

---

# Creating Links and Hot Spots

---

So far we have been talking about creating individual topics. That's certainly a necessary first step, but it isn't sufficient to create a Help file because unless you create links between those topics, users have no way to access the information. Unlike books, Help files don't have physical pages users can turn. They rely instead on *links*, the structural and navigational connections between topics. Links determine which topics are connected to each other and the relationship between connected topics. For example, a link may determine that Topic B is subordinate to Topic A or that Topic C is cross-referenced from Topic A.

Linking information together electronically is what makes normal text into *hypertext*. Normal text is what we read in most printed materials: newspapers, magazines, novels, brochures, catalogs, instruction manuals, and so on. It is structured sequentially so that people read one page after another. Hypertext simply means that you can create links between the electronic pages (topics) in such a way that users can read the material nonsequentially, or in any order they choose. If the hypertext is well-made, they should be able to choose links according to the associations they form when they read the information. In Windows Help, this notion of hypertext is the basis for all Help files.

This chapter explains how to create links between topics and topic files. It also explains how to create hot spots that run Help macros and activate pop-up

## Linking Topics Using Jumps

windows.

Help provides many ways to link topics, but most methods are implicit: they connect topics by category (browse sequences) or by subject matter (keywords). These methods don't provide a concrete connection for the user: click something and a topic appears. For that reason, the most common method for linking topics is to use *jumps*. Jumps are one-way links between topics: when the user chooses a jump, Help displays the topic connected by the link (usually referred to as the *destination topic*). Because jumps are one-directional, a link from Topic A to Topic B does not guarantee a link in the opposite direction. Jumps can occur automatically when a topic is displayed, or they can occur when the user clicks a

*hot spot*, which is a special topic region within a Help topic. A *jump hot spot* is a word or a graphic that the user chooses to move to another topic in the Help file. Hot spot text appears as green, underlined text in the Help window. Graphics used as hot spots do not change appearance.

Jumps are a flexible navigational tool. You can create many links between topics and display topics in a variety of windows. By effectively linking information in your Help file, you can make it easy for users to find specific information. You can link related topics using cross-references, inviting your users to explore related information. You can create logical paths through the information so users can browse without getting lost.

## Creating Tables of Contents

Users can only see the information you have created if you provide them with jumps to that information. If you don't provide a link to a particular topic, users will not be able to choose that information. (Of course, you can have topics that are accessed only by keyword search, but for the moment we are assuming that users are making their choices based on what is visible in the main Help window.) For that reason, it is a good idea to create a Help Contents topic as a minimal device for providing access to every topic in your Help system. That way users will have at least one jump to all primary topics.

Jumps provide a good way to create structure and organization within a Help file. For example, you can use jumps to create a hierarchical browsing order for a collection of topics. Consider the following graph, which represents the organizational structure of a Help file as defined by the topic jumps, as shown in Figure 8.1.

### Graphic

Here, the topic links create a clear information hierarchy. The four table of contents topics (C0 through C3) contain lists of hot spots that connect to information topics (Inf1 through Inf11). The contents topics let the user move from general to detailed lists of information, quickly arriving at the appropriate topic. For example, the Figure 8.2 shows two contents topics.

### Graphic

With a three-level hierarchy as shown here, users could access any topic using only two mouse clicks. Add a third level of contents topics, and you provide access to all topics using only three mouse clicks.

---

## Using Jumps to Reference Related Information

### Creating Links and Hot Spots § 8-3

In addition to creating overall structure in the Help file, you can use jumps to create links between topics that contain related information. For example, the following graph shows a few cross-reference links between topics in our sample Help file, as shown in Figure 8.3.

## Graphic

The cross-reference links let users quickly find related information without having to return to the Contents topic (C0). Cross-reference jumps can be an effective tool for providing multiple levels of detail in the same topic. A topic might contain detailed information that's interesting to some users, but not all. Using cross-reference jumps, you can hide the more detailed information in a separate topic, as shown in Figure 8.4:

## Graphic

### Jump Destinations

The topic displayed as a result of a jump is called the *destination topic*. You can create jumps to any topic in the current Help file. You can also create jumps to a specific location within a topic.

#### Note

Using Help macros and other techniques, you can create jumps to topics in other Help files. For more information, see “Creating Links Between Help Files,” later in this chapter.

You can display the destination topic in the main Help window, in a secondary window, or in a pop-up window (Figure 8.5).

## Graphic

For a complete discussion of Help windows, see Chapter 9, “Defining Topic Windows.”

### Defining Mid-Topic Jump Destinations

Normally, Help displays the first line of the topic at the top of the Help window.

However, you can define jump destinations anywhere within a topic. That way when Help displays the topic, the mid-topic destination appears at the top of the Help window. Mid-topic jumps are especially useful if you have longer topics, because you can create jumps that take users directly to the relevant information within the topic.

### To insert a mid-topic destination

1. Position the insertion point where you want the topic to appear in the Help window.
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.
4. Type a context string for the mid-topic destination.

The mid-topic destination appears at the top of the Help window, as in Figure 8.6.

