

Introduction

The WinHelp Reference section caters to those who want a more in-depth understanding of the technical aspects of WinHelp.

You'll find the following information in this section:

- [Sections of the help project file](#)
- [Help compiler options](#)
- [WinHelp compiler limits](#)
- [Topic footnote symbols](#)
- [WinHelp 4 bug list](#)
- [Macro reference](#)

WinHelp 4 only

This icon means this information pertains only to WinHelp 4 Help systems (those designed for Windows 95, 98, and NT).

WinHelp 3 only

This icon means this information pertains only to WinHelp 3 Help systems (those designed for Windows 3.1).

Topic footnote symbols

RoboHELP uses the following topic footnote symbols to define when a Help topic appears, how it looks, and how users access it. Each Help topic may have some or all of these symbols.

Only the Topic ID footnote is required.

<u>Footnote symbol</u>	<u>Name</u>	<u>Purpose</u>
Asterisk sign (*)	Build tag-indicator	Allows you to conditionally include topics in a particular version of the Help file by marking the topics with a build tag. The build tag is optional and not commonly used.
Pound sign (#) (Required footnote)	Topic ID	The Topic ID is the basis for hypertext, and it uniquely identifies a particular topic among all the topics in the Help system. When you create a link or topic reference, RoboHELP uses the Topic ID to identify the destination topic.
Dollar sign (\$)	Title	Used to list a topic in the WinHelp viewer when users perform a search. Titles are optional for topics: you may not want to use a title for popup topics.
Letter 'K' (K)	Index keywords (K-keywords)	Identifies words or phrases that users can enter to search for a particular topic in the Help system's index (on the Help system's Index tab). Index keywords are optional. You may not want to use them for popup topics.
Letter 'A' (A)	See Also keywords (A-keywords)	These are similar to Index keywords (K-keywords), except your users never see them and they aren't included in the Help system's index. These keywords are typically used with the ALink macro to provide a list of Related Topics. See Also keywords are optional. You may not want to use them for popup topics.
Plus sign (+)	Browse sequence code	Allows you to specify a suggested reading order, so users can browse through related topic content. Users access browse sequences through Browse buttons (Next >> and Prev <<) from the window's button bar. Browse strings are optional, and often omitted for topics that you

Exclamation point (!)	Topic entry macro	<p>only want to display in popup windows.</p> <p>Execute a macro as soon as the topic is displayed (selected by users).</p>
Commercial 'at' symbol (@)	Comment text	<p>Topic entry macros are similar to program scripts or batch files and can be useful when you want to customize a specific topic.</p> <p>Allows you to add internal comments to a topic, which can be helpful communication between Help authors.</p>
Greater than symbol (>)	Default topic window	<p>RoboHELP Explorer's Topic List even allows you to sort topics by comment.</p> <p>Specifies the default topic window used to display topics accessed from the Index, Find, or Search tabs in the Help system.</p> <p>For WinHelp 4 topics, if a default topic window is specified, it's automatically selected as the window for all topic references – jumps, macros, buttons, TOC pages, and so on. (Unless you specify a different window.)</p>

WinHelp compiler limits

The table below shows the capacities and limits for both WinHelp 3 and WinHelp 4 compilers.

<u>Parameter</u>	<u>WinHelp 3 (Windows 3.x)</u>	<u>WinHelp 4 (Windows 95, 98, NT)</u>
Help file size	2 gigabytes	2 gigabytes
Topics per topic file	32,700	No practical limit
Topics per Help file	42,000,000	No practical limit
Topics per keyword	32,767	64,000
Topic footnote length	1,023 characters	16,383 characters
Keyword length	255 characters	255 characters
Hotspot hidden text	Not documented	16,383 characters
Build tags	Not documented	16,383 per Help file
Unique build tags	30 per Project file	30 per Project file
Build tag string	32 characters	32 characters
Help title string	127 characters	127 characters
Topic title string	127 characters	127 characters
Custom Window title string	1 main 5 secondary	1 main 255 secondary
Custom Window name	N/A	8 characters
Copyright string	50 characters	255 characters
Browse string	Not documented	50 characters
Referenced bitmaps	32,767	65,535 per Help file
Filename	259 characters	259 characters
Number of fonts	Not documented	255 per Help file

Font name	20 characters	31 characters
Font ranges	Not documented	20 ranges
Error log file	Not documented	No limit (Note that HCW can display 64K of text in Windows 95 and 1MB in Windows NT.)
[MAP] section (Project file)	Not documented	65,535 entries
Citation string	Not documented	2,000 characters
Window definitions	Not documented	255 per Project file
Window caption	50 characters	50 characters
Macro length	512 characters	16K
Phrase file (.PH)	Not documented	65,383 characters
Contents file entries	N/A	No practical limit
Contents headings	N/A	9 levels (indented)
Contents topic strings	N/A	127 characters
Contents heading text	N/A	No practical limit

Sections of the help project file (HPJ)

The Help project file (HPJ) contains information that defines the behavior, look, and content of the Help system.

When you use RoboHELP to create a new Help project, the basic (or default) settings are taken care of for you. RoboHELP Explorer allows you to change these settings to suit your preference and design needs from the **Project Settings dialog**.

However if you're an advanced or experienced user, this section provides technical information about the following HPJ sections:

- [\[ALIAS\] section](#)
- [\[BAGGAGE\] section](#)
- [\[BITMAPS\] section](#)
- [\[BUILDTAGS\] section](#)
- [\[CONFIG\] section](#)
- [\[FILES\] section](#)
- [\[MAP\] section](#)
- [\[WINDOWS\] section](#)

[ALIAS] section

[ALIAS] Topic ID = Alias

The [ALIAS] section assigns one or more Topic IDs to a different topic. This section is optional.

Parameter	Description
Topic ID	The Topic ID that identifies a particular topic.
Alias	The alternative string or alias name that is used in the \footnote statement. An alias string has the same form and follows the same conventions as the Topic ID. That is, it is not case-sensitive, and may contain the alphabetic characters A through Z, the numeric characters 0 through 9, and the period and underscore characters.

Comments

Because Topic IDs must be unique, the [ALIAS] section provides a way to delete or combine Help topics without re-coding your files. For example, if you created a topic that replaces information in three other topics, you could manually search through your files for broken links (invalid references) to the deleted topics. An easier approach is to use Aliases or the [ALIAS] section to assign the name of the new topic to the deleted topics.

The [ALIAS] section can also be used when your application has multiple Map IDs for one Help topic, which can occur in context-sensitive help. Alias names can be used in a [\[MAP\] section](#), but only if the [ALIAS] section precedes the [MAP] section.

Example

The following example creates several Aliases:

[ALIAS]

sm_key=key_shrtcuts

cc_key=key_shrtcuts

st_key=key_shrtcuts ; combined into Keyboard Shortcuts topic clskey=us_dlog_bxs maakey=us_dlog_bxs ; covered in Using Dialog Boxes topic. chk_key=dlogprts drp_key=dlogprts

lst_key=dlogprts

opt_key=dlogprts

tbx_key=dlogprts ; combined into Parts of Dialog Box topic. frmtxt=edittxt wrptxt=edittxt

seltxt=edittxt ; covered in Editing Text topic.

[BAGGAGE] section

[BAGGAGE] filename

The [BAGGAGE] section lists files (typically multimedia sound and video files) that the Microsoft WinHelp Compiler stores within the Help file's internal file system. WinHelp can access data files stored in the Help file more efficiently than it can access files in the normal MS-DOS file system because it doesn't have to read the file allocation table from CD-ROM.

Parameter	Description
filename	The full path of a file. If a file cannot be found, the compiler reports an error.

Comments

You can store a maximum of 1,000 files as baggage files.

If you add a file other than a multimedia sound or video file in the [BAGGAGE] section, you must use or write a dynamic-link library that uses WinHelp to read these files from the Help file.

? [BITMAPS] section

[BITMAPS] filename

(WinHelp 3 only) The [BITMAPS] section specifies the names and locations of the bitmap files specified in the bmc, bml, and bmr statements.

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

filename	The full path of a bitmap file. If a file cannot be found, the compiler reports an error.
----------	---

Comments

For Windows 3.1, the [BITMAPS] section is not required if the bitmaps are located in the Image Folders (BMROOT) (RoboHELP Explorer's **Project tab** under the **Images and Multimedia** folder) or if the path containing the bitmaps is listed in the BMROOT or ROOT option. If the project file doesn't include either of these options, each bitmap filename must be listed in the [BITMAPS] section of the project file.

Example

The following example specifies three bitmap files:

```
[BITMAPS]
```

```
BMP01.BMP
```

```
BMP02.BMP
```

```
BMP03.BMP
```

[BUILDTAGS] section

[BUILDTAGS] tag

The [BUILDTAGS] section defines the build tags for the Help file. The Help compiler uses these tags to determine which topics to include when building the Help file. This section is used in conjunction with the build-tag \footnote statements, which associate a build tag with a given topic. If the build tag is also defined in the [BUILDTAGS] section, the Help compiler compiles the topic: otherwise, it ignores the topic.

Parameter	Description
tag	A build tag consisting of any combination of characters except spaces, as the Help compiler strips any space characters from the tag. The compiler treats uppercase and lowercase characters as the same characters (that is, it is not case-sensitive).

Comments

The [BUILDTAGS] section is optional, and can contain up to 30 build tags.

Build tags are specified as part of the BUILD option.

Example

The following example shows a [BUILDTAGS] section in a sample project file:

```
[BUILDTAGS]
DEMO      Topics to include in demo build
MASTER   Topics to include in master Help file
DEBUGBUILD Topics to include in debugging build
TESTBUILD Topics to include in a mini-build for testing
```

[CONFIG] section

[CONFIG] macro

The [CONFIG] section contains one or more macros that carry out actions, such as enabling browse buttons and registering dynamic-link library (DLL) functions. WinHelp executes the macros when it opens the Help file.

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

macro	A WinHelp macro.
-------	------------------

Comments

The [CONFIG] section may include any number of lines. Each line of the [CONFIG] section may be up to 254 characters long.

Example

The following example registers a DLL, creates a button, enables the browse buttons, and sets the name of the Help file containing information about how to use Help:

[CONFIG]

```
RegisterRoutine("bmp","HDisplayBmp","USSS")
```

```
RegisterRoutine("bmp","CopyBmp", "v=USS")
```

```
CreateButton("btn_up", "&Up", "JumpContents(`HOME.HLP')") BrowseButtons()
```

```
SetHelpOnFile("APPHELP.HLP")
```

[FILES] section

[FILES] filename

The [FILES] section lists all topic files used to build the Help file. Every project file requires a [FILES] section.

Parameter	Description
filename	The full or partial path of a topic file. If a partial path is given, the Help compiler uses the folders specified by the <u>ROOT</u> option to construct a full path. If a file cannot be found, the compiler reports an error.

Comments

You can also use the #include directive in the [FILES] section to indirectly specify the topic files by designating a file that contains a list of the topic files.

Example

The following example specifies four topic files:

```
[FILES]
rtftxt\COMMANDS.RTF
rtftxt\HOWTO.RTF
rtftxt\KEYS.RTF
rtftxt\GLOSSARY.RTF
```

The following example uses the #include directive to specify the topic files indirectly. In this case, the file RTFFILES.H must be in the project file (the Help compiler doesn't use the INCLUDE environment variable to search for files).

```
[FILES]
#include <rtffiles.h>
```


[MAP] section

[MAP] Topic ID map number

The [MAP] section associates Topic IDs (or Aliases) with Map numbers for context-sensitive help. The Map number corresponds to a value the parent application passes to WinHelp in order to display a particular topic. This section is optional.

Parameter	Description
Topic ID	The Topic ID of a topic in the Help file. The ID can be any combination of characters except spaces, and must also be specified in a Topic ID footnote statement in a Help file topic.
Map number	The Map number to associate with the Topic ID. The number can be in either decimal or standard C hexadecimal format. You can assign only one Map number to a Map string or Alias. Assigning the same number to more than one Map String generates a compiler error. At least one space must separate the Map number from the Map string.

Comments

You can define the Map strings listed in the [MAP] section in a Help topic or in the [\[ALIAS\]](#) section. The compiler generates a warning message if a Map string appearing in the [MAP] section is not defined in any of the topic files or in the [ALIAS] section.

If you use an Alias, the [ALIAS] section must precede the [MAP] section in the Help project file. The [MAP] section supports two additional statements for specifying map strings and their associated map numbers. The first statement has the following form:

```
#define Topic ID Map number
```

The Topic ID and Map number parameters are as described in the Parameter/Description section above. The second statement has the following form:

```
#include "filename"
```

The filename parameter, which can be enclosed in double quotation marks or angle brackets(<>), specifies the name of a file containing one or more #define statements. The file may contain additional #include statements as well, but files may not be nested in this way more than five deep.

Example

The following example assigns hexadecimal map numbers to the Map strings:



```
[MAP]
```

```
Edit_Window      0x0001
Control_Menu     0x0002
Maximize_Icon    0x0003
Minimize_Icon    0x0004
Split_Bar        0x0005
Scroll_Bar       0x0006
Title_Bar        0x0007
Window_Border    0x0008
```

OPTIONS section

[OPTIONS] option

The [OPTIONS] section includes options that control how a Help file is built and the types of feedback displayed by the compile process. If this section is included in the project file, it should be the first section listed so that the options apply during the entire compile process. The available options are:

Option	Description
<u>BMROOT</u>	The folder containing the bitmap files named in the bmc, bml, and bmr statements in topic files.
<u>BUILD</u>	Topics to include in the build (compile of the Help file).
<u>CITATION</u>	A statement appended to topic text copied from a WinHelp system.
<u>COMPRESS</u>	The type of compression to use during the build (compile of the Help file).
<u>CONTENTS</u>	The Topic ID of the Default Topic (Contents topic) for a Help file.
<u>COPYRIGHT</u>	Adds a unique copyright message for the Help file to the About dialog .
<u>ERRORLOG</u>	Places compilation errors in a file during the build. This option is new for Windows 3.1.
<u>ICON</u>	 The icon file to be displayed when the Help file is minimized.
<u>LANGUAGE</u>	Specifies a different sorting order for Help files authored in a Scandinavian language.
<u>MAPFONTSIZE</u>	Maps a font size in the topic file to a different font size in the compiled Help file.
<u>MULTIKEY</u>	An alternative keyword table to use for mapping topics.
<u>OLDKEYPHRASE</u>	Specifies whether the compiler should use the existing key-phrase table or create a new one during the build.
<u>OPTCDROM</u>	Optimizes the Help file for CD-ROM use.
<u>REPORT</u>	Controls the display of messages during the build process.
<u>ROOT</u>	 The folders containing the topic and data files listed in the project file.
<u>TITLE</u>	The text displayed in the title bar of the Help window when the file is open.
<u>TMPDIR</u>	Specifies the folder where temporary files that are created while compiling the Help file should be stored.
<u>WARNING</u>	The level of error-message reporting the compiler is to display during the build.

Comments

These options can appear in any order within the [OPTIONS] section. The [OPTIONS] section is not required.

[WINDOWS] section

[WINDOWS]

type = "caption", (x, y, width, height), sizing, (clientRGB), (nonscrollRGB), (fTop)

The [WINDOWS] section defines the size, location, and colors for the primary Help window and any secondary-window types used in a Help file. The secondary windows defined in this section are intended to be used with Windows applications that specify secondary windows when calling the WinHelp function.

Parameter	Description						
type	The type of window that uses the defined attributes. For the primary Help window, this parameter is main. For a secondary window, this parameter may be any unique name of up to 8 characters. Any jumps that display a topic in a secondary window give this type name as part of the jump.						
caption	The title for a secondary window. WinHelp places the title in the title bar of the window. To set the title for the primary Help window, use the TITLE option in the [OPTIONS] section.						
x	The x-coordinate, in Help units, of the window's upper-left corner. WinHelp always assumes the screen is 1024 Help units wide, regardless of resolution. For example, if the x-coordinate is 512, the left edge of the Help window is in the middle of the screen.						
y	The y-coordinate, in Help units, of the window's upper-left corner. WinHelp always assumes the screen is 1024 Help units high, regardless of resolution. For example, if the x-coordinate is 512, the top edge of the Help window is in the middle of the screen.						
width	The default width, in Help units, for a secondary window.						
height	The default height, in Help units, for a secondary window.						
sizing	The relative size of a secondary window when WinHelp first opens the window. This parameter can be one of the following values: <table><tr><th>Value</th><th>Meaning</th></tr><tr><td>0</td><td>Set the window to the size specified by the x, y, width, and height parameters.</td></tr><tr><td>1</td><td>Maximize the window and ignore the x, y, width, and height parameters.</td></tr></table>	Value	Meaning	0	Set the window to the size specified by the x, y, width, and height parameters.	1	Maximize the window and ignore the x, y, width, and height parameters.
Value	Meaning						
0	Set the window to the size specified by the x, y, width, and height parameters.						
1	Maximize the window and ignore the x, y, width, and height parameters.						
clientRGB	The background color of the window. This parameter is an RGB color value consisting of three 8-bit hexadecimal numbers enclosed in parentheses and separated by commas. If this parameter is not given, WinHelp uses the default window color specified by Control Panel.						
nonscrollRGB	The background color of the non-scrolling region (if any) in the Help window. This parameter is an RGB color value consisting of three 8-bit hexadecimal numbers enclosed in parentheses and separated by commas. If this parameter is not given, WinHelp uses the default window color specified by Control Panel.						
fTop	Specifies whether the secondary window is displayed on top of all other windows. When this parameter is 1, the window is displayed over all windows that do not also use this attribute. Otherwise, it should be zero. This parameter is optional.						

Example

The following example defines two windows, the main window and a secondary window named "picture." The main-window definition sets the background color of non-scrolling regions in the main Help window to (128, 0, 128), but leaves several other values empty (for which WinHelp will supply its own default values). The secondary-window definition sets the caption to "Samples" and sets the width and height of the window to about one-quarter of the width and height of the screen. The background colors for the window and non-scrolling region are (0, 255, 255) and (255, 0, 0), respectively. The sizing parameter for the main and secondary windows is zero.

[WINDOWS]

main=, (, ,), 0, (, ,), (128, 0, 128) picture = "Samples", (123,123,256,256), 0, (0,255,255), (255,0,0)

About compiler options

RoboHELP Explorer helps you organize and manage the HPJ and other source files quickly, easily, and visually. (Unless you're an advanced or experienced user, you probably don't need to work with these sections and options. Let RoboHELP take care of the complexities for you through the **Project Settings dialog**.)

However, if you're an advanced or experienced user looking for more technical information about the available compile options (the [OPTIONS] section of the HPJ), the following topics describe them:

Options

- [BMROOT option](#)
- [BUILD option](#)
- [CITATION option](#)
- [COMPRESS option](#)
- [CONTENTS option](#)
- [COPYRIGHT option](#)
- [ERRORLOG option](#)
- [ICON option](#)
- [LANGUAGE option](#)
- [MAPFONTSIZE option](#)
- [MULTIKEY option](#)
- [OLDKEYPHRASE option](#)
- [OPTCDROM option](#)
- [OPTIONS Section](#)
- [REPORT option](#)
- [ROOT option](#)
- [TITLE option](#)
- [TMPDIR option](#)
- [WARNING option](#)

Note: Some options apply only to WinHelp 3; others only to WinHelp 4. These are clearly marked with the appropriate icon.

BMROOT option

BMROOT = path[, path]

The BMROOT option indicates the folder containing the bitmap files specified in the bmc, bml, and bmr statements.

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

path	Specifies a drive and full path.
------	----------------------------------

Comments

If the project file has a BMROOT specified under **Image Folders (BMROOT)** in RoboHELP Explorer's **Project tab**, you don't need to list the bitmap files in the [BITMAPS] section. If the project file doesn't have a BMROOT, the Help compiler looks for bitmaps in the folders specified by the ROOT option. If the project file doesn't have a ROOT option or if the ROOT option doesn't specify the folder containing the bitmap files, the filename for each bitmap must be specified in the [BITMAPS] section.

Example

The following example specifies that bitmaps are in the \HELP\BMP folder on drive C:, and in the \GRAPHICS\ART folder on drive D:

[OPTIONS]

BMROOT=C:\HELP\BMP, D:\GRAPHICS\ART

BUILD option

BUILD = expression

The BUILD option specifies which topics containing build tags are included in a build. The BUILD option only applies to topics containing a build-tag \footnote statement. Topics without build tags are always compiled, regardless of the current build expression.

Parameter	Description								
expression	The build expression. This parameter consists of a combination of build tags (specified in the [BUILDTAGS] section) and the following operators: <table><tr><th>Operator</th><th>Description</th></tr><tr><td>~</td><td>Applies the NOT operator to a single tag. The Help compiler compiles a topic only if the tag is not present. This operator has the highest precedence; the compiler applies it before any other operator.</td></tr><tr><td>&</td><td>Combines two tags by using the AND operator. The Help compiler compiles a topic only if it contains both tags. The compiler applies this operator only after the ~ operator has been applied.</td></tr><tr><td> </td><td>Combines two tags by using the OR operator. The Help compiler compiles a topic if it has at least one tag. This operator has the lowest precedence; the compiler applies it only after all other operators have been applied. You can use parentheses to override operator precedence. Expressions enclosed in parentheses are always evaluated first.</td></tr></table>	Operator	Description	~	Applies the NOT operator to a single tag. The Help compiler compiles a topic only if the tag is not present. This operator has the highest precedence; the compiler applies it before any other operator.	&	Combines two tags by using the AND operator. The Help compiler compiles a topic only if it contains both tags. The compiler applies this operator only after the ~ operator has been applied.		Combines two tags by using the OR operator. The Help compiler compiles a topic if it has at least one tag. This operator has the lowest precedence; the compiler applies it only after all other operators have been applied. You can use parentheses to override operator precedence. Expressions enclosed in parentheses are always evaluated first.
Operator	Description								
~	Applies the NOT operator to a single tag. The Help compiler compiles a topic only if the tag is not present. This operator has the highest precedence; the compiler applies it before any other operator.								
&	Combines two tags by using the AND operator. The Help compiler compiles a topic only if it contains both tags. The compiler applies this operator only after the ~ operator has been applied.								
	Combines two tags by using the OR operator. The Help compiler compiles a topic if it has at least one tag. This operator has the lowest precedence; the compiler applies it only after all other operators have been applied. You can use parentheses to override operator precedence. Expressions enclosed in parentheses are always evaluated first.								

