The **bmc** command tells the Help compiler to position the SAMPLE.BMP bitmap below the text in your topic. (SAMPLE.BMP is one of the sample bitmaps provided with the Windows Help software.)

Saving the Topic File

After you've created the topic file, save it as RTF.

To save your topic file and close Word for Windows

1. From the File menu, choose Save As.
2. In the File Name box, type this filename:
sample.rtf
3. Choose the Options button.
4. From File Format, select RTF.
5. Choose OK to save the file.
6. From the File menu, choose Exit.
If word hen Word for Windows asks if you want to save changes to SAMPLE.RTF, choose No. You already saved your changes in RTF format in steps 4 and 5.

Saving the Project File

Now you're ready to save the Help project file.

To save the project file

1. From the File menu, choose Save Project As.
2. In the File Name box, type this filename:
sample.hpj
3. Choose OK to save the file.

Creating a Help Project File

Now that you've created topics for your Help file, you create a Help project file–a text file that tells the Help compiler exactly how to build the Help file. The Help

project file can include different types of information. Depending on the complexity of the Help file you are building, the Help project file may have very little information or quite a lot. At minimum, the Help project file must include a list of the topic files used to build the Help file.

The structure of the Help project file is the same as .INI files in Windows–the file is divided into sections, each of which contains information or settings for a particular aspect of an application. In the Help project file, the sections contain information about different aspects of the Help file. However, except for the [FILES] section, a section is required only if you want to include a specific feature that it contains.

The Help project file should have the same name as your Help file but with an .HPJ extension. You can use any text editor to create it, as long as you save it as unformatted text (standard ASCII format). For this exercise, save the Help project file in the SAMPLE directory.

Listing Topic Files

List the files that you're using to build your Help file in a section beginning with the following heading:

[FILES]

Under this heading, type the names of the RTF files to be included in the Help file. Because you have only one topic file, SAMPLE.RTF, the [FILES] section looks like this:

[FILES]
SAMPLE.RTF

Specifying Options

Although not required, you will usually include an [OPTIONS] section so that you can specify particular build options for the Help file. Enter these options under the following heading:

[OPTIONS]

To identify the project directory where the source files are stored, to indicate the window title you want displayed in the Help title bar, and to specify the level of warning messages to display during the build, type the following lines exactly as they are shown below:

[OPTIONS]
ROOT=C:\SAMPLE

Note

You also included a bitmap in one of the topic files. But because the bitmap resides in the same project directory as the other files, you don't have to specify anything in the Help project file.

The Complete Help Project File

Once you've finished these steps, your Help project file should look like this:

```
[OPTIONS]
ROOT=C:\SAMPLE
TITLE=SAMPLE HELP
WARNING=3

[FILES]
sample.rtf
```

The [OPTIONS] section, if used, must appear first in the Help project file.

Save this file as SAMPLE.HPJ in the SAMPLE directory. Be sure you save the file as Text Only if you are using a word processor instead of an ASCII text editor.

Building Your Help File

The final step is the simplest.

To build your Help file

n   Using the Windows Run command (or from the MS-DOS prompt), run the Help compiler from the SAMPLE directory, giving the name of your Help project file.

For the sample you created in this chapter, type the following command:

hc sample.hpj

The compiler builds your Help file and displays messages during the build. When it is complete, you have a Help file named SAMPLE.HLP.

Testing Your Help File

After you build a Help file, open it in Windows Help to test it and to make sure the features you added work correctly.

To start Windows Help

1. From Program Manager, press F1 or use the Run command to start Windows Help.
2. After Windows Help is running, choose Open from the File menu.
3. Type the name (and path if necessary) of your Help file (SAMPLE), and then choose OK.

Or use the Browse button to locate and select the SAMPLE.HLP file.

When Windows Help first opens the Help file, it appears as in Figure 2.23.

Graphic

As you can see, "Jump to Topic 2" appears in green, underlined text, indicating that it is a jump hot spot.

To test the jump

1. Move the mouse pointer over the green text.

Notice that the pointer changes to a hand shape with a pointing finger.

2. Click the left mouse button.

Your second topic appears, as in Figure 2.24.

Graphic

The SAMPLE.BMP bitmap file appears below the "Topic 2" text.

Quitting Help

Now that you have your Help file working, you can quit Windows Help.

To quit Windows Help

n   From the File menu, choose Exit.