
Introduction

The Microsoft® Windows™ Help application and compiler let you provide online Help to users of your Windows-based application. Microsoft Windows Help version 3.1 supports hypertext and context-sensitive links and integrates many advanced hypertext capabilities, making Windows Help an optimal tool for developing small Help systems and large, full-featured online documents.

About This

The *Microsoft Windows Help Authoring Guide* provides information for writers and developers of Help systems for Microsoft Windows-based applications. This section introduces the Windows Help environment and provides background information you should review before developing a Help file.

This guide is written for anyone who is creating a Help file, whether that person is a technical writer or a developer of applications for Windows. The word “you” as used in this guide refers either to the Help author or to the developer creating the Help system. The term “user,” on the other hand, refers to the person who will eventually use the Help files you create.

The *Microsoft Windows Help Authoring Guide* contains the following chapters:

- Chapter 1, “The Windows Help Application,” explains the basic features of the Windows Help application that displays Help files.
- Chapter 2, “Getting Started with Help Authoring,” explains the basic process for creating and building Help files. In this chapter you’ll create a simple Help file to learn the basics of Help authoring.
- Chapter 3, “Designing the Help System,” discusses issues and tradeoffs to consider when designing your Help system.
- Chapter 4, “Help Authoring Guidelines,” provides a set of general and specific guidelines you can follow as you create topics.
- Chapter 5, “Using Help Author,” explains Microsoft Help Author, a tool that makes creating Help files simpler and easier. The first part of the chapter explains how to use the Help Project Editor to create Help files. The second part explains how the Windows Help Authoring Templates customize Microsoft Word for Windows to

make creating topic files easier.

Chapter 6, “Creating Topics,” explains how to create Help topic files and how to add the Help-specific information that the Help compiler uses to identify topics, create a keyword index, organize topics into browse sequences, and run macros.

- ⁿ Chapter 7, “Formatting Topics,” explains how to format the text and graphics that you include in topic files. The chapter describes various formatting techniques, such as selecting fonts, creating tables, creating margins, and using special characters.
- ⁿ Chapter 8, “Creating Links and Hot Spots,” explains how to create links between topics in the Help system and how to create hot spots that run Help macros.
- ⁿ Chapter 9, “Defining Topic Windows,” explains how to display topics in different Help windows, such as pop-up, secondary , and embedded windows. It also explains how to add nonscrolling regions to topic windows.
- ⁿ Chapter 10, “Adding Graphics,” explains how to add pictures to topic files, how to control the layout of text and graphics within a topic, and how to display information activated from “hot” graphics.
- ⁿ Chapter 11, “Creating Hypergraphics,” describes hypergraphics and explains how to use Hotspot Editor to create graphics with multiple hot spots.
- ⁿ Chapter 12, “Creating Graphics for Different Displays,” explains how to use the Multi-Resolution Bitmap Compiler to combine graphics with different display resolutions into a single file format so that they will look good on different machines.
- ⁿ Chapter 13, “Customizing the Help File,” explains how to use Help features to change the way your Help file appears and how users work with it. The first part of the chapter discusses ways to customize Help menus and buttons. The second part shows other ways to change Help features.
- ⁿ Chapter 14, “Help Macros,” describes Help macros and explains the rules for constructing macros and using them in the Help file.
- ⁿ Chapter 15, “Help Macro Reference,” contains a list of all the Help

macros you can use to customize your Help file or the Windows Help feature set.

iii

- Chapter 16, “The Help Project File,” describes the format and contents of the Help project file, which defines how a Help file is built, and explains how to use it to customize Windows Help for your Help file.
- Chapter 17, “Building the Help File,” explains how to build a Help file, how to fix problems that arise during the build process, and how to display a Help file after it has been built.
- Chapter 18, “Help Error Messages,” explains each of the Help error messages that you might encounter when building a Help file with the Help compiler version 3.1 or 3.0.
- Chapter 19, “The WinHelp API,” explains the **WinHelp** API. It describes how to create context-sensitive links within the application and how to call **WinHelp** to perform Help-related operations.
- Chapter 20, “Writing DLLs for Windows Help,” explains how to write a dynamic-link library (DLL) for Windows that extends the functionality of Windows Help. This chapter shows how an application can use custom DLLs to provide additional functionality to Help authors or to control the behavior of topic elements placed in embedded windows.
- Appendix A, “Windows Virtual-Key Codes,” shows the symbolic constant names, hexadecimal values, and keyboard equivalents for the virtual-key codes used by Microsoft Windows version 3.1. You use these virtual-key codes to provide keyboard access for Help macros within the Help file.
- Appendix B, “Help RTF Statements,” describes the syntax and purpose of rich text format (RTF) statements supported by the Microsoft Help compiler. The RTF statements define the formats used to encode Help features in the source topic files.
- Appendix C, “Baggage Access Functions,” provides specialized source code that can be built into an application or custom DLL so that it can retrieve the appropriate data file from the Help file’s [BAGGAGE] section.

