



Version 8.3.2 – 17 November 2023

Published by Just Great Software Co. Ltd.

Copyright © 1996–2023 Jan Goyvaerts. All rights reserved.

“HelpScribble” and “Just Great Software” are trademarks of Jan Goyvaerts

HelpScribble Manual

Welcome to HelpScribble's manual. This manual was created by using HelpScribble's Project|Make Flat Manual function to automatically create a printable document from HelpScribble's help file. The auto-generated manual was slightly edited using Microsoft Word 2007 and then saved as a PDF. HelpScribble's help file and manual give you a good idea of what is possible with HelpScribble. The help files for all other JGsoft products were created with HelpScribble as well.

This manual consists of two parts. The first part is the "how to" part. Each "how to" section contains step by step instructions of how to accomplish certain things with HelpScribble. It does not explain everything you could possibly do with HelpScribble, but clearly explains basic tasks so you can get up to speed quickly. You can also view several self-running demonstrations explaining the basic tasks of creating a help file.

The second part is a screen by screen reference of every part of the HelpScribble software and what you can do with it. This is also the part that will be displayed when you click on one of the Help buttons in HelpScribble.

Table of Contents

HelpScribble Manual	2	Project Export to RTF+HPJ.....	18
Table of Contents	2	Project Export to HTML Help	19
How to Create a Help File	4	Project Export to Web Help.....	19
How to Add an Index	5	Project Export Topic Grid.....	21
How to Add a Table of Contents	6	Project Import Help Project	21
How to Add Images and Screen Shots	7	Project Make	21
How to Make Images Clickable	7	Compiling on The Command Line	22
How to Make a Help File Browsable	8	Project Show Help File.....	22
How to Control Fonts and Colors	8	Project Check Spelling.....	23
How to Choose Between WinHelp and HTML Help.....	9	Project Renummer Topics	23
Viewing CHM Files Across Networks	11	Project Merge.....	23
Viewing WinHelp Files on Windows Vista, 7, and 8.....	11	Project Open Delphi Unit	24
How to Link Your Help File to an Application	12	Project Exit	24
How to Distribute Your Help File	13	Topic Undo.....	25
How to Change Help Authoring Tools (Import/Export).....	13	Topic Redo.....	25
How to Write a Great Help File.....	14	Topic New	25
HelpScribble's Main Window	16	Topic Delete	25
HelpScribble's Main Menu.....	16	Topic Cut.....	26
Project New	17	Topic Copy	26
Project Open.....	17	Topic Paste	26
Project Save	17	Topic Properties	26
Project Save As.....	17	Topic Check Spelling	27
Project Make Flat Manual.....	18	Text Font	27
		Text Bold	27
		Text Italic.....	27
		Text Underline.....	27
		Text Strike Out	27

Text Left.....	28	Topic Grid Copy	44
Text Center	28	Topic Grid Paste	44
Text Right	28	Topic Grid Create Link	44
Text Bullet	28	Topic Grid Create Popup Link.....	45
Text Cut	28	Topic Grid Bookmark	45
Text Copy	28	Topic Properties	46
Text Paste.....	29	Topic Properties: ID	46
Text Paste as Plain Text.....	29	Topic Properties: Title	47
Text Delete	29	Topic Properties: Keywords	47
Text Select All.....	29	Topic Properties: Status.....	47
Text Find and Replace	30	Topic Properties: A Keywords.....	48
HelpTools Link	31	Topic Properties: Browse Sequence	48
HelpTools Create Target	32	Topic Properties: Build Tag.....	48
HelpTools Keepn	32	Topic Properties: Entry Macro	49
HelpTools Insert Bitmap	32	Topic Properties: Context String.....	49
HelpTools Insert Button	33	Project Options	50
HelpTools Internet Link.....	34	Project Options: General.....	50
HelpTools Link to File	34	Project Options: Compiler	51
HelpTools Embed Flash	34	Project Options: WinHelp.....	53
HelpTools Insert RTF Codes	34	Project Options: HTML Help	54
HelpTools Insert HTML Tags.....	35	Contents Editor	56
Go to topic Next Topic	35	Contents Editor: General	56
Go to topic Previous Topic	35	Contents Editor: Contents.....	57
Go to topic Next bookmark	36	Index Editor	59
Go to topic Previous Bookmark.....	36	SHG Editor	61
Go to topic History	36	SHG Editor: Taking Screen Shots	61
Print Printer Setup	37	SHG Editor: Adding Hotspots	62
Print Print Topic List	37	SHG Editor: Sorting Hotspots	62
View ProjectBar	38	SHG Editor: Saving SHGs	62
View TopicBar.....	38	SHG Editor: Replacing the Bitmap of an SHG.....	63
View PropertiesBar	39	Browse Sequence Editor	64
View TextBar	39	Windows Editor	65
TextMacros	39	WinHelp Macro Editor	66
View HelpToolBar	40	HelpContext Property Editor	67
View Ruler	41	Delphi Source Code Parser	68
View Topic Grid	41	Use Comments as Descriptions.....	69
View Error Log.....	41	Including or Excluding Classes.....	70
Help Download Spell Check Dictionaries	42	Including or Excluding Class Members.....	71
Topic Grid	43	Including or Excluding Types, Variables, Constants and Routines.....	72
Topic Grid Edit Topic	43	Linking to the Delphi or C++Builder documentation.....	73
Topic Grid Insert Topic	43		
Topic Grid Delete Topic.....	43		
Topic Grid Properties.....	44		
Topic Grid Cut.....	44		

How to Create a Help File

Creating a help file is very easy when you use HelpScribble. A help file consists of a series of topics. Each topic contains part of the text of your help file. When reading a help file, the help screen will show you all the text of a single topic of the help file. If the topic is too large to fit inside the help screen, scroll bars will appear so you can scroll to read the rest of the topic.

Each topic in your help text should discuss only one small subject (or “topic”, hence the name). Topics should not be longer than about twice the height of the screen. Otherwise, the help file becomes difficult to read. If a subject requires a lot of text, split it up across several topics. Right now you are reading a help topic with the title “How to Create a Help File”. Other topics discuss how to add an index, and a table of contents.

When a topic mentions something that is discussed in another topic, you should create a link to that other topic, as I did in the previous paragraph. A link is some text that is underlined and has a different color. When you click a link, the topic you were just reading will be replaced with the topic the link points to. In fact, a help system works just like a web browser. When you click on a link on a web site, your browser will load another web page.

Enough theory. Let's create our first help file in HelpScribble. You can read a detailed explanation below and watch a self-running demonstration. If you just started HelpScribble, you can start creating the new project right away. If you already opened another (sample) project, select Project|New from the menu to start with a new blank project.

The blank project has one blank topic, titled <new topic> and with Topic ID 10. You can see the title in three places: in the Topic Grid at the left side, in the TopicBar, the second toolbar from above, and in big blue letters the large blank area. The large blank area is where you will type in the text of the topic.

We will start with giving our first topic a more meaningful title. Type in the title you want in the TopicBar. When you type in the title there, HelpScribble will automatically update the first line of the topic text. The Topic Grid will also be updated automatically. Then click in the middle of the large blank area. This will put the text cursor at the end of the topic text (which is just the title right now). Type in the text of your first topic. If you do not like the default fonts for the title and body text, you can set your own defaults by changing the system TextMacros. See How to Control Fonts and Colors for more information.

To add more topics to your help file, simply select Topic|New from the menu. You will see the new topic in the Topic Grid below the first. Again, update the title in the TopicBar and type in the text. Create a few more topics, and then I will explain how you can link them.

There are three ways through which you can create links in HelpScribble: HelpTools|Link, the HelpToolBar and the Topic Grid. The Topic Grid is the easiest and most commonly used way, so I will show only that way now.

Start with selecting the part of the topic text that you want to turn into a link. Then right-click in the Topic Grid on the topic that you want to create a link to. Then select Create Link from the context menu. You will see that the selected part of the topic text is now struck out and green. It is struck out because that tells the help compiler this text should become a link in the final help file. The color doesn't matter to the help compiler, but makes it easier for you to see links while editing your help project. After the link text, you will see a maroon linkID tag. The number in the tag is the Topic ID of the topic which the link points to. If you want to see the topic itself, click on the tag with the left mouse button to put the text cursor inside the tag. Then click with the right mouse button, and pick Follow Link from the context menu.

You have already noticed that HelpScribble is not fully WYSIWYG. Links do not appear as actual links, but as tags. The benefit of this is that the tags are plain text, which you can edit just like any other text. So you can make quick changes to the text of the link, or the target Topic ID, without

having to navigate through menu items and dialog boxes. When copying and pasting text, links are copied and pasted just like the rest of the text.

Adding topics and links is all there is to creating a functional help file. Save your project with Project|Save. To test your help file, select Project|Make from the menu. When you do that the first time, the Project Options screen will appear. You need to select a help compiler on the Compiler page before the final help file can be produced. At this time, it does not matter which compiler you use.

How to Add an Index

Creating an index for your help file is very easy with HelpScribble's Index Editor which you can find in the Editors menu. If you want, you can see the Index Editor in action in a self-running demonstration.

Creating a solid index for your help file is very important. People do not like to read help files. They just want to quickly get an explanation of something and move on. So make sure they can easily find that something in the index. Every topic in your help file should appear in the index. If a concept consists of two words, such as "save project", also include the reversed concept in the index, e.g. "project,save". Since "project save" is not good English, I have made "save" a sub-keyword of "project". If a concept has synonyms, include all synonyms in the index, even if you only use one term exclusively. E.g. the HelpScribble software always uses the term "bitmap", but you will still find "image" and "picture" in the index of this help file.

To add a keyword to the index, click the New Keyword button in the Index Editor and type it in. It will be added to the index, which is shown in the left side of the Index Editor. The new keyword is indicated in red, because you did not yet assign any topics to it. To assign a topic to the keyword, double-click on the topic in the list in the lower right corner. This list holds all topics in your help project. Topics that are assigned to the currently selected keyword are shown in blue. They are also added to the list in the top right corner. You can remove a topic from a keyword by double-clicking on it in the top right list.

You can also create sub-keywords below a keyword. Select the keyword to which you want to add the sub-keyword, or select a sub-keyword of that keyword. Then click the New Sub-keyword button and type in the sub-keyword. Assign topics to the sub-keyword as you assign them to a main keyword. You cannot create sub-sub-keywords below sub-keywords.

Note that you should also assign topics directly to the main keyword, even if it has sub-keywords. Otherwise, WinHelp will not display the main keyword as you intended. HTML Help will display it as intended, but will give an error message when the user selects the main keyword instead of one of the sub-keywords. If you do not assign topics to a keyword that does not have sub-keywords, that keyword will be deleted from the index automatically when you close the Index Editor.

The Index Editor makes it easy to create an index. What it really does is assigning keywords to topics. If you want, you can also manually edit the keywords assigned to each topic. You can do so in the PropertiesBar, in the Topic Grid and by selecting Topic|Properties from the menu. Simply type in the keywords you want in the topic's keywords property. Separate keywords with a semicolon. Separate sub-keywords from their main keywords with a comma. When you open the Index Editor, it will automatically rebuild the index from the keywords that you assigned to the topics.

How to Add a Table of Contents

A help file actually needs two tables of contents. The first topic in your help file, which is the topic with the lowest Topic ID, should welcome the user to the documentation. It should tell the user what information is available, with links to the most important topics or topics that start sections of your help file. This is because when the help file is opened by the user, the first topic will be shown.

The other table of contents is the hierarchy of headers and topics. In WinHelp, this table of contents is shown on top of the help text when clicking the Contents button. In HTML Help, this table of contents is shown in the left hand part of the help window. The user can expand and collapse headers, just like folders in Windows Explorer.

To create this table of contents, use HelpScribble's Contents Editor which you can find in the Editors menu. If you want, you can see this editor in action in a self-running demonstration.

To add a header to the table of contents, click on the Header radio button. Then type in the text of the header in the Title field. Click the Add button (with the green plus sign) to add it to the table of contents.

To add topics below a header, select the header first. Then click the "link to this help file" radio button. Type in the text the topic should be referenced by in the table of contents in the Title field. Select the topic from the drop-down list and click the Add button. Another way to add a topic is to expand the Contents Editor by clicking the >> button next to the Title field. Then you can double-click on topics in the grid to add them.

To modify a topic or a header, click on it. Specify the new Title and Topic values, and click the Change button (with the yellow cross).

To create a header as a sub-header under another header, select the header under which it should appear first. If that header has topics under it, select the last topic under that header. Then add the new header as explained above. This will add the header at the same depth as the header under which you wanted to create it. To make the new header a sub-header, click on the Move Right button (with the blue arrow pointing to the right).

You can create seven levels of sub-headers under each header. Counting the top header and the topics under the last sub-header as well, the table of contents can be nine levels deep. However, I recommend that you do not create sub-sub-headers below sub-headers, because people tend to have difficulties finding what they want in deep hierarchical structures. It is better to make the table of contents longer than to make it deeper.

How to Add Images and Screen Shots

Adding images to your help file is very easy. If you want to include something other than a screen shot, you will need to use a graphics program to create the image. Save it as a .bmp (Windows bitmap) or .gif (Graphics Interchange Format, sometimes called "CompuServe bitmap") file. Windows bitmaps should not be compressed.

In HelpScribble, select HelpTools|Insert Bitmap from the menu. Select the .bmp or .gif file you want to include.

After selecting the file, you will be asked if the image should be character-aligned with no text flowing left or right, left-aligned with text flowing around the right, or right-aligned with text flowing around the left. The transparency option only has an effect when compiling into WinHelp format. It will make all pure white pixels in Windows bitmaps (.bmp) transparent, and all pixels with palette index 15 (counting from 0) in .gif files. In HTML Help, .gif files are transparent if you saved them that way in your graphics program.

Note that the image will not become visible in HelpScribble itself. All you will see is a textual tag with the file name of the image. This makes editing easier since the tag is a simple piece of text that you can edit, and allows the final help file to be better optimized.

Including screen shots in your help file is easy. If you already have a special program for taking screen shots, you can use that to save the screen shots as .bmp or .gif files. Include them in your help file as explained above. If you do not have such a program, simply press the Print Screen key on the keyboard to capture the entire screen. Press Alt+Print Screen to capture only the active window. Either way, the screen shot will end up on the Windows clipboard and you can paste it into any graphics program.

You can also paste the screen shot into HelpScribble's SHG Editor which you can find in the Editors menu. Select Edit|Paste Bitmap from the menu in the SHG Editor. After pasting, select File|Save As. Select the .bmp or .gif file type from the drop-down list at the bottom of the Save As screen, and save. Then include the file you just saved into the help text as explained above.