Comments

Only one BUILD option can be given per project file.

The Help compiler evaluates all build expressions from left to right using the specified precedence rules.

Example

The following examples assume that the [\[BUILDTAGS\]](#) section in the project file defines the build tags DEMO, MASTER, and TEST_BUILD. Although the following examples show several BUILD options on consecutive lines, only one BUILD option per project file is allowed.

Option	Description
BUILD = DEMO	Compile topics that have the DEMO tag.
BUILD = DEMO & MASTER	Compile topics with both DEMO and MASTER tags.
BUILD = DEMO MASTER	Compile topics with either DEMO or MASTER tags.
BUILD = ~DEMO	Compile topics that do not have DEMO tags.
BUILD = (DEMO MASTER) & TEST_BUILD	Compile topics that have TEST_BUILD and either DEMO or MASTER

CITATION option

CITATION = citation

The CITATION option appends a custom citation notice to the end of text copied from the Help system.

Parameter	Description
citation	Any combination of 35-75 characters.

COMPRESS option

COMPRESS = compression-level

The COMPRESS option specifies the level of compression to be used when building the Help file. Compression levels indicate either no compression, medium compression (approximately 40%), or high compression (approximately 50%).

(You can also select a type of compression on the **Project Settings: Compile tab**.)



WinHelp 3 Compression Options

Parameter	Description	
compression-level	The level of compression. This parameter can be one of the following values:	
	Value	Meaning
	0	No compression
	1	High compression
	FALSE	No compression
	HIGH	High compression
	MEDIUM	Medium compression
	NO	No compression
	TRUE	High compression
	YES	High compression



Win 95 WinHelp 4 Compression Options

Parameter	Description	
compression-level	The level of compression. This parameter can be one of the following values:	
	Value	Meaning
	0	No compression
	12	Maximum (highest) compression (Zeck and Hall)
	2	Phrase - Medium compression (for smaller Help files)
	4	Hall - Medium compression (for larger Help files)
	8	Zeck - Minimum compression
	10	Zeck and Phrase - Medium to high compression (for smaller Help files)
	12	Zeck and Hall - Maximum compression (for larger Help files)

Comments

Depending on the degree of compression requested, the build uses block compression or a combination of block and key-phrase compression. Block compression compresses the topic data into predefined units known as blocks. Key-phrase compression combines repeated phrases found within the source file(s). The compiler creates a phrase-table file with the .PH extension if one doesn't already exist. If the compiler finds a file with the .PH extension, it uses that file for the current compilation. Because the .PH file speeds up the compression process when little text has changed since the last compilation, you might want to keep the phrase file if you compile the same Help file several times with compression. However, you will get maximum compression if you delete the .PH file before starting each build.

CONTENTS option

CONTENTS = Topic ID

The CONTENTS option identifies the Topic ID of the highest-level or Contents topic. This topic is usually a table of contents or index within the Help file. WinHelp displays the Contents topic whenever the user clicks the Contents button.

Parameter	Description
Topic ID	The Topic ID of a topic in the Help file. The string can be any combination of characters except spaces, and must also be specified in a Topic ID \footnote statement in a topic in the Help file.

Comments

If the [OPTIONS] section doesn't include a CONTENTS option, the compiler assumes that the Contents topic is the first topic encountered in the first topic file in the [FILES] section of the project file.

The CONTENTS option is equivalent to the INDEX option that was available in Windows version 3.0.

Example

The following example sets the topic containing the Topic ID "main_contents" as the Contents topic:

```
CONTENTS=main_contents
```

COPYRIGHT option

COPYRIGHT = copyright-notice

The COPYRIGHT option places a custom copyright notice in the **About dialog** of WinHelp. WinHelp displays the notice immediately below the Microsoft copyright notice.

Parameter	Description
copyright-notice	Any combination of 35-75 characters.

ERRORLOG option

ERRORLOG = error-filename

The ERRORLOG option directs the Help compiler to write all error messages to the specified file. The compiler also displays the error messages on the screen.

Parameter	Description
error-filename	The name of the file to receive the error messages. This parameter can be a full or partial path if the error file is written to a folder other than the project root folder.

Example

The following example writes all errors during the build to the HLPBUGS.TXT file in the Help project root folder.
ERRORLOG=HLPBUGS.TXT

ICON option

ICON = icon-file

The ICON option identifies the icon file to display when the user minimizes WinHelp.

Parameter	Description
icon-file	Specifies the name of the icon file. This file must have the standard Windows icon-file format.

Note:

This option is not supported in WinHelp 4.

LANGUAGE option

LANGUAGE = language-name

The LANGUAGE option sets the sorting order for keywords in the Search dialog box. In Microsoft WinHelp version 3.1, the only supported options are English and Scandinavian. The default is English.

MAPFONTSIZE option

MAPFONTSIZE = m:p

The MAPFONTSIZE option maps font sizes specified in topic files to different sizes when they are displayed in the Help window. This option is especially useful if there is a significant size difference between the authoring display and the intended user display.

Parameter	Description
m	Specifies the size of the source font, which can be a single point size or a range of point sizes. A range of point sizes consists of the lowest and highest point sizes separated by a hyphen (-). If you specify a range, WinHelp changes all fonts in the range to the size specified by the p parameter.
p	Specifies the size of the desired font for the Help file.

Comments

Although the [OPTIONS] section can contain up to five font ranges, only one font size or range is allowed with each MAPFONTSIZE statement. If you include more than one MAPFONTSIZE statement, the source font size or range specified in subsequent statements cannot overlap previous mappings.

Example

The following examples illustrate how to use the MAPFONTSIZE option:

MAPFONTSIZE=8:12 ; display all 8-pt. fonts as 12-pt.

MAPFONTSIZE=12-24:16 ; display fonts from 12 to 24 pts. as 16 pts.

MULTIKEY option

MULTIKEY = footnote-character

The MULTIKEY option specifies the footnote character to use for an alternative keyword table. This option is intended to be used with topic files that contain \footnotestatements for alternative keywords.

Parameter	Description
footnote-character	Specifies the case-sensitive letter to be used for the keyword footnote.

Comments

Because keyword footnotes are case-sensitive, you should limit your keyword-table footnotes to one case, usually uppercase. If an uppercase letter is specified, the compiler will not include footnotes with the lowercase form of the same letter in the keyword table.

You can use any alphanumeric character for a keyword table except K and k, which are reserved for Help's standard keyword table. There is an absolute limit of five keyword tables, which includes the standard table. However, depending upon system configuration and the structure of your Help system, a practical limit of only two or three tables may be more realistic. If the compiler cannot create an additional keyword table, the additional table is ignored in the build.

Example

The following example illustrates how to enable the letter L for a keyword-table footnote:

MULTIKEY=L

OLDKEYPHRASE option

OLDKEYPHRASE = value

The OLDKEYPHRASE option specifies whether an existing key-phrase file should be used when compressing a Help file with phrase compression.

Parameter	Description
value	If this value is YES, the existing phrase (.ph) file is used for phrase compression. If this value is NO, a new .ph file is created during the build. The default value is NO. This parameter can be one of the following values:

Note:

If you specify phrase compression, a phrase-table file is created that has a .ph filename extension. If an existing phrase file is found, the file is deleted, and then a new one is created — unless OLDKEYPHRASE=YES is specified in your project file. If it is, your existing phrase file is used. Using an existing phrase file reduces compile time, but compression is less effective.

OPTCDROM option

OPTCDROM = yesvalue

The OPTCDROM option optimizes a Help file for display on CD-ROM by aligning topic files on predefined block boundaries.

Parameter	Description
yesvalue	One of the following values: <u>Value</u> YES TRUE 1 ON

REPORT option

REPORT = ON

The REPORT option displays messages on the screen during compile. These messages indicate when the WinHelp compiler is performing the different phases of the build, including compiling the file, resolving jumps, and verifying browse sequences.

? ROOT option

ROOT = pathname[, pathname]

(Not used in RoboHELP projects – replaced with the [BMROOT option](#).) The ROOT option specifies the folders where the Help compiler looks for files listed in the project file.

Parameter	Description
pathname	Specifies either a drive and full path, or a relative path from the project folder. If the project file has a ROOT option, all relative paths in the project file refer to one of these paths. If the project file doesn't have a ROOT option, all paths are relative to the folder containing the project file.

Comments

If the project file doesn't have a BMROOT option, the compiler looks in the folders specified in the ROOT option to find bitmaps in the bmc, bml, and bmr statements. If none of these folders contains these bitmaps, the bitmap filenames must be listed in the [BITMAPS] section of the project file.

Example

The following example specifies that the project root folder is C:\WINHELP\HELPPDIR and is found on drive C:

[OPTIONS]

ROOT=C:\WINHELP\HELPPDIR

Given this root folder, if the [FILES] section contains the entry TOPICS\FILE.RTF, the full path for the topic file is C:\WINHELP\HELPPDIR\TOPICS\FILE.RTF.

TITLE option

TITLE = titlename

The TITLE option sets the title for the Help file. WinHelp displays the title in its title bar whenever it displays the Help file.

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

titlename	The title displayed in the WinHelp title bar. The title must not exceed 50 characters.
-----------	--

Comments

If you don't specify a title using the TITLE option, WinHelp displays the title "Windows Help" in the title bar.

Example

The following example sets the Help file title to ABC Help.

[OPTIONS]

TITLE=ABC Help

WARNING option

WARNING = level

The WARNING option specifies the amount of debugging information the Help compiler is to report (REPORT option).

Parameter	Description								
level	Specifies the warning level, which can be one of the following values:								
	<table><tr><th>Value</th><th>Meaning</th></tr><tr><td>1</td><td>Report only the most severe errors.</td></tr><tr><td>2</td><td>Report an intermediate number of errors.</td></tr><tr><td>3</td><td>Report all errors and warnings.</td></tr></table>	Value	Meaning	1	Report only the most severe errors.	2	Report an intermediate number of errors.	3	Report all errors and warnings.
Value	Meaning								
1	Report only the most severe errors.								
2	Report an intermediate number of errors.								
3	Report all errors and warnings.								

Example

The following example specifies an intermediate level of error reporting:

[OPTIONS]

WARNING=2

Note:

This option is not supported in WinHelp 4.

TMPDIR option

You specify the TMPDIR option in the [OPTIONS] section of the project (.hpi) file. TMPDIR specifies the folder where temporary files that are created while compiling the Help file should be stored. Temporary files are usually not created unless your Help file exceeds 8 MB in size.

To specify the directory for storing temporary files:

1. From the Explorer View **File** menu, select **Project Settings**.
2. Click the **Advanced tab**.
3. Enter or browse for a directory in **Temp Directory**.
4. Click **OK**.
5. The temporary directory appears in the [OPTIONS] section of the help project's .HPJ file.

Note:

If the TMPDIR option is not set, the folder specified by the TEMP environment variable is used.



WinHelp 4 bug list

The following is a list of possible bugs related to WinHelp 4 (the Microsoft Help compiler for Windows 95/Windows NT) that have been reported by members of the CompuServe WinHelp section and the Internet WINHLP list. You may encounter some of these bugs while working in RoboHELP, but they are related to the Microsoft WinHelp compiler or the Microsoft WinHelp viewer, not to RoboHELP.

This list is not in any way to be considered an official bug list.

[ALink and KLink bugs](#)

[Bitmap bugs](#)

[Bulleted and numbered list bugs](#)

[CNT file bugs](#)

[Context-sensitive help bugs](#)

[Formatting bugs](#)

[Jump hotspot bugs](#)

[Macro bugs](#)

[Miscellaneous crash bugs](#)

[Other bugs](#)

[Training Card \(T-CARD\) bugs](#)

[Window bugs](#)



ALink and KLink bugs

- ALink and KLink macros that use a default topic must also have a Window parameter specified.
- An ALink jump to a secondary window in a different HLP file displays a blank main window as well as the proper secondary window.
- A KLink macro that only matches one topic jumps directly to that topic: it doesn't display the dialog box as described by the documentation. The ALink macro displays the dialog box properly.
- The KLink macro ignores any keywords after the first one unless the CNT file includes at least one additional HLP file (even if that file does not exist). The link works properly.
- KLink or ALink macros that list multiple keywords show topics more than once in the Results List if those topics contain more than one keyword. Each topic appears once for each keyword with a match.



Bitmap bugs

- 256-color bitmaps print in black and white (no grays) under Win32s 1.30a, Win 3.11, and HP LaserJet 4L.
- Transparent bitmaps pick up the background color of the scrolling region, even if they are located in the non-scrolling region.
- If a user running in 256-color mode switches the focus away from a Help file and then returns, 256-color WMFs will not have the correct colors. Note that 256-color BMPs will be redrawn properly.



Bulleted and numbered list bugs

- WinHelp doesn't use the default indent values used by Word in numbered lists. You must define a tab stop in Word to get it to display properly in WinHelp. (Unlike WinHelp 3.1, you can use the same formatting in all paragraphs.)
- The first (or sometimes last) bulleted list item shows the wrong sized bullet.
- Numbered lists frequently show incorrect numbers (such as 1, 1, 2, 2).
- Help files containing bullets formatted with Wingding (or other non-symbol) fonts and compressed with Hall compression display the wrong bullet character. Zeck compression does not alter the bullets, and Symbol bullets are not affected.
- Multi-level bulleted or numbered lists display incorrectly in ActiveTest. These types of lists only display correctly after the Help system has been compiled; this is a Microsoft Word limitation.



CNT file bugs

- If two HLP files that use a combined index through the CNT file have some identical keywords and corresponding topic titles, WinHelp appends the HLP file name to all but the first duplicated title in the results list.
- The CNT file option Nofind, used to disable the Find tab, does not work and is rejected by the Help Compiler.
- If one CNT file is included in another CNT file, WinHelp ignores the default HLP filename and default window name specifications of the "included" file.



Context-sensitive help bugs

If a topic displayed as context-sensitive help does not exist, two error message boxes are displayed if Help Author mode is on.



Formatting bugs

- Text with dotted underlining appears with a single underline.
- Double-underlined text that isn't part of a hotspot appears without underlining.
- Text beginning with or immediately following a typographical apostrophe in the non-scrolling region is raised or lowered as the font size is increased or reduced using the WinHelp Options menu.



Jump hotspot bugs

- Jumps to context strings or keywords defined in the middle of a topic only work in windows with a fixed size.
- Jumps to keywords defined in the middle of a topic always display in the default window defined in the CNT Base statement, not the window designated by the topic's > footnote.



WinHelp macro bugs

- A button in a secondary window that invokes the Exit() macro causes the buttons to be erased, and the window to remain open.
- A KLink macro that only matches one topic jumps directly to that topic; it doesn't display the dialog box as described by the documentation. The ALink macro displays the dialog box properly.
- The Escape key can no longer be used with AddAccelerator, even though it is listed in the usable keycodes. WinHelp now uses Escape to close the window.
- The ExtInsertItem macro doesn't accept the string versions of the display state field. Only numeric values compile successfully.
- HCRTF crashes when compiling an RTF containing an IfThenElse macro containing macros enclosed in quotes or apostrophes. Removing the quotes fixes the problem.
- A SetPopupColor macro chained after a PopupID macro displays the runtime error message "1031 A ")" is missing."
- The Compare macro doesn't display windows side by side.



Miscellaneous crash bugs

- Compiling certain projects with both FTS and compression causes the WinHelp compiler to crash. Turning off either compression or FTS allows the compile to complete.
- Compiling the Visual C++ file AFXCORE.RTF after deleting the "Record menu" line from the first page causes a general protection fault in HCRTF.
- You can crash HCW by defining an Include file in the MAP section, then deleting it without first saving the HPJ file.
- Specifying "bmct emdash.bmp" with compression causes the WinHelp compiler to crash. You can avoid the crash by removing the compression or the transparency option.
- HCRTF crashes when compiling an RTF containing an IfThenElse macro with macros enclosed in quotation marks or apostrophes. Removing the quotation marks fixes the problem.
- WinHelp may crash if you select a keyword defined in the beginning of a topic, and then select a keyword defined in the middle of the same topic.



Other WinHelp 4 bugs

- The external ShellExecute API cannot be used in Help files compiled under WinHelp 4, as it conflicts with the new internal ShellExecute command. ShellExecute works properly under WinHelp 4 if it is compiled with the Help 3.1 Help Compiler.
- Internal ShellExecute commands that use the fourth, fifth, or sixth parameters (operation, path, or topic ID) do not work. Only the first three parameters can be used.
- Commands in the main [CONFIG] section aren't executed until you display a topic through the WinHelp API with HELP_CONTEXT.
- Topics displayed with HELP_CONTEXTPOPUP cannot include jumps to other topics.
- If you define an Include file in the [CONFIG] section, then edit that entry, the resulting line is "#include filename."



Training Card (T-CARD) bugs

- If a user opens another WinHelp window from a T-CARD window then closes the second window, , a WMCLOSE message is also sent to the application to close the T-CARD.
- When a T-CARD instance of WinHelp is started, a hidden window is created that may not be closed when WinHelp is closed. This prevents WinHelp from reopening. You can find the hidden window with `FindWindow("MS_TCARDHELP",0)`.



Window bugs

- Autosizing for small secondary windows doesn't always provide enough height to display the entire topic.
- An ALink jump to a secondary window in a different HLP file displays a blank main window and the proper secondary window.
- If a window is positioned over another window such that a button on the top window is directly over a button on the lower window, clicking on the visible button may also execute the button in the lower window.

Macro reference list

Use this reference list to locate a specific WinHelp macro by category:

- [Button WinHelp macros](#)
- [HTML WinHelp macros](#)
- [Keyboard WinHelp macros](#)
- [Linking WinHelp macros](#)
- [Menu WinHelp macros](#)
- [Program WinHelp macros](#)
- [RoboHELP Extension WinHelp macros](#)
- [Text-Marker WinHelp macros](#)
- [Window WinHelp macros](#)

Button WinHelp macros

[Back macro](#)

[BackFlush macro](#)

[BrowseButtons macro](#)

[ChangeButtonBinding macro](#)

[ChangeEnable macro](#)

[Contents macro](#)

[CreateButton macro](#)

[DestroyButton macro](#)

[DisableButton macro](#)

[EnableButton macro](#)

[Find macro](#)

[Finder macro](#)

[Menu macro](#)

[Next macro](#)

[Prev macro](#)

[Search macro](#)

HTML WinHelp macros

[Inet macro](#)

[JumpHtml macro](#)

Keyboard WinHelp macros

[AddAccelerator macro](#)

[RemoveAccelerator macro](#)

Linking WinHelp macros

[ALink macro](#)

[Inet macro](#)

[JumpContents macro](#)

[JumpContext macro](#)
[JumpHash macro](#)
[JumpHelpOn macro](#)
[JumpHtml macro](#)
[JumpId macro](#)
[JumpKeyword macro](#)
[KLink macro](#)
[PopupContext macro](#)
[PopupHash macro](#)
[PopuId macro](#)
[UpdateWindow macro](#)

Menu WinHelp macros

[AppendItem macro](#)
[ChangeItemBinding macro](#)
[CheckItem macro](#)
[DeleteItem macro](#)
[DisableItem macro](#)
[EnableItem macro](#)
[ExtAbleItem macro](#)
[ExtInsertItem macro](#)
[ExtInsertMenu macro](#)
[InsertItem macro](#)
[InsertMenu macro](#)
[ResetMenu macro](#)
[UncheckItem macro](#)

Program WinHelp macros

[ControlPanel macro](#)
[ExecFile macro](#)
[ExecProgram macro](#)
[FileExist macro](#)
[ShellExecute macro](#)
[ShortCut macro](#)

RoboHELP Extensions WinHelp macros

[Inet macro](#)
[INETWH_Initialize macro](#)
[JumpHtml macro](#)
[RegisterRoutine WinHelp macro](#)
[RoboHelpExInitialize WinHelp macro](#)
[RoboHelpExShowSeeAlso WinHelp macro](#)
[RoboHelpExShowNavPane WinHelp macro](#)
[RoboHelpExWatermark WinHelp macro](#)

[RoboHelpExWatermarkNonScroll](#)

[Sound RHMMPLAY macro](#)

[SoundOnly RHMMPLAY macro](#)

[Video RHMMPLAY macro](#)

[VideoCaption RHMMPLAY macro](#)

[VideoMenu RHMMPLAY macro](#)

[VideoPath RHMMPLAY macro](#)

Text-Marker WinHelp macros

[DeleteMark macro](#)

[GoToMark macro](#)

[IfThen macro](#)

[IfThenElse macro](#)

[IsMark macro](#)

[IsNotMark macro](#)

[Not macro](#)

[SaveMark macro](#)

Window WinHelp macros

[CloseSecondarys macro](#)

[CloseWindow macro](#)

[FocusWindow macro](#)

[HelpOnTop macro](#)

[PositionWindow macro](#)

[SetPopupColor macro](#)

About WinHelp macro

About()

The About macro displays the **About dialog** for the WinHelp Viewer.

Parameters

None.

Comments

Using this macro in secondary windows isn't recommended.

AddAccelerator WinHelp macro

AddAccelerator(key, shift-state, "macro")

The AddAccelerator macro assigns a help macro to an accelerator key (or key combination) that, when pressed, activate(s) the macro. (An accelerator key is a keystroke that provides a keyboard equivalent to commands displayed on menus or a button bar. For example, in Word pressing **Alt+F** activates the **File** menu.)