## Graphic

### Using Jump Macros

Help provides several macros that also perform topic jumps. You might consider using these macros to create a specialized user interface for displaying topics. For information on using macros, see Chapter 14, “Help Macros.”

### Activating Jumps

You can author a jump to be activated automatically when the topic is displayed or when the user chooses a hot spot.

### Entry Macros

You can place *entry macros* in any topic and in the Help project file. Entry macros let you run one or more Help macros with the display of the Help file or a topic. Entry macros linked to topics can be activated by jump hot spots. For example, if users jump to a destination topic with an associated entry macro, the macro string for that topic is run automatically.

Entry macros are a powerful tool for configuring Help to display special topics or to provide helpful navigational feedback. Chapter 14, “Help Macros,” discusses

---

the different ways a Help file can run macros and suggests how to use them in your Help file.

## Using Entry Macros to Create Automatic Jumps

Entry macros have many uses, one of which is to create automatic jumps when a topic is displayed. For example, you can use an entry macro to display a group of topics—in the main Help window and in one or more secondary windows—when the user enters a particular topic.

When displaying groups of topics together, you should identify a *master topic* to display in the main window. By placing an entry macro in the master topic, Help can display the whole group of topics when users jump to the master topic. After displaying the master topic, Help processes the automatic jumps contained in the master topic and displays the appropriate topics in the other windows, as in Figure 8.7:

## Creating Hot Spots

### Graphic

A hot spot is a region within a Help topic that users can interact with. Help changes the mouse pointer to a hand when the pointer passes over a hot spot. The user can click the mouse to initiate the action associated with the hot spot.

You can format the following elements as hot spots.

<b>Element</b>	<b>Description</b>
Text	A contiguous series of characters.
Picture	A single picture.

---

Hypergraphic

A rectangular region defined on top of a special picture called a *hypergraphic*.

---

## How Jump Hot Spots Appear in Help

After you build the Help file, users can identify hot spots in two ways:

- By default, the text in jump hot spots (if they are not formatted as invisible) appear as green, underlined text.
- When the mouse pointer moves over a jump hot spot, the pointer changes from an arrow to a hand. This change occurs whether or not the hot spot is visible.

Figure 8.8 shows a Help topic that has all three types of hot spots. Notice the hot areas created by each type of hot spot. To choose a hot spot, users position the mouse pointer over a hot spot and click the left mouse button.

## Graphic

When the user clicks a hot spot, Help processes the associated action immediately. The action occurs at the moment the mouse button is depressed.

## Text Hot Spots

A text hot spot uses a word or phrase as the hot region. The characters in the text hot spot must be contiguous. Help can word-wrap text within a hot spot, so the hot region might extend to two lines or more. By default, Help displays text hot spots as dark green text with a single underline. To override the default formatting, you can use the method described in the “Changing the Standard Appearance of Hot Spots section”, later in this chapter. Help then uses whatever Word formatting you apply to the text.

If you change the default formatting, keep in mind the following considerations:

- Use some visual or semantic cues to identify the hot spots. If you make the hot spot look exactly like surrounding text, users will have

---

to move the mouse pointer around to find hot spots. This can frustrate users or prevent them from finding important information you've included in your Help file.

---

- The single underline used with the default hot-spot formatting helps users identify hot spots on monochrome displays. Also, the underline lets color-blind users distinguish hot spots from surrounding text.
- Usability tests with Windows Help files have shown that some users confuse italic text with hot spots.

### Bitmap Hot Spots

You can create graphical hot spots using bitmaps. With a bitmap hot spot, the user clicks anywhere on the bitmap to activate the hot spot. Bitmap hot spots use the same display mechanism as regular pictures, so you can use your 16-color bitmaps and metafiles as hot spots.

Bitmap hot spots use the same positioning options as regular pictures (except for inline bitmaps, which don't support text wrapping). For information on the alignment options for Help graphics, see Chapter 1

### Hypergraphic Hot Spots

You can create pictures with multiple hot spots so that the user can interact with different parts of the picture. Because each hot spot can initiate a different action, you can use hypergraphics to extend the usefulness of graphics in your Help file. To create hypergraphic hot spots, you use a separate application called Hot Spot Editor, which is described in Chapter 11, "Creating Hypergraphics."

### Types of Hot Spots

Help lets you define several types of hot spots.

<b>Hot spot type</b>	<b>Action</b>
Jump	When the user clicks the hot spot, Help displays the destination topic in the specified window, either main or secondary.

---

## Microsoft Windows Help Authoring Guide

---

Interfile jump	When the user clicks the hot spot, Help opens a new Help file and displays the destination topic in the specified window.
Pop-up	When the user clicks the hot spot, Help displays the topic in a pop-up window.
Macro	When the user clicks the hot spot, Help executes the specified macro.
Invisible	The hot spot has no visible formatting to indicate it is a hot spot. However, the pointer changes to a hand when moved over the hot spot.

---

## Special Hot-Spot Characters

When creating hot spots in Help topics, Windows Help assigns a specific meaning to certain standard characters. These special Help characters are described below.