A very powerful way to work with screen shots is to make them clickable. See [How to Make Images Clickable](#) to learn how to do this with HelpScribble's SHG Editor.

[HelpScribble Help Contents](#)

How to Make Images Clickable

With HelpScribble's SHG Editor, which you can find in the Editors menu, you can add clickable regions to images. When clicked, another help topic will be displayed, just like when a textual link is clicked. This can be very useful with screen shots. By adding a clickable region on top of each element on the screen, and making that region link to a topic with more information about that element, the user can learn about the software simply by clicking on the area of interest.

First, we need to get the image we want to make clickable into the SHG Editor. You can select File|Open from the menu in the SHG Editor to open an existing .bmp, .gif or .png file. You can also use Edit|Paste Bitmap to use the image from the clipboard.

To create a clickable region, use the mouse to draw a rectangle on the image in the SHG Editor. Click and hold down the mouse button in one corner, move the mouse pointer to the other corner, and

release the mouse button. After creating the region, you can move it with the mouse by clicking in the middle, holding down the mouse button, and moving the mouse pointer. You can resize the region by doing the same with the corners of the region.

Type in the name of the region or hotspot in the Name field. This name is only for your reference, to make it easier to edit the hotspots later. Select the topic the hotspot should link to from the Topic drop-down list. If the topic only contains a little bit of text, you can select the Popup Link radio button. The difference between Popup Link and Link is that popup links will display the topic in a small floating window. Regular links will replace the current topic with the topic linked to.

When you're done adding hotspots, save the file as an .shg file. HelpScribble will ask if you want to save with only 16 colors, or with true color.

After saving the .shg, you can include it in your help file by selecting HelpTools|Insert Bitmap from the menu.

How to Make a Help File Browsable

WinHelp can display <<< and >>> browse buttons that allow the user to browse through the topics in your help file. The order in which the topics are browsed through is determined by the browse sequences that you can create with HelpScribble's Browse Sequence Editor.

To create a browse sequence, double-click on the topics in the left hand list in the Browse Sequence Editor. This will move them to the right hand list. You can change the order of the topic in the right hand list by dragging and dropping them with the mouse. To save the browse sequence, type in the name of the sequence in the field labeled "browse sequence". Then click the "write sequence" button.

You can create more than one browse sequence if you want. Click the "new sequence" button to start with a blank browse sequence. Note that each topic can be part of at most one browse sequence. If a topic is not part of any browse sequence, that topic cannot be reached by clicking the <<< and >>> browse buttons. When that topic is displayed, the browse buttons will be disabled.

To edit a browse sequence, select it from the drop-down list and click the "read sequence" button. Edit the sequence as you want, and then click the "write sequence" button to save the modified sequence.

How to Control Fonts and Colors

When you create a new topic, HelpScribble automatically sets the font style for the topic text and topic title. While you can easily change them, it gets quite tedious if you want to do so for every topic. Therefore, you should modify the TextMacros HelpScribble uses in advance.

To change the default font and paragraph style for the topic title, select the *TopicTitle TextMacro from the drop-down list in the TextBar. Select a part of the topic text of the current topic that already has the desired font and paragraph style. Then click the Change TextMacro button on the TextBar to save the style of the selected text into the *TopicTitle TextMacro. To change the default font and paragraph style for the body text, change the *TopicText TextMacro in the same way.

You can also create your own TextMacros. First, select the part of the topic text that has the font and paragraph style that you want to save. Click the New TextMacro button in the TextBar. <new macro> will appear in the drop-down list. Type in a new name for the macro, and then click the Change TextMacro button. After creating a TextMacro, you can quickly give any part of the text of any topic the same font and paragraph style. Select the TextMacro from the drop-down list, and then click the Apply TextMacro button or press Ctrl+M on the keyboard.

Specifying the background color for your help file is easy when compiling to HTML Help. Simply select the color in Project Options: HTML Help.

If you are using the WinHelp format, you need to use the Windows Editor to create a new window called `main` and set the colors there. You can do so by clicking the Add button in the Windows Editor. Type in `main` in the "window name" field and unmark the "default colors" checkbox. Click the "window color" button to select the background color for your help topics. If you are using the HelpTools|Keepn feature, you will need to specify the color for the non-scrolling region as well. Click OK to close the Windows Editor.

Note that WinHelp will not use the colors that you specified in the Windows Editor if the default window background color is set to anything but white in the Windows Control Panel.

How to Choose Between WinHelp and HTML Help

There are two standard help systems on the Windows platform: WinHelp and HTML Help. WinHelp was originally developed by Microsoft to provide context-sensitive electronic documentation with their Office products. With the release of Windows 3.1, WinHelp became an integral part of Windows itself. Microsoft made a help compiler publicly available that could be used to compile a distributable .hlp file from a specially formatted .rtf file and an .hjp project text file. Microsoft did not provide easy to use tools to develop help files, creating a market for products such as HelpScribble. HelpScribble will take care of all the cumbersome and error-prone aspects of providing the help compiler with the source files it wants.

With the release of Windows 95, Microsoft created a 32-bit version of WinHelp, also known as "WinHelp 4", and released a 32-bit help compiler. To the end user, the only difference between the 16-bit and 32-bit format is the added support for a hierarchical table of contents. However, the 32-bit compiler, while it still has bugs, is more robust than the 16-bit compiler.

Windows 98, while still fully supporting both 16-bit and 32-bit WinHelp files, introduced Microsoft's latest help format: HTML Help. HTML Help is very similar to WinHelp, except that the compiler for it requires HTML files as its source, instead of RTF files. That is a benefit if you create help files by hand, since HTML files are easier to create with a plain text editor than RTF files. However, this makes no difference when using an integrated tool like HelpScribble. Some believe that the real motivation behind HTML Help is to further push Internet Explorer and drive Netscape out of the market. The HTML Help viewer, included with Windows 98 and later, uses Internet Explorer to render topics to the screen.

The most obvious difference, to the reader of the help file, is that HTML Help displays the table of contents and the help text side by side. WinHelp displays only the help text, and shows the table of contents in a pop-up after clicking the Contents button.

HelpScribble supports all three formats: 16-bit WinHelp, 32-bit WinHelp and HTML Help. You can download all three compilers from the HelpScribble website and select them in Project Options: Compiler. Since none of the three formats is clearly superior to the others, which one to use depends on what you and your customers expect of your help files. I will simply list the advantages and disadvantages of each format, and leave the choice to you. If the help file will be used for context-

sensitive help with an application, then the development tool used to create the application may restrict your choice. Not all Windows development tools support all help formats (though all tools we know of support at least one).

Note that whichever format you chose now, you are absolutely not stuck with it. With HelpScribble, switching help formats is as easy as clicking on another radio button in Project Options: Compiler and invoking Project|Make once more.

16-bit WinHelp

If you want your help file to work under Windows 3.1, you have no choice but to use 16-bit WinHelp. Note that 16-bit WinHelp files can be viewed perfectly under all later Windows versions, including NT3.1, NT3.5, NT4, 95, 98, ME, 2000 and XP. If you will be using your help file to provide context-sensitive help with a 16-bit application, you also must use 16-bit WinHelp. If you use a 32-bit help file, users running Windows NT, 2000 or XP will get an error message saying the help format is invalid. This is because these operating systems have a clear separation between the 16-bit and 32-bit world. 16-bit applications will always use the 16-bit help viewer, which cannot handle 32-bit help files. This problem does not occur on Windows 95/98/ME, because they do not separate the 16-bit from the 32-bit. When using 16-bit WinHelp, be sure to use the HCP.EXE compiler and not the HC31.EXE compiler. The latter is a real mode program, which will run out of memory quickly as it is limited to 640K.

32-bit WinHelp

This format is an improved version of 16-bit WinHelp. If you will use your help file as context-sensitive help for a 32-bit application, use 32-bit WinHelp or HTML Help. 32-bit WinHelp files work perfectly under Windows 95, 98, ME, NT4, 2000 and XP. They work on Windows Vista, 7, 8, and 8.1 if the user downloads the help viewer from Microsoft. .NET applications cannot use WinHelp files for context-sensitive help.

HTML Help

Windows 98, ME, 2000, XP, Vista, 7, 8, 8.1, 10, and 11 have built-in support for HTML Help. In theory, support for HTML Help can be added to Windows 95 and NT4 as well by installing Internet Explorer 4 or later, plus the HTML Help ActiveX controls. In practice, we've found that this can be troublesome. Not just the fact that you have to decide if you will bundle the ActiveX controls with your application (which you can), but it's not uncommon for the installation of the ActiveX controls to fail. If you want to support Windows 95 or NT4, HTML Help is not recommended for the above reasons, and because computers running those older operating systems will probably be quite slow as well, and HTML Help will work quite sluggishly on them.

HTML Help is recommended if your application needs to support Windows Vista, 7, 8, 8.1, 10, or 11. It's your only choice if you want to support Windows 10 or if your application is developed using the .NET framework. Windows 10 and 11 cannot display WinHelp files and .NET applications cannot use WinHelp files for context-sensitive help.

One restriction of HTML Help is that HTML Help CHM files cannot be reliably stored on the network. Computers with the latest Microsoft security updates applied, using the default security settings, will not display the body text of CHM files that are stored on a network drive. Unless you are able to change the security settings on all computers that will be viewing your help file, you will either need to install the HTML Help CHM file on each computer, or use WinHelp format instead. WinHelp files work flawlessly across networks.

For total compatibility with Windows 95 through Windows 11, you'll need to provide your help file in both WinHelp and HTML Help formats. With HelpScribble, you can change the format of the compiled help file at any time by making another choice in Project Options: Compiler and running the help compiler again with Project|Make.

Viewing CHM Files Across Networks

Microsoft released a security patch that makes it impossible to view CHM files that are stored on a network drive (as opposed to CHM files stored on your own computer). The help file will appear, but instead of the topic text you will see an error message "this page cannot be displayed".

The security patch and its effects are fully described on Microsoft's web site at <http://support.microsoft.com/kb/896358>

If you trust all the computers on your LAN or intranet (and the people using them), you can lower the restrictions on the Local Intranet zone to allow CHM files to be displayed with the following steps:

1. Click Start, click Run, type regedit, and then click OK.
2. Locate and then click the following subkey:
HKEY_LOCAL_MACHINE\SOFTWARE\Microsoft\HTMLHelp\1.x\ItssRestrictions
If this registry subkey does not exist, create it.
3. Right-click the ItssRestrictions subkey, point to New, and then click DWORD Value.
4. Type MaxAllowedZone, and then press Enter.
5. Right-click the MaxAllowedZone value, and then click Modify.
6. In the Value data box, type 1, and then click OK.
7. Quit Registry Editor.

If this solution does not work for you, please see <http://support.microsoft.com/kb/896358> for alternatives.

The security update may also prevent you from opening CHM files that you downloaded from the Internet. To resolve this problem, click to clear the Always ask before opening this file check box in the Open File-Security Warning dialog box, and then click Open. Alternatively, right-click the file that you want to open, click Properties, and then click Unblock. See <http://support.microsoft.com/kb/902225/> for more information.

If your customers are having problems accessing your CHM files, you have the following options:

1. Make them aware of the issues involving the security update
<http://support.microsoft.com/kb/896358>
2. Make sure your software's installer installs CHM files locally rather than on the network. Then they will always work.
3. Use the WinHelp format instead of HTML Help (CHM). You can make this choice in Project Options in HelpScribble. WinHelp HLP files are not affected by this security issue.

Viewing WinHelp Files on Windows Vista through 8.1

Windows Vista and later no longer include the WinHelp viewer. This means that .hlp files won't work on an out-of-the box installation of Windows Vista, 7, 8, or 8.1. Because of this we recommend that you compile your help file into the HTML Help format if you need to support Windows Vista or later.

If switching to HTML Help is not an option, Microsoft does make the WinHelp viewer available as a separate download for Windows Vista 7, 8, and 8.1. You'll need to instruct your users to download and install it. You cannot bundle the update with your own installer. There's a different download for different versions of Windows:

Windows Vista: <http://go.microsoft.com/fwlink/?LinkID=82148>
Windows 7: <http://go.microsoft.com/fwlink/?LinkId=166421>
Windows 8: <http://www.microsoft.com/en-us/download/details.aspx?id=35449>
Windows 8.1: <http://www.microsoft.com/en-us/download/details.aspx?id=40899>
Windows Server 2008: <http://go.microsoft.com/fwlink/?LinkId=114216>
Windows Server 2008 R2: <http://go.microsoft.com/fwlink/?LinkId=166422>

The downloadable WinHelp viewers for Windows Vista and later are not identical to the WinHelp viewer that shipped with Windows XP and earlier. In particular, Microsoft has disabled the WinHelp viewer's ability to link to external files or URLs. Thus any links created with HelpTools|Internet Link and HelpTools|Link to File will not work when you compile your help file into WinHelp format and the .hlp file is opened on Windows Vista or later.

The WinHelp viewer is not available for Windows Server 2012. Microsoft has stated that WinHelp will not be supported on Windows Server 2012 or any future versions of Windows Server.

The WinHelp viewer is also not available for Windows 10 or 11. While there has been no official statement from Microsoft, it is obvious that WinHelp is no longer supported on the desktop version of Windows either.

How to Link Your Help File to an Application

HelpScribble generates standard WinHelp .hlp files and HTML Help .chm files that can be connected to any Windows application, regardless of the development tool used. All development tools for the Windows platform support either the .hlp or the .chm format. Some support both.

Linking the help file to an application, or implementing context-sensitive help, needs to be done by the programmers developing the software. They will need a list of topics and topic IDs. Use Project|Export Topic Grid to save a list of topics and IDs to a text file. You can easily email this file to the development team, who can view it with their favorite text editor. Or, use Print|Print Topic List to create a printout on paper instead.

The actual mechanics of linking the application and the help file depend on the help authoring tool used. Our web site at <http://www.helpscribble.com/> explains the procedure for some of the more popular development tools, including Visual Basic, .NET (C#, VB.NET, etc), VBA (MS Office), Delphi, etc.

If the development tool references topics by numeric ID, that is the Topic ID in HelpScribble. If it needs string IDs, that is the Context String in HelpScribble. Select Topic|Properties from the menu to view or change the Context String. The Context String of topic 1234 is Scribble1234 by default. If you need the file name of the topic inside the .chm file (compiled HTML Help file), that is html\hs1234.htm for topic 1234, assuming the context string is set to its default value of Scribble1234. If not, the file name of the topic inside the .chm file is html\contextstring.htm.