Parameter	Description
key	The Windows virtual-key value. This parameter can be a numeric digit, a quoted character (as in 'A'), or one of the following strings:
	VK_LBUTTON VK_NUMPAD2
	VK_RBUTTON VK_NUMPAD3
	VK_CANCEL VK_NUMPAD4
	VK_MBUTTON VK_NUMPAD5
	VK_BACK VK_NUMPAD6
	VK_TAB VK_NUMPAD7
	VK_CLEAR VK_NUMPAD8
	VK_RETURN VK_NUMPAD9
	VK_SHIFT VK_MULTIPLY
	VK_CONTROL VK_ADD
	VK_MENU VK_SEPARATOR
	VK_PAUSE VK_SUBTRACT
	VK_CAPITAL VK_DECIMAL
	VK_KANA VK_DIVIDE
	VK_KANJI VK_F1
	VK_HANGEUL VK_F3
	VK_JUNJA VK_F5
	VK_HANJA VK_F6
	VK_ESCAPE VK_F7
	VK_SPACE VK_F8
	VK_PRIOR VK_F9
	VK_NEXT VK_F10
	VK_END VK_F11
	VK_HOME VK_F12
	VK_LEFT VK_F13
	VK_UP VK_F14
	VK_RIGHT VK_F15
	VK_DOWN VK_F16
	VK_SELECT VK_F17
	VK_PRINT VK_F18
	VK_EXECUTE VK_F19
	VK_SNAPSHOT VK_F20
	VK_INSERT VK_F21
	VK_DELETE VK_F22
	VK_HELP VK_F23
	VK_SCROLL VK_F24
	VK_NUMPAD0 VK_NUMLOCK

	VK_NUMPAD1																		
shift-state	The combination of ALT, SHIFT, and CTRL keys to be used with the accelerator. This parameter may be one of the following values:																		
	<table> <tr> <th>Value</th><th>Meaning</th></tr> <tr> <td>0</td><td>None</td></tr> <tr> <td>1</td><td>Shift</td></tr> <tr> <td>2</td><td>Ctrl</td></tr> <tr> <td>3</td><td>Shift+Ctrl</td></tr> <tr> <td>4</td><td>Alt</td></tr> <tr> <td>5</td><td>Alt+Shift</td></tr> <tr> <td>6</td><td>Alt+Ctrl</td></tr> <tr> <td>7</td><td>Alt+Shift+Ctrl</td></tr> </table>	Value	Meaning	0	None	1	Shift	2	Ctrl	3	Shift+Ctrl	4	Alt	5	Alt+Shift	6	Alt+Ctrl	7	Alt+Shift+Ctrl
Value	Meaning																		
0	None																		
1	Shift																		
2	Ctrl																		
3	Shift+Ctrl																		
4	Alt																		
5	Alt+Shift																		
6	Alt+Ctrl																		
7	Alt+Shift+Ctrl																		
macro	The help macro or macro string executed when the user presses the accelerator key(s). The macro must appear in quotation marks. Colons or semicolons must separate multiple macros in a string.																		

Comments

The AddAccelerator macro can be abbreviated as AA.

Example

The following macro executes the Windows Clock program when the user presses **Alt+Shift+Ctrl+F4**:

```
AddAccelerator(0x73, 7, "ExecFile('clock.exe', 1)")
```



ALink WinHelp macro

ALink(keyword[;keyword][,type[,topic-ID[,window-name]])

The ALink macro searches for topics with See Also (A-keywords) that match those specified in the macro. This macro is commonly used to display related topics from a button or hotspot.

Parameter	Description								
keyword	One or more See Also keywords (A-keywords) to search for. Separate multiple keywords with a semicolon. If any keyword contains a comma, the entire keyword string must be enclosed in quotation marks.								
type	<p>The action to perform if one or more keywords are found. If you do not specify a parameter or enter zero, WinHelp displays the Topics Found dialog containing the topic title. This parameter may be one or more of the following values, separated by spaces:</p> <table><tr><th>Value</th><th>Meaning</th></tr><tr><td>JUMP (1)</td><td>Jumps directly to the topic if only one topic is found to match the keyword.</td></tr><tr><td>TITLE (2)</td><td>Displays the title of the help file (as specified in the CNT file) for the topic title of each keyword found if a keyword is found in more than one help file.</td></tr><tr><td>TEST (4)</td><td>Specifies that the macro should return a value indicating whether or not there is at least one match.</td></tr></table>	Value	Meaning	JUMP (1)	Jumps directly to the topic if only one topic is found to match the keyword.	TITLE (2)	Displays the title of the help file (as specified in the CNT file) for the topic title of each keyword found if a keyword is found in more than one help file.	TEST (4)	Specifies that the macro should return a value indicating whether or not there is at least one match.
Value	Meaning								
JUMP (1)	Jumps directly to the topic if only one topic is found to match the keyword.								
TITLE (2)	Displays the title of the help file (as specified in the CNT file) for the topic title of each keyword found if a keyword is found in more than one help file.								
TEST (4)	Specifies that the macro should return a value indicating whether or not there is at least one match.								
topic-ID	Displays the topic in a popup window if no matches are found. If you don't specify this parameter, WinHelp displays a message indicating no additional information is available. To specify a topic in a different help file, the topic ID needs to end with the @ character followed by the name of the help file. (For example: <i>topicID @helpfilename.hlp</i> .)								
window-name	Specifies the window to display the destination topic. If you don't specify this parameter, WinHelp uses the default topic window (if one is defined) or the current window. Note that if this macro results in an inter-file jump, the window must be defined in the Project (HPJ) file for the help file containing the topic.								

Comments

The ALink macro is identical to the KLink macro, except that it searches for See Also keywords (A-keywords) instead of index keywords (K-keywords). The A topic footnote represents See Also keywords.

Annotate WinHelp macro

Annotate()

The Annotate macro displays the **Annotation dialog** from WinHelp's **Edit** menu. Provides a space for users to type any additional information they want associated with this topic. When a topic is annotated, a paper-clip icon appears at the top of the topic. Clicking the paper clip displays the **Annotation dialog** and the annotation text typed.

Parameters

None.

Comments

Don't run this macro from a popup window.

AppendItem WinHelp macro

AppendItem("menu-id", "item-id", "item-name", "macro")

The AppendItem macro appends a menu item to the end of a menu created with the [InsertMenu](#) macro.

Parameter	Description
menu-id	The name of the menu to which you're adding the new menu item. The menu name must appear in quotation marks.
item-id	The name used internally by WinHelp to identify the menu item. The item-id is also used by the DisableItem and DeleteItem macros. The name must appear in quotation marks.
item-name	The name of the new item you're placing in the menu. The name must appear in quotation marks. Within the quotation marks, place an ampersand (&) before the character you want to use for the macro's accelerator key.
macro	Specifies one or more macros to execute when users choose the menu item. The macro must appear in quotation marks. Separate multiple macros in a string by semicolons (;).

Comments

WinHelp ignores this macro if it's executed from a secondary window.

If the keyboard accelerator you choose conflicts with other menu access keys, WinHelp displays the error message "Unable to add item" and ignores the macro.

Example

The following macro appends a menu item labeled "Tips" to a popup menu that has an identifier "IDM_TIP". Choosing the menu item causes a jump to a topic with the mapstring "tpc1" in the TIP.HLP file:

```
AppendItem("IDM_BKS", "IDM_TLS", "&Tips", "JI('tls.hlp', `tpc1')")
```

Back WinHelp macro

Back()

The Back macro displays the previous topic in the history list. The history list displays a list of the last 30 topics (excluding popup topics) the user displayed since starting WinHelp.

Parameters

None.

Comments

This history list is kept separately from the list displayed in the History window.

WinHelp ignores this macro if it's executed from a secondary window.

If the Back macro is executed when the Back list is empty, WinHelp takes no action.

Users can use Back to back up to a topic only once. After Back returns to a topic, that topic is removed from the list of previously displayed topics.

BackFlush WinHelp macro

BackFlush()

Removes the back history list from the current window. This macro doesn't affect the history list displayed in the History window.

Parameters

None.

Comment

Help Workshop automatically converts the BackFlush macro to BF, which is the only form of this macro that WinHelp can use.

BookmarkDefine WinHelp macro

BookmarkDefine()

The BookmarkDefine macro displays the **Define dialog** from the **Bookmark** menu.

Parameters

None.

Comments

Using this macro in secondary windows isn't recommended.

If you execute the BookmarkDefine macro from a popup window, the bookmark is attached to the topic where you invoked the popup window.

BookmarkMore WinHelp macro

BookmarkMore()

The BookmarkMore macro displays the **More dialog** from the **Bookmark** menu when the menu lists more than nine bookmarks.

Parameters

None.

Comments

Using the macro in secondary windows isn't recommended.

BrowseButtons WinHelp macro

BrowseButtons()

The BrowseButtons macro adds the browse buttons (Next >> and Previous <<) to the windows button bar.

Parameters

None.

Comments

WinHelp ignores this macro if it's executed from a secondary window.

If the BrowseButtons macro is used with one or more CreateButton macros in the project file, the buttons appear in the same order on the button bar as the macros appear in the project file.

Example

The following macros in the project file cause the Clock button to appear immediately before the two browse buttons on the button bar:

[CONFIG]

CreateButton("&Clock", "ExecFile('clock', 0)")

BrowseButtons()

ChangeButtonBinding WinHelp macro

ChangeButtonBinding(button-ID, button-macro)

The ChangeButtonBinding macro assigns a help macro to a button bar button.

Parameter	Description
button-ID	A standard help button identifier, or an identifier assigned to the button using the <u>CreateButton</u> macro.
button-macro	The help macro assigned to the button. Separate multiple macros using colons or semicolons. (Use colons if multiple macros are specified in the project file.)

Comments

Help Workshop automatically converts the ChangeButtonBinding macro to CBB.

Example

The following macro assigns an ALink macro to a button with the ID "SeeAlso."

[CONFIG]

```
ChangeButtonBinding(SeeAlso, ALink(another topic))
```

ChangeEnable WinHelp macro

ChangeEnable("button-ID", "button-macro")

Assigns a macro to a navigation bar button and enables that button.

Parameter	Description
button-ID	A standard help button identifier, or an identifier assigned to the button using the <u>CreateButton</u> macro.
button-macro	The macro assigned to the button.

Comments

This macro is equivalent to using both the ChangeButtonBinding and EnableButton macros.

Help Workshop automatically converts this macro to CE.

Example

In the following macro, an ALink macro is assigned to a button labelled SeeAlso:

```
ChangeEnable(SeeAlso, ALink(another topic))
```

ChangeltemBinding WinHelp macro

ChangeltemBinding("item-id", "item-macro")

The ChangeltemBinding macro assigns a help macro to an item previously added to a WinHelp menu using the [AppendItem](#) macro.

Parameter	Description
item-id	The menu item appended by the AppendItem macro. The item identifier must be enclosed in quotation marks.
item-macro	Specifies the help macro to execute when the user selects the item. The macro must be enclosed in quotation marks.

Comments

WinHelp ignores this macro if it's executed from a secondary window.

You can abbreviate the ChangeltemBinding as CIB.

Example

The following macro changes the menu item identified by "time_item" to display the Windows clock:

```
ChangeltemBinding("time_item", "ExecFile(`clock`, 0)")
```

CheckItem WinHelp macro

CheckItem("item-id")

The CheckItem macro places a checkmark next to a menu item.

Parameter	Description
item-id	Identifies the menu item to check. The item identifier must be enclosed in quotation marks.

Comments

The CheckItem macro can be abbreviated as CI.

WinHelp ignores this macro if it's executed from a secondary window.

Use the UncheckItem macro to clear a checkmark.



CloseSecondarys WinHelp macro

CloseSecondarys()

The CloseSecondarys macro closes all but the current secondary windows.

Parameters

None.

Comments

WinHelp converts this macro to CS.

CloseWindow WinHelp macro

CloseWindow("window-name")

The CloseWindow macro closes a secondary window or the main help window.

Parameter	Description
window-name	Specifies the name of the window to close. The name "Main" is reserved for the main help window. For secondary windows, the window name is defined on RoboHELP Explorer's Project tab , Project folder, under the Windows folder (or the [WINDOWS] section of the project file). Enclose the window in quotation marks.

Comments

If you don't specify a window name, the macro closes the Main window.

If the Main window is closed and no other windows are open, WinHelp exits the help file and quits.

If the specified window name doesn't exist, WinHelp ignores the macro.

Help Workshop automatically converts this macro to CW.

Example

The following macro closes the secondary window named "keys":

```
CloseWindow("keys")
```



Compare WinHelp macro

Compare(HLP-filename)

The Compare macro runs a second instance of WinHelp for comparing original and translated versions of a help file.

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

HLP-filename	The path and filename of the help file to display.
--------------	--

Contents WinHelp macro

Contents()

The Contents macro displays the Contents tab if the help file has an associated CNT file. If the help file doesn't have a CNT file, this macro displays the default topic in the current help file.

The default topic (contents topic) is defined on the **Project Settings: Project tab** (the CONTENTS option in the [OPTIONS] section of the HPJ). If there isn't a default topic specified, WinHelp uses the first topic in the first help document specified in the HPJ.

Parameters

None.

ControlPanel WinHelp macro

ControlPanel(CPL_name[,panel_name,tabnum])

The ControlPanel macro opens a Control Panel applet to a specific tab.

Parameter	Description
CPL_name	The name of the Control Panel program applet.
panel_name	The name of the Control Panel to display. (Must be identical to the text under the applet's icon.)
tabnum	The number of the dialog box tab to display on top.

Comments

Not all Control Panel applets recognize the panel_name and tabnum parameters.

CopyDialog WinHelp macro

CopyDialog()

The CopyDialog macro displays the **Copy dialog** from WinHelp's **Edit** menu.

Parameters

None.

Comments

Using this macro in secondary windows isn't recommended.

CopyTopic WinHelp macro

CopyTopic()

The CopyTopic macro copies all the text in the currently displayed topic to the Windows Clipboard.

Parameters

None.

Comments

Using the macro in secondary windows isn't recommended.

This macro only copies text – not bitmaps or other images.

If the CopyTopic macro is run from a popup window, it only copies the text of the topic that invoked the popup window.

CreateButton WinHelp macro

CreateButton("button-id", "name", "macro")

The CreateButton macro adds a new button to the button bar.

Parameter	Description
button-id	The name that WinHelp uses internally to identify the button. This name must appear in quotation marks. Use the button-id in the DisableButton or DestroyButton macros if you want to remove or disable the button, or in the ChangeButtonBinding macro if you want to change the help macro executed by the button in certain topics.
name	Specifies the button name – the text that appears on the button. Within the quotation marks, place an ampersand (&) before the character you want to use for the macro's accelerator key. The button name is case-sensitive and can contain up to 96 characters – any additional characters are ignored.
macro	Specifies the help macro or macro string executed when the user clicks on the button. Separate multiple macros in a macro string using semicolons.

Comments

If you don't specify an accelerator (with the & symbol), WinHelp automatically chooses the first letter of the button name that can be used without conflicting with other buttons on the button bar. If a specified accelerator conflicts with a previous button (including standard WinHelp buttons), WinHelp attempts to move the accelerator to the first letter that doesn't conflict. If it can't find such a letter, the original accelerator specified remains. If users attempt to access the accelerator, WinHelp displays a dialog asking the user to choose which button they want.

WinHelp allows a total of 22 buttons, including the standard Browse buttons. Of those 22, you can customize 16 buttons.

If you use the [BrowseButtons](#) macro with one or more CreateButton macros in the project file, the buttons appear in the same order on the button bar as the macros appear in the project file.

WinHelp ignores this macro if it's executed from a secondary window.

You can abbreviate the CreateButton macro as CB.

Example

The following macro creates a new button labeled "Ideas" that jumps to the topic with the mapstring "dir" in the IDEAS.HLP file when clicked:

```
CreateButton("btn_ideas", "&Ideas", "JumpId('ideas.hlp', `dir`)" )
```

Deleteltem WinHelp macro

Deleteltem("item-id")

The Deleteltem macro removes a menu item that was previously added using the [AppendItem](#) macro.

Parameter	Description
item-id	The item identifier used in the AppendItem macro. The item identifier must be enclosed in quotation marks.

Comments

WinHelp ignores this macro if it's executed from a secondary window.

Example

The following macro removes the menu item "Tips" that was appended in the example for the [AppendItem](#) macro:

```
Deleteltem("IDM_TIPS")
```

DeleteMark WinHelp macro

DeleteMark("marker-text")

The DeleteMark macro removes a text marker added with the SaveMark macro.

Parameter	Description
marker-text	The text marker previously added by the <u>SaveMark</u> macro. The marker text must be enclosed in quotation marks.

Comments

If the marker doesn't exist when the DeleteMark macro is executed, WinHelp displays a "Topic not found" error message.

Example

The following macro removes the marker "Managing Projects" from a help file:

```
DeleteMark("Managing Projects")
```

DestroyButton WinHelp macro

DestroyButton("button-id")

The DestroyButton macro removes a button added with the [CreateButton](#) macro.

Parameter	Description
button-id	Identifies a button previously created by the CreateButton macro. The button identifier must be enclosed in quotation marks.

Comments

The button identifier cannot be an identifier for one of the standard help buttons. For a list of those identifiers, see the [ChangeButtonBinding](#) macro.

WinHelp ignores this macro if it's executed from a secondary window.

DisableButton WinHelp macro

DisableButton("button-id")

The DisableButton macro grays out a button added with the CreateButton macro if the EnableButton macro has not yet been executed.

Parameter	Description
button-id	The identifier assigned to the button by the <u>CreateButton</u> macro. The button identifier must be enclosed in quotation marks.

Comments

WinHelp ignores this macro if it's executed from a secondary window.

You can abbreviate the DisableButton macro as DB.

DisableItem WinHelp macro

DisableItem("item-id")

The DisableItem macro grays out a menu item added with the [AppendItem](#) macro if the [EnableItem](#) macro has not yet been executed.

Parameter	Description
item-id	The menu item previously appended with the AppendItem macro. The item identifier must be enclosed in quotation marks.

Comments

WinHelp ignores this macro if it's executed from a secondary window.

The DisableItem macro can be abbreviated as DI.

EnableButton WinHelp macro

EnableButton("button-id")

The EnableButton macro enables a button disabled with the [DisableButton](#) macro.

Parameter	Description
button-id	The identifier assigned to the button by the CreateButton macro. The button identifier must be enclosed in quotation marks.

Comments

WinHelp ignores this macro if it's executed from a secondary window.

You can abbreviate the EnableButton macro as EB.

EnableItem WinHelp macro

EnableItem("item-id")

The EnableItem macro enables a menu item disabled with the DisableItem macro.

Parameter	Description
item-id	The identifier assigned to the menu item by the <u>AppendItem</u> macro. The item identifier must be enclosed in quotation marks.

Comments

WinHelp ignores this macro if it's executed from a secondary window.

You can abbreviate the EnableItem macro as EI.



EndMPrint WinHelp macro

EndMPrint()

The EndMPrint macro dismisses the printing message box and stops the printing of multiple topics.

Parameters

None.



ExecFile WinHelp macro

ExecFile(program[,arguments[,display-state[,topic-ID]]])

Specifies a file and automatically starts the program that is associated with that file.

Parameter	Description
program	The name of the program to be run or the name of a file. (If you specify a file, the program associated with that file type is started.)
arguments	The command-line arguments to send to the program.
display-state	(Optional) A value indicating how to show the program's window. If you don't specify this parameter, Windows uses SW_SHOW, which activates the window and shows it in its current size and position. You can use one of the following values for this optional parameter: SW_HIDE Hides the window. SW_MINIMIZE Minimizes the window. SW_RESTORE Restores the window to its original size and position if the window is minimized or maximized. SW_SHOW Activates the window and displays it in its current size and position. SW_SHOWMAXIMIZED Activates the window and displays it as a maximized window. SW_SHOWMINIMIZED Activates the window and displays it as an icon. SW_SHOWMINNOACTIVE Displays the window as an icon, but WinHelp keeps the focus. SW_SHOWNA Displays a window in its current state, but WinHelp keeps the focus. (If the window was minimized before this call, it stays minimized.) SW_SHOWNOACTIVATE Displays a window in its most recent size and position, but WinHelp keeps the focus. SW_SHOWNORMAL Activates and displays the window. If the window is minimized or maximized, Windows restores it to its original size and position (same as SW_RESTORE).
topic-ID	The topic ID to display if the specified file or program cannot be started.

Comments

If a path is specified, WinHelp first searches for the file in the specified path. If the file is not found or no path was specified, WinHelp searches the same locations as it does when it searches for help files:

- The folder of the current help file.
- The current folder.
- The System subfolder in the Windows folder.
- The Windows folder.
- The folders listed in the PATH environment.
- The location specified in the Winhelp.ini file.
- The help portion of the registry.

Help Workshop automatically converts the ExecFile macro to EF.

Example

The following example opens the WIN.INI file in the program associated with .ini files:

```
ExecFile(win.ini)
```

ExecProgram WinHelp macro

ExecProgram("command-line", display-state)

The ExecProgram macro executes a Windows application. However, it's recommended that you use the [ExecFile](#) macro instead of this one.

Parameter	Description								
command-line	The command line for the application to be executed. The command line must be enclosed in quotation marks. WinHelp searches for this application in the current folder, followed by the Windows folder, the user's path, and the folder containing the currently viewed help file.								
display-state	A value indicating how the application is shown when executed. You can choose one of the following values: <table><tr><th>Value</th><th>Meaning</th></tr><tr><td>0</td><td>Normal</td></tr><tr><td>1</td><td>Minimized</td></tr><tr><td>42</td><td>Maximized</td></tr></table>	Value	Meaning	0	Normal	1	Minimized	42	Maximized
Value	Meaning								
0	Normal								
1	Minimized								
42	Maximized								

Comments

You can abbreviate the ExecProgram macro as EP.