About Windows

Windows Help lets users view Help files and other online documents in the Microsoft Windows graphical environment. Windows Help files present online information using the following elements:

- Text, with multiple fonts, type sizes, and colors
- Graphics (in several formats and using as many as 16 colors)
- Segmented hypergraphics (bitmaps with embedded hot spots)
- Cross-reference jumps for linking information
- Pop-up windows for presenting additional text and graphics in a nonintrusive way
- Secondary windows for presenting information in a controlled format
- Keyword search capability for finding specific information

Windows Help provides a practical way to combine different types of information into a format users can access easily from Windows and from Windows-based applications. Use Windows Help to:

- Create Help files for a Windows-based application.
- Supplement or replace printed product documentation.
- Create stand-alone online documents.
- Convert hard-to-access printed manuals into electronic format.

Note

Even though you can use the Windows Help tools to create different types of electronic documents, this guide refers to all created files as “Help files.” The information, however, applies equally to these other documents.

Windows Help maintains upward compatibility—Help files built with Help version 3.0 are compatible with Help version 3.1, but Help files built with the version 3.1 Help compiler will not work in version 3.0 Help applications.

What's New in Version

The basic process for developing Help files remains the same; however, the number of features has increased dramatically. You can do everything you did using version 3.0—and much more. Some of the new features^v in version 3.1 of Windows Help are:

- ⁂ You can create new menus and menu items or modify existing menus and menu items. See Chapter 14, “Help Macros,” and Chapter 16, “The Help Project File.”
- ⁂ The Help button bar has a new look and a new button called History. You can add your own buttons to the standard buttons, or you can modify the function of existing buttons. See Chapter 13, “Customizing the Help File,” Chapter 14, “Help Macros,” and Chapter 16, “The Help Project File.”
- ⁂ The Copy command has a dialog box that lets users select the exact text they want to copy to the Clipboard. See Chapter 1, “The Windows Help Application.”
- ⁂ In addition to the main Help window, you can define secondary windows that also display Help information. See Chapter 9, “Defining Topic Windows,” and Chapter 16, “The Help Project File.”
- ⁂ You can control the size, placement, and background color of all Help windows, both main and secondary. See Chapter 13, “Customizing the Heop File,” and Chapter 16, “The Help Project File.”
- ⁂ Help windows can have a nonscrolling region that does not move when the user scrolls information. The nonscrolling region can include the same elements as the main window. See Chapter 9, “Defining Topic Windows,” and Chapter 16, “The Help Project File.”
- ⁂ Help includes a set of macros that you can use to modify and extend the functionality of Windows Help. The Help macros can affect an entire Help file, or they can be limited to a single topic or a single hot spot within a topic. See Chapter 14, “Help Macros.”
- ⁂ Pop-up windows stay up until the user closes them with another action, so you can use pop-up windows for a wider variety of information with text, graphics, and hot spots. See Chapter 1, “The Windows Help Application.”

