Help Error Messages

When you build a Help file, the compiler displays messages on the screen, indicating the progress of the build and any errors or problems that may occur. This chapter describes the build and error messages for the version 3.1 and 3.0

Help Compiler 3.1 Error Messages

Help compilers.

This section lists the error messages that the version 3.1 Help compiler version 3.1 displays when it encounters problems building a Help file. Whenever possible, the Help compiler displays the name of the file that contains the error, as well as the number used to identify the specific line of the Help project file or the topic that produced the error. Since topics do not contain numbers, the topic number specified in the error message refers to the topic's sequential order in the topic file. In most cases, the topic number given in the error message will match the topic's page number in the Word for Windows document.

The Error Message File

In Windows Help version 3.1, error messages are stored in a text-only ASCII file, rather than in the compiler's resource file (as in version 3.0). Because Help error messages are stored independently of the compiler, Help authors can customize the error messages in two ways. They can:

◻ Change the text that appears in the error message.

◻ Insert tabs into the error messages.

When customizing error messages, Help authors cannot:

◻ Change the number assigned to the error message.

◻ Change the order of the error message in HC.ERR.

◻ Add, delete, or change the order of parameter (%) tags in the error messages.

The default name of the error message file is HC.ERR; however, this name may vary according to the specific version of the compiler that you are using to build the Help file: HC.ERR, HC31.ERR, or HCP.ERR, for example. To work correctly, the error message file must be located in the same directory as the compiler and its name must match the name of the compiler you are using. For example, if you rename HC31.EXE to HC.EXE, you must rename the HC31.ERR error file to HC.ERR. If the error message file is missing, the build will fail, and the compiler will display this message:

Warning 0101: Cannot open *filename***.ERR in** *filename***.EXE directory.**

If you see this error message, check to be sure you are running matched versions of the compiler and error message file and that the files are in the same directory. Also, be sure you are using the version of the compiler you think you are using. If the problem persists, reinstall the error message file from the original source disk.

Interpreting Error Messages

The Help compiler displays either warning or fatal error messages. A warning message begins with the word **Warning**. A warning error indicates a problem encountered during the build that is not severe enough to prevent the Help compiler from completing the build. Therefore, a build with warnings produces a Help file that Windows Help should be able to open, but the file may contain problems in certain topics, such as missing graphics or hot spots that aren't hot. Fatal error messages begin with the word **Error**. A fatal error indicates a problem that prevents the Help compiler from creating a Help file.

As stated previously, you use the **REPORT** option to display error messages on the screen and the **WARNING** option to specify the amount of warning information you want the Help compiler to provide. The Help compiler always reports fatal errors, regardless of the current warning level or report option specified in the Help project file, since no usable Help file results from the build.

While the Help compiler processes the Help project file, it ignores lines that contain errors and attempts to continue with the build. This means that errors encountered early in a project file may result in many more errors being reported as the build continues.

Similarly, when the Help compiler processes the RTF topic files, it reports any errors it encounters and, if the errors are not fatal, the compiler continues with the build. A single error in a topic file may result in more than one error message being displayed by the compiler. For instance, a typographic mistake in a topic's

context string will cause an error to be reported every time the compiler encounters a reference to the correct topic identifier. On the other hand, some warning errors—such as hot spots with the same hidden text codes (broken jumps) or any bitmap file errors, including "File not found"— are reported once per file, regardless of the number of times the problem occurs in the source file.

Error Message Categories

In Windows Help version 3.0, error messages included the prefixes P and R to indicate whether the error occurred in the topic file or in the Help project file. In Help version 3.1, theses prefixes have been replaced by a numbering scheme that differentiates build errors into eight categories.

Error message numbers have four digits. The first one or two digits identify the message category. The third and fourth digits establish the message order within the category. The message number prefixes and the categories they identify are listed in the following table.

Prefix	Error category
1	Problems with files used to build the Help file
2	Problems with the Help project file
30–31	Problems with build tags or build tag expressions
35–36	Problems with Help macros
40–41	Problems with context strings

You might encounter errors other than the build errors described here. For more information, see the README.TXT file.

Error Message Reference

The Error Message Reference lists all error messages by category and in the order that they appear in the HC.ERR file. The wording of the error messages is also the same as in HC.ERR, except that variables (% tags) are replaced with descriptive words. Error message descriptions contain one or more of the following types of information.

Heading	Information
Problem	Explains what the error message means.
Result	Describes how the Help compiler handles this particular error.
Solution	Offers suggestions about how to correct the problem.

File Errors

The following messages result from problems with the files used to build the Help file.

1019 Project file extension cannot be .HLP or .PH.

PROBLEM: You cannot specify a Help project file with an.HLP or a .PH extension. Project files must use the .HPJ extension.

RESULT: The compiler aborts the build.

SOLUTION: Rename the Help project file, and then recompile.

1030 File name exceeds limit of 259 characters.

PROBLEM: The combined length of the path and filename must not be more than the MS-DOS limit of 259 characters.

RESULT: The compiler ignores the file.

SOLUTION: Shorten the path, and then recompile.

1079 Out of file handles.

PROBLEM: The compiler does not have enough available file handles to continue the build. This error is likely to occur if you are compiling in an MS-DOS box or if you are using Help Author and have Word for Windows running at the same time.

RESULT: The compiler aborts the build.

SOLUTION: If possible, increase the **FILES** setting in the CONFIG.SYS file to **Files=50**. Then reboot your computer and recompile.

1100 Cannot open file *filename*: permission denied.

PROBLEM: You do not have the required file privileges to open the requested file. Requested files must have at least read privilege to be opened.

RESULT: The compiler ignores the file.

SOLUTION: Change your access privileges if you are working on a secured

network, or change the file attributes to read only or read/write.

1150 Cannot overwrite file *filename***.**

PROBLEM: The Help compiler cannot overwrite files with a read-only attribute. For example, this error occurs if you rebuild an existing .HLP file and the .HLP file is read-only.

RESULT: The compiler aborts the build.

SOLUTION: Change the file's read-only attribute if you want the Help compiler to overwrite the file; otherwise, rename the file.

1170 File *filename* **is a directory.**

PROBLEM: A directory in the Help project directory has the same name as the requested Help file. This is an MS-DOS file error.

RESULT: The compiler ignores the file.

SOLUTION: Move or rename the directory or the Help project file, and then recompile.

1190 Cannot use reserved MS-DOS file name *filename***.**

PROBLEM: The file has a reserved MS-DOS filename, such as COM1, LPT2, or PRN. This is an MS-DOS file error.

RESULT: The compiler ignores the file.

SOLUTION: Rename the file, and then recompile.

1230 File *filename* **not found.**

PROBLEM: The specified file could not be found or is unreadable. This is an MS-DOS file error or an out-of-memory condition.

RESULT: If the file not found is an RTF topic file, the compiler ignores the file. If the file is a bitmap file, the built Help file displays the "Unable to display picture" message in the topic instead of the bitmap.

SOLUTION: Check to see if the file exists, and also check the amount of available memory.

1292 File *filename* **is not a valid bitmap.**

PROBLEM: The specified bitmap file could not be found or is not in a recognizable bitmap format. This is an MS-DOS file error or an out-of-

memory condition.

RESULT: The Help file displays the "Unable to display picture" message in the compiled topic instead of the bitmap.

SOLUTION: Check to see if the file exists; if it does, check its format. If necessary save the file again in your paint or draw program, and then recompile.

1319 Disk full.

PROBLEM: There is not enough disk space to write the Help file to disk.

RESULT: The compiler aborts the build.

SOLUTION: Create more space on the destination disk, and then recompile.

1513 Bitmap name *filename* duplicated.

PROBLEM: The [BITMAPS] section contains duplicate bitmap names.

RESULT: The compiler uses the first occurrence of the name.

SOLUTION: Rename one of the duplicate bitmap files, and then recompile.

1536 Not enough memory to compress bitmap *filename*.

PROBLEM: The specified bitmaps cannot be compressed due to insufficient memory, or the compiler does not have enough memory to check bitmap files. The compiler also displays this error if any of the specified bitmaps are segmented hypergraphics (.SHG files), since the .SHG files may contain searchable hot spots.

RESULT: The Help compiler does not check or compress the bitmap files.

SOLUTION: Free more memory and then recompile. Refer to the "Getting Around Memory Problems" section in Chapter 17 for suggestions on how to free more memory for the build. If you can't overcome your memory constraints, you may have to compile without compression or remove some of the graphics in the topic files.

1000 UNKNOWN ERROR, Contact Microsoft Product Support Services

PROBLEM: This error message is the equivalent of an assertion failure.

RESULT: The compiler aborts the build.

SOLUTION: Contact Microsoft Product Support Services.

Project-File Errors

The following messages result from errors in the Help project file used to build a Help file.

2010 Include statements nested more than 5 deep.

PROBLEM: The **#include** statement on the specified line exceeds the maximum of five include levels.

RESULT: The compiler aborts the build.

SOLUTION: Simplify the #**include** statement so that it nests fewer than include levels, and then recompile.

Syntax Errors

The following messages result from syntax errors in the Help project file.

2030 Comment starting at line *linenumber* **of file** *filename* **unclosed at end of file.**

PROBLEM: The compiler has unexpectedly come to the end of the Help project file. There may be an open comment in the Help project file or in an include file.

RESULT: The compiler aborts the build.

SOLUTION: Close any open comments within the Help project file, and then recompile.

2050 Invalid #include syntax.

PROBLEM: The **#include** statement is missing the required filename.

RESULT: The compiler ignores the **#include** statement.

SOLUTION: Correct the syntax, and then recompile. The correct **#include** syntax is:

#include <filename>

2091 Bracket missing from section heading [*sectionname***].**

PROBLEM: The right bracket (]) is missing from the specified section heading.

RESULT: The compiler ignores the section heading on this line.

SOLUTION: Insert the right section heading bracket, and then recompile.

2111 Section heading missing.

PROBLEM: The section heading on the specified line is not complete, or the first entry in the Help project file is not a section heading.

RESULT: The compiler ignores the section heading on this line and all succeeding lines until it encounters a valid section heading.