How to Distribute Your Help File

Before you can distribute your help file, you need to compile it. You can do so by selecting Project|Make from the menu. You will need to decide whether you want to compile into WinHelp format, or in HTML Help format. See [How to Choose Between WinHelp and HTML Help](#) for detailed information.

If you have chosen the WinHelp format, Project|Make will produce a .hlp file through the WinHelp compiler. If you used the Contents Editor, the table of contents will be saved into a separate .cnt file. You have to distribute the .cnt file together with the .hlp file. Otherwise, the table of contents will not be available on the user's computer. However, no error message will be displayed in that case. All images used in your help project will be embedded into the .hlp file, so you do not have to distribute them separately.

When viewing a .hlp file, WinHelp will create a (hidden) .gif file in the same folder as the .hlp file. This file contains a copy of the indexing information in the .hlp file. WinHelp does this to speed up navigating between topics. However, you won't notice the difference unless you're using an old 486 computer. You should not distribute this .gid file as it will be regenerated automatically the first time the .hlp file is viewed. Not distributing it prevents the .gid and .hlp file from getting out of sync.

An important thing to note is that WinHelp identifies help files by filename only. If you have two .hlp files with the same filename on your computer, WinHelp will assume them to be identical. If they are not, WinHelp may display the wrong help file. Therefore, if you want to keep backups of your compiled help file, you should put them inside a .zip archive, so WinHelp will not show outdated help topics when you're testing the latest version. If you ever seem to get outdated text in WinHelp after compiling your help file, search for and delete any .hlp or .gid files on your computer.

If you have chosen the HTML Help format, Project|Make will produce a .chm file through the HTML Help compiler. All images used in your help project, as well as the table of contents, will be embedded into the .chm file, so you do not have to distribute them separately.

WinHelp files can be viewed on any computer running Windows 95, 98, ME, NT4, 2000 or XP. 16-bit WinHelp files also work on Windows 3.1, NT 3.1 and NT 3.5. 32-bit WinHelp files work on Windows Vista, 7, 8, and 8.1 if the user downloads the WinHelp viewer from Microsoft. WinHelp files cannot be viewed on Windows 10.

HTML Help files can be viewed on any computer running Windows 98, ME, 2000, XP, Vista, 7, 8, 8.1, 10, or 11. In all of these cases, you do not need to distribute any software along with the help file.

The compiled help files are completely independent of HelpScribble. As long as they were produced with a fully licensed copy of HelpScribble, you can do with them as you please.

How to Change Help Authoring Tools (Import/Export)

If you have previously used another help authoring tool, you can use Project|Import from RTF+HPJ to migrate your help project to HelpScribble. HelpScribble cannot import HTML Help projects, only WinHelp projects.

If you do not have the source files of the project, or the other tool cannot export to .rtf+.hpj, you can download a WinHelp decompiler from the Internet. The file is helpdc21.zip and is available at <http://www.helpscribble.com/decompiler.html>. You can use this program to decompile a .hlp file into a pair of .rtf and .hpj files.

Since the .rtf format is very loosely defined, it is possible that HelpScribble cannot properly import the project. Microsoft Word is notorious for creating messy .rtf files. In that case, use the decompiler to decompile your .hlp file into .hpl and .rtf, and import the decompiled files into HelpScribble. The decompiler generates very clean .rtf files and HelpScribble will import them just fine.

If you want to migrate your project from HelpScribble to another help authoring tool, you can use Project|Export to RTF+HPJ or Project|Export to HTML Help. Use the former command to migrate to a WinHelp-based help authoring tool, and the latter to migrate to an HTML Help-based help authoring tool.

Note that if you want somebody else to edit your help project and then send the changes back to you for further editing, you should send them your .hsc HelpScribble project file which they can edit with HelpScribble itself. That is far more reliable and practical than exporting and importing again.

How to Write a Great Help File

A lot of software these days ships with help files of very questionable quality. Even help files included with expensive software from big companies often leave a lot to be desired.

Just what is a great help file? It's simple: a great help file is a help file that provides the user with exactly that information which the user needs at any time while requiring a minimum amount of effort to locate that information.

An obvious requirement that follows from our definition of a great help file is that every aspect of the software must be documented, because you cannot know in advance which aspect of the software a particular user may want help with. However, unless your software is very limited in functionality, documenting every little detail will make it more difficult for novice users to see the forest through the trees.

Therefore, a great help file consists of two main sections. The first section is a "how to" section that explains the basic tasks of the software. It focuses on how to get something done with the software, rather than trying to explain everything. Information is presented in a logical order with respect to the user's work flow, possibly using step by step instructions. The second section is the reference section that explains each and every detail of the software. It focuses on explaining what each part of the software can be used for.

People new to the software will mostly read the "how to" section, while expert users will look up the information they want in the reference section. However, most people using the software will actually be neither beginner nor expert, but something in between. Therefore, both sections should link to each other. Whenever the "how to" section mentions a feature, that mention should be a link to the reference section. As the user gains more experience, she can easily find information in the reference section by using the "how to" topic she is already familiar with as a starting point. In the reference section, topics explaining the details of a certain part of the software should link to the relevant "how to" topics. Somebody new to the software may wonder what a certain command is for, and arrive in the reference section via the index, while a basic "how to" text would be more interesting for that person at that time.

Speaking of links: a great help file can never have enough links. Whenever you mention a feature or concept that is explained elsewhere, make that mention a link so the user can easily get more information. Remember that help files are not read cover to cover. They don't have covers. Rather, people will read a few topics when they get stuck with the software. So you cannot assume anything about what the customer already knows and what not.

Another important and related fact is that most readers will arrive at a certain topic “out of nowhere”. Maybe they found the topic through the index or they clicked on a link without really knowing what to expect. Therefore, every topic must establish a context. If a topic explains a certain menu item, like Project|Save, that topic must first mention that it is talking about a menu item, and where that menu can be found. If I look up “save project” in the index, and I get a topic that only says “click this button to save your project”, then I have learned absolutely nothing. I want to know *where* that button is. It seems that a lot of technical writers did not fully grasp this yet. A help file is not a book. I cannot flip back a few pages to establish the context. In WinHelp, browse sequences can simulate “page flipping”. But a link to the Project menu from the Save topic is just so much easier than clicking the <<< button three times and then guessing I may find the Save item in the Project menu.

I have already mentioned the use of the index several times. A great help file must have a comprehensive index. Make sure that each topic is included in the index, with every synonym you can think of, and with every possible permutation of multi-word concepts. See [How to Add an Index](#) for more information.

Screen shots are also an important part of a help file. Usually the software will be hidden behind the help window. Looking at the screen shot in the help file while reading the help text is easier than task-switching back and forth between the application and the help file. Seeing a screen shot that is identical to the actual software also assures the user that the help file is accurate. So when you add screen shots, it is very important to update them as the software gets updated. When a screen shot differs from the actual software, the user will not trust the help text.

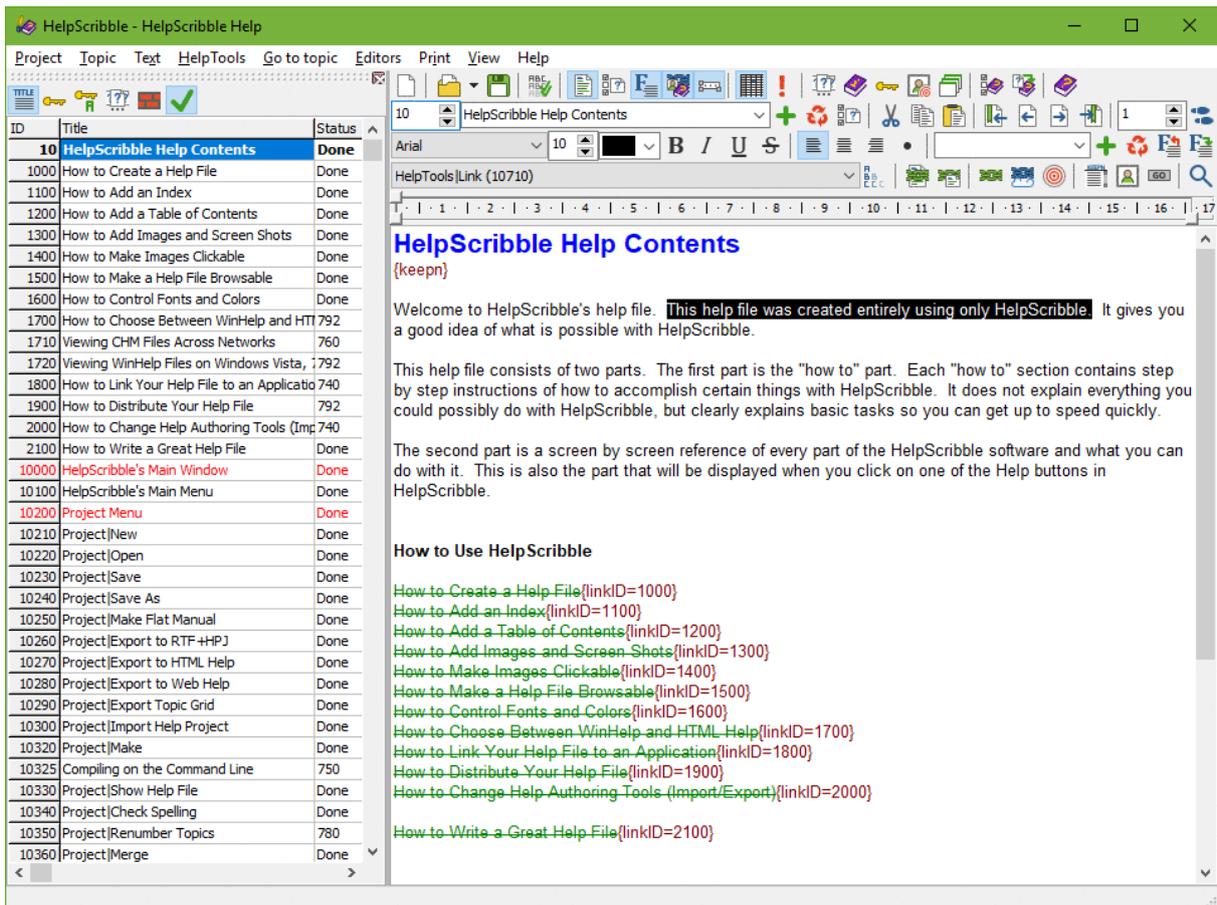
If your help file is distributed with a free evaluation version of the software, the quality of your help file will definitely influence people’s decision whether to buy this software or a competing product. Though many people will hardly use your help file at all, most will take a look at it to check if solid documentation is available, should they ever get stuck with the software. It is also far easier to verify the quality of a help file than the quality of an application. Anybody can read a help file. Compare this to taking an airplane. Most people cannot even estimate the mechanical reliability of an airplane, and the skills of its pilots. But old coffee stains on the seats and overdue paint jobs, while they have no effect on the airworthiness of the plane, are easy to spot and will determine what people think about the plane and the airline.

So be sure to write a great help file. HelpScribble takes care of all the technical details so you can focus on providing solid content.

HelpScribble's Main Window

This is the main window of HelpScribble. In this window you will perform the basic tasks of creating a help file: adding topics and linking them.

All of HelpScribble's features can be accessed through the main menu. Often-used features can be used quickly from the Topic Grid or the various toolbars. In the screen shot below, you can see the Topic Grid at the left hand side. Just below the menu, you can see two toolbars: the ProjectBar and the TextBar. The other three available toolbars, the TopicBar, the PropertiesBar and the HelpToolBar, were hidden through the View menu when the screen shot was taken.



HelpScribble's Main Menu

- Project Menu: Commands related to HelpScribble projects. Open, save, import, export and compile.
- Topic Menu: Commands related to the topic you are currently editing in HelpScribble.
- Text Menu: Commands related to the text of the topic you are currently editing in HelpScribble.
- HelpTools Menu: Commands for inserting special items such as images and links in the text of the topic you are currently editing.
- Go to topic Menu: Commands to start editing another topic.

Editors Menu	Commands for opening specialized editors built into HelpScribble for creating tables of contents, indices, clickable images, etc.
Print Menu	Commands for printing certain parts of your HelpScribble project.
View Menu	Commands for showing or hiding certain parts of HelpScribble's main window.
Help Menu	Commands to get help on how to use HelpScribble.

Project|New

Select the New item from the Project menu or click the New Project button on the ProjectBar to start with a new HelpScribble project.

Project|Open

Select the Open item from the Project menu or click the Open Project button on the ProjectBar to open an existing HelpScribble project.

You can use the Project|Reopen menu or click the downward pointing triangle to the right of the Open Project button to quickly reopen a HelpScribble project that you recently worked on.

Project|Save

Select the Save item from the Project menu or click the Save Project button on the ProjectBar to save the project you are currently editing into a .hsc HelpScribble project file. If you have previously saved the project, it will be saved again using the same file name, overwriting the previous version. If you don't want that, use Project|Save As instead. If the project is unsaved, HelpScribble will ask you under which name and in which folder you want to save it.

Project|Save As

Select the Save As item from the Project menu to save the project you are currently editing into a .hsc HelpScribble project file. HelpScribble will ask you in which folder you want to save, and under which name.

Project|Make Flat Manual

If you want to create both online help and a printed manual from the same text, you can write the text in HelpScribble and then use the Make Flat Manual command from the Project menu to create the text for the printed manual.

In the file dialog box that appears, you can select the file which the flat manual should be saved to. If you give the file an extension of .rtf it will be saved in Rich Text Format, otherwise HTML will be used. If you save as an RTF file, you will be asked if you want the topics to be separated by page breaks or not. The generated RTF will not contain any weird control codes or hyperlinks (as does the RTF created by the Export to HPJ+RTF function). The RTF will also not contain any images. You will have to re-insert them using a word processor or DTP program.

If you save as an HTML file, you will be asked if the HTML file should contain hyperlinks. If you will be distributing the HTML file electronically, you may want to include hyperlinks, otherwise, leave them out. All images in your help project will be saved in either GIF or PNG format in the same folder as the HTML file. GIFs in your help project will simply be copied over. SHGs and BMPs in your help project will be converted to GIF or PNG, depending on the choice you made in Project Options: HTML Help.

Sample strategy for creating a printed manual

1. Open your file in HelpScribble, select Project|Make Flat Manual.
2. Type in a filename with a .html extension. Choose to not include hyperlinks when asked.
3. Open the .html file in Microsoft Word 2000 or XP. (Earlier versions do not handle HTML as well.)
4. Select View|Print Layout from Word's menu.