<b>Character</b>	<b>Name</b>	<b>Purpose</b>
Asterisk (*)	Underlined hot spot	Specifies that the hot spot should be underlined only. The default green color is removed from the hot spot.
At sign (@)	Interfile jump	Specifies that the jump is to a topic located in a different .HLP file from the current file.
Exclamation point (!)	Macro hot	Specifies that the hot spot contains a Help



---

spot macro to be executed.

---

#### Creating Links and Hot Spots§ 8-9

---

Percent sign (%)	Invisible hot spot	Specifies that the hot spot should be invisible. The default green color and underlining are removed.
Right angle bracket (>)	Window name	Specifies that the topic is to be displayed in the specified window, either the main Help window or a secondary window.

---

For information about using each of these special characters, see the following

## Inserting Standard Jump Hot Spots

sections in this chapter.

The coding for a standard jump hot spot has two parts:

- A reference word or phrase formatted as double-underlined text (the jump hot-spot text) that the user chooses to make the jump
- The context string, formatted as hidden text, that identifies the destination topic

### Note

In Word for Windows, select the Hidden Text check box in the View Preferences (version 1.1) or Tools Options (version 2.0) dialog box to display hidden text. You can also assign a macro key to make selecting this option more convenient.

### To format a word or phrase as a jump hot spot

1. Select the jump hot-spot text.
2. From the Format menu, choose Character.

3. Select the Double Underline check box, and then choose OK.
  4. Position the insertion point immediately after the last letter in the double-underlined jump hot-spot text.
  5. From the Format menu, choose Character again.
  6. Clear the Double Underline check box, select the Hidden check box, and then choose OK.
  7. Type the context string assigned to the topic that is the target of the jump.
  8. From the Format menu, choose Character again, clear the Hidden check box, and then choose OK.
- Figure 8.9 shows a correctly formatted jump hot spot in a topic file.

## Graphic

### Jump Hot Spot Guidelines

When creating jump hot spots, follow these guidelines:

- You can have spaces within the hot spot text, but be sure no spaces are between the double-underlined jump text and the hidden text context string. For example, the following is correct.

## Graphic

The following is incorrect.

## Graphic

- If you place the context string at the end of a paragraph or at the end of a line with a soft carriage return (SHIFT+ENTER), be careful not to

## Creating Links Between Help Files

format the paragraph or carriage-return mark as hidden text.

If you build a Help system that has more than one Help file, you can link the information together by creating jumps from one Help file to another, or *interfile jumps*. An interfile jump is simply a jump from a topic in one Help file to a topic in a different Help file. (Jumps like the ones we've discussed so far occur within

---

a single Help file. Remember, *Help file* does not refer to the individual RTF topic files that contain the topics, but to the compiled, binary .HLP file.)

---

**Creating Links and Hot Spots § 8-11**

---

### Note

Chapter 16, “The Help Project File,” contains information on building a multifile Help system. Here, it is assumed that you’ve compiled the individual Help files and want to create a jump from one file to another.

You can use several methods to create interfile jump hot spots:

- Create an interfile jump hot spot
- Create a macro hot spot and use one of the **Jump** macros
- Have an application or DLL send an API message to Help requesting a new Help file and topic

## How Interfile Jump Hot Spots Appear in Help

In the built Help file, interfile jumps appear exactly the same as standard jumps. For our example, the interfile jump hot spot appears as the underlined text “Jump Text” (see Figure 8.x). To make the jump, users position the mouse pointer over the interfile jump hot spot text and click the left mouse button.

### Graphic

Windows Help finds and displays the topic referenced by the interfile jump hot spot, as in Figure 8.x.

### Graphic

## Creating Standard Interfile Jump Hot Spots

The simplest way to create an interfile jump is to create a standard jump hot spot and add the name of the Help file:

*context-string@.HLP-filename*

*Context-string* is the context string for the topic you are jumping to, and *HLP-*

---

*filename* is the name of the .HLP file that contains the topic.

---

### To create an interfile jump hot spot

1. Follow the steps to create a standard jump hot spot.
2. Insert an at sign (@) immediately after the last character of the context string.
3. Type the name of the file that contains the context string assigned to the topic that is the target of the interfile jump.

**Note** Both the at sign and Help filename must be formatted as hidden text.

**Note** You can also specify a path along with the Help filename. Generally, this is not recommended because the path is likely to vary from computer to computer and also because the Help file may change its location later. If you have circumstances that diminish these risks, you may want to specify a path, as in these examples:

```
ctx_string@\\server\share\file.hlp
```

```
ctx_string@c:\dir1\dir2\file.hlp
```

Figure 8.x shows a correctly formatted interfile jump hot spot in a topic file.

## Graphic

For more information about creating interfile jumps using macro hot spots, see the “Creating Macro Hot Spots” section, later in this chapter. For more information about the **Jump** macros, see Chapter 15, “Help Macro Reference.” For an explanation of the **WinHelp** API, see Chapter 19, “The WinHelp API.”