SOLUTION: Insert a valid section heading at the appropriate place in the Help project file, and then recompile.

2131 Invalid OPTIONS syntax: 'option=value' expected.

PROBLEM: An entry in the [OPTIONS] section is missing the equal sign, or it has some other syntax error.

RESULT: The compiler ignores the invalid option on this line.

SOLUTION: Check the syntax of each option used in the [OPTIONS] section. Fix the incorrect option, and then recompile. For the correct syntax, refer to Chapter 16, "The Help Project File."

2141 Invalid ALIAS syntax: 'context=context' expected.

PROBLEM: An entry in the [ALIAS] section is missing the equal sign, or it has some other syntax error.

RESULT: The compiler ignores the invalid alias string on this line.

SOLUTION: Check the syntax of the entries in the [ALIAS] section. Fix the incorrect entry, and then recompile. For the correct syntax, refer to Chapter 16, "The Help Project File."

2151 Incomplete line in [*sectionname*] section.

PROBLEM: An entry in the specified section is incomplete.

RESULT: The compiler ignores the incomplete line.

SOLUTION: Check the section for any incomplete lines and correct them. Then recompile.

2171 Unrecognized text.

PROBLEM: The compiler found some text in the Help project file that it can't

recognize.

RESULT: The compiler ignores the line with the unrecognized text.

SOLUTION: Remove the unsupported text from the Help project file, and then recompile.

2191 Section heading [*sectionname*] unrecognized.

PROBLEM: The Help project file includes a section heading that the compiler does not recognize or does not support.

RESULT: The compiler ignores the section heading.

SOLUTION: Check all bracketed headings in the Help project file for typographical or other errors. Fix the section heading, and then recompile.

2214 Line in .HPJ file exceeds length limit of 2047 characters.

PROBLEM: There is a line in the Help project file that exceeds the maximum length of 2047 characters.

RESULT: The compiler ignores the line.

SOLUTION: Shorten the line, and then recompile.

General Section Errors

The following messages result from general errors in the different sections of the Help project file.

2273 [OPTIONS] should precede [FILES] and [BITMAPS] for all options to take effect.

PROBLEM: The [OPTIONS] section should be the first section in the Help project file so that all the options will be used during compilation. Also, if the **ERRORLOG** option is used, it should be the first line in the [OPTIONS] section.

RESULT: The Help compiler does not use the **ROOT** or **BMROOT** option to locate topic or bitmap files that are listed before the [OPTIONS] section. Also, if the **ERRORLOG** option is used, the compiler does not write in the log file any errors that were generated before it processed the **ERRORLOG** option.

SOLUTION: Move the [OPTIONS] section to the beginning of the Help project file, and then recompile.

2291 Section *sectionname* **previously defined.**

PROBLEM: The compiler found more than one section with the specified heading in the Help project file.

RESULT: The compiler ignores the duplicate section and continues from the next valid section heading.

SOLUTION: Remove or rename one of the duplicate section headings, and then recompile.

2305 No valid files in [FILES] section.

PROBLEM: The [FILES] section is empty or contains only invalid files.

RESULT: The compiler aborts the build.

SOLUTION: Check the [FILES] section to see if it lists any files. If it does, make sure that the files listed are saved as RTF. After you have a list of valid topic files to include in the build, recompile.

[ALIAS] or [MAP] Section Errors

The following messages result from errors in the [ALIAS] or [MAP] section of the Help project file.

2322 Context string *context_name* **cannot be used as alias string.**

PROBLEM: A context string that has been assigned an alias cannot be used later as an alias for another context string. That is, you cannot map **a=b** and then **c=a** in the [ALIAS] section.

RESULT: The compiler ignores the attempted reassignment on this line.

SOLUTION: Correct the alias string mapping, and then recompile.

2331 Context number already used in [MAP] section.

PROBLEM: A context number in the [MAP] section of the Help project file was previously mapped to a different context string.

RESULT: The compiler ignores the line with the duplicate context number.

SOLUTION: Check all context numbers to make sure that they are unique. Remove or reassign any duplicate context numbers, and then recompile.

2341 Invalid or missing context string.

PROBLEM: A line in the [MAP] or [ALIAS] section is missing a context
string before the equal sign.

RESULT: The compiler ignores the line with the invalid context string.

SOLUTION: Check all context strings to the left of the equal sign. Correct any
missing or invalid entries, and then recompile.

2351 Invalid context identification number.

PROBLEM: A context number in the [MAP] or [ALIAS] section is empty or
contains invalid characters.

RESULT: The compiler ignores the line with the invalid context number.

SOLUTION: Check the format of all context numbers. Context numbers can
be specified as decimal or hexadecimal. Correct any invalid numbers, and then
recompile.

2362 Context string *context_name* **already assigned an alias.**

PROBLEM: The specified context string has already been assigned an alias in
the [ALIAS] section. A context string can have only one alias. That is, you
cannot map **a=b** and then **a=c** in the [ALIAS] section.

RESULT: The compiler ignores the attempted reassignment on this line.

SOLUTION: Correct the alias string mapping, and then recompile.

2372 Alias string *aliasname* **already assigned.**

PROBLEM: You cannot alias an alias. That is, an alias string cannot, in turn,
be assigned another alias. You cannot map **a=b** and then **b=c** in the [ALIAS]
section.

RESULT: The compiler ignores the attempted reassignment on this line.

SOLUTION: Correct the alias string mapping, and then recompile.

[WINDOWS] Section Errors

The following messages result from errors in the definitions of secondary
window types given in the [WINDOWS] section of the Help project file.

2391 Limit of 6 window definitions exceeded.

PROBLEM: The maximum number of window definitions is one main-window definition and five secondary-window definitions.

RESULT: The compiler ignores the additional window definitions.

SOLUTION: Reduce the number of window definitions in the Help project file to five or fewer, and then recompile.

2401 Window maximization state must be 0 or 1.

PROBLEM: The value of the *window-state* parameter is not zero or 1.

RESULT: The compiler ignores the window definition.

SOLUTION: Correct the entry in the Help project file, and then recompile. The *window-state* value must be either 0 (normal) or 1 (maximized).

2411 Invalid syntax in window color.

PROBLEM: A window definition in the Help project file has incorrectly defined a window color. A window color definition consists of three decimal numbers, representing the red, green, and blue (RGB) components of the color. The three numbers must be enclosed in parentheses and separated by commas.

RESULT: The compiler ignores the window definition.

SOLUTION: Correct the syntax, and then recompile. The correct syntax for a window color is:

(RRR, GGG, BBB)

2421 Invalid window position.

PROBLEM: The window position in a window definition consists of four decimal numbers that define the window's location on the screen and its width and height. The four numbers must be given in Help's 1024-by-1024 coordinate system, and they must be enclosed in parentheses and separated by commas.

RESULT: The compiler ignores the window definition.

SOLUTION: Correct the syntax, and then recompile. The correct syntax to indicate the predefined window position is:

(x-coord, y-coord, width, height)

2431 Missing quote in window caption.

PROBLEM: The window caption in a window definition must be enclosed in quotation marks.

RESULT: The compiler ignores the window definition.

SOLUTION: Add the missing quotation marks, and then recompile.

2441 Window name *windowname* **is too long.**

PROBLEM: The window name exceeds the maximum length of eight characters.

RESULT: The compiler ignores the window definition.

SOLUTION: Shorten the window name to eight or fewer characters, and then recompile.

2451 Window position value out of range 0–1023.

PROBLEM: One or more of the window position coordinates (*x-coord, y-coord, width, height*) exceed the limit of 1023.

RESULT: The compiler ignores the window definition.

SOLUTION: Correct the numeric values of the window position coordinates so that they fall within the valid range (0–1023), and then recompile.

2461 Window name missing.

PROBLEM: A window definition in the Help project file is missing the window name.

RESULT: The compiler ignores the window definition.

SOLUTION: Specify a name for the window, and then recompile.

2471 Invalid syntax in [WINDOWS] section.

PROBLEM: The entry for a main or a secondary window is incorrect.

RESULT: The compiler ignores the invalid window definition.

SOLUTION: Check the syntax of each window definition, and then recompile. The correct syntax for a window definition is:

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

(nonscrolling-RGB), ontop-state

2481 Secondary window position required.
PROBLEM: A window definition for a secondary window must specify all four window position parameters—*x-coord, y-coord, width,* and *height*—in the Help project file.

RESULT: The compiler ignores the window definition.

SOLUTION: Supply the missing window position parameters, and then recompile.

2491 Duplicate window name *windowname***.**

PROBLEM: The [WINDOWS] section contains window definitions with duplicate window names.

RESULT: The compiler ignores the duplicate window definition.

SOLUTION: Check the uniqueness of each window name, and then recompile.

2501 Window caption *windowcaption* **exceeds limit of 50 characters.**

PROBLEM: The caption for the window exceeds the limit of 50 characters.

RESULT: The compiler ignores the window definition.

SOLUTION: Shorten the window caption, and then recompile.

[OPTIONS] Section Errors

The following error messages are caused by problems in the [OPTIONS] section of the Help project file.

2511 Unrecognized option *optionname* **in [OPTIONS] section.**

PROBLEM: The compiler does not recognize or support the specified option.

RESULT: The compiler ignores the unrecognized option.

SOLUTION: Check the [OPTIONS] section for typing errors or invalid options, and then recompile.

2532 Option *optionname* **previously defined.**

PROBLEM: The specified option has been defined on a previous line.

RESULT: The compiler uses the first definition and ignores the attempted redefinition.

SOLUTION: Remove one of the duplicate options, and then recompile.

ROOT Option Errors

The following error messages are caused by problems with the **ROOT** or the **BMROOT** option in the [OPTIONS] section of the Help project file.

2550 Invalid path *pathname* **in** *optionname* **option.**

PROBLEM: The compiler cannot find the path specified by the **ROOT** or **BMROOT** option, or the entry is more than 128 characters long.

RESULT: The compiler uses the current working directory.

SOLUTION: Correct or shorten the path in the **ROOT** or **BMROOT** option, and then recompile.

