
The Help Project File

This chapter describes the format and contents of the Help project file (.HPJ), which is used to build the Help file. The Help project file contains all the information the Help compiler needs to combine RTF files and other elements into a Help file. Among other information, the Help project file tells the compiler:

- Where to find the files used to build the Help file.
- Which topic contains the table of contents for the Help file.
- Which custom elements (including menus, buttons, and windows) are added to the Help file.
- Which custom DLLs (if any) are used with the Help file.
- Which options to include during the build process.

Help Project File Sections

A Help project file consists of several sections, each of which specifies information about the Help file. Section names appear within square brackets using the following syntax:

[*sectionname*]

The following table describes the nine sections that can be used in a Help project file.

Section	Function
[OPTIONS]	Specifies options that control the build process. This section is optional. If this section is used, it should appear before any other section in the Help project file.
[FILES]	Specifies topic files to be included in the build. This section is required.

Microsoft Windows Help Authoring Guide

[BUILDTAGS]	Specifies valid build tags. This section is optional.
[CONFIG]	Specifies author-defined menus and buttons used in the Help file and registers DLLs and DLL functions used as macros within the Help file. This section is required if the Help file uses any of these features.
[BITMAPS]	Specifies bitmap files to be included in the build. This section is not required if the Help project file lists a path for bitmap files using the BMROOT or the ROOT option.
[MAP]	Associates context strings with context numbers for context-sensitive Help within the application. This section is optional.
[ALIAS]	Assigns one or more context strings to the same topic. This section is optional.
[WINDOWS]	Defines the characteristics of the primary Help window and secondary window types used in the Help file. This section is required if the Help file uses secondary windows.
[BAGGAGE]	Lists files that are to be placed within the Help file's .HLP file (which contains its own file system). This section is optional.

Semicolons (;) can be used to indicate a comment in the Help project file. Comments can be a single line or multiple lines. The compiler ignores all text from the semicolon to the end of the line on which it occurs.

Project-File Features

To create many features in the Help file, you modify topic files in the word processor. After you create the topic files, you build the file with the Help project file and compiler.

But that's only the simple case. In fact, you use the Help project file to do more than just pass technical information

to the compiler. You also use the Help project file to create unique features within the Help file. Some of these features can be added only by using project-file sections and options. Each section and option in the Help project file has a different purpose, and each can be used to improve the effectiveness and usability of your finished Help file.

Before you go on to the "Help Project File Reference" section, you might want to review the following table to find out what kinds of features you can create with project-file sections and options.

Project-file feature	[Section] or option
Customize the look and feel of the Help application, such as add menus and buttons and use DLLs to create new features.	[CONFIG]
Control the size and position of Help windows and add secondary windows that display information independently of the main window.	[WINDOWS]
Display a custom icon when the user minimizes the Help window.	ICON
Display the name of your Help file in the title bar of the Help window.	TITLE
Designate a certain topic as the Contents screen for the entire Help file.	CONTENTS

Microsoft Windows Help Authoring Guide

Create a custom copyright message for your Help file that users will see when viewing the Help file or copying information from it.

CITATION, COPYRIGHT

Customize the keyword search so that it uses more than one keyword table or a non-English language sorting order.

MULTIKEY, LANGUAGE

Tell the compiler which source files to include in the build and where to find them.

[FILES], [BITMAPS], BMROOT, ROOT

Compress the Help file so it uses less disk space.

COMPRESS, OLDKEYPHRASE

Control the amount of information the compiler displays during a build and where the information is displayed.

ERRORLOG, REPORT, WARNING

Create the topic files using one set of fonts and font sizes and have Help display the Help file using a different font and font size.

FORCEFONT, MAPFONTSIZE

Improve the performance of the Help file when it is delivered to users on a CD-ROM drive.

OPTCDROM

Perform a partial build of the Help file that [BUILDTAGS], **BUILD**
excludes certain topics.

The Help Project File§ 16-5

Define a set of identifiers so that the application [MAP], [ALIAS]
can display context-sensitive Help topics when the
user requests Help.

Include a group of non-standard files, such as [BAGGAGE]
sound files, in the Help file.

Sample Help Project File

The following shows a sample Help project file for the Cardfile application.
Comments in the file text indicate the purpose of each section in the file

```
; Options used to define the
; Help project root, bitmap directory,
; contents topic, title, minimized icon,
; compression, warning level,
; report, and error file
;
[OPTIONS]
ERRORLOG=CARD.BUG
ROOT=C:\HELP
BMROOT=C:\HELP\ART
CONTENTS=cont_idx_card
TITLE=Cardfile Help
ICON=CARDHLP.ICO
COMPRESS=OFF
WARNING=3
REPORT=ON

;files used to build Cardfile Help
[FILES]
\RTFTEXT\COMMANDS.RTF
\RTFTEXT\HOWTO.RTF
\RTFTEXT\KEYS.RTF
\RTFTEXT\GLOSSARY.RTF

;button macros and How to Use Help file
[CONFIG]
```

```
CreateButton("btn_up", "&Up", "JumpContents('HOME.HLP')")
BrowseButtons()
SetHelpOnFile("APPHELP.HLP")
;secondary-window characteristics
[WINDOWS]
```

Help Project File Reference

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

The Help Project File Reference describes the different sections and options in a Help project file in alphabetic order and gives examples of their use. Reference descriptions provide the following information.

Heading	Information
----------------	--------------------

Syntax	Syntax for the section or option. For information about the typographic conventions used in syntax descriptions, see the "Document Conventions" section in the Introduction to this guide.
Parameters	Describes the parameters that the section or option requires.
Comments	Notes about using the section or option, including any restrictions.
Example	Example of the section or option.
See Also	Cross-references to similar sections and options.