You can now edit the flat manual as a regular word document. Insert page breaks (Press Ctrl+Enter) where you would like them, and use File|Print to produce the printed manual. You should save the manual as .doc file in MS Word format, rather than as HTML.

Project|Export to RTF+HPJ

Select Export to RTF+HPJ from the Project menu to export your help project to a pair of RTF and HPJ files. These are the source files that the WinHelp compiler needs to create a .hlp file from your help project. They can also be imported by other help authoring tools.

This feature is useful if further editing of your help project will be done using other tools, perhaps because another person or company will take over the project. If you want somebody else to edit your help project and then send the changes back to you for further editing, you should send them your .hsc HelpScribble project file which they can edit with HelpScribble itself. That is far more reliable and practical than exporting and importing again.

Project|Export to HTML Help

Select Export to HTML Help from the Project menu to export your help project to a set of files that the HTML Help compiler needs to create a .chm file from your help project. HelpScribble will ask you for the name of the .hhp HTML Help Project file. The .hhc and .hhk files will be saved in the same folder, with the same filename (but different extension). Topics files will be saved in a sub-folder called "html", and bitmaps in a sub-folder called "images". HelpScribble will create the sub-folders if they don't already exist. Any files that already exist will be overwritten without warning.

This feature is useful if you want to tweak the help project using Microsoft's HTML Help Workshop before compiling it.

If you simply want to compile the project, use Project|Make instead to directly create the .chm file.

Project|Export to Web Help

Select Export to Web Help from the Project menu to export your help project to a series of HTML files that you can upload to your web site to provide online documentation.

There are several options to choose from.

You have three choices for the layout of the HTML pages:

- Simple pages: create one HTML file for each topic, with no navigational aids besides the links already present in the topic text.
- Simple pages with navigation header: create one HTML file for each topic with a navigation header added at the top. This header has a link to an HTML file showing the table of contents and a link to an HTML file with the index. If you have used the browse sequence editor, Next and Previous links will be added as well, implementing the browse sequence.
- Contents in frame: Create a frameset page showing the contents or index in the left frame (both with a link to each other), and the topic text in the right frame. The topic text will have the navigational header as in option 2.

If you turn on the "use style attributes for indentation", HelpScribble will use paragraph tags and style attributes in the HTML code to preserve the layout of the text as you entered it in HelpScribble. If your readers have recent browser versions installed, such as Internet Explorer 5 or later, or Netscape 6 or later, you should turn on this option.

Older browsers such as Netscape 4 or earlier do not support or only partially support cascading style sheets or the style attribute. If your readers are likely to use such browsers, you should turn off "use style attributes for indentation". HelpScribble will then use HTML code that preserves the layout of your help topics in all browsers, except that indentation will be lost. If you do not turn this off, readers with old browsers can still read your help text, but there will be excessive white space between some or all paragraphs.

You can choose whether images in BMP and SHG format in your help project should be converted to GIF or PNG format. PNG is the best choice since it does not have GIF's 256 color limitation and, more importantly, is free of any kind of patent issues. Choose GIF if compatibility with ancient (Netscape 3, MSIE 3) browser versions is important to you.

Since you cannot be sure if the person reading your documentation on your web site has all the fonts installed that you have, you may want to use the Font Substitution grid to automatically substitute fonts. In the left column of the grid you need to put the name of the font you want to substitute. In the

right column, place the alternative font names, separated by commas. The web browser will first try the fonts from left to right, until it finds one that is installed on the reader's computer. It is a good idea to put the original font name as the first one in the list, just in case the reader also has this font installed. The default settings substitute the standard Windows fonts with both those fonts and their alternatives on UNIX systems.

Since UNIX systems typically render text smaller than Windows systems, you may want to have HelpScribble use larger font sizes in the HTML output if you know you will primarily be using the generated HTML files on UNIX systems. If you are developing a cross-platform application for both Windows and Linux using Delphi and Kylix, you could use a WinHelp file on Windows, and HTML files created with "Export to Web Help" on Linux.

Finally, the text items are short bits of text used by HelpScribble to add the optional navigational headers to the HTML output. If your help project is not written in English, you will want to translate them. %s is a placeholder for the project title you specified in Project Options: General.

When you click OK, HelpScribble will ask for the filename of the main HTML file. If you will be uploading the files to a web server, `index.html` is probably a good choice since most web servers send that file when a user tries to access a directory.

Project|Export Topic Grid

Select Export Topic Grid from the Project menu to create a text file that lists all the topics in your HelpScribble project. The topics will be listed in the order that they presently have in the topic grid. The properties presently visible in the topic grid will also be written to the text file.

This can be useful if your help file will be used for context-sensitive help. You can give a copy of the exported topic grid to the application developers who can use it as a reference to insert the appropriate Topic IDs into the software. (Applications request help by specifying the filename of the help file and the Topic ID number.)

You can also print a list of topics using Print|Print Topic List.

Project|Import Help Project

If you have previously used another help authoring tool, you can select Import Help Project from the Project menu to import help files created with the other tool into HelpScribble.

If you do not have the source files of the project, or the other tool cannot export to .rtf+.hbj, you can download a WinHelp decompiler from the Internet. The file is helpdc21.zip and is available at <http://www.helpscribble.com/decompiler.html>. You can use this program to decompile a .hlp file into a pair of .rtf and .hbj files.

Since the .rtf format is very loosely defined, it is possible that HelpScribble cannot properly import the project. Microsoft Word is notorious for creating messy .rtf files. In that case, use the decompiler to decompile your .hlp file into .hbj and .rtf, and import the decompiled files into HelpScribble. The decompiler generates very clean .rtf files and HelpScribble will import them just fine.

Note that you cannot import arbitrary .rtf files. HelpScribble cannot automatically convert any document into a help project. Only .rtf files that are part of a help project defined by an .hbj file can be imported. If you want to reuse other text, open the text in a word processor or other application, and use copy and paste to get the text into HelpScribble.

Project|Make

Select Make from the Project menu, or click the Make button on the ProjectBar to create a compiled .hlp or .chm file from your HelpScribble project. This is the file that you will distribute to your customers.

Before you can compile your project, it must have been saved at least once, so it has a file name. You also have to select the help compiler that HelpScribble should use in Project Options: Compiler. The compiler selection determines whether a WinHelp .hlp file or an HTML Help .chm file will be created.

After the project has been compiled, HelpScribble will show the help compiler's report in the error log below the topic text. If the .hlp or .chm file was successfully created, HelpScribble will show it. If you created a .hlp file, HelpScribble will tell WinHelp to show the topic you are currently editing in HelpScribble. If you created a .chm file, it will be opened at the first topic.

Compiling on The Command Line

Use the following command line syntax to compile a HelpScribble project from the command line, or to integrate HelpScribble with an automated build tool.

```
HelpScr.exe project.hsc /c [/btag1;tag2;tag3] [/q]
```

The parameters must be listed in the same order as shown above. The parameters shown inside square brackets are optional.

Substitute project.hsc with the full path to your .hsc project file. If it contains spaces, remember to put double quotes around it. The /c parameter tells HelpScribble to start compiling the help file right away, as if you selected Project|Make from the menu. HelpScribble will be visible on the screen while compiling, so you can follow its progress.

If you want to use a particular output, use /c16 instead of /c to compile into 16-bit WinHelp format. Use /c32 for 32-bit WinHelp format, and /html for HTML Help.

If you have specified Build Tag properties for certain topics, you can select which help topics should be included in the compiled help file via the /b parameter. If you specify /b without any tags, all topics will be included. To include only certain topics, list the build tags you want delimited by semicolons right after the /b parameter. There should not be any space between the /b and the list of tags, nor should there be any space inside the list of tags. Only topics that have this build tag in their build tag property and topics for which you left the build tag property blank, will be included in the compiled help file. The /b parameter overrides the build tag selection in Project Options: Compiler.

If HelpScribble should terminate immediately when the compilation has been completed, add the /q parameter as the last parameter on the command line.

You can also export your project to web help from the command line using the following command line:

```
HelpScr.exe project.hsc /cweb <htmlfile> [/btag1;tag2;tag3] [/q]
```

The third parameter must be a complete path to the main HTML file of the exported web help. Specify the file name and path that you would select when using the Project|Export to Web Help menu item.

HelpScribble can also create the compiled help file from the command line. Use HelpScr.exe helpfile.hsc /c to compile helpfile.hsc from the command line. This is useful if you want to include HelpScribble in an automated build tool.

Project|Show Help File

Select Show Help File from the Project menu, or click the Show Help File button on the ProjectBar to show the compiled help file. The difference with Project|Make is that Make will always recompile the help file. The Show Help File command will only compile the help file if the compiled .hlp or .chm file does not exist yet.

If you created a .hlp file, HelpScribble will tell WinHelp to show the topic you are currently editing in HelpScribble. If you created a .chm file, it will be opened at the first topic.

Project|Check Spelling

Select Check Spelling from the Project menu, or click the Check Spelling button on the ProjectBar to spell check all the topics in your HelpScribble project. If you want to check the current topic only, use Topic|Check Spelling instead.

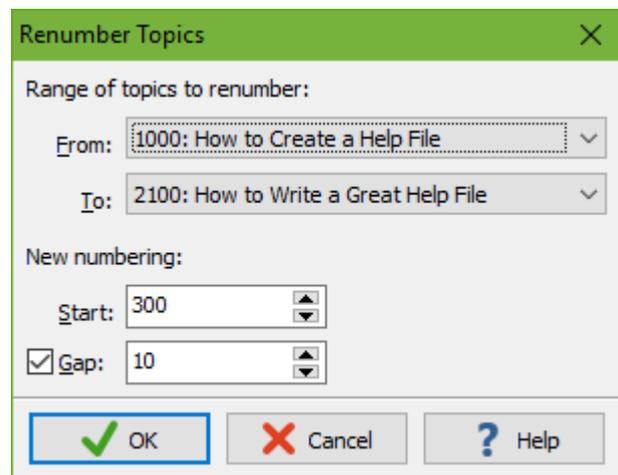
Note that you will need to have installed at least one JGsoft Spell Check dictionary. You can download these by picking Help|Download Spell Check Dictionaries from the menu, or by pointing your web browser to our web site at <http://www.helpscribble.com>.

Project|Renumber Topics

Select Renumber Topic from the Project menu if you want to change the Topic IDs of a range of topics in your help project. This can be useful if you have been inserting many new topics between existing topics and you want to make their Topic IDs equidistant again.

Indicate the range of topics you want to renumber by selecting the first topic from the From list, and the last topic from the To list.

Then specify what the new Topic IDs should be. The first topic in the range (the topic you selected in the From list), will get the Topic ID you specify in the Start box. The following topics will get the ID of the preceding topic increased with the value you specify in the Gap box. E.g. if you renumber topics 10, 17, 19, and 28 with a new starting ID of 100 and a gap of 10, the new IDs will be 100, 110, 120, and 130.



The screenshot shows the 'Renumber Topics' dialog box. The 'Range of topics to renumber:' section has 'From:' set to '1000: How to Create a Help File' and 'To:' set to '2100: How to Write a Great Help File'. The 'New numbering:' section has 'Start:' set to '300' and 'Gap:' set to '10' with the checkbox checked. The 'OK' button is highlighted with a blue border.

If you don't tick the Gap checkbox, then the Topic IDs of the selected range are shifted without changing the gaps in their ID numbers. If you renumber topics 10, 17, 19, and 28 with a new starting ID of 100 and without a gap, the new IDs will be 100, 107, 109, and 118.

Project|Merge

To merge two HelpScribble projects, select Merge from the Project menu. With this command you can add some or all of the topics of another HelpScribble project to the current project.

After selecting the project you want to merge into the current project, HelpScribble will show the screen below. It allows you to select which topics should be added to the current project.

You can select to merge all the topics, or only certain topics based on the topic properties. You can select only the topics within a certain Topic ID range, topics which are part of a certain browse sequence or topics which have the status property set to a certain value. With the last two options, the

marked browse sequences or topics with a marked status property will be added. Topics with no browse sequence or status setting will not be added.

In a Windows help file, each topic must have a unique Topic ID. So, you will have to tell HelpScribble what to do if one or more of the topics you selected to be merged with the current project, have a Topic ID that is already taken in the current project. You can decide to have the topic in the current project replaced with the topic with the same Topic ID from the project to merge. The second option is to skip the topic with the taken ID and thus leave the topic in the current project as it is. When you select the third option, the topic from the project to merge will receive a new, unique topic ID (the first one available following the one it had). Links to that topic will be automatically updated.

The merge function can be very handy in several cases. One situation could be that you need to have your help file translated into different languages. When you update your help file, it would be handy that only the updated topics would have to be translated again.

You can do so by storing version numbers in the Status property of each topic. When you have updated your help file, open the translation of the old version, select Project|Merge and select to add only the topics with a version 2 status value. The status field will also help the translator to quickly locate the topics that have been changed.

Project|Open Delphi Unit

If you want to document a VCL or CLX component written in Delphi, select Open Delphi Unit from the Project menu. This will launch HelpScribble's Delphi Source Code Parser which will automatically generate an outline help file that you can use to integrate the documentation for your component in Delphi's or C++Builder's help system.

Note that you cannot use this feature to document a regular application written in Delphi or C++Builder. If you want assistance with that, you can use the HelpContext Property Editor. In most cases you will want to create the help file as described in How to Create a Help File and then link the help file with Delphi or C++Builder using the property editor.

Project|Exit

Time to take a break!

Topic|Undo

You can use the Undo submenu of the Topic menu to undo changes made to a topic. When you select “changes to the current topic”, HelpScribble will undo all changes you made to the topic you are currently editing. You can also undo changes made to a recently edited topic by selecting it from the Undo menu. Selecting one of the topics will only undo the changes made to that topic. There is no need to undo changes made to topics edited after you edited that topic.

Undone changes can be redone using the Redo submenu of the Topic menu.

Pressing Ctrl+Y on the keyboard will undo all changes you made to the current topic since you started editing it. If you only want to undo the very last editing action, press Alt+Backspace on the keyboard.

Topic|Redo

You can redo undone changes using the Redo submenu of the Topic menu. The Redo menu works just like the Undo submenu.

Topic|New

Select New from the Topic menu or click the New Topic button in the TopicBar to add a new topic to your help project.