2570 Path in *optionname* **option exceeds** *number* **of characters.**

PROBLEM: The specified root path exceeds the limit of 128 characters.

RESULT: The compiler ignores the path and uses the current working directory.

SOLUTION: Shorten the path, and then recompile.

MAPFONTSIZE Option Errors

The following error messages are caused by problems with the **MAPFONTSIZE** option in the [OPTIONS] section of the Help project file.

2591 Invalid MAPFONTSIZE option.

PROBLEM: The font range syntax used is invalid. A font range consists of a low and high point size, separated by a hyphen (-).

RESULT: The compiler ignores the option.

SOLUTION: Correct the syntax, and then recompile. The correct syntax is:

MAPFONTSIZE=m[-n]:p

2612 Maximum of 5 font ranges exceeded.

PROBLEM: The maximum number of font ranges that can be specified is five.

RESULT: The compiler ignores additional ranges.

SOLUTION: Remove the additional font ranges so that there are five or fewer, and then recompile.

2632 Current font range overlaps previously defined range.

PROBLEM: A font size range overlaps a previously defined mapping.

RESULT: The compiler uses the first range and ignores the second mapping.

SOLUTION: Adjust one or both of the font ranges to remove any overlapping.

FORCEFONT Option Errors

The following error messages are caused by problems with the **FORCEFONT** option in the [OPTIONS] section of the Help project file.

2651 Font name exceeds limit of 20 characters.

PROBLEM: Font names cannot exceed 20 characters.

RESULT: The compiler ignores the option on this line.

SOLUTION: Shorten the font name to 20 or fewer characters, and then recompile.

2672 Unrecognized font name *fontname* **in FORCEFONT option.**

PROBLEM: The compiler has encountered a font name that it does not recognize or support.

RESULT: The compiler ignores the font name and uses the default MS Sans Serif font.

SOLUTION: Change the font name to a supported font, and then recompile. For the list of supported fonts, refer to Chapter 16, "The Help Project File."

MULTIKEY Option Errors

The following error messages are caused by problems with the **MULTIKEY** option in the [OPTIONS] section of the Help project file.

2691 Invalid MULTIKEY option.

PROBLEM: The **MULTIKEY** option must specify a single capital letter other than the letter K.

RESULT: The compiler ignores the attempted keyword table assignement on this line.

SOLUTION: Correct the syntax, and then recompile. The correct syntax is:

MULTIKEY=char

2711 Maximum of 5 keyword tables exceeded.

PROBLEM: The [OPTIONS] section contains more than five **MULTIKEY** entries, which exceeds the limit of five keyword tables.

RESULT: The compiler ignores the additional keyword tables.

SOLUTION: Reduce the number of keyword tables to five or fewer, and then recompile.

2732 Character already used.

PROBLEM: A character used for indicating the alternate keyword table (**MULTIKEY=***char*) was previously used.

RESULT: The compiler ignores the duplicate keyword table assignment on this line.

SOLUTION: Change the duplicate character to one that has not been used, and then recompile.

2752 Characters 'K' and 'k' cannot be used.

PROBLEM: These characters are reserved for Help's standard keyword table.

RESULT: The compiler ignores the attempted keyword table assignment on this line.

SOLUTION: Choose another character for the alternate keyword table, and then recompile.

Other [OPTIONS] Errors

The following error messages are caused by problems with other options in the [OPTIONS] section of the Help project file.

2771 REPORT option must be 'ON' or 'OFF'.

PROBLEM: The **REPORT** option specifies an incorrect value.

RESULT: The compiler uses the default value **off**.

SOLUTION: Correct the entry, and then recompile. The correct syntax is:

REPORT=*on/off*

2811 OLDKEYPHRASE option must be 'ON' or 'OFF'.

PROBLEM: The **OLDKEYPHRASE** option specifies an incorrect value.

RESULT: The compiler uses the default value **on**.

SOLUTION: Correct the entry, and then recompile. The correct syntax is:

OLDKEYPHRASE=[off, 0, no, false] or [on, 1, yes, true*]*

2832 COMPRESS option must be 'OFF', 'MEDIUM', or 'HIGH'.

PROBLEM: The **COMPRESS** option specifies an incorrect value.

RESULT: The compiler uses the default value **off**.

SOLUTION: Correct the entry, and then recompile. The correct syntax is:

COMPRESS=[off, 0, no, false] or [medium] or [on, 1, yes, true, high*]*

2842 OPTCDROM option must be 'TRUE' or 'FALSE'.

PROBLEM: The **OPTCDROM** option specifies an incorrect value.

RESULT: The compiler uses the default value **false**.

SOLUTION: Correct the entry, and then recompile. The correct syntax is:

OPTCDROM=[off, 0, no, false] or [on, 1, yes, true]

2852 Invalid TITLE option.

PROBLEM: The **TITLE** option defines a string that is empty or contains more than 32 characters.

RESULT: The compiler truncates the title at 32 characters.

SOLUTION: Add a title if the option is empty or shorten the title if it is too long, and then recompile.

2872 Invalid LANGUAGE option.

PROBLEM: You have specified an ordering that is not supported by the compiler.

RESULT: The compiler uses the default English (U.S.) sorting order.

SOLUTION: Correct the entry, and then recompile. The correct syntax is:

LANGUAGE=scandinavian

2893 Warning option must be 1, 2, or 3.

PROBLEM: The warning reporting level can be set only to 1, 2, or 3.

RESULT: The compiler uses full reporting (level 3).

SOLUTION: Correct the entry, and then recompile.

2911 Invalid icon file *filename*.

PROBLEM: The compiler cannot find the icon file specified in the **ICON** option, or the file is not a valid icon file.

RESULT: The compiler ignores the file.

SOLUTION: Check to see if the icon file exists and if it is a valid .ICO file. Or if the file is in not in a directory defined by the **ROOT** or **BMROOT** option, move the file so that the Help compiler can find it, and then recompile.

2932 Copyright string exceeds limit of 50 characters.

PROBLEM: The maximum length of the copyright string in the About box is 50 characters.

RESULTS: The compiler truncates the copyright string at 50 characters.

SOLUTION: Shorten the copyright notice, and then recompile.

2000 UNKNOWN ERROR, Contact Microsoft Product Support Services

PROBLEM: This error message is the equivalent of an assertion failure.

RESULT: The compiler aborts the build.

SOLUTION: Contact Microsoft Product Support Services.

Build Tag or Build Expression Errors

The following messages are caused by errors in build tag footnotes or build expressions in the [BUILDTAGS] section of the Help project file.

3011 Maximum of 32 build tags exceeded.

PROBLEM: The maximum number of build tags that can be defined is 32.

RESULT: The compiler ignores the additional tags.

SOLUTION: Reduce the number of build tags to 32 or fewer, and then recompile.

3031 Build tag length exceeds 32 characters.

PROBLEM: The build tag on the specified line exceeds the limit of 32 characters.

RESULT: The compiler ignores the build tag.

SOLUTION: Shorten the build tag to 32 or fewer characters, and then recompile.

3051 Build tag *tagname* **contains invalid characters.**

PROBLEM: Build tags can contain only alphanumeric characters or the underscore (_) character.

RESULT: The compiler ignores the line with invalid characters.

SOLUTION: Correct the build tag so that it contains only valid characters, and then recompile.

3076 [BUILDTAGS] section missing.

PROBLEM: The **BUILD** option declares a conditional build, but there is no [BUILDTAGS] section in the Help project file.

RESULT: The compiler includes all topics in the build.

SOLUTION: Add a [BUILDTAGS] section to the Help project file, if you want to perform a conditional build, and then recompile.

3096 Build expression too complex.

PROBLEM: The build expression has too many expressions (~, |, or &) or is nested too deeply.

RESULT: The compiler includes all topics in the build.

SOLUTION: Reduce the number of operators or nested levels, and then recompile. Refer to Chapter 16, "The Help Project File," for details.

3116 Invalid build expression.

PROBLEM: The syntax used in the build expression on the specified line contains one or more logical or syntax errors.

RESULT: The compiler ignores the line with the invalid expression.

SOLUTION: Correct the entry, and then recompile. For the correct syntax and information about build expressions, refer to Chapter 16, "The Help Project File."

3133 Duplicate build tag in [BUILDTAGS] section.

PROBLEM: A build tag appears more than once in the [BUILDTAGS] section.

RESULT: The compiler uses the first build tag and ignores the duplicate build tag.

SOLUTION: Remove or rename one of the duplicate build tags, and then recompile.

3152 Build tag *tagname* not defined in [BUILDTAGS] section.

PROBLEM: The specified build tag has been assigned to a topic but not declared in the Help project file.

RESULT: The compiler ignores the build tag defined in the topic.

SOLUTION: Include the build tag in the [BUILDTAGS] section, and then recompile.

3178 Build expression missing from project file.

PROBLEM: The topics have build tag footnotes, but there is no **BUILD=** expression in the Help project file.

RESULT: The compiler includes all topics in the build.

SOLUTION: Add a build expression to the Help project file and recompile if you want your build tags to work. Otherwise, you can safely ignore this message.

Macro Errors

The following messages result from errors in the use of Help macros in footnotes, hot spots, and the [CONFIG] section of the Help project file.

3511 Macro *macrostring* **exceeds limit of 254 characters.**

PROBLEM: The macro string exceeds the limit of 254 characters.

RESULT: The compiler passes the macro to the Help file, where, in most cases, it will cause the Help application to display an error when it is executed.

SOLUTION: Shorten the length of the macro string, and then recompile. In some cases, you may be able to substitute an abbreviation for the macro name to shorten the macro string. For the supported macro abbreviations, check Chapter 15, "Help Macros Reference."

3532 Undefined function in macro *macroname***.**

PROBLEM: The specified macro is not on the list of macros supported by the compiler, nor is it specified in the **RegisterRoutine** macro.

RESULT: The compiler passes the macro to the Help file, where, in most cases, it will cause the Help application to display an error when it is executed.

SOLUTION: If the function resides in a custom DLL, check the Help project file to make sure that you have registered the function using the **RegisterRoutine** macro. Otherwise, change the macro function to one that is supported or remove it, and then recompile.