## How Windows Help Locates Help Files

To perform an interfile jump, Help must be able to find the destination Help file on the user’s system. When the user chooses an interfile jump hot spot, Help looks for the Help file in the following locations:

- The directory of the Help file that initiated the interfile jump (Help’s current directory).
- The MS-DOS current directory.
- The user’s Windows directory.

- The Windows SYSTEM directory.
- The directory containing WINHELP.EXE.
- The directories listed in the user's PATH environment variable.
- The directories specified in WINHELP.INI.

If Help cannot find the Help file after searching in all these locations, it displays a "Help file not found" error message.

**Note**

Help also uses this search order when you execute Help from the MS-DOS command line or when an application sends an API call to Help.

### Creating a WINHELP.INI File for Search Paths

If you are distributing your application and Help files on removable media such as CD-ROM, you can have your setup program define the drives and directories where Help should look for Help files. The paths you define will be added to Help's regular search paths when performing interfile jumps.

The syntax for the WINHELP.INI entry is:

```
[FILES]
HLP-filename=drive:\directory-string, message
```

Parameter	Description
-----------	-------------

<i>HLP-filename</i>	The name of the Help file for which you want to define a search path.
<i>drive</i>	The letter of the drive that contains the Help file. The drive letter must be an alpha character. You can specify as many as 26 drives.
<i>directory-string</i>	The directories where Help should look for the Help file. The directory string can have as many as 256 characters.

---

The message that Help should display in a dialog box, prompting the user to insert the correct disk into the designated drive. The message can have as many as 50 characters.

---

If the application or setup program creates an invalid entry, Help will display the standard “Help file not found” error message. So, Help authors are encouraged to test the validity of each WINHELP.INI entry.

## Example

The following entry tells Help to look in the \PRODUCT\FILES directory of the P drive for the “Cardfile Help” file. If necessary Help will prompt the user to insert the CD into the drive:

```
[FILES]
CARDFILE.HLP=P:\PRODUCT\FILES, Please insert your Windows version 3.1 CD into drive P:.
```

When creating entries in the WINHELP.INI file, you must also observe the following guidelines:

- Each entry can be no longer than 512 characters.
- The first application to create the WINHELP.INI file must also create the [FILES] section heading. You can use the Windows function **WritePrivateProfileString** to create entries in the WINHELP.INI file. For more information on this function, see the Windows version 3.1 SDK.

## Creating Links to Version 3.0 Help Files

In some circumstances you may want to create a jump hot spot in your 3.1 Help file that jumps to a topic in a 3.0 Help file. If you create a standard interfile jump, Help will display the “Help topic does not exist” error message when the user chooses the jump. That is because Windows Help version 3.1 does not allow you to jump to any arbitrary topic in a 3.0 Help file based on the context string. However, you can use macros to create links to topics in a 3.0 Help file.

From a 3.1 Help file, you can:

- 
- Jump to the Contents topic in a 3.0 Help file by using the **JumpContents** macro. Creating Interfile Jump Spots § 8-15

```
!JumpContents("30file.hlp")
```

- Jump to a topic with a particular keyword in a 3.0 Help file by using the **JumpKeyword** macro. This method will work only if the topic has the keyword defined for it and that topic appears first in the list of topics that have that keyword defined.

```
!JumpKeyword("30file.hlp", "keyword")
```

- Jump to a specific topic in a 3.0 Help file that has a context number defined in the [MAP] section of the Help project file by using the **JumpContext** macro.

```
!JumpContext("30file.hlp", number)
```

- Jump to any topic within a browse sequence in a 3.0 Help file by combining one of the above **Jump** macros with the **Next** and **Prev** macros. This method will work only if the topic you want to access is on a browse sequence, and you know the context number of another topic on the same browse sequence. When you use this method, Help will not display any of the intermediate topics in the sequence.

```
!JumpContext("30file.hlp", 801);Next();Next();Next()
```

For information on how to create a macro hot spot, see the “Creating Macro Hot Spots” section, later in this chapter.

## Interfile Jump Hot Spot Guidelines

When creating interfile jump hot spots, follow these guidelines:

- You can have spaces within the hot spot text, but be sure no spaces are between the double-underlined jump text and the hidden text context string, as in Figure 8.x:

### Graphic

Figure 8.x shows an incorrect example:

---

## Graphic

**Microsoft Windows Help Authoring Guide** If you place the context string at the end of a paragraph or at the end of a line with a soft carriage return (SHIFT+ENTER), format the paragraph or carriage-return mark as normal text (don't format it as

## Creating Pop-Up Hot Spots

hidden text).

Windows Help lets you display information in a pop-up window that appears whenever the user chooses a hot spot such as a word or a phrase or a graphic (see Chapter 10, "Adding Graphics"). A *pop-up window* is a window that appears on top of the main Help window and contains a topic.

You can use pop-up windows to display hints, notes, or any other type of supplementary information. A pop-up window can contain text, graphics, or hot spots (jump hot spots, pop-up hot spots, or macro hot spots). However, if you have a hot spot within a pop-up window, the first pop-up window closes when the user chooses the hot spot. In other words, you can only display one pop-up window at a time on the screen.