The new topic will be inserted after the current topic and, if possible, before the next topic. The Topic ID of the new topic is the ID of the current topic plus the increment specified in Project Options: General, unless that would make the ID greater than that of the next topic. In that case, the new topic will get an ID between the ID of the current topic and that of the next topic.

Topic|Delete

Select Delete from the Topic menu or click the Delete button on the TopicBar to delete the current topic.

HelpScribble will not bother you with an “Are you sure?” confirmation question. If you mistakenly delete a topic, you can restore it using Topic|Undo.

Topic|Cut

Select Cut from the Topic menu or click the Cut button on the TopicBar to remove the current topic from the project, and to place it on the clipboard.

Note that this command will cut the entire topic. If you only want to cut the selected part of the topic text, use Text|Cut instead.

You can also cut a topic by right-clicking on it in the Topic Grid and selecting Cut from the context menu.

Topic|Copy

Select Copy from the Topic menu or click the Copy button on the TopicBar to place the current topic on the clipboard.

Note that this command will copy the entire topic. If you only want to copy the selected part of the topic text, use Text|Copy instead.

You can also copy a topic by right-clicking on it in the Topic Grid and selecting Copy from the context menu.

Topic|Paste

Select Paste from the Topic menu or click the Paste button on the TopicBar to replace the current topic with a topic that you previously cut or copied to the clipboard. The current topic will get the text and all the properties of the topic on the clipboard, except for the Topic ID, which will remain unchanged.

If you want to paste a piece of text into the current topic, use Text|Paste instead.

You can also paste a topic by right-clicking on it in the Topic Grid and selecting Paste from the context menu.

Topic|Properties

Select Properties from the Topic menu to edit the properties of the current topic.

Note that most, but not all, topic properties can be changed directly in the Topic Grid and the PropertiesBar.

Topic|Check Spelling

Select Check Spelling from the Topic menu to spell check the topic you are currently editing. To check the entire project, use Project|Check Spelling instead.

Note that you will need to have installed at least one JGsoft Spell Check dictionary. You can download these by picking Help|Download Spell Check Dictionaries from the menu, or by pointing your web browser to our web site at <http://www.helpscribble.com>.

Text|Font

Select Font from the Text menu to change the current font in the topic text. You will be able to select the font face, style and size in the standard Windows screen for selecting fonts. Alternatively, you can quickly change the font in the TextBar, simply by selecting the font you want from the drop-down list.

Text|Bold

Select Bold from the Text menu, press Ctrl+B on the keyboard, or click the **B** button on the TextBar to toggle the current font in the topic text between boldface and not boldface.

Text|Italic

Select Italic from the Text menu, press Ctrl+I on the keyboard, or click the *I* button on the TextBar to toggle the current font in the topic text between italics and straight.

Text|Underline

Select Underline from the Text menu, press Ctrl+U on the keyboard, or click the U button on the TextBar to toggle the current font in the topic text between underlined and not underlined.

Text|Strike Out

Select Strike Out from the Text menu, press Ctrl+S on the keyboard, or click the ~~S~~ button on the TextBar to toggle the current font in the topic text between struck out and struck out.

Text|Left

Select Left from the Text menu or click the Left Justify button on the TextBar to change the current paragraph in the topic text to be justified to the left margin.

Text|Center

Select Center from the Text menu or click the Center button on the TextBar to change the current paragraph in the topic text to be centered between the left and right margins.

Text|Right

Select Right from the Text menu or click the Right Justify button on the TextBar to change the current paragraph in the topic text to be justified to the right margin.

Text|Bullet

Select Bullet from the Text menu or click the Bullet button on the TextBar to put a bullet at the beginning of the paragraph or to remove it.

Note: You should use the ruler to put a tab stop at the position where the bulleted text begins (i.e. to determine the space between the bullets and the text). Otherwise, the bulleted text will not look as you intended when compiling your project in WinHelp format.

Text|Cut

Use the Cut command from the Text menu to delete the part of the topic text you have selected and put it on the clipboard. You can also cut text by pressing Shift+Del on the keyboard or by right-clicking on the topic text and picking Cut from the context menu. If you want to cut an entire topic, including its properties, use Topic|Cut instead.

Text|Copy

Use the Copy command from the Text menu to copy the part of the topic text you have selected to the clipboard. You can also copy text by pressing Ctrl+Ins on the keyboard or by right-clicking on the topic text and picking Copy from the context menu. If you want to cut an entire topic, including its properties, use Topic|Copy instead.

Text|Paste

Use the Paste command from the Text menu to insert the text on the clipboard in the topic text of the current topic. If you have selected a part of the topic text before issuing the paste command, it will be replaced with the text on the clipboard. You can also paste text by pressing Shift+Ins on the keyboard or by right-clicking on the topic text and picking Paste from the context menu. The text is pasted with its original formatting it had in the application it was copied from. If you want to paste just the text, without any formatting, use the Text|Paste as Plain Text menu item instead.

You can paste text copied to the clipboard with Text|Cut or Text|Copy, as well as text copied to the clipboard by another program such as a text editor or word processor. To copy and paste a whole topic, including footnotes, use Topic|Copy and Topic|Paste instead.

Text|Paste as Plain Text

Use the Paste as Plain Text command from the Text menu to insert the text on the clipboard in the topic text of the current topic. If you have selected a part of the topic text before issuing the paste command, it will be replaced with the text on the clipboard. The pasted text will use the same formatting (font, color, style, etc.) as text you would type into the help topic would have. The formatting of the text on the clipboard is ignored. This can be useful if you want to copy and paste text into HelpScribble from a document that uses different text formatting than you want to use in your help file. To paste text retaining its original formatting, use Text|Paste instead.

Text|Delete

Select Delete from the Text menu, right-click the topic text and select Delete from the context menu, or press the Del key on the keyboard to delete the currently selected part of the topic text.

Text|Select All

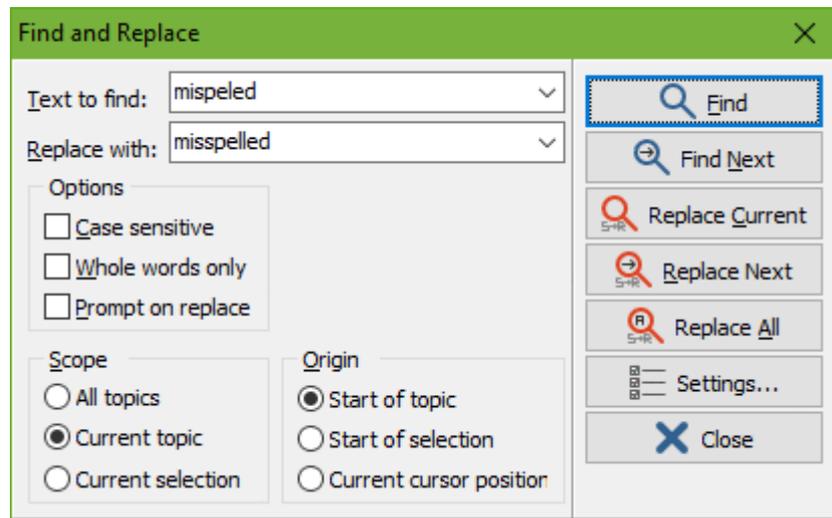
Pick Select All from the Text menu, or right-click the topic text and pick Select All from the context menu to select the entire topic text.

Text|Find and Replace

Select Find and Replace from the Text menu, or click the Find button in the HelpToolBar, to find and replace text in your HelpScribble project.

Type in the text to find in the “text to find” box. If you want to replace the found text, type in the replacement in the “replace with” box. You can also select pieces of text you recently searched for from the drop-down lists.

You can specify the following search options:



If case sensitive is checked, upper case and lower case characters are treated as being different. So searching for `hello` will not find `Hello`. If it is not checked, there is no difference and searching for `hello` will find `hello`, `Hello`, `HeLlO`, etc.

Check “whole words only” if you wish to search for entire words only. When this option is on, searching for “cat” will not find the first three letters of “category”. If it is on, those first three letters will be found, as well as the “cat” in “vacation”.

If this “prompt on replace” is checked and you click on Replace Next or Replace All, HelpScribble will first highlight the text found and ask you if you wish to replace it before actually doing so.

The “scope” determines which topics will be searched through. You can search through all topics, only the current topic, or only the selected part of the topic text of the current topic.

The “origin” determines where the search begins. This option is not available when the scope is set to all topics. “Start of topic” starts the search at the beginning of the current topic. “Start of selection” starts the search at the beginning of the selected part of the current topic. “Cursor position” starts the search at the position of the text cursor in the current topic.

Click the Find button to locate and highlight the first occurrence of the text to find using the given scope and origin. Click the Find Next button to find the next occurrence of the text to find in the scope set when you last clicked the find button.

Click the Replace Current button to replace the currently selected occurrence of the text to find with the replace text. Click the Replace Next button to find the next occurrence of the text to find in the scope set when you last clicked the find button and replace it with the replace text. Click the Replace All button to replace all occurrences of the text to find in the selected scope.

By clicking the Settings button, you can change the appearance of the Find window. With “show options”, you can show or hide the search options. With “show scope”, you can show or hide the search scope and origin radio buttons.

Select “show button text” to show or hide the text on the buttons on the find and replace dialog. Buttons without text are much smaller and allow the toolbar to be dragged to the top or bottom of the find and replace dialog. By hiding the options and scope sections, hiding the button text, and dragging

the toolbar to the top or bottom, you can make the Find window a lot smaller. That way, it does not obscure HelpScribble itself too much when you make it stay on top.

When “stay on top” is checked, the find and replace dialog stays on top of all other windows, whether it is active or not.

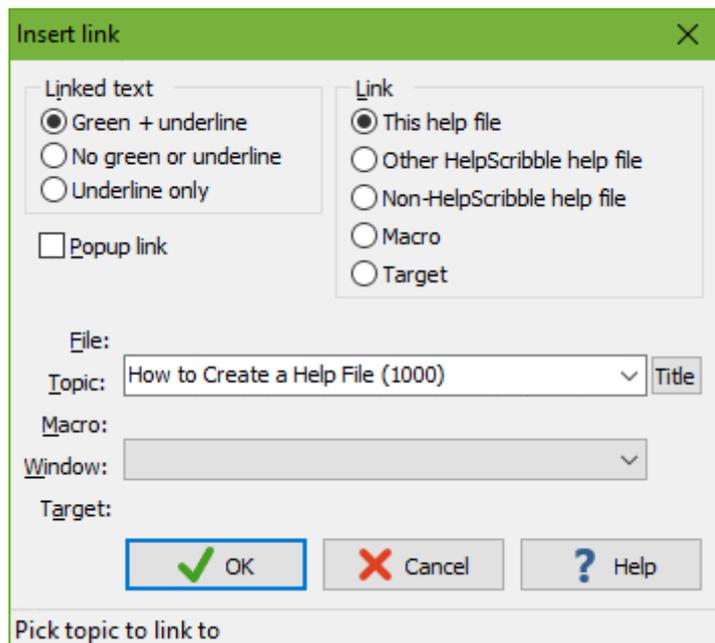
HelpTools|Link

By linking topics, people reading your help file can easily navigate from one topic to a related topic. To add a link to a topic, select the part of the topic text that you want to make clickable. Then select Link from the HelpTools menu, or click the Special Link button in the HelpToolBar. You will then be asked to specify the options for the link (see below).

If you simply want to create a link from one topic to another, without specifying any options, you can do so faster using the context menu of the Topic Grid or by using the HelpToolBar.

If you are compiling into WinHelp format, you can select if the link should be underlined and green as links usually are in WinHelp, or not. This option has no effect when compiling to HTML Help.

For HTML Help, you can specify the colors in Project Options: HTML Help.



Mark the “popup link” option if the topic linked to should be displayed in a popup window on top of the current topic, rather than replacing the current topic.

There are five types of links that you can create:

1. Link to this help file. Simply select the topic you want to link to from the drop-down list.
2. Link to another HelpScribble help file. Select the .hlp or .chm file to link to, and type in the Topic ID of the topic to link to. The link will only work reliably if both the origin help file and the target help file use the same format (.hlp or .chm). For .chm files, the link needs to specify a full absolute path if the target .chm file is not in the same folder as the origin .chm file.
3. Link to a WinHelp .hlp file created with another tool. Select the .hlp file to link to, and type in the context string of the topic to link to. If you want to link to external documentation in HTML format, use HelpTools|Internet Link instead.
4. WinHelp macro link. When the link is clicked, the WinHelp macro you construct will be executed. These links do not work with HTML Help, except for ExecFile and ExecProgram macros, which HelpScribble converts into HTML links. You can use the ExecFile macro to open any file in the default program for handling such files. If you are distributing an Excel file with your help file, use ExecFile(“spreadsheet.xls”) to start Excel and open spreadsheet.xls.
5. Link to a target created with HelpTools|Create Target. The Target drop-down list allows you to select any target in your help project.

If you are compiling into WinHelp format, you can also select the secondary window, if any, that the topic should be displayed in. You can define these windows with the Windows Editor.

HelpTools|Create Target

Targets allow you to create links to the middle of a topic, rather than the start of a topic.

To create a target, select Create Target from the HelpTools menu or click on the Create Target button in the HelpToolBar. Type in the name of the target, and click OK. Note that all the targets in your project must have unique names.

To link to a target, use HelpTools|Link. Select “target” as the type of link, and then type in the name of the target.

HelpTools|Keepn

Select Keepn from the HelpTools menu or click the Keepn button on the HelpToolBar to insert a keepn tag into the current topic. This tag will divide the topic into a non-scrolling region (the part above the keepn), and a scrolling region (the part below the keepn). The effect is that if the topic is too long to show all text at once, and the user scrolls down the topic text, the non-scrolling region will remain visible. Usually, the topic title is placed into a non-scrolling region.

Note that non-scrolling regions are only available in WinHelp format, and not in HTML Help or Web Help. When using these formats, the keepn tags will simply be removed by HelpScribble.

Also, in WinHelp, topics displayed in popup windows cannot have a scrolling region. Therefore, if you add a keepn tag to a topic and then create a popup link to it, only the part of the topic before the keepn will be visible. So do not put a keepn tag in topics that you intend to create popup links to.

HelpTools|Insert Bitmap

To insert an image into your help file, select Insert Bitmap from the HelpTools menu, or click the Insert Bitmap button on the HelpToolBar. Select the image file you want to insert. If you want to insert a screen shot, press the Print Screen key on the keyboard to capture the entire screen, or Alt+Print Screen to capture the active window. Then use HelpScribble’s SHG Editor to save the screen shot into a file, which you can then insert with HelpTools|Insert Bitmap.