Don't use the backslash character to escape double quotation-mark characters in macros. Instead, you can enclose the command line in single quotation marks and omit the backslash for the double quotation marks, as shown in the following example:

```
`command "string as parameter"
```

Note that the first single quotation mark must be an open quote, and the last single quotation mark must be a closed quote.

Example

The following example executes the Clock application. The application is minimized when it starts:

```
ExecProgram('clock.exe', 1)
```

Exit WinHelp macro

Exit()

The Exit macro exits the WinHelp application. It has the same effect as selecting **Exit** from the **File** menu.

Parameters

None.



ExtAbleItem WinHelp macro

ExtAbleItem(item-ID, display-state)

Enables or disables a menu item.

Parameter	Description						
item-ID	The identifier assigned to the menu item by the AppendItem, ExtInsertItem, or InsertItem macro.						
display-state	The state of the menu item, which can be one of the following values: <table><tr><th>Value</th><th>Meaning</th></tr><tr><td>GRAYED</td><td>Disables (grays out) the menu item.</td></tr><tr><td>CHECKED</td><td>Places a check mark next to the menu item.</td></tr></table>	Value	Meaning	GRAYED	Disables (grays out) the menu item.	CHECKED	Places a check mark next to the menu item.
Value	Meaning						
GRAYED	Disables (grays out) the menu item.						
CHECKED	Places a check mark next to the menu item.						

Comments

You cannot use this macro to change a standard WinHelp menu item.

WinHelp ignores this macro if it's run in a secondary window.



ExtInsertItem WinHelp macro

ExtInsertItem(menu-ID, item-ID, item-name, macro, position[, display-state])

Inserts a menu item at a given position on an existing menu.

Parameter	Description								
menu-ID	Identifies the menu to which you're adding the item. The menu can be a standard WinHelp menu or a menu previously created by the ExtInsertMenu or InsertMenu macros.								
item-ID	Specifies the name that WinHelp uses internally to identify the menu item.								
item-name	Specifies the name displayed on the menu. The item name is case-sensitive. Place an ampersand (&) before the character you want to use for the macro's accelerator key.								
macro	The macro or macro string to run when a user clicks the menu item. Separate multiple macros in a string using colons or semicolons. Colons are required if this macro is specified in a project file.								
position	The numeric position of the menu item in the menu. Position 0 is the first or top-most position in the menu.								
display-state	The state of the menu item, which can be one of the following values: <table><tr><th>Value</th><th>Meaning</th></tr><tr><td>GRAYED</td><td>Disables (grays-out) the menu item.</td></tr><tr><td>CHECKED</td><td>Places a check mark next to the menu item.</td></tr><tr><td>SEPARATOR</td><td>Adds a separator bar to the menu.</td></tr></table>	Value	Meaning	GRAYED	Disables (grays-out) the menu item.	CHECKED	Places a check mark next to the menu item.	SEPARATOR	Adds a separator bar to the menu.
Value	Meaning								
GRAYED	Disables (grays-out) the menu item.								
CHECKED	Places a check mark next to the menu item.								
SEPARATOR	Adds a separator bar to the menu.								

Comments

You can use the item-ID parameter in a subsequent [DisableItem](#), [EnableItem](#), [ExtAbleItem](#), or [DeletetItem](#) macro to remove or disable the item, or to change the operations that the item performs in certain topics.

WinHelp ignores this macro if it's run in a secondary window.

The specified accelerator keys must be unique. If a key conflicts with other accelerator keys, WinHelp displays the error message "Unable to add item" and ignores the macro.

Example

The following macro inserts a menu item labeled "WordPad" on WinHelp's File menu:

```
ExtInsertItem(mnu_file, itm_wpad, &WordPad, ExecFile(wordpad.exe), 3)
```

ExtInsertMenu WinHelp macro

ExtInsertMenu(parent-ID, menu-ID, menu-name, menu-position[, display-state])

Inserts a submenu in a previously defined menu.

Parameter	Description						
parent-ID	The name that WinHelp uses to identify the parent menu to which this submenu is added. For a custom menu, this parameter is the name used when the menu was created by the ExtInsertMenu or <u>InsertMenu</u> macro.						
menu-ID	The name that WinHelp uses internally to identify the menu, and can be used by the <u>AppendItem</u> , <u>ExtInsertItem</u> , or ExtInsertMenu macros to access the menu.						
menu-name	The name that WinHelp displays for this menu. Place an ampersand (&) before the character you want to use for the macro's accelerator key.						
menu-position	The integer position on the menu bar of the new menu name. Position 0 is the first position.						
display-state	The state of the menu item. This parameter may be one of the following values: <table><tr><th>Value</th><th>Meaning</th></tr><tr><td>CHECKED</td><td>Places a check mark next to the menu item.</td></tr><tr><td>GRAYED</td><td>Disables (grays out) the menu item.</td></tr></table>	Value	Meaning	CHECKED	Places a check mark next to the menu item.	GRAYED	Disables (grays out) the menu item.
Value	Meaning						
CHECKED	Places a check mark next to the menu item.						
GRAYED	Disables (grays out) the menu item.						

Comment

WinHelp ignores this macro if you run it in a secondary window.

Example

The following macro adds a Wizard submenu to the standard help menu:

```
ExtInsertMenu(mnu_help, mnu_wizard, &Wizard, 2)
```



FileExist WinHelp macro

FileExist(filename)

The FileExist macro checks to see if a specified file or program exists on the user's computer.

Parameter	Description
filename	The name of the file to find on the user's computer. You can also specify a path.

Comments

This macro can be used in conjunction with macros like IfThenElse, which use the result of a Boolean macro to determine what action to take.

If you specify a path, WinHelp first searches for the file in the specified path. If the file isn't found or no path was specified, WinHelp searches the same locations it does to find a help file:

- The folder of the current help file.
- The current folder.
- The System subfolder in the Windows folder.
- The Windows folder.
- The folders listed in the PATH environment.
- The location specified in the Winhelp.ini file.
- The help portion of the registry.

Help Workshop automatically converts the FileExist macro to FE.

Example

The following macro checks to see if "Myapp.exe" has been installed. If the file is present, WinHelp runs it. If the file is not present, WinHelp displays a topic:

```
IfThenElse(FileExist(myapp.exe), ExecFile(myapp), JumpId(install_my_app))
```

FileOpen WinHelp macro

FileOpen()

The FileOpen macro displays the Open dialog box from the File menu.

Parameters

None.

Comments

Using the macro in secondary windows isn't recommended.

Find WinHelp macro

Find()

Displays the **Find** tab in the **Help Topics** dialog.

Parameters

None.

Comment

The Find macro is new for WinHelp version 4.0.



Finder WinHelp macro

Finder()

The Finder macro displays the **Help Topics dialog**.

Parameters

None.

Comments

Help Workshop automatically converts the Finder macro to FD.



FloatingMenu WinHelp macro

FloatingMenu()

Displays the context (floating) menu at the current mouse cursor position. This menu also appears when users click a topic using their right mouse button.

Parameters

None.

Comments

You can add as many as 20 menu items to the context menu.



Flush WinHelp macro

Flush()

The Flush macro causes WinHelp to process any pending messages, including previously called macros.

Parameters

None.

FocusWindow WinHelp macro

FocusWindow("window-name")

The FocusWindow macro changes the focus to the window you specify, which can be the main help window or a secondary window.

Parameter	Description
window-name	The name of the window to receive the focus. The name "Main" is reserved for the main help window. For secondary windows, the window name is defined in the [WINDOWS] section of the project file. This name must be enclosed in quotation marks.

Comments

This macro is ignored if the specified window doesn't exist.

Example

The following macro changes the focus to the secondary window "keys":

```
FocusWindow("keys")
```



Generate WinHelp macro

Generate(message, wParam, lParam)

Posts a message to the currently active help window.

Parameter	Description
message	The message to send to the currently-active help window.
wParam	The first argument of the message.
lParam	The second argument of the message.

GoToMark WinHelp macro

GoToMark("marker-text")

The GoToMark macro jumps to a marker set with the SaveMark macro.

Parameter	Description
marker-text	A text marker previously defined using the <u>SaveMark</u> macro.

Example

The following macro jumps to the marker "Managing Projects".

```
GoToMark("Managing Projects")
```

HelpOn WinHelp macro

HelpOn()

The HelpOn macro displays the help file for the WinHelp application. The macro carries out the same action as choosing the **How to Use Help** command on the **Help** menu.

Parameters

None.

HelpOnTop WinHelp macro

HelpOnTop()

The HelpOnTop macro toggles the on-top state of WinHelp. It's equivalent to checking or unchecking the Always On Top command in the help menu.

Parameters

None.

Comments

WinHelp doesn't provide a macro to check the current state of the Always On Top command. It's up to the user to determine whether the macro should be used to change the state of the command.

History WinHelp macro

History()

The History macro displays the history list, which shows the last 40 topics the user has viewed since opening a help file in WinHelp. It has the same effect as choosing the **History** button.

Parameters

None.

Comments

WinHelp ignores this macro if it's executed from a secondary window.

IfThen WinHelp macro

IfThen(`macro("marker-text")`, `macro`)

The IfThen macro executes a help macro if a given marker exists.

Parameter	Description
macro	Specifies any macro that returns a True or False value (FileExist, InitMPrint, IsBook, IsMark, IsNotMark, TestALink, and TestKLink).
marker-text	A text marker previously created using the <u>SaveMark</u> macro. The marker must be enclosed in quotation marks.
macro	A help macro or macro string to be executed if the marker exists. You must separate multiple macros in a macro string with semicolons.

Example

The following macro jumps to the topic with the mapstring "man_mem" if a marker named "Managing Memory" has been set by the SaveMark macro:

```
IfThen(IsMark("Managing Memory"), "JI(`trb.hlp', `man_mem')")
```


IfThenElse WinHelp macro

IfThenElse(**macro**("marker-text"), **macro1**, **macro2**)

The IfThenElse macro executes one of two help macros depending on whether or not a marker exists. The IsMark macro is used to determine if the marker exists.

Parameter	Description
macro	Any macro that returns a True or False value (FileExist, InitMPrint, IsBook, IsMark, IsNotMark, TestALink, and TestKLink).
marker-text	A text marker previously created by using the IsMark macro. You must enclose the marker in quotation marks.
macro1	A help macro or macro string to execute if the marker exists. You must use semicolons to separate multiple macros in either macro string.
macro2	Specifies a help macro or macro string to be executed if the marker doesn't exist. You must use semicolons to separate multiple macros in either macro string.

Example

The following macro jumps to the topic with the mapstring "mem" if a marker named "Memory" has been set by the SaveMark macro. If the marker doesn't exist, WinHelp jumps to the next topic in the browse sequence.

```
IfThenElse(IsMark("Memory"), "JI('trb.hlp', `mem')", "Next()")
```

Inet WinHelp macro

Inet("URL")

The Inet WinHelp macro jumps to an HTML topic or page and displays that page inside the user's default Internet browser.

Parameter	Description
URL	Specifies the URL (Uniform Resource Locator) address for the destination HTML page on the Internet or intranet. Make sure to enter the full http:// statement for the URL.

Comments

When you use the Inet macro, you need to ship the following files with your compiled help file to ensure that the Inet macro functions properly. You can find these files on your PC in the \WinHelp\RoboHELP\Internet\Distrib folder. Install the appropriate DLL in the user's \Windows\system folder (\WinNT\system for Windows NT users), and install the SETBROWSE.EXE file in the same folder as the help file.

SETBROWS.EXE	May be included to allow users to specify the browse path for an Internet browser or navigator other than the default.
INETWH16.DLL	Adds Internet access capabilities to Windows 3.1 help systems using the HC31.EXE or HCP.EXE help Compiler.
INETWH32.DLL	Adds Internet access functionality to Windows 95, 98 and NT help systems using the HCW.EXE help Compiler.

Examples

The following macro command initiates a jump to an HTML page (Internet or intranet):

Inet ("http://www.address")

Example: Inet("http://www.blue-sky.com")

The following macro command downloads the "filename.xxx" file from the Internet:

Inet("ftp://ftp.address/directory/filename.xxx")

Example: Inet("ftp://ftp.microsoft.com/developr/MSDN/JulyCD/help.zip")

The following macro command browses a folder on the Internet:

Internet("ftp://ftp.address/folder/")

Example: Internet("ftp://ftp.microsoft.com/developr/MSDN/")

The following macro command sends email:

Internet("mailto:EmailAddress")

Example: Internet("mailto:robohelpdocs@blue-sky.com")



INETWH_Initialize WinHelp macro

INETWH_Initialize()

This macro is required whenever you use the [Inet](#) or [JumpHtml](#) macro to provide Internet/intranet access from within a WinHelp help file.

RoboHELP automatically places this macro in the Startup Macros section (**Project tab, Project folder Startup Macros** folder) of your help project whenever you select any of these features.

Parameters

None.



InitMPrint WinHelp macro

InitMPrint()

The InitMPrint prints multiple topics.

Parameters

None.

InsertItem WinHelp macro

InsertItem("menu-id", "item-id", "item-name", "macro", position)

The InsertItem macro inserts a standard menu item or custom menu item that you create with the [InsertMenu](#) macro at a given position on an existing menu.

Parameter	Description										
menu-id	Identifies a standard WinHelp menu or a menu previously created using the InsertMenu macro in which to insert the new menu item. The standard menu items are: <table><tr><th>Name</th><th>Menu</th></tr><tr><td>MNU_FILE</td><td>File</td></tr><tr><td>MNU_EDIT</td><td>Edit</td></tr><tr><td>MNU_BOOKMARK</td><td>Bookmark menu</td></tr><tr><td>MNU_HELP</td><td>Help</td></tr></table> To add the menu item to a menu not listed, use the name of a menu created using the InsertMenu macro. In all cases, enclose the menu identifier in quotation marks.	Name	Menu	MNU_FILE	File	MNU_EDIT	Edit	MNU_BOOKMARK	Bookmark menu	MNU_HELP	Help
Name	Menu										
MNU_FILE	File										
MNU_EDIT	Edit										
MNU_BOOKMARK	Bookmark menu										
MNU_HELP	Help										
item-id	The name that WinHelp uses internally to identify the menu item. Enclose the item identifier in quotation marks.										
item-name	The name WinHelp displays in the menu. This name is case-sensitive and must be enclosed in quotation marks. Place an ampersand (&) before the character you want to use for the macro's accelerator key.										
macro	A help macro or macro string to be executed when the user chooses the menu item. Enclose the macro in quotation marks. Separate multiple macros in a string using semicolons (;).										
position	An integer identifying the position of the menu item in the menu. Position 0 is the first position in the menu.										

Comments

To remove or disable the item, or change the operations that the item performs in certain topics, use the [DisableItem](#) or [DeleteItem](#) macro.

WinHelp ignores this macro if you execute it in a secondary window.

The keyboard access keys you specify must be unique. If a key conflicts with other menu access keys, WinHelp displays the error message "Unable to add item" and ignores the macro.

Example

The following macro inserts the "Tips" menu item as the third item on a menu with the identifier "MNU_BKS." Selecting the menu item causes a jump to a topic with the mapstring "tls1" in the TIP.HLP file:

```
InsertItem("mnu_bks", "m_tls", "&Tips", "JI('tls.hlp', 'tls1')", 3)
```

InsertMenu WinHelp macro

InsertMenu("menu-id", "menu-name", menu-position)

The InsertMenu macro inserts a new menu in the WinHelp menu bar.

Parameter	Description
menu-id	The name that WinHelp uses internally to identify the menu. You must enclose the menu identifier in quotation marks.
menu-name	The name that WinHelp displays on the menu bar. This name must be enclosed in quotation marks. Place an ampersand (&) before the character you want to use for the macro's accelerator key.
menu-position	An integer identifying the position of the new menu on the menu bar. Positions are numbered from left to right, with position 0 being the left-most menu.

Comments

WinHelp ignores this macro if it's executed from a secondary window.

To add macros to the menu you've created, use the [AppendItem](#) macro.

If you specify an accelerator key, it must be unique. If a specified accelerator key conflicts with other current accelerator keys, WinHelp displays the error message "Unable to add menu" and ignores the macro.

Example

The following macro adds the "Tools" menu as the fourth item on the WinHelp menu bar, and specifies that the user can press ALT+T to open the menu.

```
InsertMenu("IDM_TOOL", "&Tools", 3)
```



IsBook WinHelp macro

IsBook()

The IsBook macro determines whether WinHelp runs as a standalone system, or from a program.

Parameters

None.

IsMark WinHelp macro

IsMark("marker-text")

The IsMark macro is used as a parameter in the conditional macros IfThen and IfThenElse to test whether or not a marker set by the SaveMark macro exists. . The IsMark macro returns nonzero if the mark exists, or zero if it doesn't.

Parameter	Description
marker-text	Specifies a text marker previously created using the <u>SaveMark</u> macro.

Comments

Use the Not macro to reverse the results of the IsMark macro.

Example

The following macro jumps to the topic with the mapstring "man_mem" if a marker named "Managing Memory" has been set by the SaveMark macro:

```
IfThen(IsMark("Managing Memory"), "JI(`trb.hlp', `man_mem')")
```


IsNotMark WinHelp macro

IsNotMark("marker-text")

The IsNotMark macro tests whether or not a marker set by the SaveMark macro exists.

Parameter	Description
marker-text	Specifies a text marker, enclosed in quotation marks, previously created using the SaveMark macro.

JumpContents WinHelp macro

JumpContents("filename")

The JumpContents macro jumps to the contents topic of a specified file in the help file. The contents topic is indicated by the CONTENTS option entry in the [OPTIONS] section of project file. If the CONTENTS option isn't specified, WinHelp jumps to the first topic in the help file.

Parameter	Description
filename	The name of the destination file for the jump. The filename must be enclosed in quotation marks. If WinHelp cannot find this file, it displays an error message and doesn't perform the jump.

Comments

WinHelp ignores this macro if it's executed from a secondary window.

Example

The following macro jumps to the contents topic of the PROGMAN.HLP file:

```
JumpContents("PROGMAN.HLP")
```

JumpContext WinHelp macro

JumpContext("filename", map number)

This macro jumps to a topic identified by a Map number

Parameter	Description
filename	The name of the destination file for the jump. The filename must be enclosed in quotation marks. If WinHelp cannot find this file, it displays an error message and doesn't perform the jump.
map number	The map number of the topic in the destination file. The map number must be defined in the [MAP] section of the project file. If the map number isn't valid, WinHelp jumps to the contents topic or to the first topic in the file instead, and displays an error message.

Comments

You can abbreviate the JumpContext macro as JC.

Example

The following macro jumps to the topic mapped to map number 801 in the PROGMAN.HLP file:

```
JumpContext("PROGMAN.HLP", 801)
```



Win 95 JumpHash WinHelp macro

JumpHash([filename>window-name,] hash-code)

Jumps to a topic identified by a hash number.

Parameter	Description
filename	The name of the help file containing the hash number. This is an optional parameter that you use to jump to a topic that isn't in the current help file.
window-name	The window type in which to display the topic.
hash-code	The <u>hash number</u> of the topic in the destination file.

Example

The following macro displays this topic:

```
JumpHash(hcw, 4104506004)
```

JumpHelpOn WinHelp macro

JumpHelpOn()

The JumpHelpOn macro jumps to the contents topic of the "How to Use Help" file. The How To Use Help file is the default WINHELP.HLP file shipped with Windows 3.1, or the help file designated by the SetHelpOnFile macro in the [CONFIG] section of the project file.

Parameters

None.

Comments

If WinHelp cannot find the specified help file, it displays an error message and doesn't perform the jump.

Example

The following macro jumps to the contents topic of the designated How to Use Help file:

JumpHelpOn()



JumpHtml WinHelp macro

JumpHtml("URL")

The JumpHtml WinHelp macro jumps to an HTML page and displays that page inside the help window, creating the impression that the user is still within the WinHelp system. This WinHelp macro requires Internet Explorer 4 or higher be set as the default Internet browser on the end user's system.

Parameter	Description
URL	Specifies the URL (Uniform Resource Locator) address for the destination HTML page on the Internet or intranet. Make sure to enter the full http:// statement for the URL.

Comments

When you use the JumpHtml macro, you need to do the following to ensure the macro functions properly:

Make sure your end users have Internet Explorer 4 or higher on the systems and that IE is set as the default Internet browser. (You can ship Internet Explorer along with your help system, just in case.)

Ship a DLL with your compiled help file to ensure that the macro functions properly. This DLL is located on your PC in the \WinHelp\RoboHELP\Internet\Distrib folder. Install the DLL in the user's \Windows\system folder (\WinNT\system for Windows NT users).

HTMLWH.DLL	Adds Internet access functionality within the help system's window to WinHelp 4 help systems. Without the appropriate DLL, users cannot use a macro to access an HTML topic.
------------	--

Example

The following macro command jumps to an HTML page (Internet or intranet) and displays it within the help window:

```
JumpHtml("http://www.address")
```

Example: `JumpHtml("http://www.blue-sky.com")`

JumpId WinHelp macro

JumpId("filename", "Topic ID")

The JumpId macro jumps to the topic with the specified topic ID in the help file.

Parameter	Description
Filename	Specifies the name of the help file that contains the topic ID. (This parameter is optional and only used when jumping to an external topic in current help file.)
window-name	Specifies the window to use to display the destination topic.
topic-ID	Specifies the ID of the topic you want to jump to.

Comments

You can abbreviate the JumpId macro as JI.

Example

The following macro jumps to a topic with "second_topic" as its mapstring in the SECOND.HLP file:

```
JI("second.hlp", "second_topic")
```

JumpKeyword WinHelp macro

JumpKeyword("filename", "keyword")

The JumpKeyword macro loads the indicated help file, searches through the K keyword table for a particular keyword, and displays the first topic containing the index keyword specified in the macro.