### Note

The procedures for creating pop-up hot spots are exactly the same as those for creating jump hot spots with one exception: the text for pop-up hot spots is formatted with a single underline and jump hot spot text is formatted as double-underline. Therefore, if you are familiar with the material in the previous section, you may want to skim the information in this section.

### How Pop-Up Hot Spots Appear in Help

After you build the Help file, users can identify pop-up hot spots in two ways:

- n By default, pop-up hot spots (if they are not invisible) appear as green, dotted-underlined text.



- n When the mouse pointer moves over a pop-up hot spot, the pointer changes from an arrow to a hand. This change occurs whether or not the hot spot is visible.

In the sample pop-up window in Figure 8.x, the first paragraph includes a pop-up hot spot that displays a pop-up window when chosen. The words “Pop-up text” appear in the pop-up window.

## Graphic

To see the pop-up window, the user clicks the hot spot, which causes the window to appear directly on top of the current topic, as in Figure 8.x.

## Graphic

The pop-up window includes a shadow to make it more distinguishable from the main window, and it remains open until the user clicks the left mouse button again or presses any key (which includes choosing a hot spot within the pop-up window).

## Creating Standard Pop-Up Hot Spots

The coding for a standard pop-up hot spot has two parts:

- n A word or phrase, formatted as underlined text (the hot-spot text), that the user chooses to display the pop-up window
- n The context string, formatted as hidden text, that identifies the pop-up topic (what the user sees in the pop-up window)

### **To format a word or phrase as a pop-up hot spot**

1. Create the content of the pop-up topic.
2. Define a context string footnote for the pop-up topic.  
 For example, the following context string identifies the pop-up window topic for “Topic A”:  

```
# ctx_string_topic_a
```
3. Go to the topic from which you want to display the pop-up window.
4. Select the hot spot text that will display the pop-up window.
5. From the Format menu, choose Character.

6. Select the Underline check box, and then choose OK.
  7. Position the insertion point immediately after the last letter in the underlined hot spot text.
  8. From the Format menu, choose Character again.
  9. Clear the Underline check box, select the Hidden check box, and then choose OK.
  10. Type the context string assigned to the pop-up topic.
  11. From the Format menu, choose Character again, clear the Hidden check box, and then choose OK.
- Figure 8.x shows a correctly coded pop-up hot spot in a topic file.

## Graphic

### Pop-Up Hot Spot Guidelines

The same guidelines for creating jump hot spots apply to creating pop-up hot spots:

- Be sure no spaces are between the underlined pop-up hot spot text and the hidden text context string, as in Figure 8.x:

## Graphic

Figure 8.x shows an incorrect example:

## Graphic

- If you place the context string at the end of a paragraph or at the end of a line with a soft carriage return (SHIFT+ENTER), format the paragraph or carriage-return mark as normal text (don't format it as

## Creating Macro Hot Spots

hidden text).

In addition to jump hot spots and pop-up hot spots, you can also create *macro hot spots*, a word or a phrase or a graphic (see Chapter 10, “Adding Graphics”) that the user chooses to execute a Help macro. Text formatted as macro hot spots appear as green, underlined text in the Help window—exactly the same as jump hot spots.

---

You can use macro hot spots to customize Windows Help or individual Help topics in many ways. For example, you could have macro hot spots that:

---

**Creating Links and Hot Spots § 8-19**

- Start another application
- Create a new button or menu item
- Change what happens when a user chooses an existing button or menu item
- Display a topic in a secondary window
- Change the size and location of a Help window
- Print the current topic
- Display a custom Help file in a second Help window
- And perform many other actions

Macros provide a convenient way to create many interesting effects and refine the presentation of your Help file. How you use them is really up to you.

**Note**

The procedures for creating macro hot spots are exactly the same as those for creating jump hot spots with one exception: the hidden text following the hot-spot text contains the macro string(s) to be executed instead of a topic context string. Therefore, if you are familiar with the material in the “Creating Links Between Topics” section, you may want to skim the information in this section.

## How Macro Hot Spots Appear in Help

In the built Help file, macro hot spots appear exactly the same as standard jump hot spots. For our example, the macro hot spot appears at the underlined text “Macro hot spot Text” (see Figure 8.x). To choose the macro hot spot, users position the mouse pointer over the macro hot-spot text and click the left mouse button.

## Graphic

Windows Help executes the macro referenced by the macro hot spot, as in Figure 8.x.

---

# Graphic

## Creating Standard Macro Hot Spots

The coding for a standard macro hot spot has two parts:

- A reference word or phrase formatted as double-underlined text (the macro hot spot text) that the user chooses to make the macro hot spot
- The macro string, formatted as hidden text, that identifies the macro that Help executes when the user chooses the hot spot

### Note

In Word for Windows, select the Hidden Text check box in the View Preferences (version 1.1) or Tools Options (version 2.0) dialog box to display hidden text. You can also assign a macro key to make selecting this option more convenient.