Syntax

[ALIAS] Section

[ALIAS]

context_string=alias context string

·
·
·

The [ALIAS] section associates one set of context strings with an alternate set of context strings. The alias strings correspond to context strings assigned to topics in the # footnotes of the Help file. This section is optional; however, if it is

Parameters

included, it must precede the [MAP] section in the Help project file.

context_string

Specifies the application ID or other context ID that you want to reassign.

alias context string

Specifies the context string that appears in the # footnote of the topic you want Help to recognize. An alias context string has the same form and follows the same conventions as standard context string. That is, it is not case-sensitive and may contain the alphabetic characters A through Z, the numeric characters 0 through 9, and the

Comments

period (.) and underscore (_) characters.

Because context strings must be unique for each topic and cannot be used for any other topic in the Help project, the [ALIAS] section provides a way to remap context strings that are no longer used or invalid. For example, suppose the application defines a context ID for each field in a dialog box, but your Help file only provides one topic for all the fields. You can use the [ALIAS] section to map all the application context IDs to your one Help topic. In this way, no matter which field the user has selected in the dialog box, Help will display your Help

topic when the user requests context-sensitive Help.

Microsoft Windows Help Authoring Guide

You can also use the [ALIAS] section to combine Help topics without recoding your files. For example, if you create a topic that replaces the information in three other topics, you could manually search through your files for invalid cross-references to the deleted topics. The easier approach, however, would be to use the [ALIAS] section to assign the name of the new topic to the deleted topics.

You can use alias names in the [MAP] section of the Help project file. If you do,

Example

however, the [ALIAS] section must precede the [MAP] section.

The following example creates several aliases within an [ALIAS] section:

```
[ALIAS]
sm_key=key_shrcuts
cc_key=key_shrcuts
st_key=key_shrcuts           ;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
```

See Also

```
seltxt=edittxt           ;covered in editing text topic
```

[MAP] Section

Syntax

[BAGGAGE] Section

[BAGGAGE]

filename

.

.

The [BAGGAGE] section lists files that the Windows Help compiler stores within the .HLP file's internal file system. Windows Help can access data files stored in the Help file system more efficiently than it can access files stored in the normal MS-DOS file system because it doesn't have to read the file allocation table from CD-ROM when it accesses the files. Any type of DOS file can be stored in baggage.

In some circumstances, a Help file may use many DLLs or execute several programs that use external data elements, such as graphics, animations, or audio files. In those situations, the Help file will benefit if it can include all the data elements needed by those programs in one MS-DOS file. To store Help-related data files within the Help file, Help authors can include a [BAGGAGE] section in

Parameter

the Help project file.

filename

Specifies the full path of a baggage file. If a file cannot be found, the

Comments

compiler reports an error.

The Help compiler stores all files listed in the [BAGGAGE] section exactly as they are typed. In other words, baggage filenames are case sensitive. To retrieve a baggage file, Help uses the MS-DOS filename without the path. This means that the Help author must specify the filename exactly as it appears in the [BAGGAGE] section, as in the following example.

GRAPHIC

Help does not limit the number of baggage files that you can define; however, adding too many baggage files will waste disk space and decrease performance when building and displaying the Help file. For that reason, you should define no more than 1000 baggage files. If you have more than 1000 files to include, you should store the data outside the .HLP file or concatenate the files into a few baggage files that you list in the [BAGGAGE] section.

To access the data from the Help file's internal file system, Windows Help provides specialized source code. This source code can be built into an

application or custom DLL so that it can retrieve the appropriate data file from the Help file's [BAGGAGE] section. For more information and the baggage include file, see Appendix C, "Baggage Access Functions."

To avoid having to specify a full path, use the **ROOT** option in the [OPTIONS] section to specify the path. Then all filenames that you give in the topic file are

Example

relative to the **ROOT** path.

The following [BAGGAGE] section stores multimedia files in the Help file:

```
[BAGGAGE]
\\ANIM\MASTER.AMF
\\AUDIO\APATO.WAV
\\AUDIO\PACHY.WAV
\\AUDIO\STAR.WAV
\\AUDIO\TREX.WAV
\\AVT\APATO.AVI
\\AVT\PACHY.AVI
```

See Also

\\AVT\TREX.AVI

ROOT Option

Syntax

[BITMAPS] Section

[BITMAPS]

filename

.
. .
.

The [BITMAPS] section tells the Help compiler where to find bitmaps placed using the **bmc**, **bml**, or **bmr** reference.

Parameter *filename*

~~Specifies the full path of a bitmap file. If a file cannot be found, the~~

~~The Help Project Files 16-11~~

Comments

compiler reports an error.

The [BITMAPS] section is not required if the bitmaps are located in the Help project directory or if the path containing the bitmaps is listed in the **BMROOT** or **ROOT** option. If the Help project file does not include either of these options, each bitmap filename must be listed in the [BITMAPS] section of the Help

Example

project file.

The following example specifies three bitmaps:

```
[BITMAPS]
BMP01.BMP
BMP02.BMP
```

See Also

BMP03.BMP

BMROOT Option, ROOT Option

Syntax

BMROOT Option

BMROOT=*pathname*[, *pathname*]...

The **BMROOT** option designates the bitmap root directory, which tells the Help compiler where to find the bitmap files to include in the build.

If the bitmap files reside in the root directory of the project, you don't need to include a **BMROOT** directory; use the **ROOT** option instead. However, if you

set up your Help project so that bitmap files are not stored in the project root directory, you may want to designate one or more bitmap root directories in the

Parameter

BMROOT option.

pathname

Specifies a drive and full path to each bitmap directory or to a path relative to the project directory. If you give more than one path for this option, use a comma (,) to separate the paths. The entire **BMROOT** entry in the Help project file can be up to 128 characters

Comments

long. If the path is invalid or too long, the compiler reports an error.

If the Help project file has a **BMROOT** option, you don't need to list bitmap files in the [BITMAPS] section.