- In addition to links between topics, you can create links to information within the same topic or to topics in other Help files. See Chapter 6, “Creating Topics,” Chapter 8, “Creating Links and Hot Spots,” and Chapter 14, “Help Macros.”
- Topics can include dynamically sized, multiple-column tables. See Chapter 7, “Formatting Topics.”
- Help includes the Hotspot Editor application so that your Help graphics can include hot spots for pop-up windows, hypertext jumps, or macros. See Chapter 10, “Adding Graphics,” and Chapter 11, “Creating Hypergraphics.”
- Help includes the Multi-Resolution Bitmap Compiler so that you can create graphics that display correctly on different video resolutions. See Chapter 12, “Creating Graphics for Different Display.”
- The Help project file includes several new sections and many new options, which gives you increased control over how the Help file is created and built. See Chapter 16, “The Help Project File.”
- The Help compiler has been rewritten for improved performance; error messages have also been revised and rewritten. See Chapter 17, “Building the Help File,” and Chapter 18, “Help Error Messages.”
- The Help compiler supports three levels of compression. See Chapter 16, “The Help Project File.”
- To make developing Help files easier and more efficient, Windows Help includes the Help Authoring Templates and the Help Project Editor. These additions let you use menu commands and dialog boxes to create Help files within the Windows graphical environment instead of entering Help codes in your RTF editor. See Chapter 5, “Using Help Author.”
- You can write custom DLLs that extend Help’s functionality by including features that you want to be part of your Help system. See Chapter 14, “Help Macros,” and Chapter 20, “Writing DLLs for Windows Help.”
- You can create an embedded window within a Help topic and use a DLL to display an object, such as an animation or a 256-color bitmap, in the window. See Chapter 20, “Writing DLLs for Windows

Help.”

What Should You Know to

Because Windows Help integrates text and graphics, you might use many skills during the creation of a Help file—document analysis, writing, editing, graphic design and

production, and for programming and compiling Windows. This guide assumes you have this expertise or that you work with others who do.

The actual skills necessary to create the RTF files from which the Help file is generated, however, are much simpler. To start using Windows Help and this authoring guide, you should have:

- Experience with MS-DOS®.

Although most of your development work takes place within the Windows graphical environment, you should know the basics of the MS-DOS operating system. Managing a Help project requires some understanding of MS-DOS commands and directory structures.

- Experience using Windows and an understanding of the Windows user interface.

Before starting development on your Help system, you should install Windows version 3.0 or 3.1 on your computer and learn how to use it. Be sure to learn the name, purpose, and operation of each part of a Windows-based application (such as windows, dialog boxes, menus, controls, and scroll bars). Because the Windows Help application incorporates these features, it is very important that you understand them so you can implement them properly in your application and Help system.

- Experience using Word for Windows or another word processor.

Microsoft Word for Windows is the preferred word processor for creating Windows Help files. Many of the features in Windows Help were designed with Word for Windows in mind. You can develop Help files using other editors, but they may present challenges that you typically avoid when using Word for Windows.

- An understanding of the user-interface style guidelines for Windows.

One goal of Microsoft Windows is to provide a common user interface for all applications, including the Help application. This ultimately helps the user by reducing the effort required to learn the user interface, and it helps you by clarifying the choices you have to make when designing your application and Help system. Even though the content of the Help system varies from application to application, the user expects Help to be the same. Therefore, it is important that you build in a certain amount of consistency with other Help systems to make it easier for users to learn your Help system.

System Requirements

The system required to create a Windows Help file is different from the system required to view the Help file.

Authoring System Requirements

The following is the recommended configuration to create and test a Help file.

Hardware

- ▮ 80386-based computer running at 25 MHz
- ▮ 4 megabytes RAM
- ▮ 70 megabytes hard-disk storage
- ▮ VGA (16-color) monitor
- ▮ Tape drive or other storage device to create backups

Software

- ▮ MS-DOS version 3.3 or later
- ▮ Microsoft Windows version 3.0 or 3.1
- ▮ Microsoft Windows Help version 3.1
- ▮ Microsoft Word for Windows version 1.1 or 2.0

Viewing System Requirements

The following is the recommended configuration to display a Windows Help file.

Hardware

- 80286-based computer running at 10 MHz
- 1 megabyte RAM
- 20 megabytes hard-disk storage
- EGA (monochrome or color) monitor

Software

- MS-DOS version 3.3 or later
- Microsoft Windows version 3.0 or 3.1
 - Microsoft Windows Help version 3.1

Windows Help version 3.1 includes the following tools and files that you need to build your Help files.

What Tools Do You

Tool	Description
Microsoft Windows Help: WINHELP.EXE	The Windows Help application is the application that users open to display the Help files you create. Help is built into the Windows operating system; therefore, it is a shared resource available to all applications running in the Windows environment.
Microsoft Windows Help Compiler: HC31.EXE	The Help compiler compiles RTF files into binary Help files (.HLP) that can then be displayed in the Windows Help application.