After selecting the file, you will be asked if the image should be character-aligned with no text flowing left or right, left-aligned with text flowing around the right, or right-aligned with text flowing around the left.

The transparency option only has an effect when compiling into WinHelp format. For .bmp files, it will make all pure white (RGB 255,255,255) pixels in the bitmap transparent. If you want to have white pixels in a transparent image, make them almost white (e.g. RGB 250,250,250). For .gif files, it will make all pixels that have the color of the .gif palette index 15 (counting from zero). WinHelp ignores the transparency setting saved into the .gif file.

If you want to have transparent bitmaps in HTML Help files, use a graphics program to create a transparent .gif and insert that into your help project. The transparency option of HelpTools|Insert Bitmap does not affect the transparency of images when compiling to HTML Help.

Note that the image will not become visible in HelpScribble itself. All you will see is a textual tag with the file name of the image. This makes editing easier since the tag is a simple piece of text that you can edit, and allows the final help file to be better optimized. When compiling the help file, the help compiler will insert the image file into the compiled .hlp or .chm file. So you do not need to distribute the image files to the recipients of the .hlp or .chm file.

Which image file formats are supported depends on the help compiler that you are using:

16 bit WinHelp

BMP: Uncompressed 16, 256 and true color BMPs are supported.
WMF: Supported
SHG: Supported
GIF: Not supported
PNG: Not supported
JPEG: Not supported

32 bit WinHelp

BMP: Uncompressed 16, 256 and true color BMPs are supported.
WMF: Supported
SHG: Supported
GIF: Supported, but buggy. Some valid GIF files will appear as "unable to display graphic". In our tests, all GIFs saved by HelpScribble's SHG Editor worked fine.
PNG: Not supported
JPEG: Supported.

HTML Help

BMP: Not supported, but automatically converted to GIF or PNG by HelpScribble.
WMF: Not supported
SHG: Not supported, but automatically converted to GIF or PNG plus an HTML image map by HelpScribble.
GIF: Supported
PNG: Supported
JPEG: Supported

HelpTools|Insert Button

You can insert a button into the current topic by selecting Insert Button from the HelpTools menu, or by clicking the Insert Button button on the HelpToolBar.

There are two kinds of buttons. The simple kind simply displays another topic, just like a link does. This kind of button is supported by 32-bit WinHelp, HTML Help and Web Help. 16-bit WinHelp does not support buttons. To create this kind of button, type in the caption for the button, click on the "link to topic" radio button, and select the topic to link to from the drop-down list. If you have added targets to your help file, you can also create buttons that link to those targets. Select the "link to target" radio button and choose the target from the drop-down list.

The other kind is only supported by 32-bit WinHelp. This button executes a WinHelp macro. Type in the caption for the button, click on the "WinHelp macro" radio button, and type in the WinHelp macro. You can easily create the WinHelp macro by clicking on the (...) button to invoke HelpScribble's WinHelp Macro Editor.

Note: The 32-bit WinHelp help compiler cannot handle more than 50 buttons in a single topic.

HelpTools|Internet Link

Select Internet Link from the HelpTools menu, or click the Internet Link button on the HelpToolBar to create a link to an URL. Specify the hotspot text that will be inserted into the help text and be turned into a link, and specify the URL to link to. To open a web page in the user's default web browser, type in an URL such as <http://www.helpscribble.com/>. To open the user's default email program ready to send an email to a particular address, type in a mailto: URL such as <mailto:support@helpscribble.com>.

Internet links do not work in WinHelp files (.hlp files) on Windows Vista, 7, 8, and 8.1. Microsoft has disabled the ability to link to URLs in the WinHelp viewers for Windows Vista, 7, 8, and 8.1. To create a link to another topic in your help file, use HelpTools|Link instead.

HelpTools|Link to File

Select Link to File from the HelpTools menu to create a link to an external file. This can be a Word or PDF document, or any other file. You will need to distribute that file along with the compiled .hlp or .chm file. The file must be placed into the same folder as the .hlp or .chm file, or it will not be found. When the reader of your help file clicks this link, the application that is associated with this kind of file on the readers's computer will be used to open the file.

When you compile into HTML Help format, the link will appear as a button rather than a textual link. This is due to a technical limitation of the .chm file format. Regular links can only reference files that are compiled into the .chm file, and can be viewed in Internet Explorer.

Links to files do not work in WinHelp files (.hlp files) on Windows Vista, 7, 8, and 8.1. Microsoft has disabled the ability to open external files in the WinHelp viewers for Windows Vista, 7, 8, and 8.1.

HelpTools|Embed Flash

Select Embed Flash from the HelpTools menu to embed a Flash movie into the help topic. When compiling to HTML Help, the .swf file that contains the Flash movie is compiled into the .chm help file. So you do not need to distribute the .swf file along with the .chm file. When exporting to Web Help format, the .swf file must be distributed along with the other image and HTML files.

Flash movies cannot be embedded into WinHelp files. The Embed Flash menu item will be disabled if you selected the WinHelp compiler in Project Options.

HelpTools|Insert RTF Codes

WinHelp files internally use RTF or Rich Text Format to store and display the text of your help topics. HelpScribble takes care of the task of creating the proper RTF, so you do not have to worry about this.

However, if you know how RTF works, and you want to insert some RTF tags that HelpScribble does not support into your help topics, you can do so by picking "Insert RTF Codes" from the HelpTools menu. HelpScribble will insert an rtf= tag into the help text. Type the RTF tag you want to insert after the equals sign. E.g. type the RTF tag \brdrb to have a line drawn below the paragraph.

When compiling or exporting to HTML Help or Web Help, the rtf= tags will be automatically filtered out by HelpScribble.

HelpTools|Insert HTML Tags

HTML Help and Web Help internally use HTML to store and display the text of your help topics. HelpScribble takes care of the task of creating the proper HTML, so you do not have to worry about this.

But if you are familiar with HTML and want to use some special formatting that HelpScribble does not support, you can insert any HTML code you want into your help topics. To do so, select Insert HTML Tags from the HelpTools menu. You can add whichever tags you want. HelpScribble will insert them unchanged into the HTML code for the topic.

If you compile your project into WinHelp, HelpScribble will automatically filter out the HTML tags you entered this way.

Go to topic|Next Topic

Select Next Topic from the Go to topic menu, or click the Next Topic button in the TopicBar to make the next topic the current one, so you can edit it. The order of the topics is determined by the Topic Grid. By default, topics are ordered from lowest Topic ID to highest Topic ID. You can change this by clicking on the column headers in the Topic Grid.

A faster way to make another topic the current one, is to double-click on it in the Topic Grid.

Go to topic|Previous Topic

Select Previous Topic from the Go to topic menu, or click the Previous Topic button in the TopicBar to make the previous topic the current one, so you can edit it. The order of the topics is determined by the Topic Grid. By default, topics are ordered from lowest Topic ID to highest Topic ID. You can change this by clicking on the column headers in the Topic Grid.

A faster way to make another topic the current one, is to double-click on it in the Topic Grid.

Go to topic|Next bookmark

Select Next Bookmark from the Go to topic menu, or click the Next Bookmark button in the TopicBar to make the next bookmarked topic the current one, so you can edit it. If there are no bookmarked topics following the current topic, the last topic will be activated. You can bookmark topics by right-clicking on them in the Topic Grid and selecting Bookmark from the context menu.

The order of the topics is determined by the Topic Grid. By default, topics are ordered from lowest Topic ID to highest Topic ID. You can change this by clicking on the column headers in the Topic Grid.

A faster way to make another topic the current one, is to double-click on it in the Topic Grid.

Go to topic|Previous Bookmark

Select Previous Bookmark from the Go to topic menu, or click the Previous Bookmark button in the TopicBar to make the previous bookmarked topic the current one, so you can edit it. If there are no bookmarked topics following the current topic, the last topic will be activated. You can bookmark topics by right-clicking on them in the Topic Grid and selecting Bookmark from the context menu.

The order of the topics is determined by the Topic Grid. By default, topics are ordered from lowest Topic ID to highest Topic ID. You can change this by clicking on the column headers in the Topic Grid.

A faster way to make another topic the current one, is to double-click on it in the Topic Grid.

Go to topic|History

Select History from the Go to topic menu, or right-click the Go to topic button in the TopicBar to see a list of topics you recently viewed or edited, ordered from most recent to least recent. Select one of the topics from the menu to activate it for viewing and editing.

Note that if you want to access the history from TopicBar, you need to click on the Go to topic button with the right mouse button, and not with the left button. Clicking the left button will activate the topic with the ID that you can type in immediately to the left of the Go to topic button on the TopicBar.

Print|Printer Setup

Select Printer Setup from the Print menu to configure your printer. You should do this before using any other command from the Print menu.

Print|Print Topic List

Select Print Topic List from the Print menu to print a list of all the topics in your help project. The printout will show the Topic ID, Title and Status of each topic. The topics will be ordered by ascending ID. Bookmarked topics will be indicated by a dotted line.

If you want to configure the printer, use Print|Printer Setup first.

You can also save a list of all topics to a file using Project|Export Topic Grid.

View|ProjectBar

You can show or hide ProjectBar using the View menu. When shown, the ProjectBar is the topmost toolbar in HelpScribble's main window.

On the ProjectBar, you will see many buttons that are shortcuts to various menu items. From left to right, the buttons will invoke the following menu items when clicked:

Project|New

Project|Open, Project|Save

Project|Check Spelling

View|TopicBar, View|PropertiesBar, View|TextBar, View|HelpToolBar, View|Ruler

View|Topic Grid, View|Error Log

Editors|Browse Sequence Editor, Editors|Contents Editor, Editors|Index Editor, Editors|SHG Editor, Editors|Windows Editor

Project|Options, Project|Make, Project|Show Help File

View|TopicBar

You can show or hide the TopicBar by using the View menu.

- From left to right, the following controls are available on the TopicBar:
- Spinner box for editing the Topic ID of the current topic.
- Drop-down list showing the Topic Title of the current topic, and keeping an alphabetically sorted list of all topics in your help file. To change the title of the current topic, type in a new title when the drop-down list is closed. To make another topic the current one, open the drop-down list and select the topic you want to edit.
- Button for Topic|New
- Button for Topic|Delete
- Button for Topic|Properties
- Button for Topic|Cut
- Button for Topic|Copy
- Button for Topic|Paste
- Button for Go to topic|Previous Bookmark
- Button for Go to topic|Previous Topic
- Button for Go to topic|Next Topic
- Button for Go to topic|Next bookmark
- Spinner box for typing in the Topic ID of the topic you want to view or edit next. Click on the button to the right of it to actually go to that topic.
- Button for use with the spinner box to the left of it. If you right-click this button, you will see a list of recently edited topics like Go to topic|History.

View|PropertiesBar

You can show or hide the PropertiesBar using the View menu. On this toolbar, you can edit the most common topic properties of the current topic. From left to right, you can edit:

- Keywords
- A Keywords
- Browse Sequence
- Build Tag
- Status

Two other important topic properties, namely the Topic ID and Topic Title, can be edited on the TopicBar. The remaining topic properties can be edited by selecting Properties from the Topic menu.

View|TextBar

You can show or hide the TextBar using the View menu. This toolbar allows you to change font and paragraph styles in the text of the current topic.

From left to right, you can:

- Change the current font by selecting it from the drop-down list
- Change the size of the font by typing in a new font size, or by clicking the up and down spinner buttons.
- Change the color of the current font by selecting it from the drop-down list
- Toggle boldface on and off
- Toggle italics on and off
- Toggle underlining on and off
- Toggle striking out on and off
- Align the current paragraph to the left margin
- Center the current paragraph between the left and right margins
- Align the current paragraph to the right margin
- Put a bullet in front of the current paragraph, or remove that bullet

The remaining controls at the right side of the TextBar allow you to work with TextMacros.

TextMacros

A TextMacro remembers font style and paragraph alignment. Using TextMacros makes it easier for you to give your help file a consistent lay-out. Do not confuse them with WinHelp macros.

The macro's that start with a * are system TextMacros. You cannot rename or delete them, though you can change the text style they hold. The system TextMacros are used when you add topics and when you run the Delphi source code parser.

You can use a TextMacro by selecting it from the TextBar and clicking on the Apply TextMacro button, the rightmost button in the TextBar. The TextBar is the toolbar with the font-style controls. Select

View|TextBar from the menu if it is hidden. You can also apply the currently selected TextMacro quickly by pressing Ctrl+M on the keyboard. If you only want to apply the font style saved in the TextMacro, and not the paragraph style, right-click the Apply TextMacro button instead of left-clicking it. Clicking the Apply TextMacro button will move keyboard focus to the topic text, so you can continue typing right away.

You can create your own TextMacros by clicking on the New TextMacro button on the TextBar. Type in a new name in the drop-down list and press the Change TextMacro button on the TextBar to rename the TextMacro. Clicking the Change TextMacro button will also assign the font style of the selected text to the TextMacro. To change an existing TextMacro, select it from the drop-down list, select the part of the topic text that already has the desired style, and then click the Change TextMacro button.

Note: If you create a Project|New, it will initially have exactly the same TextMacros as the project you were editing just before you created the new project. So if you want to create two projects that have the same look, open the first project when you create the second to make it “inherit” the TextMacros. If you change the TextMacros in the new project, the previous project will *not* be updated.

View|HelpToolBar

You can show or hide the HelpToolBar using the View menu.

The left part of this toolbar is occupied by a drop-down list containing all the topics in your help file. You can sort this list by Topic ID or Topic Title by clicking on the Title/ID button to the right of it. When you select a topic from this list, the two graphical buttons to the right of it will become enabled. The first button will create a link in the current topic, to the topic you selected from the drop-down list. The second button creates a popup link. A regular link will replace the current topic with the topic linked to, when clicked by the user. A popup link will show the topic linked to in a floating popup window. If you selected a part of the topic text before clicking these buttons, that part will be turned into a link pointing. If you did not select a part of the topic text, the buttons will insert the Topic Title of the topic linked to and use that to create the link.

The next button on the HelpToolBar is HelpTools|Link. This button, like corresponding menu item, will not be enabled unless you select a part the topic text. This text will be used to create the link.

The remaining buttons are shortcuts for, from left to right:

- HelpTools|Internet Link
- HelpTools|Create Target
- HelpTools|Keepn
- HelpTools|Insert Bitmap
- HelpTools|Insert Button
- Text|Find and Replace

View|Ruler

The ruler in the HelpScribble main window works like the rulers in many popular word processors. You can show or hide the ruler using the View menu.