### To format text as a macro hot spot

1. Select the macro hot-spot text.
2. From the Format menu, choose Character.
3. Select the Double Underline check box, and then choose OK.
4. Position the insertion point immediately after the last letter in the double-underlined macro hot spot text.
5. From the Format menu, choose Character again.
6. Clear the Double Underline check box, select the Hidden check box, and then choose OK.
7. Insert an exclamation point (!) as the first character of the macro string.

**Note** The exclamation point must be formatted as hidden text.

8. Type the macro string that you want Help to execute when the user chooses this hot spot text.
9. From the Format menu, choose Character again, clear the Hidden check box, and then choose OK.

Figure 8.x shows a correctly coded macro hot spot in a topic file.

# Graphic

---

## Macro Hot Spot Guidelines

Creating Links and Hot Spots § 8-21

---

When creating macro hot spots, follow these guidelines:

- The macro hot-spot text can have as many as 512 characters, including any embedded spaces.
- Be sure no spaces exist between the double-underlined macro hot spot text and the hidden text macro string. For example, the following is correct:

### **Graphic**

The following is incorrect:

### **Graphic**

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

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

- If you place the macro string at the end of a paragraph or at the end of a line with a soft carriage return (SHIFT+ENTER), format the paragraph or carriage-return mark as normal text (don't format it as hidden text).
- You can execute any macro listed in Chapter 15, "Help Macro Reference," from a macro hot spot; however, if the topic is to be displayed in a secondary window, the macro may be ignored or cause undesirable behavior.

## Creating Links to Online Tutorials

Help version 3.1 includes limited support for creating links between an application's Help file and an online tutorial. This means that you can create hot spots within the Help file that link to specific tutorial lessons. When the user chooses a tutorial hot spot, Help starts the tutorial application and informs it that

the user has requested a tutorial lesson. Help then turns control over to the tutorial application so that it can run the lesson.

Creating tutorial links within a Help file requires cooperation from all sources:

- The Help author is responsible for creating the hot spots that the user chooses to see a tutorial lesson.
- Windows Help is responsible for sending a message to the tutorial application and informing it that the user has requested a particular lesson. To send this message, Help may also have to start the tutorial application.
- The tutorial application is responsible for starting and running the tutorial lesson and for returning control to Help when the user has finished using the tutorial.

## Creating the Hot Spots

If your application includes an online tutorial, you can use the **ExecProgram** macro to create hot spots within the Help file that link to tutorial lessons.

### Note

For detailed information about the **ExecProgram** macro, see Chapter 15, “Help Macro Reference.”

### To create a macro hot spot that links to a tutorial lesson

1. Follow the steps to create a standard macro hot spot.
2. Type the tutorial information for this hot spot in the **ExecProgram** macro string.

**Note** The exclamation point (!) and macro string must be formatted as hidden text.

For example, the following macro would start the tutorial application TUTOR.EXE, which is located in the LESSONS subdirectory, and display a lesson on the application’s Toolbox in a maximized window:

```
!ExecProgram("c:\\lessons\\tutor.exe toolbox.cbt", 1)
```

**Note** If your tutorial application supports command-line options, you can also include them when starting the lesson. For example, a command-line option may allow you to start the lesson in a specific location. Refer to the documentation that accompanies your tutorial application for information about the command-line options that it supports.

Figure 8.x shows a correctly formatted macro hot spot to a tutorial lesson.

## Graphic

### Tutorials on Windows Help

Because Windows Help is included in so many different software packages, users often have their Help system replaced by the install program they are using to install a new software program. This makes it virtually impossible for any single application to know which version of Windows Help users have installed on their system. For that reason, Help authors at Microsoft have decided not to create online tutorial lessons about Windows Help. Therefore, Windows Help 3.1 has removed the “CBT aware” functionality that was required to support tutorials

## Changing the Standard Appearance of Hot Spots

about Help.

A standard hot spot in Windows Help means that the hot spot appears as green, solid- or dotted-underlined text to the user. You can change the default appearance of hot spots by:

- Removing both the green color and underline (invisible hot spots).
- Removing just the green color (underlined hot spots).
- Changing the green color and removing the underline (custom hot spots).

No matter how you change the appearance of hot spots, the Windows Help cursor

changes to a hand pointer when it is over the hot-spot text. You may want to change the default appearance of hot spots to create a custom look for your Help file.

**Note**

Changing the standard appearance of hot spots may or may not be a good idea. It depends on why you are creating the Help file. If you are creating Help for a standard Windows application, you may not want to have your hot spots look different from the standard that Windows users recognize. On the other hand, if you are creating a Help file for your personal use, changing the hot spots may be essential to getting the look you want. For more discussion about these issues, see Chapter 4, “Help Authoring Guidelines.”.

**Note**

Changing the appearance of jump hot spots, pop-up hot spots and macro hot spots involves essentially the same procedures.

## Creating Invisible Hot Spots

You can make some or all of your hot spots invisible. Invisible hot spots appear as normal text to the user, but the Windows Help cursor changes to a hand pointer when it is over an invisible hot spot. Invisible hot spots are a useful way to create hot spots that look like regular text. For example, you may want to use invisible jump hot spots in a Contents topic to have better control over the appearance of the list. Or, you may want to use invisible pop-up hot spots in a Master Glossary topic where all the hot spots display a pop-up window with a definition.