If the Help project file does not have a **BMROOT** option, the Help compiler looks for bitmaps in the directories specified by the **ROOT** option. If the Help project file doesn't have a **ROOT** option, or if none of the **ROOT** directories contain the bitmap files, you can:

- List the bitmap filenames in the [BITMAPS] section of the Help project file.

or

Example

- Place all bitmap files in the same directory as the Help project file.

The following entry specifies that bitmap files reside in the \HELP\BMP1 and \HELP\BMP2 directories on drive C and in the \GRAPHICS\ART directory on drive D:

```
[OPTIONS]  
BMROOT=C:\HELP\BMP1, C:\HELP\BMP2, D:\GRAPHICS\ART
```

See Also [BITMAPS] Section, [OPTIONS] Section, ROOT Option

The Help Project File§ 16-13

Syntax

BUILD Option

BUILD=*expression*

The **BUILD** option specifies which topics containing build tags are included in or excluded from a build. Use this option only if the RTF topic files have build tags.

A topic contains a build tag if it includes a build tag footnote (*). Topics without

Parameter

build tags are always compiled, regardless of the current build expression.

expression

A logical statement that specifies which topics to include in or exclude from the build. This parameter consists of a combination of build tags (specified in the [BUILDTAGS] section) and the following logical operators.

Operator Description

-
- | | |
|---|--|
| ~ | Applies the NOT operator to a single build tag. The Help compiler compiles a topic only if the tag is <i>not</i> present. This operator has the highest precedence; the compiler applies it before any other operator. |
| & | Combines two build tags by using the AND operator. The Help compiler compiles a topic only if it contains both build tags used in the expression. The compiler applies this operator only after the ~ operator has been applied. |
| | Combines two build tags by using the OR operator. The Help |

compiler compiles a topic if it has at least one of the build tags used in the expression. This operator has the lowest precedence; the compiler applies it only after all other operators have been applied.

Parentheses may be used to override operator precedence.

Comments

Expressions enclosed within parentheses are always evaluated first. Only one **BUILD** option can be given per Help project file.

The Help compiler evaluates all build expressions from left to right, using the

Example

specified precedence rules.

The following examples assume that the [BUILDTAGS] section in the Help project defines the build tags DEMO, MASTER, and TEST_BUILD. Although the following examples show several **BUILD** options on consecutive lines using these build tags, only one **BUILD** option would be allowed in the Help project file.

Example

Compile all topics

BUILD=DEMO

That have the DEMO tag

BUILD=DEMO & MASTER

With both the DEMO and MASTER tag

BUILD=DEMO | MASTER

With either the DEMO or MASTER tag

BUILD=(DEMO | MASTER) & TEST_BUILD That have either the DEMO or MASTER

tag and also the TEST_BUILD tag

The Help Project File§ 16-15

BUILD=~MASTER

That do not have the MASTER tag

See Also

[BUILDTAGS] Section, [OPTIONS] Section

Syntax

[BUILDTAGS] Section

[BUILDTAGS]

tag

·
·
·

The [BUILDTAGS] section defines the valid build tags for a Help file. The Help compiler uses the build tags to determine which topics to include when building the Help file.

This section is used in conjunction with the build tag footnote (*) and the **BUILD** option. The build tag footnote associates a particular build tag with a given topic. If the build tag is included in the [BUILDTAGS] section and defined in the **BUILD** expression, the Help compiler compiles the topic; otherwise, it excludes

Parameter

the topic from the build.

tag

Specifies a build tag consisting of any combination of characters except spaces. The Help compiler strips any spaces it finds between

the start of the build tag and the end of the tag. Build tags are case insensitive, so the compiler treats uppercase and lowercase characters

Comments

as the same. Each build tag can have as many as 32 characters. The [BUILDTAGS] section is optional. If used, it can include as many as 30

Example

build tags.

The following example defines four build tags in a sample Help 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
```

See Also

```
TESTBUILD     ;topics to include in a mini-build for testing
```

BUILD Option

Syntax

CITATION Option

CITATION=*copyright-notice*

The **CITATION** option appends a custom copyright notice to the end of any information that is copied from the Help file using the Copy command. You can use this option if a publisher owns some material in your Help file and requires that you attach a copyright notice to that material.

Parameter *copyright-notice*

~~The Help Project Files 16-17~~
Specifies the copyright information you want to append to copied material. The notice appears at the end of text displayed in the Copy dialog box (the notice is appended whenever the user copies text

Comments

from the original Help file).

The **CITATION** option differs from the **COPYRIGHT** option in two ways: a citation does not appear in the About dialog box, and a citation can be much larger than the standard copyright notice. The maximum size of the citation

Example

copyright notice is about 2K (compared to 50 characters for **COPYRIGHT**).

The following shows a sample citation:

CITATION=Information in this document is subject to change without notice and does not represent a commitment on the part of Microsoft Corporation. The software, which includes information contained in any databases, described in this document is furnished under a license agreement or nondisclosure agreement and may be used or copied only in accordance with the terms of that agreement. It is against the law to copy the software except as specifically allowed in the license or nondisclosure agreement. No part of this document may be reproduced in any form or by any means, electronic or mechanical, for any purpose without the express written permission of Microsoft

See Also

Corporation.

COPYRIGHT Option, [OPTIONS] Section

Syntax

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 percent), or high compression (approximately 50 percent). The higher the compression level, the smaller the

Parameter

Help file. However, the higher the compression, the longer the file takes to build.

compression-level

Specifies the level of compression. This parameter can be one of the following values.

Value	Meaning
OFF	The Help compiler does not compress the Help file.
MEDIUM	The compiler uses a medium level of compression (block compression).
HIGH	The compiler uses high compression (block and key-phrase compression).

Comments

Depending on the level of compression requested, the Help compiler uses either *block compression* or a combination of *block* and *key-phrase compression*:

- ⁿ *Block compression* compresses the topic data into predefined units known as *blocks*.
- ⁿ *Key-phrase compression* combines repeated phrases found within the source file(s).

The Help compiler creates a phrase-table file with a .PH extension if it doesn't find one in the project root directory. If the Help compiler finds a file with a .PH extension, it uses the file for the current build. Because the .PH file speeds up the compression process when little text has changed since the last build, you might want to keep the phrase file around if you compile the same Help file several times with compression. However, you get maximum compression if you delete the .PH file before starting each build.

Because of these improvements to Help compression, the resulting Help files you build will take up less space on users' disks, but your setup program may not be able to compress the smaller Help files as much as version 3.0 files. Therefore, when estimating disk-space requirements, you should use sizes relatively close to those achieved with high compression.

Key-phrase compression is the only compression method supported by the

Example

version 3.0 Help compiler.

The following shows a typical compression entry in the Help project file:

See Also

COMPRESS=off

[OPTIONS] Section

Syntax

[CONFIG] Section

[CONFIG]

macro

-
-
-

The [CONFIG] section contains one or more Windows Help macros that carry out actions, such as creating buttons or menus. This section can also contain

macros that register routines in external DLLs as Windows Help macros. These routines can then be used the same as Windows Help macros. Windows Help

Parameter

executes the macros when it opens the Help file.

macro

Specifies a Windows Help macro or a DLL function registered as a Help macro. For descriptions of the standard Help macros and the

Comments

RegisterRoutine macro, see Chapter 15, “Help Macro Reference.”

The [CONFIG] section can include any number of lines, and each line can have as many as 512 characters. When listing macros in the [CONFIG] section, include only one macro per line, instead of stringing them together and separating them with semicolons as you do in macro hot spots and macro footnotes.

When opening a Help file, Help does not necessarily execute the macros listed in

Example

the [CONFIG] section in the same order.

The following example registers two DLLs, creates a button, enables the browse buttons, and sets the name of the How To Use Help file:

```
[CONFIG]
RegisterRoutine("bmp", "HDisplayBmp", "USSS")
RegisterRoutine("bmp", "CopyBmp", "v=USS")
CreateButton("btn_up", "&Up", "JumpContents('HOME.HLP')")
BrowseButtons()
SetHelpOnFile("APPHELP.HLP")
```

Syntax **CONTENTS**=*context-string*

The Help Project File§ 16-21

The **CONTENTS** option identifies the context string of the highest-level or “home” topic (usually a Table of Contents or index within the Help file). Windows Help displays the Contents whenever a user chooses Contents from the Help menu, clicks the Contents button, or presses F1 in the application without a

Parameter

specific context.

context-string

Specifies the context string of a topic in the Help file. The string can be any combination of characters, except spaces, and must also be specified in a context string footnote (#) in some topic in the Help

Comments

file.

If the [OPTIONS] section does not include a **CONTENTS** option, the compiler assumes that the first topic it encounters in the first listed topic file in the [FILES]

Example

section of the Help project file is the contents topic.

The following example sets the topic containing the “main_contents” context string as the Contents topic for this Help file:

See Also

CONTENTS=main_contents

[FILES] Section, [OPTIONS] Section

Syntax

COPYRIGHT=*copyright-notice*

Microsoft Windows Help Authoring Guide

The **COPYRIGHT** option places a custom copyright notice in the About dialog box of Windows Help. Help displays the notice immediately below the Microsoft copyright notice. You can use this option to copyright the material found in your

Parameter

Help file.

copyright-notice

Specifies the copyright notice you want to display for users of your Help file. The notice can be any combination of characters, and can have from 35 to 75 characters depending on the characters you use. A

Comments

notice of 50 characters generally fits in the dialog box.

The copyright notice also appears at the end of text displayed in the Copy dialog box. The notice is appended whenever the user copies text from the original Help file using the Copy command.

Example

If you need to display a longer copyright notice, use the **CITATION** option.

The following example adds a short copyright notice to the Help file:

See Also

COPYRIGHT="Copyright (C) 1992, Microsoft Corporation."

CITATION Option, [OPTIONS] Section

ERRORLOG Option

Syntax **ERRORLOG=***error-filename*

The Help Project File§ 16-23

The **ERRORLOG** option directs the compiler to write all error messages generated during the build to an error file. The compiler also displays the error

Parameter

messages on the screen.

error-filename

Specifies the name of the file to which the compiler will write the error messages. The filename can be any valid MS-DOS filename. This parameter can be a full or partial path if you want the file to be

Comments

written to a directory other than the Help project root directory.

If you use the **ERRORLOG** option, it should be the first line in the [OPTIONS] section.

The error file contains the Windows Help copyright notice and the name of the Help project file at the top of the file, followed by any build errors that occurred. Error messages are listed on separate lines. The periods representing compiler progress are included only in the screen display, not in the written file.

If the Help compiler cannot create or open the error file, it displays an error

Example

message on the screen and continues the build.

The following example writes all errors during the build to the HLPBUGS.TXT file in the Help project root directory:

See Also

ERRORLOG=HLPBUGS.TXT

[OPTIONS] Section, REPORT Option, WARNING Option

Syntax

[FILES] Section

[FILES]

filename

.
. .
.

The [FILES] section lists all RTF topic files used to build the Help file. A Help

Parameter

project file must have a [FILES] section.

filename

Specifies the full or partial path of a topic file. If a partial path is given, the help compiler uses the directories specified by the **ROOT** option to construct a full path. If a file is not on the defined path and

Comments

cannot be found, the compiler reports an error.

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

The **#include** directive has the following syntax:

#include <*filename*>

The *filename* must reside in the Help project directory, or it must include a complete path specification. The Help compiler does not use the INCLUDE environment variable to search for files.

Example

The following example specifies four topic files:

The Help Project File§ 16-25