Microsoft Hotspot Editor:
SHED.EXE

Hotspot Editor lets you create graphics with multiple hot spots. Using Hotspot Editor, you can define hot spots that link to other Help graphics, to Help topics, or to multimedia events (if provided for by external DLLs).

Microsoft Multi-Resolution
Bitmap Compiler: **MRBC.EXE**

The Multi-Resolution Bitmap Compiler lets you create bitmaps with different resolutions and combine them into a single graphic to compensate for differences between the aspect ratio of bitmaps you create and the user's display.

Microsoft Windows Help
Authoring Templates:
WHAT30.DOT and
WHAT31.DOT

The Help Authoring Templates are word-processing templates that modify Word for Windows. You use the templates to create and edit Help topic files and save them as RTF so they can be compiled. The templates offer a simplified way to add Help features to your Help file.

Microsoft Windows Help Project
Editor: **WHPE.EXE**

The Help Project Editor is a tool that you use to create and edit Help project files (.HPJ). You can also compile Help files from within the Windows environment using the editor.

Microsoft Help Example:
HELPEX.EXE

The Help Example application is written in the C programming language and conforms to the user-interface style recommended by Microsoft for Windows-based applications. It is an example of a simple Windows-based application that uses context-sensitive Help created with the Windows Help development tools. It is a good idea to review the source code for this application if you are planning to include context-sensitive Help for your application.

Document Conventions

This guide uses the following typographical conventions.

Convention**Use**

Bold

Indicates C language-specific terms and options (for example, **WinHelp**), and options and commands specific to Windows Help (for example, the **ROOT** option).

Italic

Introduces new terms and indicates placeholders. You provide the actual value for the placeholder.

FULL CAPS

Represent filenames (for example, WINHELP.EXE) and commands specific to Windows Help (for example, HELP_CONTENTS).

small caps

Represent key names (such as enter).

key+key

Indicates that you must press the keys at the same time. For example, “Press alt+f4” means to hold down alt and press f4.

[brackets]

Enclose optional items in syntax statements. For example, [window-name] indicates a window name may be needed in the syntax statement. Type only the information within the brackets, not the brackets themselves.

{braces}

Enclose required items in syntax statements. Type only the information within the braces, not the braces themselves.

Semicolon (;)

Introduces comments in the Help project file. The Help compiler ignores these lines.

Monospace

Represents program code and Help examples.

Horizontal ellipsis (...)	Indicates that the item shown might appear more than once.
---------------------------	--

Vertical ellipsis (. . .)	Indicates that the example omits a portion of the file or program.
------------------------------------	--

In addition to this guide, there are several other documentation sources that relate to Help. They vary in the type of information they contain, but you may find one or more of them useful.

Documentation

Note

The Microsoft Windows version 3.1 Software Development Kit (SDK) includes information on Windows Help that is substantially different from the information in this guide in one respect: it is intended primarily for a programming audience. Therefore, you or the developers in your company should refer to the indicated SDK manuals for additional technical information about Windows Help.

Title	Contents
<i>Programming Tools</i> (SDK): Chapter 3, “Creating Help Files,” and Appendix B, “Help Compiler Error Messages”	Detailed information about the RTF file format and syntax used in Help topics and the Help project file. Use this information to learn the technical structure of Help or to develop your own custom application that reads and writes RTF files.
<i>Programmer’s Reference, Volume 2: Functions</i> (SDK)	Information about the <code>HELP_*</code> constants of the WinHelp function.

*Programmer's Reference,
Volume 4: Resources (SDK):
Chapter 15, "Windows Help
Statements and Macros"*

Syntax and purpose of RTF statements and macros used in Help topics and project files. Use this information to learn the technical structure of Help or to develop your own custom application that reads and writes RTF files.

*Microsoft Windows version
3.1 Guide to Programming:
Chapter 20, "Dynamic-Link
Libraries"*

Step-by-step explanation of how to create a Windows-based dynamic-link library.

*The Windows Interface: An
Application Design Guide*

Principles of user interface design. This information will help you make decisions about the Help system you are creating and will ensure that the changes you make to the standard Windows Help interface are consistent with Microsoft Windows and applications for Windows.