**Note**

Because invisible hot spots do not have any visible formatting to inform users, it is a good idea to tell them that there are hot spots on the screen.

### To create an invisible hot spot

1. Follow the steps to create a standard jump hot spot, pop-up hot spot,



---

or macro hot spot.

2. Insert a percent sign (%) as the first character of the context string (just before the exclamation point if it is a macro hot spot).

**Note** The percent sign must be formatted as hidden text.

Figure 8.x shows a correctly formatted invisible hot spot in a topic file.

## Graphic

### Creating Underlined Hot Spots Without Color

You can create *underlined hot spots* in your Help topics. Underlined hot-spots appear to the user as normal, solid- or dotted-underlined text without the standard green hot spot color. However, the Windows Help cursor changes to a hand pointer when it is over the hot spot. Underlined hot spots are a useful way to remove the default green color but retain the underlining as the hot-spot indicator.

#### To create an underlined hot spot

1. Follow the steps to create a standard jump hot spot, pop-up hot spot, or macro hot spot.
2. Insert an asterisk (\*) as the first character of the context string (just before the exclamation point if it is a macro hot spot).

**Note** The asterisk must be formatted as hidden text.

Figure 8.x shows a correctly formatted underlined hot spot in a topic file.

## Graphic

### Creating Hot Spots with Custom Colors

You can create *custom-color hot spots* in your Help topics by changing the default color of the hot spots. Custom-color hot spots are a useful way to create a unique look for your Help file. However, you should be aware of the issues surrounding the decision to change Help's standard hot spot color.

For guidelines on using color in Help files, see Chapter 3, "Designing Your Help System."

### Issues to Consider Before Changing Hot Spot Colors

When deciding whether to use colored text in your Help file, consider the

---

following issues:

---

**Microsoft Windows Help Authoring Guide**

» Author-defined colors may conflict with the user's default screen colors.

Most Microsoft applications do not override the user's default colors, and users expect Help to be consistent with that behavior.

» In some circumstances (displaying on gray-scale monitors, for example), colored text may not be visible on the user's machine.

Help cannot guarantee that colors the user hasn't chosen will have enough contrast to be visible.

» You cannot rely on color to make important distinctions of meaning.

If the product is localized for an international market, the colors will have no intrinsic meanings in the various countries and cultures.

Moreover, on monochrome monitors the color disappears, so the Help file loses any meaning associated with color.

You can solve all these problems only if users are in complete control of their screen colors.

### **The Windows Color Palette**

The Windows palette provides 16 colors, many of which are not good choices as text colors. Windows will only display text in solid colors, so you can't use any dithered colors. If we assume that the background color of the Help window is white (not necessarily a good assumption), this eliminates using any bright colors such as yellow or cyan. The dark colors, like dark green and dark red, are all readable, but some, like dark blue, have less contrast with black. Dark green is readable and not visually distracting, but brighter than dark blue. If you want to give the text more emphasis, you might choose dark green.

On the other hand, if you want to give the colored text less emphasis but have some differentiation from black, you might choose dark blue. The effect of the blue text would be more subtle than the dark green. However, blue text is less recognizable than black text because we have fewer receptors in our eyes that can detect blue. Therefore, blue fades to the background and does not stand out. Blue is high-contrast, but isn't easy to distinguish from black for everyone. So generally, blue text should be used sparingly as a secondary color.

---

## Conflicts with the User's Screen Colors

The Windows default color scheme includes black text on a white background. However, the Windows Control Panel lets users change their default screen colors, including the text and background colors. Because the Help application does not interfere with a user's selection, Help authors cannot assume that Help will always display black text on a white background.

Depending on a user's choice, colored text may be difficult or impossible to see. For example, if a user chooses yellow text on a black background, and the Help author uses dark blue text in the Help topics, the colored text may be very difficult to read. Individual computers may also have hardware or other display-related problems, such as an inability to resolve contrast between some colors—light grey and dark grey, for example.

## Problems with Color on Different Displays

Help cannot predict how the text colors are actually displayed on different monitors. For example, on some gas-plasma displays that use VGA color drivers, Help displays colors as various shades of red (gray-scale red).

Help can predict the readability of a given text color on a given background color only under two circumstances:

- The text color is the user's default text color.

In this circumstance, Help relies on the user to make the right choice. If the text isn't readable, the user must change the color in Control Panel.

- The author defines a text color that is exactly the same as the background color.

In this circumstance, Help uses the default text color and displays it as normal text.

## Problems with Assigning Meaning to Color

In addition to the technical aspects of color, color also affects a user's emotions. In fact a big part of human color perception is psychological and emotional. For example, dark green and dark blue are fairly bland colors but appeal to a wide audience. Dark red is brighter and hence more attractive to the eye, but it is often associated with anger or strong emotion.