Parameter	Description
filename	The name of the help file containing the desired keyword table. The filename must be enclosed in quotation marks. If this file doesn't exist, WinHelp displays an error message and doesn't perform the jump.
keyword	The keyword for which to search. The keyword must be enclosed in quotation marks. If WinHelp finds more than one match, it displays the first matched topic. If it doesn't find any matches, it displays a "Not a keyword" message and displays the contents topic of the destination file instead.

Comments

You can abbreviate the JumpKeyword macro as JK.

Example

The following macro initiates a search for the keywords "hands" in the CLOCK.HLP file:

```
JumpKeyword("clock.hlp", "hands")
```




KLink WinHelp macro

KLink("keyword[;keyword]"[,type[, "topic-ID"[,window-name]]])

The KLink macro searches for matching keywords in the current help file.

Parameter	Description
keyword	One or more keywords for which to search. You must separate multiple keywords with a semicolon. If a keyword contains a comma, the entire keyword string must be enclosed in quotation marks. Note: As of this release, you can only search for one keyword at a time.
type	The action to perform if one or more keywords is found. If you don't specify an action or enter "0," WinHelp displays the Topics Found dialog box with a list of the topics containing the keyword. You can choose one or more of the following actions, separated by spaces: JUMP (1) Directs WinHelp to jump directly to the topic if only one topic is found that matches any part of the keyword. TITLE (2) Directs WinHelp to display the title of the help file (as specified in the CNT file) for the topic title of each keyword found if a keyword is found in more than one help file. TEST (4) Specifies that the macro should return a value indicating whether or not there is at least one match. The TestKLink macro is converted by Help Workshop into an KLink macro with this parameter.
topic-ID	If no matches are found, specifies the topic to display in a popup window. If you don't specify this parameter, WinHelp displays a message box indicating that no topics were found.
window-name	The window in which to display the topic. If you don't specify this parameter, WinHelp uses the window specified for the topic (if one is defined), or the default or current window. Note that if this macro results in an inter-file jump, the window must be defined in the Project (HPJ) file for the help file to which the topic jumps.

Menu WinHelp macro

Menu()

The Menu macro displays the map menu for the current help window.

Parameters

None.

MPrintHash WinHelp macro

MPrintHash(hash-code)

The MPrintHash macro prints a topic identified by a specific hash number .

Parameter	Description
hash-code	The <u>hash number</u> of the topic to be printed.

MPrintID WinHelp macro

MPrintID(topic-ID)

The MPrintID macro prints a topic identified by a specific topic ID.

Parameter	Description
topic-ID	The topic ID of the topic to print.

Next WinHelp macro

Next()

The Next macro displays the next topic in the browse sequence for the help file.

Parameters

None.

Comments

If the currently-displayed topic is the last topic of a browse sequence, nothing happens.

WinHelp ignores this macro if it's executed from a secondary window.



NoShow WinHelp macro

NoShow()

The NoShow macro prevents the current help window from being displayed.

Parameters

None.

Not WinHelp macro

Not(IsMark("marker-text"))

The Not macro reverses the result (nonzero or zero) returned by the IsMark macro, and is used with the IsMark macro as a parameter to the conditional macros IfThen and IfThenElse.

Parameter	Description
marker-text	Specifies a text marker previously created by using the <u>SaveMark</u> macro. You must enclose the marker text in quotation marks.

Example

The following macro jumps to the topic with the mapstring "doc1" if a marker named "Documents" has not been set by the SaveMark macro:

```
IfThen(Not(IsMark("Documents")), "JI(`trb.hlp', `doc1')")
```

PopupContext WinHelp macro

PopupContext("filename", map number)

The PopupContext macro displays the topic identified by a specific map number in a popup window.

Parameter	Description
filename	The name of the file that contains the topic to be displayed. You must enclose the filename in quotation marks. If WinHelp cannot find this file, it displays an error message.
mapnumber	Specifies the map number of the topic to be displayed. The map number must be specified in the [MAP] section of the project file. If the map number isn't valid, WinHelp displays the contents topic or the first topic in the file instead.

Comments

You can abbreviate the PopupContext macro as PC.

Example

The following macro displays in a popup window the topic mapped to map number 801 in the PROGMAN.HLP file:

```
PopupContext("progman.hlp", 801)
```


PopupHash WinHelp macro

PopupHash([filename,] hash-code)

Displays the topic identified by a hash number in a pop-up window.

Parameter	Description
filename	The name of the destination help file for the jump. Use this optional parameter if the pop-up isn't in the current help file.
hash-code	The <u>hash number</u> of the topic to be displayed.

PopupId WinHelp macro

PopupId("filename", "Topic ID")

The PopupId macro displays a topic from a specified file in a popup window.

Parameter	Description
filename	Specifies the name of the help file containing the topic ID. Use this optional parameter if the topic is not in the current help file.
topic-ID	Specifies the ID of the topic to display.

Comments

You can abbreviate the PopupId macro as PI.

Example

The following macro displays a popup window with the map string "second_topic" from the SECOND.HLP file:

```
PopupId("second.hlp", "second_topic")
```

PositionWindow WinHelp macro

PositionWindow(x, y, width, height, state, "name")

The PositionWindow macro sets the size and position of a window. (Also called absolute positioning. This is much easier to achieve using RoboHELP's **Window Properties dialog**.)

Parameter	Description																						
x	Specifies in help units the x-coordinate of the upper-left corner of the window. WinHelp always assumes that the screen (regardless of resolution) is 1024 help units wide. For example, if the x-coordinate is 512, the left edge of the help window is in the middle of the screen.																						
y	Specifies in help units the y-coordinate of the upper-left corner of the window. WinHelp always assumes that the screen (regardless of resolution) is 1024 help units high. For example, if the y-coordinate is 512, the top edge of the help window is in the middle of the screen.																						
width	Specifies in help units the default width of the window.																						
height	Specifies in help units the default height of the window.																						
state	Specifies how the window is displayed. You can use one of the following values: <table><tr><th>Value</th><th>Meaning</th></tr><tr><td>SW_HIDE</td><td>Hides the window.</td></tr><tr><td>SW_MINIMIZE</td><td>Minimizes the window.</td></tr><tr><td>SW_RESTORE</td><td>Restores the window to its original size and position if the window is minimized or maximized.</td></tr><tr><td>SW_SHOW</td><td>Activates the window and displays it in its current size and position.</td></tr><tr><td>SW_SHOWMAXIMIZED</td><td>Activates the window and maximizes it.</td></tr><tr><td>SW_SHOWMINIMIZED</td><td>Activates the window and displays it minimized.</td></tr><tr><td>SW_SHOWMINNOACTIVE</td><td>Displays the window minimized, but WinHelp keeps the focus.</td></tr><tr><td>SW_SHOWNA</td><td>Displays a window in its current state, but WinHelp keeps the focus. (If the window was minimized before this call, it will stay minimized.)</td></tr><tr><td>SW_SHOWNOACTIVATE</td><td>Displays a window in its most recent size and position, but WinHelp keeps the focus.</td></tr><tr><td>SW_SHOWNORMAL</td><td>Activates and displays the window. If the window is minimized or maximized, Windows restores it to its original size and position (same as SW_RESTORE).</td></tr></table>	Value	Meaning	SW_HIDE	Hides the window.	SW_MINIMIZE	Minimizes the window.	SW_RESTORE	Restores the window to its original size and position if the window is minimized or maximized.	SW_SHOW	Activates the window and displays it in its current size and position.	SW_SHOWMAXIMIZED	Activates the window and maximizes it.	SW_SHOWMINIMIZED	Activates the window and displays it minimized.	SW_SHOWMINNOACTIVE	Displays the window minimized, but WinHelp keeps the focus.	SW_SHOWNA	Displays a window in its current state, but WinHelp keeps the focus. (If the window was minimized before this call, it will stay minimized.)	SW_SHOWNOACTIVATE	Displays a window in its most recent size and position, but WinHelp keeps the focus.	SW_SHOWNORMAL	Activates and displays the window. If the window is minimized or maximized, Windows restores it to its original size and position (same as SW_RESTORE).
Value	Meaning																						
SW_HIDE	Hides the window.																						
SW_MINIMIZE	Minimizes the window.																						
SW_RESTORE	Restores the window to its original size and position if the window is minimized or maximized.																						
SW_SHOW	Activates the window and displays it in its current size and position.																						
SW_SHOWMAXIMIZED	Activates the window and maximizes it.																						
SW_SHOWMINIMIZED	Activates the window and displays it minimized.																						
SW_SHOWMINNOACTIVE	Displays the window minimized, but WinHelp keeps the focus.																						
SW_SHOWNA	Displays a window in its current state, but WinHelp keeps the focus. (If the window was minimized before this call, it will stay minimized.)																						
SW_SHOWNOACTIVATE	Displays a window in its most recent size and position, but WinHelp keeps the focus.																						
SW_SHOWNORMAL	Activates and displays the window. If the window is minimized or maximized, Windows restores it to its original size and position (same as SW_RESTORE).																						
name	The name of the window to position. The name "Main" is reserved for the main help window. For secondary windows, the window name must be defined in the [WINDOWS] section of the project file. You must enclose the name in quotation marks.																						

Comments

If the window to be positioned doesn't exist, WinHelp ignores the macro.

The PositionWindow macro can be abbreviated as PW.

Example

The following macro positions the secondary window "Samples" in the upper-left corner (100, 100) with a width and height of 500 (in help units):

```
PositionWindow(100, 100, 500, 500, 0, "Samples")
```

Prev WinHelp macro

Prev()

The Prev macro displays the previous topic in the browse sequence for the help file. If the currently displayed topic is the first topic of a browse sequence, this macro does nothing.

Parameters

None.

Comments

WinHelp ignores this macro if it's executed from a secondary window.

Print WinHelp macro

Print()

The Print macro sends the currently displayed topic to the printer. It should be used only to print topics in windows other than the main help window (for example, topics in a secondary window).

Parameters

None.

PrinterSetup WinHelp macro

PrinterSetup()

The PrinterSetup macro displays the Printer Setup dialog box from the File menu.

Parameters

None.

Comments

Using the macro in secondary windows isn't recommended.

RegisterRoutine WinHelp macro

RegisterRoutine("DLL-name", "function-name", "format-spec")

The RegisterRoutine macro registers a function within a dynamic link library (DLL). Once registered, the function can be used the same as any other WinHelp macro.

Parameter	Description
DLL-name	The filename of the DLL. The filename must be enclosed in quotation marks. If WinHelp cannot find the library, it displays an error message.
function-name	The name of the function to execute in the designated DLL.
format-spec	A string indicating the format of parameters passed to the function. The format string must be enclosed in quotation marks. Characters in the string represent C parameter types.

Character	Description
u	unsigned short (WORD)
U	unsigned long (DWORD)
i	short int
l	int
s	near char * (PSTR)
S	far char * (LPSTR)
v	void

If the function is used as a help macro, WinHelp ensures that the macro parameters match the parameter types given in this macro.

Comments

You can abbreviate the RegisterRoutine macro as RR.

Example

The following call registers a routine named PlayAudio in MMLIB.DLL. PlayAudio takes arguments of the far char *, int, and unsigned long types:

```
RegisterRoutine("MMLIB", "PlayAudio", "SIU")
```

RemoveAccelerator WinHelp macro

RemoveAccelerator(key, shift-state)

The RemoveAccelerator macro removes the assignment of a help macro to an accelerator key (or key combination) that was previously made using the [AddAccelerator](#) macro.

Parameter	Description																		
key	The Windows virtual-key value. See the Virtual key code topic for a list of virtual-key codes that can be used for this parameter.																		
shift-state	Specifies the combination of ALT, SHIFT, and CTRL keys used with the accelerator represented by one of the following values: <table><tr><th>Value</th><th>Meaning</th></tr><tr><td>0</td><td>None</td></tr><tr><td>1</td><td>Shift</td></tr><tr><td>2</td><td>Ctrl</td></tr><tr><td>3</td><td>Shift+Ctrl</td></tr><tr><td>4</td><td>Alt</td></tr><tr><td>5</td><td>Alt+Shift</td></tr><tr><td>6</td><td>Alt+Ctrl</td></tr><tr><td>7</td><td>Alt +Shift+Ctrl</td></tr></table>	Value	Meaning	0	None	1	Shift	2	Ctrl	3	Shift+Ctrl	4	Alt	5	Alt+Shift	6	Alt+Ctrl	7	Alt +Shift+Ctrl
Value	Meaning																		
0	None																		
1	Shift																		
2	Ctrl																		
3	Shift+Ctrl																		
4	Alt																		
5	Alt+Shift																		
6	Alt+Ctrl																		
7	Alt +Shift+Ctrl																		

Comments

You can abbreviate the RemoveAccelerator macro as RA. WinHelp doesn't return an error when you use this macro with an accelerator for which a macro isn't defined.

Example

The following macro removes a macro from the **Alt+Shift+Ctrl+F4** key combination:

```
RemoveAccelerator(0x73, 7)
```


ResetMenu WinHelp macro

ResetMenu()

Deletes all custom menus and menu items, restores and enables all standard menu items, and restores the item bindings of all standard menu items to their defaults.

Parameters

None.

SaveMark WinHelp macro

SaveMark("marker-text")

The SaveMark macro saves the location of the currently-displayed topic and file and associates a text marker with that location. This allows you to use the GotoMark macro to jump to this location.

Parameter	Description
marker-text	The text marker used to identify the topic location. You must enclose this text in quotation marks, and the name must be unique. If you use the same name for more than one marker, WinHelp uses the most recently entered marker.

Comments

You can use text markers with the [GotoMark](#), [DeleteMark](#), [IfThen](#) , and [IfThenElse](#) macros.

WinHelp deletes all text markers when the user exits WinHelp.

Example

The following macro saves the marker "Managing Projects" in the current topic:

```
SaveMark("Managing Projects")
```

Search WinHelp macro

Search()

The Search macro displays the **Help Topics dialog** with the **Index tab** on top.

Parameters

None.

Comments

WinHelp ignores this macro if it's executed from a secondary window.

SetContents WinHelp macro

SetContents("filename", map number)

The SetContents macro designates a specific topic as the default topic (contents topic) in the specified help file.

Parameter	Description
filename	Specifies the name of the help file that contains the contents topic. The filename must be enclosed in quotation marks. If WinHelp cannot find this file, it displays an error message and doesn't perform the jump.
mapnumber	Specifies the mapnumber of the topic in the specified file. The mapnumber must be defined in the [MAP] section of the project file. If the mapnumber isn't valid, WinHelp displays an error message.

Example

The following example sets the topic mapped to the mapnumber 801 in the PROGMAN.HLP file as the contents topic. After executing this macro, clicking the contents button causes a jump to the topic specified by the map number parameter:

```
SetContents("PROGMAN.HLP", 801)
```

SetHelpOnFile WinHelp macro

SetHelpOnFile("filename")

Parameter	Description
filename	Specifies the name of the replacement "How to Use Help" file. The filename must be enclosed in quotation marks. WinHelp displays an error message if it cannot find this file.

Comments

If this macro appears in a help file topic, the replacement file is set after the macro is executed. If this macro appears in the [CONFIG] section of the project file, the replacement file is set when the help file is opened.

Example

The following macro sets the "How to Use Help" file to MYHELP.HLP:

```
SetHelpOnFile("myHelp .hlp")
```



SetPopupColor WinHelp macro

SetPopupColor(r,g,b)

The SetPopupColor macro sets the background color for popup windows.

Parameter	Description
r	Specifies the red component of the color. This value is an integer in the range 0 to 255.
g	Specifies the green component of the color. This value is an integer in the range 0 to 255.
b	Specifies the blue component of the color. This value is an integer in the range 0 to 255.

Comments

Once this macro is executed, the set color applies to all topics displayed in popup windows.

Example

The following macro sets the background color of all popup windows to pale yellow:

```
SetPopupColor(255,255,225)
```



ShellExecute WinHelp macro

ShellExecute(filename,[options[,show-flag[,operation[,path[,topic-id]]]])

This macro opens or prints the specified file.

Parameter	Description
filename	The name of the file to open or print.
options	Specifies parameters passed to the program when the filename parameter specifies an executable (EXE) file. If the filename parameter specifies a document file, this parameter is empty.
show-flag	Specifies whether the program is shown when it's opened. If the filename parameter specifies a document file, this parameter is zero.
operation	Specifies the operation to perform. This parameter can be open, opencpl, or print. The default operation is open.
path	The default folder containing the file.
topic-ID	The topic ID of the topic to display if this macro fails.



ShortCut WinHelp macro

ShortCut(window-class,program[,wParam[,lParam[,topic-ID]]])

This macro runs a program.

Parameter	Description
window-class	The main window class name of the program. WinHelp uses this class name to determine if the program is already running.
program	The name of the executable to run if the window class name cannot be found. You don't need to include the .EXE extension.
wParam	The first argument to the WM_COMMAND message that is sent to the program. If you don't specify this value, WinHelp starts the program, but doesn't activate it or send a message.
lParam	The second argument to the WM_COMMAND message.
topic-ID	Specifies the topic ID of an alternate topic to display if the program cannot be found. If you don't specify a topic ID, WinHelp displays a message box indicating that the program could not be found.

Sound RHMMPLAY macro

Sound("filename.wav")

The Sound macro allows you to insert a macro link which plays a sound (*.WAV) file inside a WinHelp topic.

Comments

This macro automatically registers [eHelp Corporation's RHMMPLAY.DLL](#), since the macro uses the extended multimedia capabilities in this DLL. For the sound to work in the final help file, you'll need to ship the *.WAV file and install it in the same directory as the HLP file. You'll also need to [ship Blue Sky's RHMPLAY.DLL](#) and install it in the user's system directory (usually, \System or \System32).

Example

The following macro plays the LIONROAR.WAV sound file:

```
Sound("LIONROAR.wav")
```

SoundOnly RHMMPLAY macro

SoundOnly("filename.wav")

The SoundOnly macro allows you to insert a macro link which plays a sound (*.WAV) file without displaying the Video/Sound Player inside the help topic.

Comments

This macro automatically registers [eHelp Corporation's RHMMPLAY.DLL](#), since the macro uses the extended multimedia capabilities in this DLL. For the sound to work in the final help file, you'll need to ship the *.WAV file and install it in the same directory as the HLP file. You'll also need to [ship Blue Sky's RHMPLAY.DLL](#) and install it in the user's system directory (usually, \System or \System32).

Example

The following macro plays the LIONROAR.WAV sound file without displaying the Video/Sound Player:

```
SoundOnly("LIONROAR.wav")
```



TCard WinHelp macro

TCard(command)

The Tcard macro sends a message to the program that WinHelp is opening as a training card.

Parameter	Description																		
command	<p>One of the following commands or a numeric value. If you specify one of the following values, its numeric equivalent is sent as the wParam value of the WM_TCARD message:</p> <table><tr><td>IDABORT</td><td>The user clicked an authorable Abort button.</td></tr><tr><td>IDCANCEL</td><td>The user clicked an authorable Cancel button.</td></tr><tr><td>IDCLOSE</td><td>The user closed the training card.</td></tr><tr><td>IDHELP</td><td>The user clicked an authorable Help button.</td></tr><tr><td>IDIGNORE</td><td>The user clicked an authorable Ignore button.</td></tr><tr><td>IDOK</td><td>The user clicked an authorable OK button.</td></tr><tr><td>IDNO</td><td>The user clicked an authorable No button.</td></tr><tr><td>IDRETRY</td><td>The user clicked an authorable Retry button.</td></tr><tr><td>IDYES</td><td>The user clicked an authorable Yes button.</td></tr></table> <p>If a numeric value is used, the training card program is sent HELP_TCARD_DATA for the wParam parameter, and the numeric value is passed as the lParam value of the WM_TCARD message.</p>	IDABORT	The user clicked an authorable Abort button.	IDCANCEL	The user clicked an authorable Cancel button.	IDCLOSE	The user closed the training card.	IDHELP	The user clicked an authorable Help button.	IDIGNORE	The user clicked an authorable Ignore button.	IDOK	The user clicked an authorable OK button.	IDNO	The user clicked an authorable No button.	IDRETRY	The user clicked an authorable Retry button.	IDYES	The user clicked an authorable Yes button.
IDABORT	The user clicked an authorable Abort button.																		
IDCANCEL	The user clicked an authorable Cancel button.																		
IDCLOSE	The user closed the training card.																		
IDHELP	The user clicked an authorable Help button.																		
IDIGNORE	The user clicked an authorable Ignore button.																		
IDOK	The user clicked an authorable OK button.																		
IDNO	The user clicked an authorable No button.																		
IDRETRY	The user clicked an authorable Retry button.																		
IDYES	The user clicked an authorable Yes button.																		



Test WinHelp macro

Test(test-num)

The Test macro enables an author of a program to test the topics in a help file.

Parameter	Description
test-num	<p>Specifies the testing option. You can use one of the following values:</p> <ol style="list-style-type: none">1 Displays in the current window all the topics in the help file, starting with the current topic.2 Displays in the current window all the topics in the help file, starting with the first topic.3 Continuously displays in the current window all the topics in the help file, starting with the first topic.4 Displays in the current window all the topics in the help file starting with the first topic, and then exits.5 Jumps to all topics specified in the contents (CNT) file, excluding macros.6 Jumps to all topics specified in the contents (CNT) file, excluding macros, and then exits.7 Sets "Help Author=1" (in your WIN.INI file) for the current instance of WinHelp only, and then launches a second instance of WinHelp on the same help file. Both instances of WinHelp are resized so they can appear side-by-side, and both display the first topic in the current help file. Moving through one file (using CTRL+SHIFT commands) updates both windows.



TestALink WinHelp macro

TestALink("keyword[;keyword]")

This macro tests whether an ALink macro has an effective link to at least one topic.