```
[FILES]
\\rtftxt\COMMANDS.RTF      ;comment 1
\\rtftxt\HOWTO.RTF        ;comment 2
\\rtftxt\KEYS.RTF         ;comment 3
\\rtftxt\GLOSSARY.RTF     ;comment 4
```

The following example uses the **#include** directive to specify the topic files indirectly:

```
[FILES]
```

See Also

```
#include <rtfiles.h>
```

ROOT Option

Syntax

FORCEFONT Option

FORCEFONT=*fontname*

The **FORCEFONT** option forces the Help file to substitute the specified font for all requested fonts. Use this option to create Help files that can be viewed on

Parameter

systems that do not have all fonts available.

fontname

Specifies the name of an available font. Font names must be spelled the same as they are in the Fonts dialog box in Control Panel. Font names cannot exceed 20 characters. If an invalid font name is given, the Help compiler uses the MS Sans Serif font as the default.

Comments

The fontname can be any of the following standard fonts installed in Windows version 3.1:

- n Courier 10,12,15
- n Modern
- n MS Sans Serif 8,10,12,14,18,24
- n MS Serif 8,10,12,14,18,24
- n Roman
- n Script
- n Small
- n Symbol 8,10,12,14,18,24

Windows version 3.1 also includes the following scalable TrueType™ fonts; however, you cannot use this option to specify a TrueType font:

- n Arial®
- n Arial Bold
- n Arial Bold Italic
- n Arial Italic
- n Courier
- n Courier Bold
- n Courier Bold Italic
- n Courier Italic
- n Times New Roman®
- n Times New Roman Bold
- n Times New Roman Bold Italic
- n Times New Roman Italic
- n Symbol

Example

The following example forces all fonts to be displayed in the MS Serif font:

The Help Project File§ 16-27

See Also

FORCEFONT=MS SERIF

MAPFONTSIZE Option, [OPTIONS] Section

ICON Option**Syntax**

ICON=*icon-file*

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

Parameter

Windows Help application.

icon-file

Specifies the name of the icon file. This file must have the standard Windows icon-file format (.ICO file). You must create this file in an application such as Microsoft Windows Image Editor (IMAGEDIT.EXE) or other application that generates the .ICO file format. You can specify either an absolute or relative path if the file

Comments

resides in a directory other than the Help project root directory.

If you do not include the **ICON** option in your Help project file, Help will use the standard question-mark icon when the user minimizes the Help file.

If the icon file is in an invalid format, or if Help cannot find the icon file, the Help compiler displays an error message on the screen during the build and

ignores this option. In that case, Help will display the standard question-mark icon.

The **ICON** option only appears when the user minimizes Help. To have your custom icon appear in a Program Manager group, you must provide users with the icon file and then instruct them to use the Properties command in Program

Example

Manager to change the standard Help icon to your custom icon.

The following example creates a custom icon for the Help file:

See Also

ICON=HYPER.ICO

[OPTIONS] Section

Syntax

LANGUAGE Option

LANGUAGE=*language-name*

The **LANGUAGE** option sets the sort order for keywords in the Search dialog

Parameter

box.

language-name

Specifies the language on which to base sorting. This parameter can have only one value.

Value	Meaning
--------------	----------------

The Help Project File§ 16-29

scandinavian	Sets the sorting order to the Scandinavian-language order.
---------------------	--

Comments

The default sorting order is the English-language order.

Microsoft Windows Help version 3.1 supports only English and Scandinavian

Example

sorting.

The following example changes the sorting order to Scandinavian:

See Also

LANGUAGE=SCANDINAVIAN

[OPTIONS] Section

Syntax

[MAP] Section

[MAP]
context_string *context number*

·
·
·

The [MAP] section associates context strings (or aliases) to context numbers for

context-sensitive Help. The context number corresponds to a value the parent application passes to Windows Help to display a particular topic. This section is

Parameters

optional.

context-string

Specifies the context string of a topic in the Help file. The string can be any combination of characters, except spaces, and must also be specified in a context string footnote (#) in some topic in the Help file.

context-number

Specifies the context number to associate with the context string. The

Comments

number can be in either decimal or standard C hexadecimal format.

Only one context number may be assigned to a context string or alias. Assigning the same number to more than one context string generates a compiler error.

You can separate context numbers and context strings by an arbitrary amount of white space using space characters or tabs, but there must be at least one space between the context number and the context string.

If you do not explicitly assign context numbers to topics, the Help compiler generates default values by converting topic context strings into context numbers.

You can define the context strings listed in the [MAP] section either in a Help topic or in the **[ALIAS]** section. The compiler generates a warning message if a context 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.

If you remove a Help topic that the application defines as context sensitive, users will get a “Topic does not exist” error message when they request Help on the item. To prevent the error from occurring, you must either change the application so that it no longer sends the context number to Help or map that context number

to an existing topic.

The [MAP] section supports two additional statements for specifying context strings and their associated context numbers: **#include** and **#define**.

The **#include** statement has the following form:

```
#include <filename>
```

The *filename* parameter, which can be enclosed in either angle brackets (<>) or double quotation marks, 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.

The **#define** statement has the following form:

```
#define context-string context-number
```

The *context-string* and *context-number* parameters are the same as those described in the “Parameters” section above.

If context numbers use the **#define** directive and the file containing the **#define** statements is included in both the application code and the Help file, updates made to the context numbers by the application programmers are reflected in the next Help build.

When using the **#define** statement in the [MAP] section, observe these rules:

- You can use C-style comments (*/** open comment and **/* close comment) with the **#define** directive. The comments can occur anywhere in the line.