3552 Undefined variable in macro *macroname***.**

PROBLEM: The specified macro contains a variable that is not recognized by the compiler.

RESULT: The compiler passes the macro to the Help file, where, in most cases, it will cause the Help application to display an error when it is executed.

SOLUTION: Change the macro so that it uses only defined variables, and then recompile. If the variable is defined in a custom DLL, check the **RegisterRoutine** macro in the Help project file to be sure that the variable is defined and that you are using the function correctly. Otherwise, check Chapter 15, "Help Macro Reference."

3571 Wrong number of parameters to function in macro *macroname***.**

PROBLEM: There are too many or too few parameters in the macro.

RESULT: The compiler passes the macro to the Help file, where, in most cases, it will cause the Help application to display an error when it is executed.

SOLUTION: Change the macro so that it uses the correct number of parameters, and then recompile. If the macro is defined in a custom DLL, check the **RegisterRoutine** macro in the Help project file for the correct number of parameters. Otherwise, check Chapter 15, "Help Macro Reference."

3591 Syntax error in macro *macroname*.

PROBLEM: The syntax of the macro is invalid.

RESULT: The compiler passes the macro to the Help file, where, in most cases, it will cause the Help application to display an error when it is executed.

SOLUTION: Correct the syntax, and then recompile. For the correct syntax of the macro you are using, refer to Chapter 15, "Help Macro Reference."

3611 Function parameter type mismatch in macro *macroname*.

PROBLEM: There is a type mismatch in the function call. In other words, a string is given for a numeric parameter or a number is given for a string parameter.

RESULT: The compiler passes the macro to the Help file, where, in most cases, it will cause the Help application to display an error when it is executed.

SOLUTION: Change the macro so that it uses the correct parameter types, and then recompile. If the macro is defined in a custom DLL, check the **RegisterRoutine** macro in the Help project file for the correct parameter types. Otherwise, check Chapter 15, "Help Macro Reference."

3631 Bad macro prototype.

PROBLEM: The prototype string passed to **RegisterRoutine** is invalid.

RESULT: The compiler passes the macro to the Help file, where, in most cases, it will cause the Help application to display an error when it is executed.

SOLUTION: Change the macro so that it uses a valid prototype string, and then recompile.

3652 Empty macro string.

PROBLEM: The ! footnote or a hidden text code starting with ! does not contain a valid macro.

RESULT: The compiler passes the macro to the Help file, where, in most cases, it will cause the Help application to display an error when it is executed.

SOLUTION: Correct the macro string in the topic footnote, and then recompile.

3672 Macro *macroname* **nested too deeply.**

PROBLEM: Macro strings may not contain more than three other macro strings as parameters.

RESULT: The compiler passes the macro to the Help file.

SOLUTION: Remove one or more of the nested macro strings, and then recompile. Or ignore the error if you want to include the macro in the Help file. The following macro correctly nests three macros:

IfThen(1, `IfThen(1, `IfThen(1, `BrowseButtons()')')')

The following macro string is nested too deeply:

IfThen(1, `IfThen(1, `IfThen(1, `IfThen(1, `BrowseButtons()')')')')

The Help application will not display an error message if a macro string is nested too deeply. Therefore, both of the above macros will work, even though one generates an error during compilation and one does not.

3000 UNKNOWN ERROR, Contact Microsoft Product Support Services

PROBLEM: This error message is the equivalent of an assertion failure.

RESULT: The compiler aborts the build.

SOLUTION: Contact Microsoft Product Support Services.

Context String Errors

The following messages are caused by problems with context string footnotes or with context strings specified in jumps or in Help project file options.

4011 Context string *contextname* **already used.**

PROBLEM: The specified context string was previously assigned to another topic.

RESULT: The compiler ignores the duplicate context string, and the topic has

no identifier.

SOLUTION: Change the second context string so that it is unique, and then recompile.

4031 Invalid context string *contextname*.

PROBLEM: The context string footnote contains non-alphanumeric characters or is empty.

RESULT: The compiler does not assign the topic an identifier.

SOLUTION: Change the characters in the context string so that it is valid, and then recompile.

4056 Unresolved context string specified in CONTENTS option.

PROBLEM: The Contents topic defined in the Help project file could not be found.

RESULT: The compiler uses the first topic in the build as the Contents.

SOLUTION: Compare the context string in the **CONTENTS** option to the context string used in the Contents topic to be sure they are the same. Change one of the strings to match the other, and then recompile.

4072 Context string exceeds limit of 255 characters.

PROBLEM: The context string hidden text cannot exceed 255 characters.

RESULT: The compiler ignores the context string.

SOLUTION: Shorten the context string, and then recompile.

4098 Context string(s) in [MAP] section not defined in any topic.

PROBLEM: The compiler cannot find a context string listed in the [MAP] section in any of the topics in the build.

RESULT: The compiler ignores the line in the [MAP] section. If the application uses that number in the WinHelp function, Help will display the "Help topic not found" error message.

SOLUTION: Check to be sure that all the context strings listed in the [MAP] section are assigned to topics included in the build and that they are all spelled correctly. Then recompile.

4113 Unresolved jump or pop-up *contextname***.**

PROBLEM: The specified topic contains a context string that identifies a nonexistent topic. **Building the Help File§ 18-27**

RESULT: The Help application displays the "Help topic not found" error message when the user chooses a hot spot with the unresolved context string.

SOLUTION: Check the topic for spelling errors in the context string, and also check to see if the requested topic is included in the build.

4131 Hash conflict between *contextname* **and** *contextname***.**

PROBLEM: The hash algorithm has generated the same hash value for both of the listed context strings.

RESULT: The Help application displays the topic with the first context string when the user chooses a hot spot containing either context string.

SOLUTION: Change one of the context strings, and then recompile.

4151 Invalid secondary window name *windowname***.**

PROBLEM: The hot spot contains a window name that is not defined in the Help project file.

RESULT: The hot spot is not active (not hot) in the built Help file.

SOLUTION: Change the window name to a unique name that is eight or fewer characters, and then recompile.

4171 Cannot use secondary window with pop-up.

PROBLEM: The hidden text defining the pop-up identifier contains a secondary window name.

RESULT: The hot spot is not active (not hot) in the built Help file.

SOLUTION: Remove the right angle bracket and secondary window name from the pop-up hot spot, and then recompile.

4196 Jumps and lookups not verified.

PROBLEM: Due to low memory conditions, the Help compiler cannot verify all jump and pop-up hot spots. (The reference to "lookups" (keywords) in the error message is incorrect.)

RESULT: The compiler continues the build but does not verify the validity of

jump and pop-up hot spots. The resulting Help file may still be valid.

SOLUTION: Free more memory and then recompile. For suggestions on how to free more memory for the build, refer to the "Getting Around Memory Problems" section in Chapter 17. If you can't overcome your memory constraints, you may have to compile and test the hot spots manually.

Footnote Errors

The following messages are caused by problems with footnotes in topic files.

4211 Footnote text exceeds limit of 1023 characters.

PROBLEM: The footnote text cannot exceed the limit of 1023 characters.

RESULT: The compiler ignores the footnote.

SOLUTION: Reduce the length of the footnote text, and then recompile.

Browse Sequence Errors

4251 Browse sequence not in first paragraph.

PROBLEM: The browse sequence footnote is not in the first paragraph of the topic.

RESULT: The compiler ignores the browse sequence footnote.

SOLUTION: Move the browse sequence footnote so that it is first, and then recompile.

4272 Empty browse sequence string.

PROBLEM: The browse sequence footnote for the specified topic contains no sequence characters.

RESULT: The compiler ignores the browse sequence footnote.

SOLUTION: Add sequence characters to the browse sequence footnote, and then recompile.

4292 Missing sequence number.

PROBLEM: A browse sequence number ends with a colon (:) for the specified topic.

RESULT: The compiler ignores the browse sequence footnote.

SOLUTION: Remove the colon or enter a minor sequence number, and then recompile.

4312 Browse sequence already defined.

PROBLEM: A browse sequence footnote already exists for the specified topic.

RESULT: The compiler uses the first browse sequence footnote and ignores the duplicate footnote.

SOLUTION: Remove the duplicate browse sequence footnote, and then recompile.

Topic Title Errors

4331 Title not in first paragraph.

PROBLEM: The title footnote ($) is not in the first paragraph of the topic.

RESULT: The compiler ignores the title footnote, and the topic does not have a title.

SOLUTION: Move the title footnote to the beginning of the topic, and then recompile.

4352 Empty title string.

PROBLEM: The title footnote for the specified topic contains no characters.

RESULT: The compiler ignores the title footnote, and the topic does not have a title.

SOLUTION: Open the footnote window and type the title string next to the topic title footnote character for the topic, and then recompile.

4372 Title defined more than once.

PROBLEM: There is more than one title footnote in the specified topic.

RESULT: The compiler uses the first title footnote and ignores the duplicate title footnote.

SOLUTION: Remove the duplicate title footnote, and then recompile.

4393 Title exceeds limit of 128 characters.

PROBLEM: The title for the specified topic exceeds 128 characters.

RESULT: The compiler truncates the title string at 128 characters.

SOLUTION: Shorten the length of the title string, and then recompile.

Keyword Errors

4412 Keyword string exceeds limit of 255 characters.

PROBLEM: The keyword string exceeds the limit of 255 characters.

RESULT: The compiler ignores the keyword footnote.

SOLUTION: Shorten the length of the keyword string, or add another keyword footnote for the extra keywords, and then recompile.

4433 Empty keyword string.

PROBLEM: There are no characters in the keyword footnote.

RESULT: The compiler ignores the keyword footnote, and the topic does not have any keywords.

SOLUTION: Open the footnote window and type some keywords next to the keyword footnote character for the topic, and then recompile.

4452 Keyword(s) defined without title.

PROBLEM: The topic has a keyword assigned to it, but no title.

RESULT: The topic will appear as >>Untitled Topic<< in the Search dialog box and in the history list.