Parameter	Description
keyword	Specifies one or more See Also keywords (A-keywords) to search for. Separate multiple keywords with semicolons. If a keyword contains a comma, the entire keyword string must be enclosed in quotation marks.



TestKLink WinHelp macro

TestKLink("keyword[;keyword]")

This macro tests whether a KLink macro has an effective link to at least one topic.

Parameter	Description
keyword	Specifies one or more index keywords (K-keywords) to search for. Separate multiple keywords with semicolons. If a keyword contains a comma, the entire keyword string must be enclosed in quotation marks.

UncheckItem WinHelp macro

UncheckItem("item-id")

The UncheckItem macro removes the check mark from a menu item.

Parameter	Description
item-id	The menu item to uncheck. The item identifier must be enclosed in quotation marks.

Comments

You can abbreviate the UncheckItem macro as UI.



UpdateWindow WinHelp macro

UpdateWindow([filename>]window-name,topic-ID)

This macro jumps to the topic with the specified topic ID in the specified window and then returns the focus to the window that called the macro.

Parameter	Description
filename	The name of the help file to jump to, if it's not the current help file. This parameter must be followed by a greater than (>) sign.
window-name	The window to jump to.
topic-ID	Specifies the topic ID of the topic to jump to.

Video RHMMPLAY macro

Video("filename.avi")

The Video macro allows you to insert a macro link that plays an animation or video (*.AVI) file.

Comments

This macro automatically registers [eHelp Corporation's RHMMPLAY.DLL](#), since the macro uses the extended multimedia capabilities in this DLL. For the video or animation file to work in the final help file, you'll need to ship the *.AVI file and place it in the same directory as the help file. You'll also need to [ship Blue Sky's RHMPLAY.DLL](#) and install it in the user's system directory (usually, \System or \System32).

Example

The following macro plays the SAIL.AVI video:

```
Video("SAIL.AVI")
```

VideoCaption RHMMPLAY macro

VideoCaption("any caption")

The VideoCaption macro allows you to change the caption displayed on the titlebar of the Video/Sound Player. The default caption is (playing) when the *.AVI file is playing and (stopped) when the file is stopped. Use it in along with the [Video RHMMPLAY macro](#).

Comments

This macro automatically registers [eHelp Corporation's RHMMPLAY.DLL](#), since the macro uses the extended multimedia capabilities in this DLL. For the video or animation file to work in the final help file, you'll need to ship the *.AVI file and install it in the same directory as the help file. You'll also need to [ship Blue Sky's RHMPLAY.DLL](#) and install it in the user's system directory (usually, \System or \System32).

Example

The following macros plays the SAIL.AVI video with the caption "Sailing the Deep Blue Seas":

```
VideoCaption("Sailing the Deep Blue Seas");Video("SAIL.AVI");
```

VideoMenu RHMMPLAY macro

VideoMenu("on\off")

The VideoMenu macro allows you to specify whether the Video/Sound Player menu is displayed or not for your end-user. This menu allows users more control over playing the *.AVI file. Use it in along with the [Video RHMMPLAY macro](#).

Comments

This macro automatically registers [eHelp Corporation's RHMMPLAY.DLL](#), since the macro uses the extended multimedia capabilities in this DLL. For the video or animation file to work in the final help file, you'll need to ship the *.AVI file and install it in the same directory as the help file. You'll also need to [ship Blue Sky's RHMPLAY.DLL](#) and install it in the user's system directory (usually, \System or \System32).

Example

The following macros play the SAIL.AVI video without displaying the Video/Sound Player's menu:

```
VideoMenu("off");Video("SAIL.AVI")
```

VideoPath RHMMPLAY macro

VideoPath (path)

The VideoPath macro allows you to specify a path for a sound (*.WAV) or animation/video (*.AVI) file specified using one of the following RHMMPLAY macros: [Sound](#), [SoundOnly](#), or [Video](#).

Comments

This macro automatically registers [eHelp Corporation's RHMMPLAY.DLL](#), since the macro uses the extended multimedia capabilities in this DLL.

Examples

The following macros play the SAIL.AVI video file located in the \Video folder:

```
VideoPath("\Video");Video("SAIL.AVI")
```

The following macros play the LIONROAR sound file located in the \Sound folder:

```
VideoPath("\Sound");Sound("LIONROAR.WAV")
```



RoboHelpExInitialize WinHelp macro

RoboHelpExInitialize()

This macro is required whenever you use any of the RoboEx32.DLL features, which include [WinHelp 4 Explorer View](#), [Watermarks](#) (for topics or windows), the [Smart See Also button](#), the [JumpHtml WinHelp macro](#) (a jump to an HTML topic displayed in the help window), and HTML topics.

RoboHELP automatically places this macro in the Startup Macros section (**Project tab, Project folder Startup Macros** folder) of your help project whenever you select any of these features.

Comments

This macro works with eHelp Corporation's RoboEx32.DLL and requires the DLL be properly installed on end-users' systems. For more information, refer to [Distributing WinHelp 2000 Help systems](#).



RoboHelpExShowSeeAlso WinHelp macro

RoboHelpExShowSeeAlso()

This macro invokes the Smart See Also button on the selected window of your WinHelp 4 help system. You select this button on the Project Settings: Windows tab.

RoboHELP automatically places this macro in the Startup Macros section (**Project tab, Project folder Startup Macros** folder) of your help project whenever you select this feature.

Parameters

None.

Comments

This macro works with eHelp Corporation's RoboEx32.DLL and requires the DLL be properly installed on end-users' systems. For more information, refer to [Distributing WinHelp 2000 Help systems.](#)



RoboHelpExShowNavPane WinHelp macro

RoboHelpExShowNavPane()

This macro provides the functionality for the WinHelp 4 Explorer view – including the show/hide feature for the navigation pane.

RoboHELP automatically places this macro in the Startup Macros section (**Project tab, Project folder Startup Macros** folder) of your help project whenever you select this feature.

Parameters

None.

Comments

This macro works with eHelp Corporation's RoboEx32.DLL and requires the DLL be properly installed on end-users' systems. For more information, refer to [Distributing WinHelp 2000 Help systems](#).



RoboHelpExWatermark WinHelp macro

RoboHelpExWatermark(filename, location, attributes)

This macro adds a watermark to the background region of the entire project or a topic. RoboHELP allows you to specify a watermark for your help project, or for a specific topic.

Comments

To specify a watermark for the project, use the **Project Settings: WinHelp 2000 tab**. To specify a watermark for a topic, use the **Topic Properties: Watermarks tab**.

Parameter	Description																								
filename	The name of the .BMP image file. This bitmap must be in the help project directory. If it isn't, RoboHELP automatically copies it.																								
location	The position and alignment of the watermark. You can choose from one of the following values: <table><tr><th>Value</th><th>Meaning</th></tr><tr><td>Bottom</td><td>Bottom of the background region.</td></tr><tr><td>Center</td><td>Center of the background region.</td></tr><tr><td>Left</td><td>Left margin of the background region.</td></tr><tr><td>LowerLeft</td><td>Lower left corner of the background region.</td></tr><tr><td>LowerRight</td><td>Lower right corner of the background region.</td></tr><tr><td>Right</td><td>Right margin of the background region.</td></tr><tr><td>Stretch</td><td>Stretches the image to fill the background region.</td></tr><tr><td>Tile</td><td>Repeats the image to fill the background region.</td></tr><tr><td>Top</td><td>Top of the background region.</td></tr><tr><td>UpperLeft</td><td>Upper left corner of the background region.</td></tr><tr><td>UpperRight</td><td>Upper right corner of the background region.</td></tr></table>	Value	Meaning	Bottom	Bottom of the background region.	Center	Center of the background region.	Left	Left margin of the background region.	LowerLeft	Lower left corner of the background region.	LowerRight	Lower right corner of the background region.	Right	Right margin of the background region.	Stretch	Stretches the image to fill the background region.	Tile	Repeats the image to fill the background region.	Top	Top of the background region.	UpperLeft	Upper left corner of the background region.	UpperRight	Upper right corner of the background region.
Value	Meaning																								
Bottom	Bottom of the background region.																								
Center	Center of the background region.																								
Left	Left margin of the background region.																								
LowerLeft	Lower left corner of the background region.																								
LowerRight	Lower right corner of the background region.																								
Right	Right margin of the background region.																								
Stretch	Stretches the image to fill the background region.																								
Tile	Repeats the image to fill the background region.																								
Top	Top of the background region.																								
UpperLeft	Upper left corner of the background region.																								
UpperRight	Upper right corner of the background region.																								
attributes	Indicates where to apply the watermark. <table><tr><th>Value</th><th>Meaning</th></tr><tr><td>Topic</td><td>Applies to the background region of the specified topic only.</td></tr><tr><td>Window</td><td>Applies to the background region of the specified window.</td></tr></table>	Value	Meaning	Topic	Applies to the background region of the specified topic only.	Window	Applies to the background region of the specified window.																		
Value	Meaning																								
Topic	Applies to the background region of the specified topic only.																								
Window	Applies to the background region of the specified window.																								

Comments

This macro works with eHelp Corporation's RoboEx32.DLL and requires the DLL be properly installed on end-users' systems. For more information, refer to [Distributing WinHelp 2000 Help systems](#).



RoboHelpExWatermarkNonScroll

RoboHelpExWatermarkNonScroll(filename, location, attributes)

This macro adds a watermark to the non-scrolling region of the project or topic. RoboHELP allows you to specify a watermark for a your entire help project, or for a specific topic.

Parameter	Description																								
filename	The name of the .BMP image file. This bitmap must be in the help project directory. If it isn't, RoboHELP automatically copies it.																								
location	The position and alignment of the watermark. You can choose from one of the following values: <table><tr><th>Value</th><th>Meaning</th></tr><tr><td>Bottom</td><td>Bottom of the background region.</td></tr><tr><td>Center</td><td>Center of the background region.</td></tr><tr><td>Left</td><td>Left margin of the background region.</td></tr><tr><td>LowerLeft</td><td>Lower left corner of the background region.</td></tr><tr><td>LowerRight</td><td>Lower right corner of the background region.</td></tr><tr><td>Right</td><td>Right margin of the background region.</td></tr><tr><td>Stretch</td><td>Stretches the image to fill the background region.</td></tr><tr><td>Tile</td><td>Repeats the image to fill the background region.</td></tr><tr><td>Top</td><td>Top of the background region.</td></tr><tr><td>UpperLeft</td><td>Upper left corner of the background region.</td></tr><tr><td>UpperRight</td><td>Upper right corner of the background region.</td></tr></table>	Value	Meaning	Bottom	Bottom of the background region.	Center	Center of the background region.	Left	Left margin of the background region.	LowerLeft	Lower left corner of the background region.	LowerRight	Lower right corner of the background region.	Right	Right margin of the background region.	Stretch	Stretches the image to fill the background region.	Tile	Repeats the image to fill the background region.	Top	Top of the background region.	UpperLeft	Upper left corner of the background region.	UpperRight	Upper right corner of the background region.
Value	Meaning																								
Bottom	Bottom of the background region.																								
Center	Center of the background region.																								
Left	Left margin of the background region.																								
LowerLeft	Lower left corner of the background region.																								
LowerRight	Lower right corner of the background region.																								
Right	Right margin of the background region.																								
Stretch	Stretches the image to fill the background region.																								
Tile	Repeats the image to fill the background region.																								
Top	Top of the background region.																								
UpperLeft	Upper left corner of the background region.																								
UpperRight	Upper right corner of the background region.																								
attributes	Indicates where to apply the watermark. <table><tr><th>Value</th><th>Meaning</th></tr><tr><td>Topic</td><td>Applies to the background region of the specified topic only.</td></tr><tr><td>Window</td><td>Applies to the background region of the specified window.</td></tr></table>	Value	Meaning	Topic	Applies to the background region of the specified topic only.	Window	Applies to the background region of the specified window.																		
Value	Meaning																								
Topic	Applies to the background region of the specified topic only.																								
Window	Applies to the background region of the specified window.																								

Comments

For a watermark to display, you must define a non-scrolling region as part of a heading style or topic.

To specify a watermark for the project, use the **Project Settings: WinHelp 2000 tab**. To specify a watermark for a topic, use the **Topic Properties: Watermarks tab**.

This macro works with eHelp Corporation's RoboEx32.DLL and requires the DLL be properly installed on end-users' systems. For more information, refer to [Distributing WinHelp 2000 Help systems](#).

Hash number

A hash number is a unique number generated from a topic ID. It's this number, and not the topic ID itself, that WinHelp uses. You can use the JumpHash macro to jump to a topic identified by a hash number.

Glossary Terms

A
B
C
D
E
F
G
H
I
J
K
L
M
N
O
P
Q
R
S
T
U
V
W
X
Y
Z

A

[A-keyword](#)
[Absolute path](#)
[Absolute positioning](#)
[Absolute URL](#)
[Action Type](#)
[ActiveEdit](#)
[ActiveTest](#)
[Alias](#)
[ALink](#)
[Array](#)
[ASCII](#)
[Associative topic jumps](#)
[Audio-Video Interleave](#)
[Authorable buttons](#)

[Auto size](#)

[Automatic button accelerators](#)

[Auxiliary macros](#)

[.AVI file](#)

B

[Background region](#)

[Baggage](#)

[Base](#)

[Bitmap](#)

[.BMP file](#)

[Book](#)

[Bookmarking](#)

[Broken link](#)

[Browse number](#)

[Browse sequence](#)

[Browse Sequence Editor](#)

[Browse string](#)

[Browser](#)

[Build](#)

[Build expression](#)

[Build tag](#)

[Button macros](#)

[{Button} statement](#)

C

[Clear](#)

[Click](#)

[Client area](#)

[CNT file](#)

[Compile](#)

[Compiler](#)

[Compression](#)

[Contents database file](#)

[Contents file](#)

[Contents tab](#)

[Context ID](#)

[Context number](#)

[Context string](#)

[Context string mapping file](#)

[Context-sensitive](#)

[Control menu](#)

[Current project](#)

D

[Debugging](#)

[Deep menu](#)
[Dialog](#)
[Dialog box](#)
[DLL](#)
[.DOC file](#)
[Document template](#)
[Double-click](#)
[Drag and drop](#)
[Dynamic Link Library \(DLL\)](#)
[Dynamic WYSIWYG](#)

E

[Email](#)
[Embedded graphics](#)
[Explorer View Help window](#)
[Exported document](#)
[External topic](#)
[External Topic hotspot](#)

F

[Find tab](#)
[Folder](#)
[Footnote anchors](#)
[FTG file](#)
[FTS file](#)
[Full-text search](#)
[Full-text search group file](#)
[Full-text search index file](#)

G

[GID file](#)
[Graphical buttons](#)

H

[Hard disk capacity](#)
[HCRTF.EXE](#)
[HCW.EXE](#)
[Help author](#)
[Help authoring](#)
[Help compiler](#)
[Help document](#)
[Help file](#)
[Help graphical buttons](#)
[Help jump](#)
[Help message](#)
[Help popup](#)

[Help project](#)
[Help project file](#)
[Help topic](#)
[Help window](#)
[HELP_CONTEXTMENU](#)
[HELP_FINDER](#)
[HELP_SET_POPUP_POSITION](#)
[HELP_TCARD](#)
[HELP_WM_HELP](#)
[Hotspot](#)
[Hotspot definition](#)
[Hotspot images](#)
[HTML](#)
[HTML Help](#)
[HTML Jump](#)
[HTTP](#)
[Hypergraphic](#)
[Hypermedia](#)
[Hypertext link](#)

I

[Icon](#)
[Image Workshop](#)
[Index](#)
[Index Designer](#)
[Index file](#)
[Index keyword](#)
[Interface](#)
[Internet](#)
[Intranet](#)

J

[Jump](#)

K

[Keyboard macros](#)
[Keyword](#)
[Keyword search](#)
[KLink](#)

L

[Link](#)
[Link View](#)
[Linking macros](#)

M

[Macro](#)
[Macro Editor](#)
[Macro hotspot](#)
[Macro Wizard](#)
[Map file](#)
[Map ID](#)
[Map number](#)
[Memory](#)
[Menu macros](#)
[Mid-Topic IDs](#)
[Mini buttons](#)
[Multi-level index](#)
[Multimedia](#)

N

[Nested information](#)
[Non-scrolling region](#)

O

P

[Page](#)
[Pane](#)
[Parse](#)
[Popup](#)
[Primary Target](#)
[Project Manager](#)

Q

R

[RAM](#)
[Random Access Memory](#)
[Relative path](#)
[Relative positioning](#)
[Relative URL](#)
[ReSize](#)
[Rich text format file](#)
[ROBOHELP.DOT](#)
[.RTF file](#)

S

[Secondary window](#)
[See Also hotspot](#)
[See Also keyword](#)
[Segmented hypergraphic](#)

[Select](#)
[SHED image](#)
[Shell program](#)
[Shortcut buttons](#)
[Smart Index Wizard](#)
[Smart See Also button](#)
[Source files](#)
[Specify](#)
[Standalone Help System](#)
[Style](#)
[System menu](#)

T

[Table of contents](#)
[Template](#)
[Text-marker macros](#)
[Title](#)
[TOC](#)
[TOC Composer](#)
[Tools tab](#)
[Topic](#)
[Topic ID](#)
[Topic List](#)
[Training cards](#)
[Transparent bitmap](#)
[TrueCode](#)
[Tutorial](#)

U

[Uniform Resource Locator](#)
[URL](#)
[User](#)

V

W

[Watermark](#)
[.WAV file](#)
[WebHelp](#)
[What's This Help Composer](#)
[What's This? Help](#)
[Window macros](#)
[Windows Help Viewer](#)
[WinHelp 2000](#)
[WinHelp 3](#)
[WinHelp 4](#)

[WinHelp API](#)

[WYSIWYG](#)

X

Y

Z

Absolute path

The full location of a file. It includes the drive letter it is stored on, such as "C," the folder name, subfolder name(s), and file name.

For example: C:\Office\Documents\Files.doc is an absolute path. See also [Relative path](#).

Absolute positioning



Win 95 Fixes the size and location of a WinHelp 4 window regardless of the end user's screen resolution. This method uses absolute pixel coordinates based on a 1024x1024 pixel coordinate grid. Windows positioned with absolute positioning do not move from their fixed location – even when the user chooses a new screen resolution. The other method of positioning (placing) WinHelp windows is called [relative positioning](#).

Absolute URL

The full Internet address of a Web page. It includes a protocol (such as "http://"), network location, path, and file name. For example, <http://www.blue-sky.com> is an absolute URL. See also [URL](#).

Action Type

One of six possible links used in WinHelp systems, which identifies the action initiated when users click the hotspot, button, or area on a hotspot SHED image.

Action types include:

- [Jumps](#)
- [Popups](#)
- [External Topic jumps](#)
- [HTML jumps](#)
- [See Also hotspots](#)
- [Macro hotspots](#)

ActiveEdit

This RoboHELP compile option attaches the ActiveEdit button to the windows button bar in the compiled Help file. This button allows you to quickly jump from viewing the compiled Help file directly into the source topic, so you can make any changes or corrections on the spot.

ActiveEdit is designed as a tool used during the Help development testing and debugging process.

ActiveTest

This RoboHELP feature allows you to view a Help topic as it will appear in the compiled Help system – without compiling.

ActiveTest displays the topic based on the following criteria:

- If the topic hasn't been compiled into the Help file or if the topic has changed since the last compile, ActiveTest simulates compiling the topic and then display its in the selected ActiveTest window.
- If the topic hasn't changed since the last time you compiled the Help file, ActiveTest displays the topic in the current compiled Help file.

A-keyword




(Also known as See Also keyword.) A type of keyword that allows you to create topic links to display See Also or related topics in WinHelp 4 Help systems. A-keywords are associated with topic properties and are represented by the **A** topic footnote. A-keywords don't appear in the Help index and are basically invisible to end users.

In RoboHELP, you can specify either the ALink WinHelp macro or the See Also hotspot action type, or select WinHelp 2000's Smart See Also button to implement A-keywords.

Alias

A pseudonym for a Topic ID used for two reasons:

- **Fix broken links.** If a Topic ID has changed or is missing, you can create an Alias to redirect the link to the new Topic ID.
- **Connect Map IDs to context-sensitive help topics.** There are two main reasons to use Aliases in context-sensitive help:
 - You need to connect an existing topic to a Map ID. For example, you and your programming team are developing simultaneously. When you receive the Map files from your programmer, you connect the Map IDs to your existing topics by creating an Alias for each topic. (You alias the Map string to the appropriate Topic ID.)
 - You need to connect more than one Map number to the same topic. For example, your programming team provides you with seven Map IDs for a dialog. But you've planned one topic to cover the entire dialog, so you need to connect all seven Map IDs to the same topic.

RoboHELP uses the  symbol to represent Aliases.

ALink

(Also known as See Also hotspots.) A specific WinHelp linking macro that searches for a See Also keyword (or A-keyword) and then lists all the topics containing that See Also keyword.

RoboHELP's See Also hotspot Action Type and WinHelp 2000's Smart See Also button automatically use the ALink macro to implement See Also A-keywords.

See also [KLink](#).

Array

Simple, common type of structured data within a program. Contains a group of items that are all of the same type and are directly accessed through the use of an array index.

ASCII

Stands for American Standard Code for Information Interchange. An ASCII text file is a standard text file that can be viewed across operating systems. In Windows, you can view and edit ASCII files using Notepad.

Associative topic jumps



Dynamic hotspots created by the Help author, usually containing a list of See Also or related topics.

Audio-Video Interleave

(.AVI) A Windows multimedia file format for sound and moving pictures that uses the Microsoft RIFF (Resource Interchange File Format) specification.

This is the standard video format for Microsoft Windows and WinHelp.

Auto size

Help window option that automatically adjusts the height of a secondary Help window based on the length (content length) of the topic it displays.