These emotional responses are also related to cultural associations—red means stop, warning, passion; green equals go, ripe, money. In American culture, red text might be good for warnings, but it can be a very bad color for Help text that contains neutral information. Moreover, you cannot use color to create absolute meaning because color associations vary within a culture, and from culture to culture. If you're writing Help for an international product, the English version will be used by many non-Americans who will interpret your color associations differently.

## Overriding Custom Hot Spot Colors

If your Help file contains a custom color for hot spots, the user cannot use the standard WIN.INI entries—**JumpColor**, **PopUpColor**, **MacroColor**, **IFJumpColor**, **IFPopUpColor**—to override your custom colors. To override an author-defined hot spot color users must type the following entry in the WIN.INI file:

[Windows Help]

**Colors=NONE**

When Help finds this entry in the WIN.INI file, it uses the system default colors for foreground and background colors, regardless of how the Help file was authored.

If you change the standard color of hot spots in your Help file, it is your responsibility to instruct users how to override your custom colors.

## Creating Custom Hot Spots

There are four ways to create a custom-color hot spot. You can:

- Create an underlined hot spot and change the color of the hot spot text to one of the Windows standard 16 colors (Windows color, standard solid- or dotted-underline).
- Create an underlined hot spot and change the color of the hot spot text to a custom RGB color (custom RGB color, standard solid- or dotted-underline).
- Create an invisible hot spot and change the color of the hot spot text to one of the Windows standard 16 colors (Windows color, no solid- or dotted-underline).
- Create an invisible hot spot and change the color of the hot spot text

---

to a custom RGB color (custom RGB color, standard solid- or dotted-underline).

### **To create a hot spot with a Windows color and standard underline**

1. Follow the steps to create a standard jump hot spot, interfile jump hot spot, pop-up hot spot, or macro hot spot.
2. Insert an asterisk (\*) as the first character of the context string (just before the exclamation point if it is a macro hot spot).

**Note** The asterisk must be formatted as hidden text.

3. Select the hot spot text.
4. From the Format menu, choose Character.
5. In the Color box, choose the color that you want for the hot spot text.
6. Choose OK.

Figure 8.x shows a correctly formatted custom underlined hot spot in a topic file.

## **Graphic**

### **To create a hot spot with a custom RGB color and standard underline**

1. Follow the steps to create a standard jump hot spot, interfile jump hot spot, pop-up hot spot, or macro hot spot.
2. Insert an asterisk (\*) as the first character of the context string (just before the exclamation point if it is a macro hot spot).

**Note** The asterisk must be formatted as hidden text.

3. Select the hot spot text.
4. From the Format menu, choose Character.
5. In the Color box, choose the color that approximates the custom color you want for the hot spot text.

For example, if you want to create a custom blue, choose one of the blues from the Color box.

6. Choose OK.
7. Save the file as RTF.
8. Open the RTF file as Text; do not convert it to .DOC format.
9. In the `\colortbl` section of the RTF header, change the RGB value of the hot spot color.

For example, if your hot spots are blue (`\red0\green0\blue255`), change

the RGB value of blue in the `\colortbl` to your custom color, `\red0\green0\blue64`, for example.

**Note** Help uses `\red0\green128\blue0` for the standard hot spot color. Figure 8.x shows a correctly formatted custom hot spot in a topic file.

## Graphic

10. Save the RTF file as text.

When you build the Help file, the hot spots will appear in the custom color.

### To create a hot spot with a Windows color and no underline

1. Follow the steps to create a standard jump hot spot, interfile jump hot spot, pop-up hot spot, or macro hot spot.
2. Insert a percent sign (%) as the first character of the context string (just before the exclamation point if it is a macro hot spot).

**Note** The percent sign must be formatted as hidden text.

3. Select the hot spot text.
4. From the Format menu, choose Character.
5. In the Color box, choose the color that you want for the hot spot text.
6. Choose OK.

Figure 8.x shows a correctly formatted custom hot spot in a topic file.

## Graphic

### To create a hot spot with a custom RGB color and no underline

1. Follow the steps to create a standard jump hot spot, interfile jump hot spot, pop-up hot spot, or macro hot spot.
2. Insert a percent sign (%) as the first character of the context string (just before the exclamation point if it is a macro hot spot).

**Note** The percent sign must be formatted as hidden text.

3. Select the hot spot text.
4. From the Format menu, choose Character.
5. In the Color box, choose the color that approximates the custom color you want for the hot spot text.

For example, if you want to create a custom red, choose one of the reds from the Color box

6. Choose OK.
7. Save the file as RTF.
8. Open the RTF file as Text; do not convert it to .DOC format.

---

9. In the `\colortbl` section of the RTF header, change the RGB value of the hot spot color.

**Creating Links and Hot Spots** § 8-31

---

For example, if your hot spots are red (`\red255\green0\blue0`), change the RGB value of red in the `\colortbl` to your custom color, `\red64\green0\blue0`, for example.

**Note** Help uses `\red0\green128\blue0` for the standard hot spot color. Figure 8.x shows a correctly formatted custom hot spot in a topic file.

## Graphic

10. Save the RTF file as text.

When you build the Help file, the hot spots will appear in the custom color.