SOLUTION: Add a topic title footnote to the topic if you want a title to appear in the Search dialog box when the user selects this keyword.

Build Tag Errors

4471 Build tag footnote not at beginning of topic.

PROBLEM: The build tag footnote marker, if used, has to be the first character in the topic.

RESULT: The compiler ignores the build tag footnote, and the topic is not assigned a build tag.

SOLUTION: Move the build tag footnote to the beginning of the topic, before other footnotes, and then recompile.

4492 Build tag exceeds limit of 32 characters.

PROBLEM: A build tag for the specified topic exceeds the limit of 32 characters.

RESULT: The compiler ignores the build tag assigned to the topic.

SOLUTION: Shorten the length of the build tag to 32 or fewer characters, and then recompile.

Entry Macro Errors

4551 Entry macro not in first paragraph.

PROBLEM: The ! footnote (for executing a macro) is not in the first paragraph of the topic.

RESULT: The compiler ignores the macro footnote.

SOLUTION: Move the macro footnote to the beginning of the topic, next to the other footnotes, and then recompile.

Topic File Errors

The following messages result from problems in the use of Word for Windows formatting in one or more topic files.

4616 File *filename* **is not a valid RTF topic file.**

PROBLEM: The specified file is not an RTF file.

RESULT: The compiler ignores the file.

SOLUTION: Make sure that you save the topic as RTF from your word processor.

4639 Error in file *filename* **at byte offset 0x***offset***.**

PROBLEM: The specified file contains unrecognized RTF at that byte offset. Unrecognized RTF includes any formatting the Help compiler does not recognize, including mismatched braces or nonstandard RTF produced by applications other than Word for Windows. This message should not appear if Word for Windows, Word for the Macintosh, or PCWord is used.

RESULT: The compiler aborts the build.

SOLUTION: Open the file in Word for Windows and save the file again as RTF, and then recompile. If you are using Word for the Macintosh, transfer the file to the PC again, and then recompile. If the problem persists, open the file without converting it to .DOC format, and look for errors in the source RTF.

4649 File *filename* **contains more than 32767 topics.**

PROBLEM: The maximum number of topics allowed in one RTF file is 32,767.

RESULT: The compiler aborts the build.

SOLUTION: Reduce the number of topics in the file by dividing it into two or more topic files, and then recompile.

4652 Table formatting too complex.

PROBLEM: The compiler encountered a table with borders, shading, or right justification.

RESULT: The compiler ignores the formatting.

SOLUTION: Remove the unsupported formatting, and then recompile.

4662 Side by side paragraph formatting not supported.

PROBLEM: The side-by-side paragraph formatting is not supported in Windows Help version 3.1.

RESULT: The compiler ignores the side-by-side paragraphs.

SOLUTION: Convert the side-by-side paragraphs to a table by opening the file in Word for Windows and saving it.

4671 Table contains more than 32 columns.

PROBLEM: The maximum number of columns in one table is 32. Some word processors may have different limits for the number of columns supported.

RESULT: The compiler treats the additional columns as one table cell.

SOLUTION: Reduce the number of columns to 32 or fewer, and then recompile.

4680 Font *fontname* **in file** *filename* **not in RTF font table.**

PROBLEM: A font not defined in the RTF header has been entered into the topic.

RESULT: The compiler uses the default system font.

SOLUTION: Check to be sure the font you are using is supported by the word processor. Then save the file again as RTF and recompile.

4692 Unrecognized graphic format.

PROBLEM: The compiler supports only Windows bitmaps (.BMP), Windows device-independent bitmaps (.DIB), Windows metafiles (.WMF), segmented hypergraphics (.SHG), and multiresolution bitmaps (.MRB).

RESULT: The compiler ignores the graphic.

SOLUTION: Make sure that you have not used Macintosh picture format or some other unsupported graphic format.

4733 Hidden page break.

PROBLEM: The page break is part of the hidden text.

RESULT: The compiler ignores the page break formatted as hidden text, so the two topics will not be separated.

SOLUTION: Reformat the page break as plain text and recompile.

4753 Hidden paragraph.

PROBLEM: A paragraph marker is part of the hidden text.

RESULT: The compiler ignores the paragraph marker formatted as hidden text, so the two paragraphs will run together.

SOLUTION: Reformat the paragraph marker as plain text and recompile.

4763 Hidden carriage return.

PROBLEM: A carriage return is part of the hidden text.

RESULT: The compiler ignores the carriage return formatted as hidden text and does not create a new line.

SOLUTION: Reformat the carriage return as plain text and recompile.

4774 Paragraph exceeds limit of 64K.

PROBLEM: A single paragraph has more than 64K of text or 64K of graphics. If the paragraph contains an inline graphic (pasted directly into the topic in Word for Windows) or a bitmap reference stored with data (**bmcwd**, for example), the paragraph size cannot exceed 64K. This limit does not apply to graphics stored separately from the topic data using the standard bitmap references **bmc**, **bml**, or **bmr**.

RESULT: The compiler aborts the build.

SOLUTION: Replace any inline graphics with bitmap references, or reduce the amount of text in the paragraph, or reduce the size of the graphic, and then recompile.

4792 Non-scrolling region defined after scrolling region.

PROBLEM: A paragraph that was authored as Keep With Next is not the first paragraph in the topic.

RESULT: The compiler ignores the Keep With Next attribute; the paragraph is treated as regular text and will be part of the topic text displayed in the scrolling region.

SOLUTION: Format the first paragraph(s) as Keep With Next and remove the Keep With Next formatting from the subsequent paragraphs. Then recompile.

4813 Non-scrolling region crosses page boundary.

PROBLEM: The Keep With Next paragraph formatting crosses a page break boundary.

RESULT: The topic includes a nonscrolling region, whether or not one is intended.

SOLUTION: Check the file to be sure that all topics end with a hard carriage return immediately before the page break. Remove the Keep With Next formatting attribute from any unwanted paragraphs, and then recompile.

4000 UNKNOWN ERROR, Contact Microsoft Product Support Services

PROBLEM: This error message is the equivalent of an assertion failure.

RESULT: The compiler aborts the build.

SOLUTION: Contact Microsoft Product Support Services.

Miscellaneous Errors

The following messages are caused by conditions such as MS-DOS file errors or out-of-memory conditions.

5035 File *filename* **not created.**

PROBLEM: There are no topics to compile, or the build expression is false for all topics.

RESULT: The compiler does not create a Help file.

SOLUTION: Add a topic filename to the [FILES] section in the Help project file, or change the build expression so that all topics are not excluded from the build, and then recompile.

5059 Not enough memory to build help file.

PROBLEM: The Help compiler does not have enough memory to complete the build process.

RESULT: The compiler does not create a Help file.

SOLUTION: To free memory, unload any unneeded applications, device drivers, and memory-resident programs. You can also turn off compression in the Help project file, replace any inline bitmaps pasted directly in the topic files with bitmap references, or compile under OS/2 rather than MS-DOS.

5075 Help Compiler corrupted. Please reinstall HC.EXE.

PROBLEM: Virus-checking code has detected a corruption in the compiler.

RESULT: The compiler does not run.

SOLUTION: Reinstall the compiler from the original source disk.

5098 Using old key-phrase table.

PROBLEM: The compiler found an existing key-phrase table for the Help file you are building, and the Help project file does not specify not to use the table.

RESULT: The compiler uses the key-phrase table that was generated from an earlier build, instead of creating a new table.

SOLUTION: To achieve maximum compression during a build, you must delete the old key-phrase table (.PH file) before each recompilation, or set the **OLDKEYPHRASE** option to *off.*

5115 Write failed.

PROBLEM: Write to disk failed.

RESULT: The compiler aborts the build.

SOLUTION: Contact Microsoft Product Support Services.

5139 Aborted by user.

PROBLEM: The user ended the build prematurely by pressing CTRL+C or CTRL+BREAK. Note: The extended version of the compiler (HCP.EXE) does not display this error message.

RESULT: The Help compiler aborts the build.

SOLUTION: Restart the build when ready.

5000 UNKNOWN ERROR, Contact Microsoft Product Support Services

PROBLEM: This error message is the equivalent of an assertion failure.

RESULT: The compiler aborts the build.

Help Compiler 3.0 Error Messages

SOLUTION: Contact Microsoft Product Support Services.

This section lists the error messages that the Help compiler version 3.0 displays when it encounters problems building a Help file. Although many of the messages are similar to those displayed by the version 3.1 Help compiler, the 3.0 message scheme and numbering system are different. This section describes the version 3.0 error messages.

Whenever possible, the Help compiler displays the name of the file that contains the error, as well as the number used to identify the specific line of the Help project file or the topic that produced the error. Since topics do not contain numbers, the topic number specified in the error message refers to the topic's sequential order in the topic file. In most cases, the topic number given in the error message will match the topic's page number in the Word for Windows document.

Interpreting Help 3.0 Error Messages

The Help compiler version 3.0 displays either warning or fatal error messages. A
warning message begins with the word **Warning**. A warning error indicates a
problem encountered during the build that is not severe enough to prevent the
Help compiler from completing the build. Therefore, a build with warnings
produces a Help file that Windows Help should be able to open, but the file may
contain problems in certain topics, such as missing graphics or hot spots that
aren't hot. Fatal error messages begin with the word **Error**. A fatal error
indicates a problem that prevents the Help compiler from creating a Help file.

As stated previously, you use the **REPORT** option to display error messages on
the screen and the **WARNING** option to specify the amount of warning
information you want the Help compiler to provide. The 3.0 compiler always
reports fatal errors, regardless of the current warning level or report option
specified in the Help project file, since no usable Help file results from the build.

While the Help compiler processes the Help project file, it ignores lines that
contain errors and attempts to continue with the build. This means that errors
encountered early in a project file may result in many more errors being reported
as the build continues.

Similarly, when the Help compiler processes the RTF topic files, it reports any
errors it encounters and, if the errors are not fatal, the compiler continues with the
build. A single error in a topic file may result in more than one error message
being displayed by the compiler. For instance, a typographic mistake in a topic's
context string will cause an error to be reported every time the compiler
encounters a reference to the correct topic identifier. On the other hand, some
warning errors—such as hot spots with the same hidden text codes (broken
jumps) or any bitmap file errors, including "File not found"— are reported once
per file, regardless of the number of times the problem occurs in the source file.