Authorable buttons



Authorable buttons allow you to create customized text buttons using the WinHelp 4 {button} command. If you include text, the button is the size of the specified text. If you don't include text, the button is 12x12 pixels (the same size as a mini button).

Automatic button accelerators

Key characters specified by the Help author that activate button functions. For example, you can specify that the C character activates the Contents function.

Auxiliary macros

WinHelp macros that provide access to programs and functionality not usually available in typical WinHelp. For example, RoboHELP's JumpHTML and Inet WinHelp macros allow you to add links to HTML topics and Web site pages in your Help system.

.AVI file

(Audio-Video Interleave) This is the standard video format for Microsoft Windows and WinHelp.

Background region

Area of the WinHelp window below the titlebar that contains the body of the topic – usually the largest part of the Help window. If a [non-scrolling region](#) is set, the background region begins below it.

Baggage

WinHelp's internal filing system used to store multimedia, images, and other types of miscellaneous files. By using a [DLL](#), you can use baggage to store and retrieve any kind of file, which can simplify distributing Help systems and prevent end users from accidentally deleting needed files.

In RoboHELP Explorer's Project Manager (**Project tab**), you'll find all baggage items stored in the Baggage folder under the main Project folder.

Base

The Base: statement in the HPJ defines the name of the compiled Help (HLP) file. It's used with CNT files to establish the base Help file. This statement can also include the name of the default window used to display topics in the Help system.

Bitmap

Windows bitmap files (also called raster graphics) break images into a grid of equally-sized pieces called pixels. When you create or edit bitmaps, you work with the individual pixels.

Bitmaps are the most popular and most supported image format in Windows operating systems and are the general, all-purpose Help image, even though they can be large and difficult to resize. They use the .BMP file extension.

.BMP file

(Bitmap) Windows bitmap files (also called raster graphics) break images into a grid of equally-sized pieces called pixels. When you create or edit bitmaps, you work with the individual pixels.

Bitmaps are the most popular and most supported image format in Windows operating systems and are the general, all-purpose Help image, even though they can be large and difficult to resize. They use the .BMP file extension.

Book

A category of topics within a Help system. The book icon is used in the Help system's table of contents to organize topics and provide a hierarchical view of the Help system. It's similar to the concept of grouping information into chapters in a printed book.

Bookmark

1. **Word.** An item or location in a document that you identify and name for future reference. Use bookmarks to jump to a specified location, create cross-references, mark page ranges, and so on.
 2. **HTML Help.** A named area in an HTML Help topic that can be used to jump to different sections with the same or in other HTML Help topics. Bookmarks in HTML topics are the equivalent of WinHelp's Mid-Topic IDs.
 3. **Compiled WinHelp (HLP) files.** In compiled WinHelp, bookmarks act as electronic placeholders that allow users to quickly return to specified Help topics. The Bookmark menu allows users to mark Help topics with a bookmark, which places the topic title on the Bookmark menu in an alphabetical list for easy selection.
- (For information about jumping to a specific location inside a topic, see [Mid-Topic IDs](#).)

Broken link

A link that doesn't correctly point to a topic or link destination. Broken links usually indicate a missing topic or a changed Topic ID.

Browse number

A unique number (an integer) assigned to a topic along with a browse string to identify the topic's order in the browse sequence.

Browse sequence

Browse sequences allow users to move forward and backward through a series of topics in an order defined by the Help author. They consist of a browse string and a browse number and are part of an included topic's properties. Browse sequences are represented by the + (plus symbol) footnote. Each topic can only be part of one browse sequence.

There are two types of browse sequences:

- **Topic order.** Topic order browse sequences determine browse order by how topics are physically placed (their physical order) in a Help document.
- **Numbered order.** Numbered order browse sequences determine browse order by the assigning a browse string name and a unique browse number. (The browse number determines the placement of the topic in the browse sequence.)

The easiest way to create and manage browse sequences in RoboHELP is to use the [Browse Sequence Editor](#).

Browse Sequence Editor

RoboHELP's Browse Sequence Editor allows you to visually create and maintain foolproof [browse sequences](#) for all your Help projects. When you open the Browse Sequence Editor, you see a visual picture of the most current browse sequences.

Using the Browse Sequence Editor you can:

- Create new browse sequences
- Automatically create browse sequences (based on table of contents topic order)
- Rename browse sequences
- Add available topics to browse sequences
- Drag and drop topics from one browse sequence to another
- Remove topics from browse sequences
- Remove browse sequences

The Browse Sequence Editor uses numbered browse sequences. Whenever you use the Browse Sequence Editor to modify or create browse sequences, those sequences are updated in numbered order. All you need to do is place topics into a browse sequence in the order you want them to appear and save your changes. The Browse Sequence Editor takes care of the behind-the-scenes work for you, updating topics and browse sequences.

Browse string

A unique name given to a browse sequence. RoboHELP suggests browse string names based on the name of the current Help document, but you can select any name you want.

You can create browse sequences across multiple documents by assigning topics to the same browse string.

Browser

Software that runs on the client-side (Help authors' and end-users' systems) designed to communicate with Web and intranet servers and interpret the data received from them.

Build

In Help authoring, the process of compiling a Help file (.HLP). This is also a common term used in software development to refer to the compiled Help file or to compiled software code during the development cycle.

See also [compile](#).

Build expression

Advanced way to indicate which topics (based on [build tags](#)) are included or excluded from a compiled Help file during the compile process. Build expression syntax is specified on the **Project Settings: Advanced tab** and must follow exact build syntax parameters.

Build tag

Advanced way to control which topics are part of a compiled Help file. Help authors mark topics with build tags and then use a [build expression](#) to specify whether to include or exclude topics using that build tag. By using build tags, you can design and create different Help files from one set of source files.

Build tags are represented by the * (asterisk symbol) topic footnote and are part of a topic's properties.

Button macros

Used to provide access to standard Help window buttons, allow you to create new window buttons, or modify window button functionality.

{Button} statement

Creates a button in a Help topic that runs the selected macro when the user clicks on it.

Clear

To unmark an option, button, check box, or item in a list. (Opposite of [select](#).)

Click

An instruction to click the left mouse button once.

Client area

(Now called background region.) Area of the WinHelp window below the titlebar that contains the body of the topic – usually the largest part of the Help window. If a [non-scrolling region](#) is set, the background region begins below it.

CNT file

(Also known as the Contents file.) An ASCII text file that contains the table of contents for the Help file (.HLP). CNT files use the books and pages metaphor to represent the Help file's organization – books allow you to group and organize topics; pages provide access to the Help file's topics.

The Contents (CNT) file serves several purposes:

- Provides access to the Contents tab in the Help file.
- Combines topics from multiple Help systems into a "Master" Help file.
- Combines indexes and See Also keyword groups from multiple Help systems into a "Master" Help file.
- Allows users to navigate to Web sites, product demonstrations, related applications, tool wizards, and even online tutorials.
- Allows users to print topics from the compiled Help file in groups.

Compile

The process of combining all project source files into one, compiled Help file. When you compile WinHelp, RoboHELP uses the selected WinHelp compiler (based on your project's [Primary Target](#) selection).

Compiler

WinHelp compilers process Help source files and components (topics, images, windows, and so on) located in a Help project into one compiled Help file (.HLP). The compiler used depends on the primary target operating system of the Help system.

Compression

A method used to decrease the size of a file. The WinHelp compiler provides several compression settings that can significantly reduce the size of the Help file (.HLP). The greater the compression, the longer it takes to compile the Help file.

Contents file

(Also known as the CNT file.) An ASCII text file that contains the table of contents for the Help file (.HLP). Contents files use the books and pages metaphor to represent the Help file's organization – books allow you to group and organize topics; pages provide access to the Help file's topics.

The Contents (CNT) file serves several purposes:

- Provides access to the Contents tab in the Help file.
- Combines topics from multiple Help systems into a "Master" Help file.
- Combines indexes and See Also keyword groups from multiple Help systems into a "Master" Help file.
- Allows users to navigate to Web sites, product demonstrations, related applications, tool wizards, and even online tutorials.
- Allows users to print topics from the compiled Help file in groups.

Contents database file

(GID file) The hidden database file generated dynamically when a WinHelp 4 Help system is viewed in Windows 95 or higher.

Contents tab

One of the tabs in the compiled Help system, the Contents tab contains the table of contents hierarchy for the Help system.

Context ID

See [Map ID](#).

Context number

See [Map number](#).

Context-sensitive

Help topics users access from within an application, usually at a dialog box. Context sensitive help topics provide information relevant to the user's particular location inside the software program.

Context string

See [Map string](#).

Context string mapping file

See [Map file](#).

Control menu

A window menu that contains commands to resize, close, or switch the window. You can access the Control menu by clicking the Control menu button or by typing **ALT+SPACE**.

Current project

The Help project associated with a given session of RoboHELP.

Debugging

The process of removing errors from a Help system. Debugging involves reading and interpreting compiling messages, modifying the project source files, and recompiling the project until there are no such errors.

Deep menu

A set of related menus presented as a series of user choices. For example, the first menu might prompt users to select between printing, copying, or pasting, while subsequent menus provide options specific to their first choice.

Dialog

Contain command buttons and various kinds of options through which users can carry out a particular command or task. For example, in the **Save As** dialog box, the user must indicate in which folder and under what name the project or document should be saved.

Dialog box

Contain command buttons and various kinds of options through which users can carry out a particular command or task. For example, in the **Save As** dialog box, the user must indicate in which folder and under what name the project or document should be saved.

DLL

Dynamic Link Library. File that contains executable content that can be loaded by an application to complete or extend its functionality. An "application" can actually be an executable (.EXE) combined with one or more DLLs. DLLs can also contain images, strings, or other resources that can be used by the application. Multiple applications can use the same .DLL simultaneously.

.DOC file

A file extension that identifies document files formatted for a word processor. This is the default file extension for Microsoft Word document files.

Document template

In Microsoft Word, document templates are special types of documents that store styles, macros, AutoText entries, and custom toolbar, menu, and shortcut key settings. Word's NORMAL.DOT template is a global template that applies to all Word documents – Help documents or not.

RoboHELP integrates Help authoring functionality with Word through the ROBOHELP.DOT template. This global template contains the commands, toolbars, menus, and shortcut keys necessary to complete authoring tasks, as well as pre-defined styles to assist you in formatting your Help topics.

Double-click

To quickly click the left mouse button two times.

Drag and drop

Program feature where a topic, file, or item can be dragged from one component to another to create a desired effect. For example, you can drag a topic from the Topic List into the TOC Composer to add a new page to the table of contents.

Dynamic Link Library (DLL)

(DLL) File that contains executable content that can be loaded by an application to complete or extend its functionality. An "application" can actually be an executable (.EXE) combined with one or more DLLs. DLLs can also contain images, strings, or other resources that can be used by the application. Multiple applications can use the same .DLL simultaneously.

Dynamic WYSIWYG

(What You See Is What You Get) The development environment option that allows you to view everything in a Help project as if the Help system was already compiled. For example, images in WYSIWYG mode display as they will in a compiled Help system instead of by reference code.

The other development environment option is [True Code](#).

Email

Electronic mail. An interactive method for sending electronic messages to mail recipients.

Embedded graphics

A visible image that is stored in the document as a graphic file instead of being referenced from another file location.

Explorer View Help window

Tri-pane WinHelp window similar to the HTML Help window. The Explorer View Help window is enabled with [WinHelp 2000](#) and gets its settings (size, location, window buttons) from the Main window, as defined in the current Help project.

The three panes (tri-pane) of the Explorer View Help window are:

- **Button bar.** The button bar displays across the top of the Explorer View Help window. You can select or create almost any window buttons on the Main window (used to create the Explorer View Help window). In the compiled Explorer View Help system, the Help Topics button is a 3-D button. Users can show or hide the Navigation pane by toggling the Help Topics button. For best results, don't select the Contents, Index, and Find buttons on the Main Help window – WinHelp 2000 takes care of these functions for you in the Explorer View Help window. (Clear these options, if necessary, and recompile.)
- **Navigation pane.** The navigation pane takes up approximately one-third of the left side of the Explorer View Help window (or one-third of the total window size). It contains three tabs – the Contents tab, the Index tab, and the Search tab. The Contents tab uses the books and pages metaphor to display contents and remains synchronized with the topic displayed in the Topic pane. The Index tab provides access to the Help index, and the Search tab provides access to full-text search. Users don't have to leave the Contents, Index, or Search tabs to see their selected topic – it appears in the Topic pane on the right.
- **Topic pane.** Help topics appear on the right side of the Explorer View Help window. If the topic extends beyond the window, scroll bars allow users to see the rest of the topic.

Exported document

RoboHELP can convert Help documents to printed documents. The resulting printed document can be referred to as an "exported document."

External topic

A topic not part of the current Help project. External topics can be WinHelp topics in other Help systems or HTML topics in compiled HTML Help systems (CHM files).

External Topic hotspot

Jump where the destination is a topic in another Help system. You can jump to WinHelp topics or HTML topics contained in compiled HTML Help systems (CHM files).

Find tab

One of the tabs (sections) under the Help Topics window, the Find tab provides a full-text search for all keywords within the Help system.

Folder

A method for naming and organizing files and projects. Sometimes folders indicate a location – such as the Help project directory.

Footnote anchors

Characters that appear at the beginning of each topic in a Help document (i.e. *, #, \$, K, A, +, !). Each footnote represents one or more attributes of a Help topic to the Help Compiler.

FTG file

See [Full-text search group file](#).

FTS file

See [Full-text search index file](#).

Full-text search group file

(FTG file) Generated dynamically by Windows 95 Help, the FTG file provides for the full-text search by group feature.

Full-text search index file

(FTS file) Generated dynamically by Windows 95 Help, the FTS file provides for the full-text search by keyword feature.

Full-text search

The ability to search an entire Help system for a particular word or keyword.

GID file

The hidden database file generated dynamically when a WinHelp 4 Help system is viewed in Windows 95 or higher.

Graphical buttons

Three-dimensional topic buttons that can contain two different images – one on the face of the button and another when users click the button. The button size adjusts automatically according to the size of the image on the face of the button.

You assign an [Action Type](#) to the button that executes when users click the button inside a Help topic.

Hard disk capacity

The amount of memory available on a PC fixed media storage device, usually measured in Mega (million) bytes (Mb) of memory.

HCRTF.EXE

The WinHelp 4 compiler. Compiles Help projects into WinHelp 4 Help files.

HCW.EXE

The user interface for the WinHelp 4 Compiler. (Uses HCRTF.EXE to compile WinHelp 4 projects into WinHelp 4 Help files.)

Help file

(Sometimes called Help system.) A hypertext file created by processing Help project files (topics, images, multimedia, and so on) using a compiler into one file. WinHelp uses a [WinHelp compiler](#) to create a Help file with the .HLP extension.

Help window

Help windows display topics in the compiled Help system. WinHelp windows are compatible with Microsoft Windows operating systems. Help windows contain navigational buttons to help users move around the Help system.

In WinHelp, there is one Main window and custom windows called secondary windows. You can customize the appearance and attributes Help windows to fit into your overall Help system design.

HELP_CONTEXTMENU

Displays context-sensitive Help when the user clicks the right mouse button.

HELP_FINDER

Displays the Help Topics dialog box with the same tab selected (Table of Contents, Index, or Full Text Search) as the last time you used Help.

HELP_SET_POPUP_POSITION

Sets the position for the next popup window.

HELP_TCARD

Indicates that a command is for a training card instance of WinHelp.

HELP_WM_HELP

Displays context-sensitive Help when the user presses the F1 or (?) button.

Help author

The person creating the Help system.

Help authoring

The process of planning, developing, and organizing information to produce a Help system.

Help compiler

WinHelp compilers process Help source files and components (topics, images, windows, and so on) located in a Help project into one compiled Help file (.HLP). The compiler used depends on the primary target operating system of the Help system.

Help document

A Microsoft Word document (DOC file) that contains Help topics and topic content. Help documents are and associated with and part of the RoboHELP project source files. Help authors work develop topics and content using Help authoring features integrated into Microsoft Word. RoboHELP saves the Word DOC file and an RTF file – the RTF file is used by the WinHelp compiler to create the final compiled Help (HLP) system.

Help graphical buttons

Three-dimensional topic buttons that can contain two different images – one on the face of the button and another when users click the button. The button size adjusts automatically according to the size of the image on the face of the button.

You assign an [Action Type](#) to the button that executes when users click the button inside a Help topic.

Help jump

Jumps link text or images to topics. When users click on the hotspot text, image, or button, the Help system "jumps" from the current topic to the destination topic. You can create jumps to any new or existing topic in your Help system. When you use a jump, you choose which window the destination topic displays in – the Main window or any secondary window defined in your Help project. If the destination topic has a defined window, RoboHELP suggests that window by default.

Help message

Information displayed on the screen to assist the user in a particular task.

Help popup

Popup links display the destination topic in a small, temporary window that "pops up" on the screen. This popup window is a small, self-sizing window. While popup links are mainly used for popup topics, you can display any new or existing topic in your Help system using a popup link.

Popups are often used to provide additional information without interfering with the continuity of the current screen of information.

Help project

A collection of files that bring together all the elements of a Help system. The project file contains information about the content and properties of your Help system:

- **Content.** Help project files assist you in managing and organizing the files and components that make up your Help system. It contains information about the location of your topics, images, and other files.
- **Navigation.** Projects also consist of tables of contents and indexes – methods used to navigate (or move around) the Help system.
- **Properties.** The project file also contains the settings information – such as window location – that makes your Help system look the way you designed it. When you create a new project, the basic (or default) settings are already made for you. You can modify these settings – such as adding custom windows or changing the size or location of a window – to suit your preferences and design needs.

Help project file

(Also known as HPJ) An ASCII text file that stores information about the location and contents of a particular Help project. The Help project file has the same name as the Help project with an .HPJ extension.

Help topic

The basic unit of information – the building block of a Help system. Topics are the message – the information you want to communicate through your Help system. When users view your Help system, they get information (help) by finding and reading specific topics.

Hotspot

Electronic links that allow users to navigate Help systems. Hotspots can be text, buttons, or areas on images that initiate a specific [Action Type](#) when clicked. Hotspots allows a non-linear way of navigating information – closer to the user's thought processes.

Hotspots got their name because the text or image is formatted in a special way: this special formatting signals users the text is "hot" – or clickable. The action of clicking on the link activates the link.

Hotspot images

Any image that contains one or more "clickable" [hotspot](#) areas.

More commonly, hotspot SHED images contain multiple hotspots and use a unique image file format with the .SHG extension.

Hotspot images are commonly used to explain screen shots, flowcharts, diagrams, illustrations, and so on.

Hotspot text

Text associated with a [hotspot](#) – or the "hot" text users click on to execute the hotspot's action.

HTML

Hypertext Markup Language. A set of tags used to mark the structural elements of text (ASCII) files. HTML files include tags that create hyperlinks to other documents on the Internet or documents displayed using Internet browser functionality. HTML can be regarded as a programming language because not only does it define what part each piece of text plays in a topic, but it also specifies to load other files, run scripts, and find and run external programs.

HTML Help

A Help technology developed by Microsoft to create and display Help topics in HTML format.

The Microsoft HTML Help format was introduced in August 1997. Microsoft announced that HTML Help will be its Help standard beginning with Windows 98 and NT 5.

Like WinHelp, Microsoft HTML Help is compiled into one file – the .CHM file. This Help format is used primarily as application Help, but you can create just about any type of Help system as long as your audience uses a 32-bit Windows operating system (Windows 95 and later) and has access to Internet Explorer 4. (While Microsoft HTML Help can run on Internet Explorer 3, Internet Explorer 4 is recommended.)

HTML Jump

Jumps where the destination is an HTML page on the Internet/intranet or an HTML topic in uncompiled HTML Help systems.

In RoboHELP, you can display the HTML page inside your WinHelp system's Help window, so users have extended access but remain "inside" your Help system. Or you can choose to launch your users' default Internet browser to display the HTML page separately inside their preferred browser's window.

HTTP

Hypertext Transfer Protocol. The communications protocol on which the Web is based. HTTP sets rules for how information is passed between the server and the browser software. Help authors deal with HTTP when they link to URLs on Web sites.

Hypergraphic

See [Hotspot image](#).

Hypermedia

A term used to represent the computer media and its possibilities for user interaction.

Hypertext link

Electronic links that allow users to navigate Help systems. Hypertext is text that initiates a specific [Action Type](#) when clicked. Hypertext allows a non-linear way of navigating information – closer to the user's thought processes.

See also [Hotspot](#) and [Link](#).

Icon

A picture or symbol that represents an object, task, command, or choice users can select by pointing and clicking with a mouse.

Image Workshop

The Image Workshop (Image Workshop tab) allows you to work with the images in your Help file and accomplish the most common image tasks:

- Create hotspot images (SHED images) and work with hotspot image properties (like replace the base image in a hotspot SHED image)
- Resize images (only available with RoboHELP Office)
- Crop buttons, menus, toolbars, and more from an image
- Copy and paste images from the Clipboard
- Change the color depth of images
- Edit the selected image in your preferred paint/image editor and import the changes back to RoboHELP
- View images and image properties
- Create new images by dragging image files from your desktop or Windows Explorer into the Image Workshop and saving them as new image files.

Index

A multilevel list of keywords or phrases that help users find topics within a Help system.

Index Designer

Left-pane component of RoboHELP Explorer that looks just like the index in a compiled Help system. Use it to add, edit, and delete Index keywords (K-keywords) to your Help system.

Index file

(Also called the FTS file) Generated dynamically by the WinHelp 4 viewer, the Index file provides for the full-text search by keyword feature.

Index keyword

Also K-keyword. A key word or phrase in the Help system's index. These are the words and phrases users are most likely to think of when searching for information. Help authors associated Index keywords with topics to help users find the topics they're looking for in the Help system.