```
#define context_string context_number /* comment */
```

- The Help compiler supports 32-bit constants in **#define** statements. It also accepts (as 32-bit constants) numbers that end with L and are accepted by the C compiler for a long constant:

```
#define vscroll 1234000L
```

- The Help compiler does not perform arithmetic on the object of the **#define** statement. It does not support the following forms of **#define**:

```
#define A 1
```

```
#define B (A+1)
```

```
#define C (A+2)
```

ⁿ The Help compiler only accepts **#define** statements; it does not support other forms such as **#ifdef** and **#endif**.

Examples

The following example uses a decimal number to specify the context number:

```
dtb_scr    34    ;document title bar
```

The following example uses a hexadecimal number to specify the context number:

```
Minimize_Icon  0x0004
```

The following example uses an **#include** directive to point to another file containing the context strings and context numbers:

```
#include <sample.h>
```

The following example uses a **#define** statement to specify the mapping:

See Also

```
#define up_scroll  0x0112  /* up scroll arrow */
```

[ALIAS] Section

Syntax

MAPFONTSIZE

```
MAPFONTSIZE=m[-n]:p
```

The **MAPFONTSIZE** option maps font sizes specified in topic files to different sizes when displayed in the Help window. You can use one font size in your topic files and have the compiler change them to an appropriate size for the actual Help file display.

This option is especially useful if there is a significant size difference between the authoring display and the intended user display, as there is if the RTF file is

The Help Project File§ 16-33

Parameters

created using Word for the Macintosh.

m[-*n*]

Specifies the size of the source font. This parameter is either a single point size or a range of point sizes, as indicated by the optional parameter *n*, which specifies a font range to be mapped. A range of point sizes consists of the low and high point sizes separated by a hyphen (-). If a range is specified, all fonts in the range are changed to the size specified by the *p* parameter.

p

Comments

Specifies the size of the desired font for the Help file.

Although you can specify as many as five font ranges in the [OPTIONS] section of the Help project file, you can map only one font size or range with each **MAPFONTSIZE** statement. If you include more than one **MAPFONTSIZE** statement, the source font size or range specified in subsequent statements cannot

Examples

overlap previous mappings.

The following examples illustrate the use of 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 pt.
```

The following two statements show an incorrect use of the **MAPFONTSIZE** option because the second statement contains a point size already mapped in the preceding statement (14 falls in the 12—24 range):

```
MAPFONTSIZE=12-24:16  
MAPFONTSIZE=14:20
```

See Also

FORCEFONT Option, [OPTIONS] Section

Microsoft Windows Help Authoring Guide

Syntax

MULTIKEY Option

MULTIKEY=*footnote-character*

The **MULTIKEY** option specifies the footnote character to use for an alternate keyword table. This option is intended to be used in conjunction with topic files

Parameter

that contain keyword footnotes for alternative keyword tables.

footnote-character

Specifies the case-sensitive letter to be used for the keyword

Comments

footnote.

Multiple keyword tables enable Help authors to differentiate terminology from different sources. For example, the standard keyword table can be used to define search queries for the parent application, and a second keyword table can map commands in an auxiliary application. Users can then look up topics using both sets of keywords.

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 may 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, including 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.

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

See Also

MULTIKEY=L

[OPTIONS] Section

Syntax**OLDKEYPHRASE Option**

OLDKEYPHRASE=*yes/no*

The **OLDKEYPHRASE** option specifies whether to use an existing key-phrase

Parameters

file for the current build.

yes/no

Comments

Specifies whether the existing file should be used.

The Help compiler creates a phrase-table file with a .PH extension if it doesn't find one in the project root directory. If the Help compiler finds a file with a .PH extension, it uses the file for the current build. Because the .PH file speeds up the compression process when little text has changed since the last build, you might want to keep the phrase file around if you compile the same Help file several times with compression. However, you get maximum compression if you delete

the .PH file before starting each build.

Microsoft Windows Help Authoring Guide If you do not include this option in the Help project file, the Help compiler will

Example

use the old keyphrase file by default.

The following example illustrates this option:

See Also

OLDKEYPHRASE=NO

[OPTIONS] Section

Syntax

OPTCDROM Option

OPTCDROM=*yes/no*

The **OPTCDROM** option optimizes a Help file for CD-ROM display by aligning

Parameters

topic files on predefined 2K block boundaries.

yes/no

Comments

Specifies whether the file should be optimized for CD-ROM.

The CD-ROM optimization allows Windows Help to read data from the CD-ROM drive faster and more efficiently. On average, sequential reads from the CD are twice as fast when the topics are aligned using the **OPTCDROM** option.

The disadvantage to using the **OPTCDROM** option is that it slightly increases the size of the built Help file (approximately 10K). If you are delivering your Help file on CD-ROM, the extra file size should not be significant, given the large storage capacity of a CD-ROM disc.

Using the **OPTCDROM** option will also improve performance when Help files are read from a standard hard disk; however, because of the size increase you may not want to use this option if you are shipping your Help file on floppy disks. Help authors must evaluate the size difference and performance improvement of

Example

their individual files to decide whether to use this option.

The following example illustrates this option:

See Also

OPTCDROM=YES

[OPTIONS] Section

Syntax

[OPTIONS] Section

[OPTIONS]

option

-
-
-

The **[OPTIONS]** section includes options that control how a Help file is built and what feedback the build process displays. If you include this section in the Help project file, list it first so that the options apply during the entire build process.

Parameters

option

Specifies one of the following project-file options.

Option	Description
BMROOT	Specifies the directory containing the bitmap files named in bmc , bml , and bmr references in the Help topic files.
BUILD	Determines which topics to include in the build.
CITATION	Adds a unique copyright message whenever users copy information in the Help file using the Copy command.
COMPRESS	Specifies the type of compression to use during the build.
CONTENTS	Specifies the context string of the Help file's Contents topic.
COPYRIGHT	Adds a unique copyright message for the Help file to the About dialog box.
ERRORLOG	Puts compilation errors in a file during the build.
FORCEFONT	Forces all authored fonts in the topic files to appear in a different font when displayed in the Help file.

ICON Specifies the icon file to be displayed when the Help file is minimized.
The Help Project File§ 16-39

LANGUAGE Specifies a different sorting order for Help files authored in a Scandinavian language.