Help 3.0 Error Message Categories

In Windows Help version 3.0, errors that occur during processing of the Help
project file are prefixed with the letter P and appear as in the following examples:

Error P1025: line...7 of filename.HPJ : Section heading sectionname unrecognized.

Warning P1039: line...38 of filename.HPJ : [BUILDTAGS] section missing.

Errors that occur during processing of the RTF topic file(s) are prefixed with the
letter R and appear as in the following examples:

Error R2025: File environment error.

Help 3.0 Error Message Reference

This reference lists all Help 3.0 error messages by category. The wording of the error messages is the same as that displayed by the 3.0 Help compiler, except that variables (% tags) are replaced with descriptive words. Error message descriptions contain one or more of the following types of information.

Heading	Information
Problem	Explains what the error message means.
Result	Describes how the Help compiler handles this particular error.
Solution	Offers suggestions about how to correct the problem.

Project File Error Messages

This section lists all the possible errors that might occur when the 3.0 Help compiler is processing the Help project file.

P1001   Unable to read file *filename*.

PROBLEM: The specified file could not be found or is unreadable. This is an MS-DOS file error or an out-of-memory condition.

RESULT: If the file not found is an RTF topic file, the compiler ignores the file. If the file is a bitmap file, the built Help file displays the "Unable to display picture" message in the topic instead of the bitmap.

SOLUTION: Check to see if the file exists, and also check the amount of

available memory.

P1003   Invalid path specified in Root option.
PROBLEM: The compiler cannot find the path specified by the **ROOT** option.

RESULT: The compiler uses the current working directory.

SOLUTION: Correct the path in the **ROOT** option, and then recompile.

P1005   Path and filename exceed limit of 79 characters.

PROBLEM: The combined length of the path and filename must not be more than the MS-DOS limit.

RESULT: The compiler ignores the file.

SOLUTION: Shorten the path, and then recompile.

P1007   Root path exceeds maximum limit of 66 characters.

PROBLEM: The specified root path exceeds the MS-DOS limit.

RESULT: The compiler ignores the path and uses the current working directory.

SOLUTION: Shorten the path, and then recompile.

P1009   [FILES] section missing.

PROBLEM: The [FILES] section is required.

RESULT: The compiler aborts the build.

SOLUTION: Create a [FILES] section that lists the topic files to include in the build, and then recompile.

P1011   Option *optionname* **previously defined.**

The specified option was defined previously. The compiler ignores the attempted redefinition.

P1013   Project file extension cannot be .HLP.

PROBLEM: You cannot specify a Help project file with a .HLP extension. Project files must use the .HPJ extension.

RESULT: The compiler aborts the build.

SOLUTION: Rename the Help project file, and then recompile.

P1015   Unexpected end-of-file.

PROBLEM: The compiler has unexpectedly come to the end of the Help project file. There may be an open comment in the Help project file or in an include file.

RESULT: The compiler aborts the build.

SOLUTION: Close any open comments within the Help project file, and then recompile.

P1017   Parameter exceeds maximum length of 128 characters.

PROBLEM: An option, context name or number, build tag, or other parameter on the specified line exceeds the limit of 128 characters.

RESULT: The compiler ignores the line.

SOLUTION: Shorten the line, and then recompile.

P1021   Context number already used in [MAP] section.

PROBLEM: A context number in the [MAP] section of the Help project file was previously mapped to a different context string.

RESULT: The compiler ignores the line with the duplicate context number.

SOLUTION: Check all context numbers to make sure that they are unique. Remove or reassign any duplicate context numbers, and then recompile.

P1023   Include statements nested too deeply.

PROBLEM: The **#include** statement on the specified line exceeds the maximum of five include levels.

RESULT: The compiler aborts the build.

SOLUTION: Simplify the #**include** statement so that it nests fewer than five include levels, then recompile.

P1025   Section heading *sectionname* **unrecognized.**

PROBLEM: The Help project file includes a section heading that the compiler does not recognize or does not support.

RESULT: The compiler ignores the section heading.

SOLUTION: Check all bracketed headings in the Help project file for typographical or other errors. Fix the section heading, and then recompile.

P1027   Bracket missing from section heading *sectionname*.

PROBLEM: The right bracket (]) is missing from the specified section
heading.   **Building the Help File§   18-41**

RESULT: The compiler ignores the section heading on this line.

SOLUTION: Insert the right section heading bracket, and then recompile.

P1029   Section heading missing.

PROBLEM: The section heading on the specified line is not complete, or the
first entry in the Help project file is not a section heading.

RESULT: The compiler ignores the section heading on this line and all
succeeding lines until it encounters a valid section heading.

SOLUTION: Insert a valid section heading at the appropriate place in the Help
project file, and then recompile.

P1030   Section *sectionname* **previously defined.**

PROBLEM: The compiler found more than one section with the specified
heading in the Help project file.

RESULT: The compiler ignores the duplicate section and continues from the
next valid section heading.

SOLUTION: Remove or rename one of the duplicate section headings, and
then recompile.

P1031   Maximum number of build tags exceeded.

PROBLEM: The maximum number of build tags that can be defined is 32.

RESULT: The compiler ignores the additional tags.

SOLUTION: Reduce the number of build tags to 32 or fewer, and then
recompile.

P1033   Duplicate build tag in [BUILDTAGS] section.

PROBLEM: A build tag appears more than once in the [BUILDTAGS]
section.

RESULT: The compiler uses the first build tag and ignores the duplicate build
tag.

SOLUTION: Remove or rename one of the duplicate build tags, and then

recompile.

P1035   Build tag length exceeds maximum.

PROBLEM: The build tag on the specified line exceeds the limit of 32 characters.

RESULT: The compiler ignores the build tag.

SOLUTION: Shorten the build tag to 32 or fewer characters, and then recompile.

P1037   Build tag *tagname* **contains invalid characters.**

PROBLEM: Build tags can contain only alphanumeric characters or the underscore (_) character.

RESULT: The compiler ignores the line with invalid characters.

SOLUTION: Correct the build tag so that it contains only valid characters, and then recompile.

P1039   [BUILDTAGS] section missing.

PROBLEM: The **BUILD** option declares a conditional build, but there is no [BUILDTAGS] section in the Help project file.

RESULT: The compiler includes all topics in the build.

SOLUTION: Add a [BUILDTAGS] section to the Help project file, if you want to perform a conditional build, and then recompile.

P1043   Too many tags in Build expression.

PROBLEM: The Build expression on the specified line has used more than the maximum of 20 build tags.

RESULT: The compiler ignores the line.

SOLUTION: Reduce the number of build tags to 20 or fewer, and then recompile.

P1045   [ALIAS] section found after [MAP] section.

PROBLEM: When used, the [ALIAS] section must precede the [MAP] section in the Help project file.

RESULT: The compiler ignores the [ALIAS] section.

SOLUTION: Move the [ALIAS] section so that it precedes the [MAP] section, and then recompile.

P1047 Context string *contextname* **already assigned an alias.**

PROBLEM: The specified context string has already been assigned an alias in the [ALIAS] section. A context string can have only one alias. That is, you cannot map **a=b** and then **a=c** in the [ALIAS] section.

RESULT: The compiler ignores the attempted reassignment on this line.

SOLUTION: Correct the alias string mapping, and then recompile.

P1049 Alias string aliasname already assigned.

PROBLEM: You cannot alias an alias. That is, an alias string cannot, in turn, be assigned another alias. You cannot map **a=b** and then **b=c** in the [ALIAS] section.

RESULT: The compiler ignores the attempted reassignment on this line.

SOLUTION: Correct the alias string mapping, and then recompile.

P1051 Context string *contextname* **cannot be used as alias string.**

PROBLEM: A context string that has been assigned an alias cannot be used later as an alias for another context string. That is, you cannot map **a=b** and then **c=a** in the [ALIAS] section.

RESULT: The compiler ignores the attempted reassignment on this line.

SOLUTION: Correct the alias string mapping, and then recompile.

P1053 Maximum number of font ranges exceeded.

PROBLEM: The maximum number of font ranges that can be specified is five.

RESULT: The compiler ignores additional ranges.

SOLUTION: Remove the additional font ranges so that there are five or fewer, and then recompile.

P1055 Current font range overlaps previously defined range.

PROBLEM: A font size range overlaps a previously defined mapping.

RESULT: The compiler uses the first range and ignores the second mapping.

SOLUTION: Adjust one or both of the font ranges to remove any overlapping.

P1056   Unrecognized font name in Forcefont option.

PROBLEM: The compiler has encountered a font name that it does not recognize or support.

RESULT: The compiler ignores the font name and uses the default MS Sans Serif font.

SOLUTION: Change the font name to a supported font, and then recompile. For the list of supported fonts, refer to Chapter 16, "The Help Project File."

P1057   Font name too long.

PROBLEM: Font names cannot exceed 20 characters.

RESULT: The compiler ignores the option on this line.

SOLUTION: Shorten the font name to 20 or fewer characters, and then recompile.

P1059   Invalid multiple-key syntax.

PROBLEM: The **MULTIKEY** option must specify a single capital letter other than the letter K.

RESULT: The compiler ignores the attempted keyword-table assignement on this line.

SOLUTION: Correct the syntax, and then recompile. The correct syntax is:

MULTIKEY=char

P1061   Character already used.

PROBLEM: A character used for indicating the alternate keyword table (**MULTIKEY=***char*) was previously used.

RESULT: The compiler ignores the duplicate keyword-table assignment on this line.

SOLUTION: Change the duplicate character to one that has not been used, and then recompile.

P1063   Characters 'K' and 'k' cannot be used.

PROBLEM: These characters are reserved for Help's standard keyword table.

RESULT: The compiler ignores the attempted keyword-table assignment on this line.

SOLUTION: Choose another character for the alternate keyword table, and then recompile.

P1065 Maximum number of keyword tables exceeded.