Interface

The elements of an application that allow the user to interact with the computer.

Internet

The global computer network, composed of thousands of Wide Area Networks (WANs) and Local Area Networks (LANs) that uses TCP/IP to provide world-wide communications. The World Wide Web runs on the Internet.

Intranet

A computer network available only to members within an organization as a company-wide Web resource.

Jump

Jumps link text or images to topics. When users click on the hotspot text, image, or button, the Help system "jumps" from the current topic to the destination topic. You can create jumps to any new or existing topic in your Help system.

Keyword

Also Index keyword or K-keyword. A collection of words or phrases defined by the Help author, which comprise the Help system's index. These keywords are associated with topics and are designed to be the words and phrases users will use when searching for information.

Keyboard macros

WinHelp macros used to assign or remove keyboard accelerators.

Keyword search

A method of searching online documentation in which a user enters a keyword or Boolean string to get a list of related topics.

KLink



Win 95 WinHelp 4 macro link used in the index to search for a particular Index keyword. If found, the KLink displays a list of all topics containing that keyword. (This topic list is dynamically generated at the time the link is executed.)

Link

(Also called hotspots.) Electronic links that allow users to navigate Help systems. Links can be text, buttons, or areas on images that initiate a specific [Action Type](#) when clicked. They allow a non-linear way of navigating information – closer to the user's thought processes.

The Link statement also defines the Help files to be included in the [ALink](#) and [KLink](#) keyword searches.

See also [Jump](#), [Popup](#), [Macro](#), [External Topic Hotspot](#), [See Also Hotspot](#), or [HTML Jump](#).

Link View

Right-pane RoboHELP Explorer tab that graphically displays topic links and browse sequences for the selected topic. When you open Link View, the selected topic appears in the middle of the Link View, outlined in blue. Topics that have links to this topic display on the left (inbound links). Links from this topic to other topics display on the right (outbound links).

Linking macros

WinHelp macros which create links to specific Help topics.

Macro

Special scripts that enable Help authors to customize and closely control their Help system's functionality. All hotspots use macros to accomplish links, and there are several linking macros. RoboHELP makes use of the more common linking WinHelp macros to link text and images and assists you in completing the necessary macro syntax and selecting macro parameters. Sometimes, though, to accomplish a special link – such as a link that starts another program – you may need to select your own macro. RoboHELP's Macro Wizard and Editor provide the tools you need and simplify entering macro syntax and parameters.

Macro Editor

RoboHELP component that allows you to edit macro statements by hand or use the Macro Wizard to assist in building macro syntax.

Macro hotspot

Macros are special scripts that enable Help authors to customize and closely control their Help system's functionality. All hotspots use macros to accomplish links, and there are several linking macros. RoboHELP makes use of the more common linking WinHelp macros to link text and images and assists you in completing the necessary macro syntax and selecting macro parameters. Sometimes, though, to accomplish a special link – such as a link that starts another program – you may need to select your own macro. RoboHELP's Macro Wizard and Editor provide the tools you need and simplify entering macro syntax and parameters.

Macro Wizard

RoboHELP component that simplifies adding macros to your Help projects and topics. The Macro Wizard leads you through every step of choosing macros and macro parameters and completes the necessary macro syntax automatically, based on your choices.

Map file

A text file containing [Map IDs](#) used to provide context-sensitive links from applications to contextual Help topics. Map files are written into the Help project, referenced in the [MAP] section, and generally have an .H, .HH, or .HM file extension. In RoboHELP, Map files can be generated by Help authors or programmers.

Map ID

Map IDs provide context-sensitive links from applications to help topics and contain two basic elements:

- **Map string.** The Map string is a "string" of characters that uniquely identify a Map ID. (Similar to a Help topic's Topic ID.)
- **Map number.** The Map number is a unique number (an integer) that identifies the Map ID.

Map IDs are contained in [Map files](#).

Map number

A unique integer that identifies a context-sensitive topic. Map numbers provide context-sensitive links between help topics and applications. Map numbers, together with Map strings, make up [Map IDs](#).

Map numbers can be created and assigned by both programmers and Help authors.

Map string

A "string" of characters that uniquely identify a [Map ID](#), similar to a Help topic's Topic ID, used to connect context-sensitive help topics to applications.

Memory

The part of a computer that stores information or instructions. Memory can be RAM (Random Access Memory) or ROM (Read Only Memory), and is usually measured in Kilobytes (thousand) or Megabytes (million) bytes of memory.

Menu macros

Macros that provide access to standard Help menu items, create new menus and new menu items, or modify menus and menu items.

Mid-Topic IDs

Uniquely identify a marked area inside a topic. Mid-Topic IDs allow users to jump to different sections within the same WinHelp topic – like a bookmark or inter-topic jump. They're helpful to break up long topics, so users go right to the information they want.

Mini buttons

A type of topic button. Mini buttons are small buttons that automatically size to 12x12 pixels. Text doesn't appear on the face of the button, but rather as a hotspot next to the button. Users can select either the button or the hotspot text next to the button to launch the action associated with the button.

Multi-level index

A Help index that contains two or more levels of keywords: top-level keywords and subkeywords.

For example:

Creating	keyword
Aliases	subkeyword
Topics	subkeyword

Multimedia

Using more than one medium in a Help system – audio, video, or animation clips. Allowable multimedia formats are sound (.WAV) and video/animation (.AVI).

In RoboHELP, you choose from two ways to include multimedia: Microsoft's {mci} command (WinHelp 4 only) or RoboHELP's RHMMPLAY multimedia extensions, which allows you to include multimedia in both WinHelp 3 and WinHelp 4 Help systems.

Nested information

See [Context-sensitive](#).

Non-scrolling region

Unique to Help windows, this is an area at the top of a WinHelp topic that remains in place when the rest of the topic scrolls. Non-scrolling regions are separated from the rest of the topic by a line.

Help authors must add (or set) the non-scrolling region for topics, since they don't automatically appear. In RoboHELP, you use Word's Keep with Next paragraph attribute to mark non-scrolling region text.

You can quickly add non-scrolling regions to topics two ways:

- **Individual topic.** Select the topic title and any text to include in the non-scrolling region, then open RoboHELP's Set Non-Scrolling Region dialog and set the non-scrolling region. (This automatically applies Word's Keep with Next paragraph attribute to the selected text.)
- **Customize a heading style.** Typically, non-scrolling regions include just the topic title, so you can modify the Heading 1 style (or other topic title style) to include the Keep with Next paragraph attribute. This automatically applies a non-scrolling region to every topic created with customized topic heading style.

Popup windows don't support non-scrolling regions, so topics with non-scrolling regions will only display the topic title in the popup window.

Page

Icon used in the Help system's table of contents to represent a single piece of information or function within the Help system. Usually pages display individual Help topics (local or external), but pages can also display Internet/intranet pages or execute macro functions – like launch other applications, tutorials, or Help systems.

Pane

A frame used to display a left- or right-hand RoboHELP Explorer program component, such as the Project Manager (Project tab), TOC Composer (TOC tab), Index Designer (Index tab), Tools tab, Image Workshop, Topic List, or Link View.

Parse

To turn Help document footnote codes into meaningful references in other Help project files.

Popup

Popup links display the destination topic in a small, temporary window that "pops up" on the screen. This popup window is a small, self-sizing window. While popup links are mainly used for popup topics, you can display any new or existing topic in your Help system using a popup link.

Popups are often used to provide additional information without interfering with the continuity of the current screen of information.

Primary Target

Determines which WinHelp compiler is used to create the final compiled Help (HLP) file – WinHelp 3 or WinHelp 4.

Project Manager

(Project tab) Left-hand RoboHELP Explorer program component that displays an expandable/collapsible hierarchical view of folders that organize topics and files into manageable groups.

RAM

See [Random Access Memory](#).

Random Access Memory

(RAM) The primary memory in a computer, where all software is run.

Relative path

A path that is implied by the current working directory. When a user enters a command that refers to a file, if the full pathname is not entered, the current working directory becomes the relative path of the file referred to. In WinHelp projects, the path is relative to the Help project folder (directory containing the WinHelp HPJ Help project file and other source files.)

For example: If you create an Images folder named "Images" as a subfolder of your project directory, the relative path for the images in that folder is \project folder\Images\bitmap.bmp

See also [Absolute path](#).

Relative positioning

Determines the size and position of a Help window based on a percentage of the display screen.

The other method of positioning (placing) WinHelp windows is called [absolute positioning](#).

Relative URL

Relative Uniform Resource Locator. A form of [URL](#) in which the domain and some or all directory names are omitted, leaving only the document name and extension (and perhaps a partial list of directory names). The indicated file is found in a location relative to the pathname of the current document

ReSize

An eHelp tool distributed with RoboHELP Office that allows Help authors to resize images and change color depth. ReSize functionality has been integrated into Image Workshop

Rich text format file

(RTF file) A language for describing text that contains multiple fonts, images, and other visual objects. Rich text format is the source format for Windows WinHelp Help systems.

The RTF file acts a mediator between a Help document ([DOC file](#)) and the Help project file ([HPJ file](#)).

ROBOHELP.DOT

Integrates Help authoring functionality with Word. This global template contains the commands, toolbars, menus, and shortcut keys necessary to complete authoring tasks – create topics and links, insert images and multimedia, test topics, compile the Help system, and so on. ROBOHELP.DOT also contains various pre-defined styles to assist you in formatting your Help topics.

.RTF file

Rich text format. A language for describing text that contains multiple fonts, images, and other visual objects. Rich text format is the source format for Windows WinHelp Help systems.

The RTF file acts a mediator between a Help document ([DOC file](#)) and the Help project file ([HPJ file](#)).

Secondary window

A Help window that is independent of the main window, which means that the main window and a secondary window can be displayed at the same time in the Windows Help Viewer.

See Also keyword



[Win 95](#) (Also known as A-keywords.) Similar to Index keywords, except never visible to users, See Also keywords group related topics together. See Also keywords are activated in the compiled Help file (.HLP) through the ALink WinHelp macro. In RoboHELP, you can either create the macro link manually or select the [See Also hotspot](#) or the [Smart See Also button](#) to create the link automatically.

When users select the See Also link (usually through a hotspot or button), the Topics Found dialog appears listing related topics associated with the See Also keyword used in the link. Users select a topic from this dialog and "jump" to the destination topic.

See Also hotspot



RoboHELP Action Type that uses the ALink WinHelp macro to dynamically link topics associated with selected [See Also keywords](#) (A-keywords). When users click the See Also link, a list of related topics appears. You can use the See Also hotspot to create See Also links as hotspots, buttons, or on hotspot images.

Segmented hypergraphic

Hotspot SHED image. A bitmapped image with hotspots drawn on the bitmap itself that, when clicked, display additional information about the area selected.

Select

To mark an option, button, check box, or item in a list by highlighting it. Selecting an item does not start an action.

SHED image

(Also known as hotspot SHED image). A bitmapped image with hotspots drawn on the bitmap itself that, when clicked, display additional information about the area selected.

Shell program

A graphical user interface (an environment in which the user and computer communicate) such as Microsoft Windows.

Shortcut buttons

A type of topic button. Shortcut buttons are actually images that appear as buttons and are commonly used to take users to an application, demonstration, or directions to complete a task. They can appear with or without hotspot text next to them. The size of the shortcut button depends upon the size of the image you select for the button face.

Smart Index Wizard

The Smart Index Wizard is an online indexing tool that greatly simplifies index preparation – whether you need to enhance an existing index or create an entirely new one. It automates indexing tasks you normally do by hand, streamlining the process and helping you become more productive.

The Smart Index Wizard searches your topics based on criteria you specify and suggests existing keywords used in other topics, as well as new keywords, based on the topic content. You can create an entire index at once or review suggested keywords on an individual topic basis.

Smart See Also button



One of the features of WinHelp 2000, the Smart See Also button appears on the Help window's button bar. If the current topic is associated with one or more [See Also keywords](#), the Smart See Also button is automatically enabled – allowing users quick access to related topics. RoboHELP takes care of creating the See Also hotspot link for you automatically.

Source files

The files that comprise a Help system for the Windows Help Viewer after they are compiled.

Specify

To choose from several choices in a listbox, or to type text or numbers into a text box to change a preference.

Standalone Help System

A Help system that is not connected to a program.

Style

Provide a way to help you define the appearance of the elements of a document. Styles control fonts, paragraphs, spacing, topic formatting, topic heading styles, and so on. The benefits of using styles include consistent formatting across documents and global updates – just change the attributes of the style.

Because RoboHELP integrates its Help authoring functionality into Microsoft Word, you use Word's style functionality to create, apply, and change styles – including maintaining styles in [ROBOHELP.DOT](#), the template that includes pre-defined Help authoring styles and integrates RoboHELP functionality with Word.

System menu

(Also known as Control menu) A window menu that contains commands to resize, close, or switch the window. You access the Control menu by clicking the Control menu button or by typing **ALT+SPACE**.

Table of contents

(TOC) A tree outline capable of displaying topics in an expandable/collapsible hierarchy. Tables of contents are commonly used to navigate through the structure of a Help system and locate information. Help tables of contents use the book and page metaphor to display and organize topics.

Users can browse through the TOC to see what topics are covered and select a topic to get the information they want. The table of contents also provides users with a home base – a safe, reliable place to start from and return to when they aren't sure where to go next.

Template

In Microsoft Word, templates are special types of documents that store styles, macros, AutoText entries, and custom toolbar, menu, and shortcut key settings. Word's NORMAL.DOT template is a global template that applies to all Word documents – Help documents or not.

RoboHELP integrates Help authoring functionality with Word through the [ROBOHELP.DOT](#) template. This global template contains the commands, toolbars, menus, and shortcut keys necessary to complete authoring tasks, as well as pre-defined styles to assist you in formatting your Help topics.

Text-marker macros

Macros used to create and manipulate text markers.

Title

1. **Project title.** The statement that defines the text that appears in the title bar of the Contents window when the Help file appears. In software applications, this is usually the name of the software.
2. **Topic title.** The statement that defines the text that appears at the top of topic text.

TOC

(Table of contents) A tree outline capable of displaying topics in an expandable/collapsible hierarchy. Tables of contents are commonly used to navigate through the structure of a Help system and locate information. Help tables of contents use the book and page metaphor to display and organize topics.

Users can browse through the TOC to see what topics are covered and select a topic to get the information they want. The table of contents also provides users with a home base – a safe, reliable place to start from and return to when they aren't sure where to go next.

TOC Composer

(TOC tab) Left-pane RoboHELP Explorer tab that enables Help authors to visually create tables of contents. It shows the results of the table of contents as it's being created and provides a feature for automatically constructing a table of contents.

Tools tab

Left-pane RoboHELP Explorer tab that provides shortcuts to the applications, folders, or files Help authors use the most during development. RoboHELP provides shortcuts to eHelp Corporation tools such as What's This Help Composer, but Help authors can add shortcuts to their favorite applications, such as Paint Shop Pro.

Topic

See [Help topic](#).

Topic ID

Unique identifiers for each topic in a WinHelp file – sort of like an address. Topic IDs are represented by the # (pound) topic footnote, and every topic is required to have a Topic ID. Each time a topic is referenced (hotspots, buttons, macros, table of contents, keywords, and so on), the reference uses the Topic ID to locate the "right" topic. The Topic ID can consist of letters, digits and the underscore character (_).

Some topics contain inter-topic Topic IDs, called [Mid-Topic IDs](#), which allow users to jump to text within a topic.

Topic List

Right-pane RoboHELP Explorer tab that provides a quick list of topics and their attributes in the current Help project. You can choose the columns of information that appear in the Topic List, as well as sort information by clicking on a particular column heading.

You can use the Topic List to drag and drag topics into the TOC Composer and Index Designer to quickly create tables of contents and Index keywords.

Training cards

Small tutorial like cards that guide the user through a particular procedure, one step at a time. They require a cooperative effort between the programmer and Help author.

Transparent bitmap

A transparent bitmap is one where all the white pixels in the image are replaced with the background color of the currently active window.

TrueCode

This is the "traditional" development view mode where you see the coding statements rather than the objects inside your Help topics. Choosing this view when working means you see the statement the WinHelp compiler uses to create the object – button, image, or multimedia object – rather than the object itself.

For example, a Help image would appear something like this: `{bmc graphic.bmp}`.
The other development view mode is called [Dynamic WYSIWYG](#).

Tutorial

Information conveyed in a step-by-step breakdown with many examples and graphics. Tutorials are usually shipped with a Help system and are designed for the user who wants to walk through a feature or a program.

Uniform Resource Locator

(URL) An address for a resource on the Internet. URLs are used by Web browsers to locate Internet resources. A URL specifies the protocol to be used in accessing the resource (such as http: for a World Wide Web page or ftp: for an FTP site), the name of the server on which the resource resides (such as //www.blue-sky.com), and, optionally, the path to a resource (such as an HTML document or a file on that server).

URL

Uniform Resource Locator. An address for a resource on the Internet. URLs are used by Web browsers to locate Internet resources. A URL specifies the protocol to be used in accessing the resource (such as http: for a World Wide Web page or ftp: for an FTP site), the name of the server on which the resource resides (such as //www.blue-sky.com), and, optionally, the path to a resource (such as an HTML document or a file on that server).

User

Refers to the end-user (or intended user) of a Help system.

Watermark



Win 95 Images that appear in the background of the [Explorer View Help window](#), behind text and other images in the non-scrolling or background regions of the window. With watermarks, you can add unique textures, logos, images, and more to your Help systems to give them a distinctive look. You can use any 16-color or 256-color bitmap (.BMP) image as a watermark.

.WAV file

The native file format in which Windows stores sounds as waveforms. Such files have the extension .WAV.

WebHelp

eHelp's cross-platform, browser-independent Help file format. It allows Help authors to create HTML Help systems and intranet sites for most platforms and browsers.

With WebHelp, users can view HTML Help with Internet Explorer or Netscape Navigator on Windows, Macintosh, Linux, Sun Solaris, and several other UNIX platforms.

WebHelp provides standard HTML-based features without platform or browser restrictions, which means Help authors can distribute the same WebHelp system on any combination of platforms and browsers.

What's This? Help

Context-sensitive help available for each field or control on a dialog box. This type of context-sensitive help was introduced with Windows 95. It appears when users press **F1**, click the **?** button, or right-click buttons and controls at dialogs.

What's This Help Composer

eHelp application that allows Help authors to create What's This-style context-sensitive help for fields and controls on dialogs.

WinHelp 3

WinHelp 3 is the original Windows Help. It was primarily developed for Windows 3.x and Windows 3 applications, which use 16-bit technology. WinHelp 3 features and functionality are fully supported on all Windows operating systems – Windows 95 and newer.

WinHelp 4

WinHelp 4 was introduced with Windows 95. It uses an expanded 32-bit technology and includes newer features and added functionality not available through WinHelp 3. WinHelp 4 Help systems only run on Windows 95, 98, and NT operating systems. WinHelp 4 contains expands the features in WinHelp 3 and offers several additional features – such as the Help Topics browser user interface (containing the Contents, Index, and Search tabs), a hierarchical table of contents that uses the books and pages metaphor, See Also links to easily display related topics, and more.

WinHelp 2000

RoboHELP Classic feature that allows Help authors to add the following functionality to WinHelp 4 Help systems:

- **Explorer View Help tri-pane window.** A new WinHelp window similar to Microsoft HTML Help's tri-pane window – Contents, Index, and Search tabs appear to the left of the topic pane, so users never lose their place.
- **Smart See Also button.** The Smart See Also button provides "smart" access to related topics by placing a See Also button on your Help window's button bar. If a topic is part of one or more See Also keyword groups, clicking the Smart See Also button automatically displays the list of other topics associated with the same See Also keyword groups.
- **Watermarks.** Watermarks allow you to add background textures, colors, images, and even logos to the Explorer View Help window to give your Help system a distinctive look.
- **Links to HTML topics and Web sites from inside the WinHelp system.** WinHelp 2000 allows you to embed links to HTML topics and Web pages inside the WinHelp system. When users click the hotspot link, WinHelp 2000 launches the selected HTML topic or Web page in the appropriate browser, but within the Explorer View Help window frame.
- **Insert HTML topics and Web sites inside the WinHelp system.** WinHelp 2000 lets you integrate HTML topics and Web pages in WinHelp systems by inserting a reference to the HTML topic or Internet or intranet Web page as a "topic" in your WinHelp system. This topic can even be included in your WinHelp system's browse sequence and index.

WinHelp 2000's functionality is enabled by the Robo32Ex.DLL.

Windows Help Viewer

The program that allows users to open and display the compiled Help file. The WinHelp viewers used to display the Help system are automatically built-in to every Microsoft Windows operating system. Windows opens the appropriate Help viewer to display the Help system, depending on the compiler used to process it and the features the Help system contains.

Window macros

Macros that control or modify the behavior of the various Help windows.

WYSIWYG

(What You See Is What You Get) The development environment option that allows you to view everything in a Help project as if the Help system was already compiled. For example, images in WYSIWYG mode display as they will in a compiled Help system instead of by reference code.

The other development environment option is [TrueCode](#).

WinHelp API

WinHelp Application Programming Interface. This function makes it possible to display a context-sensitive help topic from inside an application.

This is the general form used to send the command to the WinHelp viewer to call a context-sensitive help topic as documented in Microsoft Help Compiler's online Help Author's Guide:

WinHelp(HWND hwnd, LPCTSTR lpszHelpFile, UINT fuCommand, DWORD dwData)

- HWND is the Window requesting help
- LPCTSTR is the Help file name containing the context-sensitive topic
- UINT is the type of help
- DWORD is additional information

The Microsoft Foundation Class (MFC) simplifies the syntax, but it is more limited because it requires the Help file to have the same name as the application. If your Programmer needs to send a command specifying a different help window or a different Help file, he or she will have to use the general form listed above.

{ewl roboex32.dll, WinHelp2000, }