The ruler shows a measurement in centimeters by default. If you prefer inches, right-click on the ruler and pick Inches from the context menu. Note that these measurements are relative to your display settings and may not be 100% exact depending on the size of your monitor and the screen resolution you are using.

The ruler has three small square buttons and possibly a few L-shaped symbols.

The top square determines how much the first line of a paragraph is indented. The left bottom square determines the indent of the other lines of the paragraph. The right bottom square places the right margin.

The L-shaped symbols are the tab positions. If you do not place tabs, inserting tabs in the text (by pressing the TAB button on the keyboard) will still work, but the indents will be the default ones.

- To place a tab: click anywhere on the ruler space vertically between the square buttons.
- To move a tab: click on it, hold the mouse button down and move the mouse.
- To delete a tab: move it all the way to the left or the right.

Note that HTML does not support tabs in the way WinHelp does. If you are compiling for HTML Help, you should not use tabs in your help text. Tabs will be converted to spaces, and the layout will not be what you intended. If you want to create special layouts in HTML Help, select a fixed-width font such as Courier New, and arrange the text using spaces.

View|Topic Grid

Select Topic Grid from the View menu to show or hide the Topic Grid.

View|Error Log

Select Error Log from the View menu to show or hide the output produced by the help compiler when you last used Project|Make. The error log is shown automatically after using Project|Make.

If the help compiler found any problems when compiling your help file, they will be noted in the error log. The WinHelp compiler will report the topic as "topic #123". The WinHelp compiler counts the topics 1, 2, 3, ... sorted by Topic ID, starting with the topic with the lowest Topic ID. To quickly locate the offending topic, type the number reported by the WinHelp compiler in the small box to the right of the error log. Then click the "Go to topic" button below the small box.

The HTML Help compiler reports topics as "hs123.htm" where 123 is the Topic ID of the topic. To quickly locate the topic, type the Topic ID in the rightmost spinner box on the TopicBar, and then click the Go to topic button next to it on the TopicBar.

Help|Download Spell Check Dictionaries

Before you can use Project|Check Spelling or Topic|Check Spelling, you need to download and install spell check dictionaries for one or more languages. You can easily do so right within HelpScribble. Just pick Help|Download Spell Check Dictionaries from the menu. HelpScribble will then connect to the Internet to fetch a list of available languages.

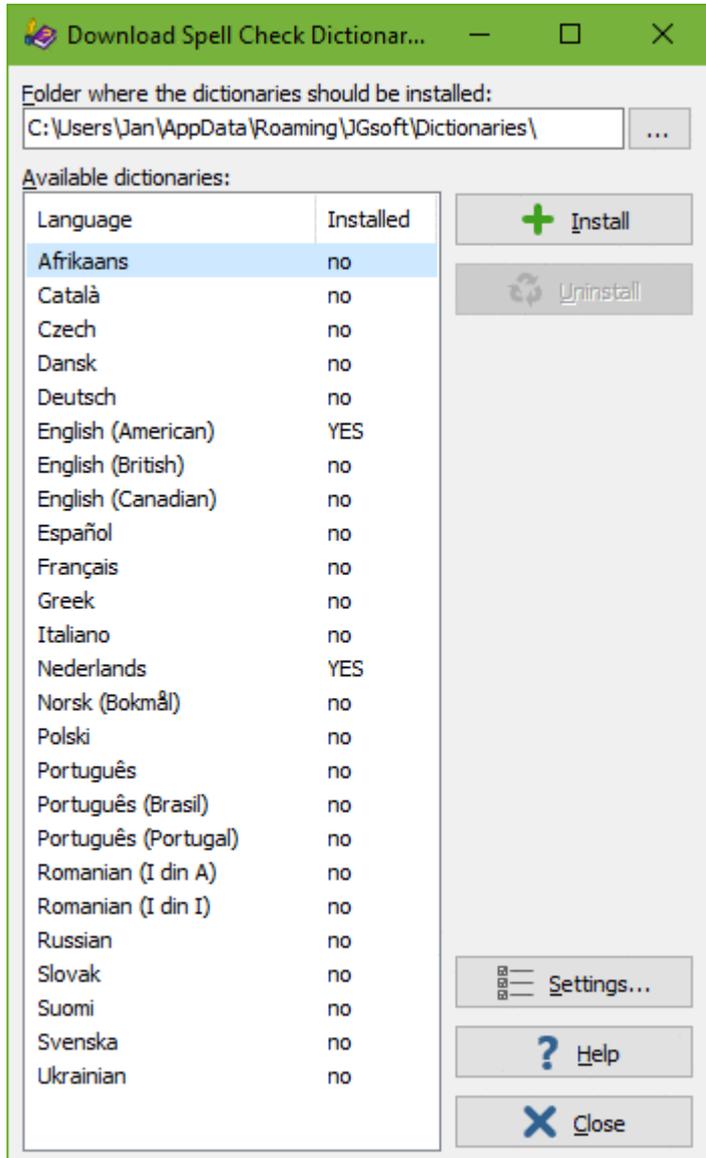
If you want, you can choose the folder into which the dictionaries should be installed. If you specify a folder that does not exist, HelpScribble will create it. All dictionaries must be installed into the same folder. If you already installed some dictionaries, and then install another dictionary into a different folder, HelpScribble will move the previously installed dictionaries to the new folder.

To install a dictionary, click on the language you want in the list, and click the Install button. HelpScribble will then automatically download the dictionary, and install it into the folder you specified. A progress meter will appear while downloading.

You can have as many dictionaries installed as you want. There is no need to uninstall dictionaries. However, if you want, you can select a language and click the Uninstall button to delete a spell check dictionary.

If you are behind a proxy server, and HelpScribble is unable to detect your proxy settings, you can change them by clicking the Settings button.

Due to firewall and other network settings, it is possible that HelpScribble cannot connect to the Internet directly. In that case, you can manually download and install the dictionaries from <http://www.helpscribble.com/spell.html>.



Topic Grid

By default, HelpScribble will show all the topic in your help file in a grid at the left hand side of its main window. You can hide or show this grid with View|Topic Grid. You can dock the grid to the right edge of HelpScribble's main window, or make it float, by dragging the double beveled horizontal line at the top of the Topic Grid. When the grid is floating, drag its caption bar to the left or right edge of HelpScribble's main window. It will snap into place when the mouse pointer is near the edge of the main window.

The grid can show the most important properties of each topic. The Topic ID is always visible. The other properties can be shown or hidden by clicking the buttons at the top of the topic grid. The buttons will show, from left to right: Topic Title, Keywords, Build Tag, A Keywords, Browse Sequence, Status. All properties visible in the grid, except for the Topic ID, can be edited directly in the Topic Grid. Click once on the item you want to edit, and type in the new value. You can edit a topic's properties in the grid, whether that topic is the current topic or not. The build tag and status properties can be changed quickly by right-clicking on them and selecting a new value from the context menu.

You can change the order in which topics are listed in the grid by clicking on the grid's column headers. Click once on a column header to sort by that column in ascending order. Click again to sort by descending order. You can resize columns by using the mouse to drag the vertical bevel that separates the column headers.

Right-click anywhere on the grid to show the context menu. The commands in this menu are explained in the following sections.

Topic Grid|Edit Topic

Select Edit Topic from the context menu of the Topic Grid to make the topic that you right-clicked on in the grid the current one. This allows you to edit that topic. You can achieve the same by double-clicking on the topic in the Topic Grid.

Topic Grid|Insert Topic

Select Insert Topic from the context menu of the Topic Grid to insert a new topic in your help project, after the topic on which you right-clicked in the grid.

Topic Grid|Delete Topic

Select Delete Topic from the context menu of the Topic Grid to delete the topic on which you right-clicked in the grid.

Topic Grid|Properties

Select Properties from the context menu of the Topic Grid to edit all the properties of the topic on which you right-clicked in the grid. Note that all properties, except the Topic ID, visible in the grid can be edited directly by clicking once on them, and then typing in the new value.

Topic Grid|Cut

Select Cut from the context menu of the Topic Grid to put the topic on which you right-clicked in the grid on the clipboard, and delete the topic from your project. Note that this will cut the entire topic. If you only want to cut a piece of text, use Text|Cut instead.

Topic Grid|Copy

Select Copy from the context menu of the Topic Grid to put the topic on which you right-clicked in the grid on the clipboard. Note that this will copy the entire topic. If you only want to copy a piece of text, use Text|Copy instead.

Topic Grid|Paste

Select Paste from the context menu of the Topic Grid to replace the topic on which you right-clicked in the grid with a topic that you previously cut or copied to the clipboard. The topic you right-clicked on will get the text and all the properties of the topic on the clipboard, except for the Topic ID, which will remain unchanged.

If you want to paste a piece of text into the current topic, use Text|Paste instead.

Topic Grid|Create Link

Select Create Link from the context menu of the Topic Grid to create a link from the current topic to the topic you right-clicked on in the grid. If you selected part of the text of the current topic, that text will be used to create the link. If not, the title of the topic you right-clicked on will be inserted into the topic text of the current topic, and the title will be used to create the link.

Topic Grid|Create Popup Link

Select Create Popup Link from the context menu of the Topic Grid to create a popup link from the current topic to the topic you right-clicked on in the grid. If you selected part of the text of the current topic, that text will be used to create the link. If not, the title of the topic you right-clicked on will be inserted into the topic text of the current topic, and the title will be used to create the link.

The difference between a regular link and a popup link is that when a regular link is clicked, the topic the link points to will replace the topic the user is currently viewing. A popup link will cause the linked topic to be shown in a window floating on top of the current topic.

Topic Grid|Bookmark

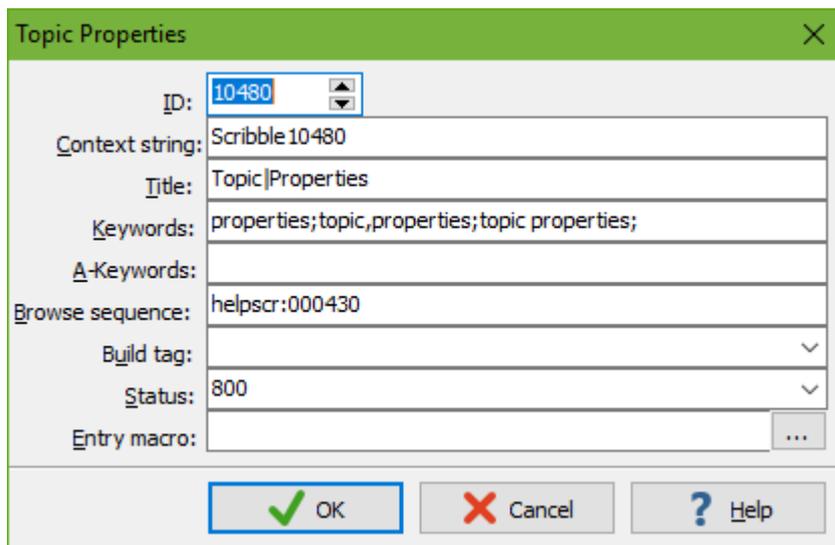
Select Bookmark from the context menu of the Topic Grid to bookmark the topic you right-clicked on in the grid. Bookmarked topics are shown in red in the Topic Grid. If your project has many topics, you should bookmark the first topic of each section to make it easier to select topics from the grid. You can quickly jump to bookmarked topics using Go to topic|Next bookmark and Go to topic|Previous Bookmark.

Topic Properties

A HelpScribble project consists of a series of topics. Each topic holds one page of text in the help file, plus a series of properties that determine how the topic is included in the help file. Some of these properties, such as the title and keywords, are very basic and you will use them for each topic. Other properties are very specialized, and you may never use them.

Basic properties: ID, Title, Keywords, Status

Special properties: A Keywords, Browse Sequence, Build Tag, Entry Macro, Context String



ID:	10480
Context string:	Scribble 10480
Title:	Topic Properties
Keywords:	properties;topic,properties;topic properties;
A-Keywords:	
Browse sequence:	helpscr:000430
Build tag:	
Status:	800
Entry macro:	

Topic Properties: ID

Each topic must have a Topic ID that is unique in your help project. It must be an integer number equal to or greater than one. By default, HelpScribble will give the first topic an ID of 10, and number the following ones 20, 30, 40, etc... You can change the gap in Project Options: General. You can change the ID of the current topic in the TopicBar and in Topic|Properties. You can change the IDs of a range of topics by using Project|Renummer Topics.

When you create a link from one topic to another, HelpScribble will insert a linkID tag into the topic text. The tag references the Topic ID of the target topic. If you change the ID of the topic linked to, the links will be automatically updated.

If you want to use your help file for context-sensitive help, you will most likely need to reference the Topic ID. Check your development tool's documentation. If it talks about a context number or mapping number for a topic, this is the Topic ID in HelpScribble. In Borland Delphi or C++Builder, you need to assign the Topic ID to the corresponding HelpContext properties. In Visual Basic, you will assign the Topic IDs to HelpContextId properties. A few tools expect a string or textual ID. That is the context string in HelpScribble.

Topic Properties: Title

If you change the title of the topic in the View|TopicBar, HelpScribble will automatically update the first line of the topic text to keep it in sync with the new topic title. However, the topic title can differ from what's in the topic text if you want to. When the user looks up a keyword in the index of your help file, and there is more than one topic associated with that keyword, the user will need to choose a topic based on their titles. So you should assign meaningful titles to each topic. In most cases, you will want to use the same text for the topic title and the first line of the topic text as we did in HelpScribble's own help file.

Topic Properties: Keywords

The Index of your help file is built from the keywords that you assign to each topic. There is no limit to the number of keywords you assign to a topic. You can assign the same keyword to as many topics as you want. You will probably want to assign meaningful keywords to all topics in your help project, so the user can easily locate the topics of interest in the index of your help file.

You can edit the keywords directly in the PropertiesBar and in the Topic Grid. Simply type in the keywords, separated by semicolons. You can also specify sub-keywords below a keyword. Separate the main keyword and the sub-keyword by a comma. You cannot create sub-keywords below sub-keywords.

For example, this topic has three keywords. Two of the keywords are a sub-keyword below another keyword: `topic properties,keywords;properties,keywords;keywords`

Another way to assign keywords to a topic is to use the Index Editor. When you open the Index Editor, it will read all the keywords assigned to all topics. When you confirm the changes you made in the Index Editor, it will update all the keywords for all topics. So you can build and maintain the index of your help file by editing the keywords of a topic directly, by using the Index Editor and by using both methods together. I prefer to assign keywords directly when I add new topics and to use the Index Editor to fine-tune the final index.