PROBLEM: The [OPTIONS] section contains more than five **MULTIKEY** entries, which exceeds the limit of five keyword tables.

RESULT: The compiler ignores the additional keyword tables.

SOLUTION: Reduce the number of keyword tables to five or fewer, and then recompile.

P1067 Equal sign missing.

PROBLEM: An entry in the [OPTIONS] section is missing the equal sign, or it has some other syntax error.

RESULT: The compiler ignores the invalid option on this line.

SOLUTION: Check the syntax of each option used in the [OPTIONS] section. Fix the incorrect option, and then recompile. For the correct syntax, refer to Chapter 16, "The Help Project File."

P1069 Context string missing.

PROBLEM: A line in the [MAP] or [ALIAS] section is missing a context string before the equal sign.

RESULT: The compiler ignores the line with the invalid context string.

SOLUTION: Check all context strings to the left of the equal sign. Correct any missing or invalid entries, and then recompile.

P1071 Incomplete line in *sectionname* section.

PROBLEM: An entry in the specified section is incomplete.

RESULT: The compiler ignores the incomplete line.

SOLUTION: Check the section for any incomplete lines and correct them. Then recompile.

P1073 Unrecognized option in [OPTIONS] section.

PROBLEM: The compiler does not recognize or support the specified option.

RESULT: The compiler ignores the unrecognized option.

SOLUTION: Check the [OPTIONS] section for typing errors or invalid options, and then recompile.

P1075   Invalid build expression.

PROBLEM: The syntax used in the build expression on the specified line contains one or more logical or syntax errors.

RESULT: The compiler ignores the line with the invalid expression.

SOLUTION: Correct the entry, and then recompile. For the correct syntax and information about build expressions, refer to Chapter 16, "The Help Project File."

P1077   Warning level must be 1, 2, or 3.

PROBLEM: The warning reporting level can be set only to 1, 2, or 3.

RESULT: The compiler uses full reporting (level 3).

SOLUTION: Correct the entry, and then recompile.

P1079   Invalid compression option.

PROBLEM: The **COMPRESS** option specifies an incorrect value.

RESULT: The compiler uses the default value **off.**

SOLUTION: Correct the entry, and then recompile. The correct syntax is:

COMPRESS=[off, 0, no, false] or [on, 1, yes, true]

P1081   Invalid title string.

PROBLEM: The **TITLE** option defines a string that is empty or contains more than 32 characters.

RESULT: The compiler truncates the title at 32 characters.

SOLUTION: Add a title if the option is empty or shorten the title if it is too long, and then recompile.

P1083   Invalid context identification number.

PROBLEM: A context number in the [MAP] or [ALIAS] section is empty or contains invalid characters.

RESULT: The compiler ignores the line with the invalid context number.

SOLUTION: Check the format of all context numbers. Context numbers can be specified as decimal or hexadecimal. Correct any invalid numbers, and then recompile.

P1085   Unrecognized text.

PROBLEM: The compiler found some text in the Help project file that it can't recognize.

RESULT: The compiler ignores the line with the unrecognized text.

SOLUTION: Remove the unsupported text from the Help project file, and then recompile.

P1086   Invalid font-range syntax.

PROBLEM: The font range syntax used is invalid. A font range consists of a low and high point size, separated by a hyphen (-).

RESULT: The compiler ignores the option.

SOLUTION: Correct the syntax, and then recompile. The correct syntax is:

MAPFONTSIZE=m[-n]:p

P1089   Unrecognized sort ordering.

PROBLEM: You have specified an ordering that is not supported by the compiler.

RESULT: The compiler uses the default English (U.S.) sorting order.

SOLUTION: Correct the entry, and then recompile. The correct syntax is:

LANGUAGE=scandinavian

RTF Topic File Error Messages

This section lists all the possible errors that might occur when the 3.0 Help compiler is processing the RTF topic file(s).

R2001   Unable to open bitmap file *filename*.

PROBLEM: The specified bitmap file could not be found or is not in a recognizable bitmap format. This is an MS-DOS file error or an out-of-memory condition.

RESULT: The Help file displays the "Unable to display picture" message in the compiled topic instead of the bitmap.

SOLUTION: Check to see if the file exists; if it does, check its format. If necessary save the file again in your paint or draw program, and then recompile.

R2003 Unable to include bitmap file *filename*.

PROBLEM: The specified file could not be found or is unreadable. This is an MS-DOS file error or an out-of-memory condition.

RESULT: The built Help file displays the "Unable to display picture" message in the topic instead of the bitmap.

SOLUTION: Check to see if the file exists, and also check the amount of available memory.

R2005 Disk full.

PROBLEM: There is not enough disk space to write the Help file to disk.

RESULT: The compiler aborts the build.

SOLUTION: Create more space on the destination disk, and then recompile.

R2009 Cannot use reserved DOS device name for file *filename*.

PROBLEM: The file has a reserved MS-DOS filename, such as COM1, LPT2, or PRN. This is an MS-DOS file error.

RESULT: The compiler ignores the file.

SOLUTION: Rename the file, and then recompile.

R2013 Output file *filename* **already exists as a directory.**

PROBLEM: A directory in the Help project directory has the same name as the requested Help file. This is an MS-DOS file error.

RESULT: The compiler ignores the file.

SOLUTION: Move or rename the directory or the Help project file, and then recompile.

R2015 Output file *filename* **already exists as read-only.**

PROBLEM: The Help compiler cannot overwrite files with a read-only attribute. For example, this error occurs if you rebuild an existing .HLP file

and the .HLP file is read-only.

RESULT: The compiler aborts the build.

SOLUTION: Change the file's read-only attribute if you want the Help compiler to overwrite the file; otherwise, rename the file.

R2017   Path for file *filename* **exceeds limit of 79 characters.**

PROBLEM: The combined length of the path and filename must not be more than the MS-DOS limit of 79 characters.

RESULT: The compiler ignores the file.

SOLUTION: Shorten the path, and then recompile.

R2019   Cannot open file *filename*.

PROBLEM: The specified file could not be found or is unreadable. This is an MS-DOS file error or an out-of-memory condition.

RESULT: The compiler ignores the file.

SOLUTION: Check to see if the file exists, and also check the amount of available memory.

R2021   Cannot find file *filename*.

PROBLEM: The specified file could not be found or is unreadable. This is an MS-DOS file error or an out-of-memory condition.

RESULT: The compiler ignores the file.

SOLUTION: Check to see if the file exists, and also check the amount of available memory.

R2023   Not enough memory to build Help file.

PROBLEM: The Help compiler does not have enough memory to complete the build process.

RESULT: The compiler does not create a Help file.

SOLUTION: To free memory, unload any unneeded applications, device drivers, and memory-resident programs. You can also turn off compression in the Help project file, replace any inline bitmaps pasted directly in the topic files with bitmap references, or compile under OS/2 rather than MS-DOS.

R2025 File environment error.

PROBLEM: The compiler does not have enough available file handles to continue the build. This error is likely to occur if you are compiling in an MS-DOS box or if you are using Help Author and have Word for Windows running at the same time.

RESULT: The compiler aborts the build.

SOLUTION: If possible, increase the **FILES** setting in the CONFIG.SYS file to **FILES=50**. Then reboot your computer and recompile.

R2027 Build tag *tagname* **not defined in [BUILDTAGS] section of project file.**

PROBLEM: The specified build tag has been assigned to a topic but not declared in the Help project file.

RESULT: The compiler ignores the build tag defined in the topic.

SOLUTION: Include the build tag in the [BUILDTAGS] section, and then recompile.

R2033 Context string in Map section not defined in any topic.

PROBLEM: The compiler cannot find a context string listed in the [MAP] section in any of the topics in the build.

RESULT: The compiler ignores the line in the [MAP] section. If the application uses that number in the WinHelp function, Help will display the "Help topic not found" error message,

SOLUTION: Check to be sure that all the context strings listed in the [MAP] section are assigned to topics included in the build and that they are all spelled correctly. Then recompile.

R2035 Build expression missing from project file.

PROBLEM: The topics have build tag footnotes, but there is no **BUILD=** expression in the Help project file.

RESULT: The compiler includes all topics in the build.

SOLUTION: Add a build expression to the Help project file and recompile if you want your build tags to work. Otherwise, you can safely ignore this message.

R2037   File *filename* **cannot be created, due to previous error(s).**

PROBLEM: The compiler cannot continue because there are topics remaining to be processed.

RESULT: The compiler does not create a Help file.

SOLUTION: Correct the errors that preceded this error, and then recompile.

R2039   Unrecognized table formatting in topic *topicnumber* **of file** *filename***.**

PROBLEM: The compiler ignores table formatting that is unsupported in Help.

RESULT: The compiler aborts the build.

SOLUTION: Reformat the entries as text paragraphs with hanging indents or as side-by-side paragraphs, and then recompile.

R2041   Jump *context_string* **unresolved in topic** *topicnumber* **of file** *filename***.**

PROBLEM: The specified topic contains a context string that identifies a nonexistent topic.

RESULT: The Help application displays the "Help topic not found" error message when the user chooses a hot spot with the unresolved context string.

SOLUTION: Check the topic for spelling errors in the context string, and also check to see if the requested topic is included in the build.

R2043   Hotspot text cannot spread over paragraphs.

PROBLEM: A paragraph marker is part of the hidden text.

RESULT: The compiler ignores the paragraph marker formatted as hidden text, so the two paragraphs will run together.

SOLUTION: Reformat the paragraph marker as plain text, and then recompile.

R2045   Maximum number of tab stops reached in topic *topicnumber* **of file** *filename***.**

PROBLEM: The maximum number of custom tab stops that can be defined for a paragraph is 32.

RESULT: The compiler uses the default tab stops after the 32nd tab.

SOLUTION: Reduce the number of tab stops to 32 or fewer, and then recompile.

R2047   File *filename* **not created.**

PROBLEM: There are no topics to compile, or the build expression is false for all topics.

RESULT: The compiler does not create a Help file.

SOLUTION: Add a topic filename to the [FILES] section in the Help project file, or change the build expression so that all topics are not excluded from the build, and then recompile.