MAPFONTSIZE Maps a font size in the topic file to a different font size in the compiled Help file.

MULTIKEY Specifies an alternate keyword table to use for mapping topics.

OLDKEYPHRASE Specifies whether the compiler should use the existing key phrase table or create a new one during the build.

OPTCDROM Optimizes the Help file for CD-ROM use.

REPORT Controls the display of messages during the build process.

ROOT Specifies the directories containing the topic and data files listed in the Help project file.

TITLE Specifies the text that is displayed in the title bar of the Help window when the file is open.

WARNING Indicates 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

Example

[OPTIONS] section is not required.

The following shows the [OPTIONS] section in a sample Help project file:

```
[OPTIONS]
ERRORLOG=CARD.LOG
ROOT=C:\HELP\PROJECT
BMROOT=C:\HELP\PROJECT\BMP1;C:\HELP\PROJECT\BMP2;C:\HELP\PROJECT\BMP3;
TITLE=My Help File
CONTENTS=IDX_CONTENTS
COMPRESS=HIGH
OLDKEYPHRASE=NO
ICON=C:\HELP\PROJECT\BMP2\HLPFILE.ICO
WARNING=3
REPORT=ON
```

This sample [OPTIONS] section indicates the following to the Help compiler:

- Write messages displayed during the build to the CARD.LOG file.
- The project directory is C:\HELP\PROJECT.
- Bitmap files reside in three subdirectories off of the project root directory: \BMP1, \BMP2, and \BMP3.
- The title bar caption for the Help file is My Help File.
- The context string of the Help file's Contents topic is IDX_CONTENTS.
- Use high compression during the build.
- Do not use the old key-phrase table during compression.
- Use HLPFILE.ICO as the minimized Help icon.
- Report all errors and warnings.
- Display messages throughout the build to indicate when processes are occurring.

BMROOT Option, BUILD Option, CITATION Option, COMPRESS Option, CONTENTS Option, COPYRIGHT Option, ERRORLOG Option, FORCEFONT Option, ICON Option, LANGUAGE Option, MAPFONTSIZE Option, MULTIKEY Option, OLDKEYPHRASE Option, OPTCDROM Option, REPORT Option, ROOT Option, TITLE Option, WARNING Option

Syntax

REPORT Option

REPORT=*on/off*

The **REPORT** option displays messages on the screen during the build. These messages indicate when the compiler is performing the different phases of the build, including scanning the file for compression, compiling the file, verifying

Parameters

context strings, and resolving jumps, keywords, and browse sequences.

on/off

Specifies whether the compiler should display messages on the

Comments

screen during the build.

Unlike the **ERRORLOG** option, messages displayed by using the **REPORT** option are not written to a file.

Example

The following example illustrates this option:

See Also

REPORT=ON

[OPTIONS] Section, WARNING Option

ROOT Option

Syntax

ROOT=*pathname*[, *pathname*]..

The **ROOT** option specifies the project root directory where the Help compiler

Parameter

looks for the source files to include in the build.

pathname

Specifies either a drive and full path or a relative path from the project directory. If you specify more than one project directory, a comma (,) separates each path. The entire **ROOT** entry can be up to 128 characters long. If the path is invalid or too long, the compiler reports an error. If the Help project file has a **ROOT** option, all relative paths in the Help project file refer to one of these paths. If the Help project file does not have a **ROOT** option, all paths are

Comments

relative to the directory containing the Help project file.

If the Help project file does not have a **BMROOT** option, the compiler looks in the directories specified in the **ROOT** option to find bitmaps positioned by using the **bmc**, **bml**, and **bmr** references. If the Help project file doesn't have a **ROOT**

option or if none of the **ROOT** directories contain the bitmaps, the bitmap

The Help Project File§ 16-43

Example

filenames must be listed in the [BITMAPS] section of the Help project file.

The following entry specifies that the project root directory is C:\HELP\PROJECT and is found on drive C:

```
[OPTIONS]
ROOT=C:\HELP\PROJECT
```

Given this root directory, topic files can be listed relative to the project root directory in the [FILES] section:

```
[FILES]
\RTF\TOPICS1.RTF
\RTF\TOPICS2.RTF
\RTF\TOPICS3.RTF
\RTF\TOPICS4.RTF
\RTF\TOPICS5.RTF
```

See Also

The full path for a topic file is C:\HELP\PROJECT\RTF\TOPICS1.RTF.

[BITMAPS] Section, BMROOT Option, [OPTIONS] Section

Syntax

TITLE Option

TITLE=*titlename*

The **TITLE** option assigns a title to the Help file. Windows Help displays this title in the title bar of the Help window whenever it displays the Help file.

Parameter *titlename*

Specifies the title displayed in the title bar of the Help window. The

Comments

title can have as many as 50 characters.

If you do not specify a title using the **TITLE** option, the title bar displays

Example

Microsoft Help.

The following example sets the Help file title to Cardfile Help:

[OPTIONS]

See Also

TITLE=Cardfile Help

[OPTIONS] Section

Syntax

WARNING Option

WARNING=level

The **WARNING** option specifies the amount of debugging information that the

Parameter

compiler is to report during the build.

level

Specifies the warning level. This parameter may be any one of the

following values.

Value	Meaning
--------------	----------------

1	Report only the most severe errors.
---	-------------------------------------

2	Report an intermediate number of errors.
---	--

3	Report all fatal errors and warnings.
---	---------------------------------------

Comments

This option only specifies the amount of information to report. It should be used in conjunction with the **REPORT** option and the **ERRORLOG** option if you

Example

want to see the error messages on the screen and save them in a file.

The following example specifies an intermediate level of error reporting:

[OPTIONS]

See Also

WARNING=2

ERRORLOG Option, [OPTIONS] Section, REPORT Option

Syntax

[WINDOWS]

window-name="caption", (x-coord, y-coord, width, height), window-state, (scrolling-RGB), (nonscrolling-RGB), ontop-state

.
. .
.