R2049   Context string text too long in topic *topicnumber* **of file** *filename***.**

PROBLEM: The context string hidden text cannot exceed 64 characters.

RESULT: The compiler ignores the context string.

SOLUTION: Shorten the context string, and then recompile.

R2051   File *filename* **is not a valid RTF topic file.**

PROBLEM: The specified file is not an RTF file.

RESULT: The compiler ignores the file.

SOLUTION: Make sure that you save the topic as RTF from your word processor, and then recompile.

R2053   Font *fontname* **in file** *filename* **not in RTF font table.**

A font not defined in the RTF header has been entered into the topic. The compiler uses the default system font.

R2055   File *filename* **is not a usable RTF topic file.**

PROBLEM: The specified file contains a valid RTF header, but the content is not RTF or is corrupted.

RESULT: The compiler ignores the file.

SOLUTION: Resave the topic as RTF from your word processor, and then recompile.

R2057   Unrecognized graphic format in topic *topicnumber* **of file** *filename***.**

PROBLEM: The compiler supports only Windows bitmaps (.BMP) and Windows device-independent bitmaps (.DIB).

RESULT: The compiler ignores the graphic.

SOLUTION: Make sure that you have not used a metafile, a Macintosh picture format, or some other unsupported graphic format.

R2059 Context string identifier already defined in topic *topicnumber* **of file** *filename***.**

PROBLEM: There is more than one context-string identifier footnote for the specified topic.

RESULT: The compiler uses the identifier defined in the first # footnote.

SOLUTION: Remove one of the context string identifiers, and then recompile.

R2061 Context string *contextname* **already used in file** *filename***.**

PROBLEM: The specified context string was previously assigned to another topic.

RESULT: The compiler ignores the duplicate context string, and the topic has no identifier.

SOLUTION: Change the second context string so that it is unique, and then recompile.

R2063 Invalid context-string identifier for topic *topicnumber* **of file** *filename***.**

PROBLEM: The context string footnote contains non-alphanumeric characters or is empty.

RESULT: The compiler does not assign the topic an identifier.

SOLUTION: Change the characters in the context string so that it is valid, and then recompile.

R2065 Context string defined for index topic is unresolved.

PROBLEM: The Index topic defined in the Help project file could not be found.

RESULT: The compiler uses the first topic in the build as the Index.

SOLUTION: Compare the context string in the **INDEX** option to the context string used in the Index topic to be sure they are the same. Change one of the strings to match the other, and then recompile.

R2067 Footnote text too long in topic *topicnumber* **of file** *filename***.**

PROBLEM: The footnote text cannot exceed the limit of 1000 characters.

RESULT: The compiler ignores the footnote.

SOLUTION: Reduce the length of the footnote text, and then recompile.

R2069 Build tag footnote not at beginning of topic *topicnumber* **of file** *filename***.**

PROBLEM: The build tag footnote marker, if used, has to be the first character in the topic.

RESULT: The compiler ignores the build tag footnote, and the topic is not assigned a build tag.

SOLUTION: Move the build tag footnote to the beginning of the topic, before other footnotes, and then recompile.

R2071 Footnote text missing in topic *topicnumber* **of file** *filename***.**

PROBLEM: The specified topic contains a footnote that has no characters.

RESULT: The compiler ignores the footnote.

SOLUTION: Add characters to the footnote, and then recompile.

R2073 Keyword string is null in topic *topicnumber* **of file** *filename***.**

PROBLEM: There are no characters in the keyword footnote.

RESULT: The compiler ignores the keyword footnote, and the topic does not have any keywords.

SOLUTION: Open the footnote window and type some keywords next to the keyword footnote character for the topic, and then recompile.

R2075 Keyword string too long in topic *topicnumber* **of file** *filename***.**

PROBLEM: The keyword string exceeds the limit of 255 characters.

RESULT: The compiler ignores the keyword footnote.

SOLUTION: Shorten the length of the keyword string, or add another keyword footnote for the extra keywords, and then recompile.

R2077 Keyword(s) defined without title in topic *topicnumber* **of file** *filename***.**

PROBLEM: The topic has a keyword assigned to it, but no title.

RESULT: The topic will appear as >>Untitled Topic<< in the Search dialog box and in the history list.

SOLUTION: Add a topic title footnote to the topic if you want a title to appear in the Search dialog box when the user selects this keyword.

R2079 Browse sequence string is null in topic *topicnumber* **of file** *filename***.**

PROBLEM: The browsesequence footnote for the specified topic contains no sequence characters.

RESULT: The compiler ignores the browse sequence footnote.

SOLUTION: Add sequence characters to the browse sequence footnote, and then recompile.

R2081 Browse sequence string too long in topic *topicnumber* **of file** *filename***.**

PROBLEM: The browse sequence string exceeds the limit of 128 characters.

RESULT: The compiler ignores the browse sequence footnote.

SOLUTION: Shorten the length of the sequence string, and then recompile.

R2083 Missing sequence number in topic *topicnumber* **of file** *filename***.**

PROBLEM: A browsesequence number ends with a colon (:) for the specified topic.

RESULT: The compiler ignores the browse sequence footnote.

SOLUTION: Remove the colon or enter a minor sequence number, and then recompile.

R2085 Sequence number already defined in topic *topicnumber* **of file** *filename***.**

PROBLEM: A browse-sequence footnote already exists for the specified topic.

RESULT: The compiler uses the first browse sequence footnote and ignores the duplicate footnote.

SOLUTION: Remove the duplicate browse sequence footnote, and then recompile.

R2087 Build tag too long.

PROBLEM: A build tag for the specified topic exceeds the limit of 32 characters.

RESULT: The compiler ignores the build tag assigned to the topic.

SOLUTION: Shorten the length of the build tag to 32 or fewer characters, and then recompile.

R2089 **Title string null in topic** *topicnumber* **of file** *filename***.**

PROBLEM: The title footnote for the specified topic contains no characters.

RESULT: The compiler ignores the title footnote, and the topic does not have a title.

SOLUTION: Open the footnote window and type the title string next to the topic title footnote character for the topic, and then recompile.

R2091 **Title too long in topic** *topicnumber* **of file** *filename***.**

PROBLEM: The title for the specified topic exceeds 128 characters.

RESULT: The compiler truncates the title string at 128 characters.

SOLUTION: Shorten the length of the title string, and then recompile.

R2093 **Title titlename in topic** *topicnumber* **of file** *filename* **used previously.**

PROBLEM: The specified title has previously been assigned to another topic.

RESULT: The compiler uses the first title and ignores the duplicate title.

SOLUTION: Remove the duplicate title, and then recompile.

R2095 **Title defined more than once in topic** *topicnumber* **of file** *filename***.**

PROBLEM: There is more than one title footnote in the specified topic.

RESULT: The compiler uses the first title footnote and ignores the duplicate title footnote.

SOLUTION: Remove the duplicate title footnote, and then recompile.

R2501 **Using old key-phrase table.**

PROBLEM: The compiler found an existing key-phrase table for the Help file you are building, and the Help project file does not specify not to use the table.

RESULT: The compiler uses the key-phrase table that was generated from an earlier build, instead of creating a new table.

SOLUTION: To achieve maximum compression during a build, you must delete the old key-phrase table (.PH file) before each recompilation.

R2503 Out of memory during text compression.

PROBLEM: The compiler encountered a memory limitation during
compression.   **Building the Help File§   18-57**

RESULT: The Help compiler continues the build without compressing the
Help file.

SOLUTION: Free more memory and then recompile. For suggestions on how
to free more memory fo the build, refer to the "Getting Around Memory
Problems" in Chapter 17. If you can't overcome your memory constraints, you
may have to compile without compression or remove some of the graphics in
the topic files.

R2505 File environment error during text compression.

PROBLEM: The compiler does not have enough available file handles to
compress the Help file. This error is likely to occur if you are compiling in an
MS-DOS box or if you are using Help Author and have Word for Windows
running at the same time.

RESULT: The Help compiler continues the build without compressing the
Help file.

SOLUTION: If possible, increase the **FILES** setting in the CONFIG.SYS file
to **FILES=50**. Then reboot your computer and recompile.

R2507 DOS file error during text compression.

PROBLEM: The compiler encountered a problem accessing a disk file while it
was compressing the Help file. This is an MS-DOS file error.

RESULT: The Help compiler continues the build without compressing the
Help file.

SOLUTION: Check to make sure you have enough free disk space or look for
other MS-DOS related problems, and then recompile.

R2509 Error during text compression.

PROBLEM: One of the three compression errors—R2503, R2505, or R2507—
has occurred.

RESULT: The Help compiler continues the build without compressing the
Help file.

SOLUTION: Check the solutions section for each of the other errors to see

what the problem might be. When you have corrected the problem, recompile.

R2701   Internal error.

PROBLEM: The compiler is unable to create the data structure (b-tree) containing all the titles in the Help file.

RESULT: The compiler aborts the build.

SOLUTION: Contact Microsoft Product Support Services.

R2703   Internal error.

PROBLEM: The compiler cannot create the bitmap file because of a low-memory condition, lack of sufficient free disk space, or insufficient file handles.

RESULT: The compiler aborts the build.

SOLUTION: Contact Microsoft Product Support Services.

R2705   Internal error.

PROBLEM: The phrase file used for compression has disappeared between the time it is created (or last checked) and when the compiler needs to use it. This will likely occur if the hard disk drive is low on disk space or if you are building the Help file over a network.

RESULT: The compiler aborts the build.

SOLUTION: Contact Microsoft Product Support Services.

R2707   Internal error.

PROBLEM: The compiler has encountered a disk error other than "out of memory," "out of disk space," or "out of file handles." This error may indicate a problem with your hard disk drive, such as bad sectors or too many files in a directory.

RESULT: The compiler aborts the build.

SOLUTION: Contact Microsoft Product Support Services.

R2709   Internal error.

PROBLEM: The compiler can't open the output system file. This problem should only occur if you are using a nonstandard release or beta release of Help version 3.0.

RESULT: The compiler aborts the build.

SOLUTION: Contact Microsoft Product Support Services.