The [WINDOWS] section defines the size, location, and colors for the primary Help window and any secondary window types used in a Help file. You can

Parameters

define as many as five secondary window types.

window-name

Specifies the name of the 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 (other than **main**) with as many as eight characters. Any jumps that display a topic in a secondary window give this window name as part of the jump.

caption

Specifies the text that appears in the title bar of the window. The caption can have as many as 50 characters. If no caption is specified, Windows Help is displayed. Generally, to set the caption of the main window, you use the **TITLE** option in the [OPTIONS] section of the Help project file. If the caption specified here differs from the **TITLE** option, the [WINDOWS] caption determines the title displayed in the title bar.

x-coord

Specifies the x-coordinate, in Help units, of the window's upper-left corner. The horizontal position is defined in terms of Windows Help's 0—1023 coordinate system. (Help always assumes the screen is 1024 units wide, regardless of resolution. For information about how to determine actual coordinates for different video resolutions, see the "Comments" section.) The x-coordinate is relative to the upper-left corner of the screen, which is 0,0.

y-coord

Specifies the y-coordinate, in Help units, of the window's upper-left corner. The vertical position is defined in terms of Help's 0—1023 coordinate system. (Help always assumes the screen is 1024 units wide, regardless of resolution. For information about how to determine actual coordinates for different video resolutions, see the "Comments" section.) The y-coordinate is relative to the upper-left corner of the screen, which is 0,0.

width

Specifies the window's default width in Help's 0—1023 coordinate system.

height

Specifies the window's default height in Help's 0—1023 coordinate system.

window-state

Specifies how the window is sized when Windows Help first opens it. This parameter can be one of the following values.

Value Meaning

0 Size the window according to the values specified in the *x-coord*, *y-coord*, *width*, and *height* parameters. The default is 0 or normal size.

1 Maximize the window, and ignore the *x-coord*, *y-coord*, *width*, *height*, and *ontop-state* parameters given in the type definition.

scrolling-RGB

Specifies the background color for the window's scrolling region. Colors are given as standard RGB values—where RRR, GGG, and BBB are three-digit numbers in the range 0 to 255 representing the red, green, and blue components of the color. If this parameter is not given, Help uses the default Windows system color specified by the

end-user in Control Panel.

Microsoft Windows Help Authoring Guide *nonscrolling-RGB*

Specifies the background color for the window's nonscrolling region (if any). Colors are given as standard RGB values—where RRR, GGG, and BBB are three-digit numbers in the range 0 to 255 representing the red, green, and blue components of the color. If this parameter is not given, Help uses the default Windows system color specified by the end-user in Control Panel.

ontop-state

Specifies whether a secondary window type stays on top of other windows. The main Help window cannot be authored as a topmost window. This parameter can be one of the following values.

Value Meaning

0	Display the window normally, not on top of other windows. The default is 0 or normal behavior.
1	Display the window on top of other windows. If this value is given, the user cannot change the window behavior using the Always On Top command in Help.

Comments

Window attributes defined in the [WINDOWS] section follow these rules:

- A single comma may be substituted for an entry or a group of entries enclosed by parentheses. Preceding commas are required if you want to use the default settings. Trailing commas are optional. (See the “main” example below.)
- Windows attributes are set for the Help windows whenever a file containing predefined attributes is opened.
- If a user performs an operation, such as an interfile jump, that opens

a file with predefined window attributes, the Help windows will adjust to the new settings, — change size or location, for example.

- n When an application requests Help using the **WinHelp** function, Help uses the values defined in the [WINDOWS] section to determine window attributes. If the Help file does not include settings in the Help project file, Help uses the default settings.
- n When Help is closed, the window settings are updated in the [Windows Help] section of the WIN.INI file.

The Help coordinate system ranges from 0 through 1023 in both directions, so the vertical position plus the height must be less than or equal to 1023. Similarly, the horizontal position plus the width must be less than or equal to 1023. This 1024-by-1024 coordinate system is mapped to the horizontal and vertical resolutions of the video card. To convert from pixels to Windows Help coordinates, you invert the ratio between Help's resolution and the video resolution. Assuming the video card resolution is *horiz* by *vert* pixels, and the horizontal and vertical locations (or dimensions) you want are in pixels, the x-coordinate (or width), in Help coordinates, is as follows:

$$x\text{-coord} = \text{pixel location} * (1024/\text{horiz})$$

$$\text{width} = \text{number of pixels} * (1024/\text{horiz})$$

The y-coordinate (or height), in Help coordinates, is:

$$y\text{-coord} = \text{pixel location} * (1024/\text{vert})$$

$$\text{height} = \text{number of pixels} * (1024/\text{vert})$$

For example, if you want the window's upper-left corner to appear at horizontal pixel 320 and at vertical pixel 120, and the Help file is being displayed on a standard VGA monitor with 640 by 480 resolution, the *x-coord* is $(320 * (1024/640)) = 512$, and *y-coord* is $(120 * (1024/480)) = 256$.

Example

$(1024/640)) = 512$, and *y-coord* is $(120 * (1024/480)) = 256$.

The following example defines two windows, the main Help window and a secondary window named "picture." The main window definition sets the background color to magenta (128, 0, 128) and leaves the other values empty (for which Help will supply its default values). The secondary window definition sets the caption to Samples, positions it in the upper-left part of the screen (123, 123),

and sets the window's width and height to about one-quarter of the screen's width and height (256). The window will not be maximized (0). The window's background colors are set to cyan (0, 255, 255) for the scrolling region and light gray (192, 192, 192) for the nonscrolling region. The secondary window will appear on top of other windows when it is open (1).

[WINDOWS]
main=, , (128, 0, 128)

See Also

picture="Samples", (123, 123, 256, 256), 0, (0, 255, 255), (192, 192, 192), 1

[OPTIONS] Section, TITLE Option