Topic Properties: Status

Each topic has a status property. You can edit it in the PropertiesBar and in the Topic Grid. You can put whichever text you want into the status properties. You could specify values like "writing", "testing" and "done" to mark which topics are finished, and which require attention. By sorting the Topic Grid by the status properties, you can easily see the topics you still have to work on.

You can also specify version numbers like "1.0" and "2.0". That is useful when somebody else will be proofreading version 2.0 of your help file, and they don't want to waste time rechecking what you did not change. Status properties are also useful in combination with Project|Merge. You could, say, merge version 2.0 topics into a translated version of the 1.0 help file, so the translator only has to translate modified topics.

Topic Properties: A Keywords

A Keywords are a special kind of keywords that are only used by the 32-bit WinHelp compiler. They are useful to do lookups with the ALink WinHelp Macro.

You can edit the A Keywords directly in the PropertiesBar and in the Topic Grid. To specify more than one A Keyword for a topic, separate them with semicolons.

You can then call the ALink WinHelp Macro and specify an A Keyword as its parameter. WinHelp will then display a list of all topics that have this keyword in their A Keywords property and the user can then select which topic should be displayed. If there is only one such topic, it will be displayed immediately.

The most useful place to implement an ALink macro is in a “see also” or “related topics” button or link. To create a button, use HelpTools|Insert Button and type in `ALink("keyword")` as the macro. To create a link, use HelpTools|Link, select the “macro” radio button, and type in `ALink("keyword")` as the macro.

Topic Properties: Browse Sequence

WinHelp can display <<< and >>> browse buttons that allow the user to browse through the topics in your help file. The order in which the topics are browsed through is determined by the value of the browse sequence properties of each topic.

In HelpScribble, you can easily create browse sequences using the Browse Sequence Editor. It will assign appropriate values to the browse sequence properties in your help project. However, you can also edit the properties manually in the Topic Grid and in the PropertiesBar.

A browse sequence property must consist of a browse sequence name (letters a-z and underscores only), a colon, and a number. Since the sorting method used by WinHelp is alphanumerical and not numerical, you must make sure that you always use the same amount of digits for the number and pad with zeros as needed.

You can use more than one browse sequence in your project, but each topic can be part of only one sequence. By clicking the <<< and >>> buttons, the user can browse through all topics in the same browse sequence as the current topic. If you leave this property blank, the topic won't be included in any browse sequence. In that case, the browse buttons will be disabled in WinHelp.

Topic Properties: Build Tag

You can create different compiled help files from a single HelpScribble project by using build tags. If you are writing a help file for an application that exists in Lite, Standard and Pro versions, you could create three help files from a single HelpScribble project. The HelpScribble project would document all features. Using build tags, the compiled help file would only include features relevant to Lite, Standard or Pro.

You can specify the build tag for a topic in the PropertiesBar, by selecting Topic|Properties from the menu, or by right-clicking on the Topic Grid and selecting Properties from the context menu. A build tag should consist of the letters a-z. You can specify more than one build tag for a topic by separating them by a semicolon. If a topic only applies to the Pro version, you can specify "pro" as the build tag. If it applies to both Standard and Pro, specify "standard;pro". If a topic should always be included, leave its build tag blank.

You can also use build tags to create different versions of the same topic. If you want to put a list of features in the help file, that list will be different for Standard, Lite and Pro. What you should do is create three topics in HelpScribble, and set their build tags so that at most one of the three topics is included in the compiled help file. Use Topic|Properties to give all three topics the same context string. If you then create a link to one of the three topics, that link will point to the topic that is actually included in the help file, even if it is not the one you linked to. The reason is that internally, the help system uses the context string to identify topics. That is also why context strings must be unique in the compiled help file. But you can have more than one topic with the same context string in HelpScribble, as long as you use build tags to exclude all but one of them from the compiled help file.

Once you have specified build tags for one or more topics, you will be able to select which help topics should be included in the compiled help file in Project Options: Compiler. If you select all build tags, or none of the build tags, all topics will be included. If you select some but not all build tags, only topics that have this build tag in their build tag property and topics for which you left the build tag property blank, will be included in the compiled help file.

Topic Properties: Entry Macro

When compiling to WinHelp format, the Entry Macro of a topic is executed each time that topic is displayed. You can edit the macro by selecting Topic|Properties from the menu. Click the (...) button to invoke the WinHelp Macro Editor.

Topic Properties: Context String

Each topic must have a context string that is unique throughout your help project. Context strings are used to internally reference help topics in .hlp files. HelpScribble also bases the file names of help topics inside .chm files on the context string.

Most likely, you will not need to worry about the context string of a topic. Therefore, HelpScribble will automatically give each topic a context string of "Scribble1234", where 1234 is the Topic ID of the topic. Since Topic IDs must be unique too, the context string is guaranteed to be unique this way. If you change the Topic ID, HelpScribble will automatically keep the context string in sync. If you prefer to specify your own context strings, you can do so by selecting Topic|Properties from the menu. If you change the context string, HelpScribble will not update it when you change the Topic ID.

One development tool that uses the context string is Clarion. To specify the topic with ID 1234 in Clarion, you will need to use ~Scribble1234 (a tilde followed by the context string).

Project Options

Select Options from the Project menu to edit project-wide settings.

The Project Options screen consists of four pages: General, Compiler, WinHelp and HTML Help

The General page lists very basic options which you should specify. You must select a help compiler on the Compiler page before you can compile your help project. You can specify advanced options on the WinHelp page if you selected the WinHelp compiler. If you selected the HTML Help compiler, you should configure the compiled help file on the HTML Help page.

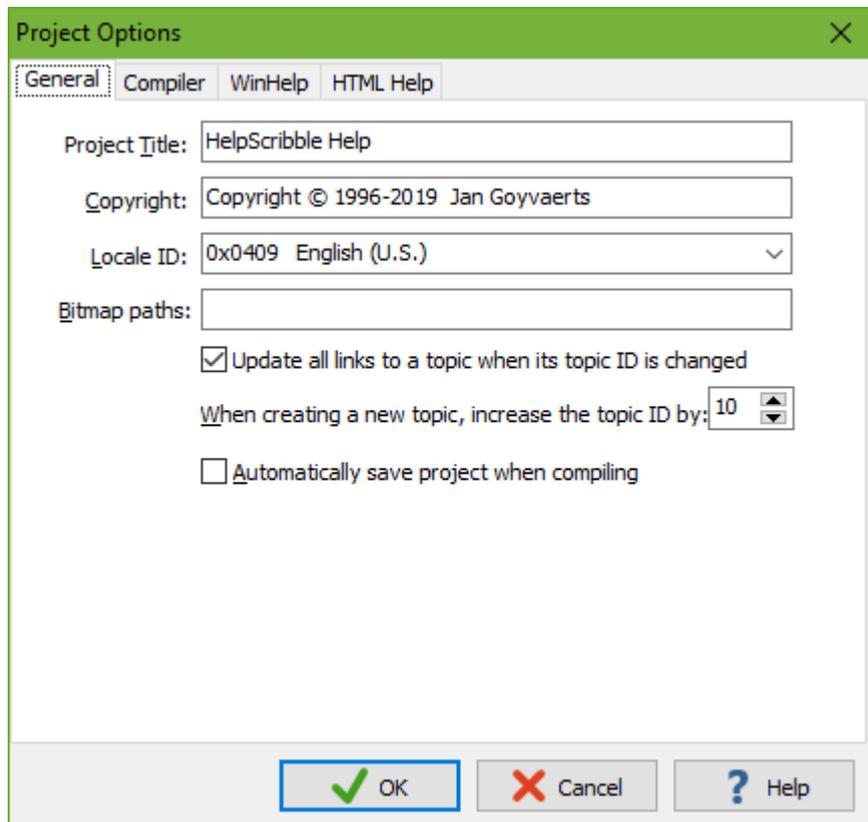
Project Options: General

Select Options from the Project menu to edit project-wide settings.

The Project Title is displayed in the caption bar of the help window when the user is reading your help file.

The Copyright is displayed in the about box of WinHelp. It is not displayed in HTML Help.

In the Locale ID, specify the language that your help file is written in. This will make sure that the help file uses the correct character set, so it is properly displayed on other people's computers (which may have different default character sets than your computer).



In Bitmap Paths, specify all the folders that contain images included in your help file, delimited by commas. HelpScribble will automatically add folders from which you add images using HelpTools|Insert Bitmap. If you move images to different folders, or add image tags manually, you will need to update Bitmap Paths. The only folder that may be omitted from bitmap paths is the folder containing the .hsc project file. HelpScribble always searches that folder for images.

You can specify the folders using absolute paths like c:\HelpProject\Extra\Images or as relative paths. You can only use relative paths if the folder containing the images is a subfolder of the folder containing the .hsc project file. If you saved the project into c:\HelpProject, then you can use Extra\Images instead of c:\HelpProject\Extra\Images in Bitmap Paths.

Mark "update all links to a topic when its Topic ID is changed" to make sure that all links are automatically updated when you change the Topic ID of a topic.

When you create a new topic, its Topic ID will be the ID of the current topic incremented by a certain value. The default is 10, but you can change it in Project Options: General.

The first time you use Project|Make, HelpScribble will ask you to save the project. This is because it must have a file name. When you use Project|Make again, HelpScribble will not automatically save the project unless you marked that option in Project Options: General.

Project Options: Compiler

Select Options from the Project menu to edit project-wide settings.

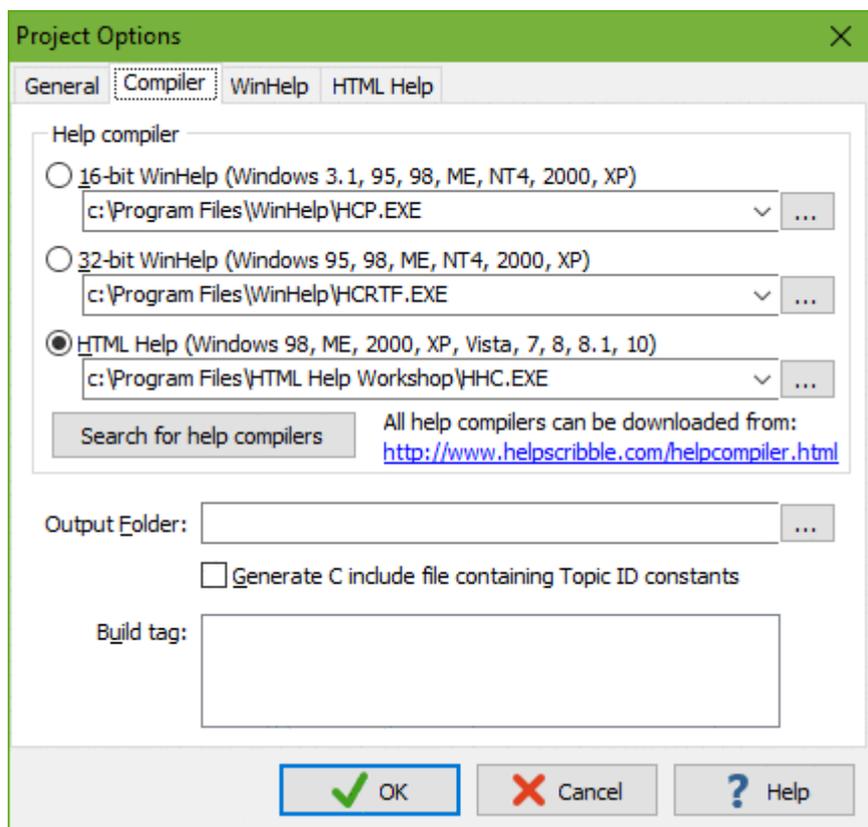
Before you can use Project|Make to create a distributable help file from your HelpScribble project, you need to select which help compiler you want to use. The WinHelp .hlp and HTML Help .chm file formats are designed by Microsoft. Since Microsoft did not publish the file formats, HelpScribble has to rely on a Microsoft help compiler to create a compiled .hlp or .chm file from your help project.

If you have any major development tools like those from Microsoft or Borland installed on your computer, you probably already have a help compiler. Click the

“Search for help compilers” button, and HelpScribble will find out for you. If you do not have a help compiler yet, you can download the Microsoft help compilers for free from our web site at <http://www.helpscribble.com/helpcompiler.html>.

There are three different kinds of help compilers. HelpScribble can use them all. Which of them you should use depends on your and your customers’ expectations of the final help files. Each help format has its advantages and disadvantages. See [How to Choose Between WinHelp and HTML Help](#) for more information.

The 16-bit WinHelp compiler will generate help files that work with Windows 3.1, 95, 98, ME, NT4, and XP. You should only use this option when maintaining an old system that still needs to support Windows 3.1. There are two versions of this compiler: HCP.EXE and HC31.EXE. Both are old DOS programs, but the HC31.EXE is limited to the base 640K memory, so it will run out of memory quickly when you include images in your help file. HCP.EXE does not have this constraint and works just fine, even under Windows XP.



The 32-bit WinHelp compiler will generate help files that work with Windows 95, 98, ME, NT4, and XP. 32-bit WinHelp files also work with Windows Vista, 7, 8, and 8.1 if the user downloads and installs the WinHelp viewer from Microsoft. Windows 10 does not support WinHelp. The help compiler is HCRTF.EXE, a 32-bit console application which requires HWDLL.DLL (included with the download mentioned above). You will get an error trying if you try to run HCRTF manually, but HelpScribble knows the parameters it wants (-x helpfile.hpj).

The HTML Help compiler generates .chm files in Microsoft's HTML Help format. This format was introduced with Windows 98 and is also supported by Windows ME, 2000, XP, Vista, 7, 8, 8.1, 10, and 11. This format is recommended as it provides the best compatibility with the latest versions of Windows. The help compiler is HHC.EXE, a 32-bit console application that is included with Microsoft's HTML Help workshop. Note that HTML Help files cannot be used to provide online documentation on a web site. For that, use Project|Export to Web Help.

If you want to support both WinHelp and HTML Help, simply change the compiler selection before using Project|Make. With HelpScribble, you can create WinHelp and HTML Help files from the same HelpScribble project. Depending on the compiler you chose, you can specify additional options on the WinHelp and HTML Help pages in Project Options.

The Output Folder is the folder where HelpScribble will put the compiled help file. If you leave this blank, HelpScribble will put it in the folder where you saved the .hsc project file. The WinHelp compilers generate a .hlp file. If you used the Contents Editor and are compiling a 32bit WinHelp file, the table of contents will be written to a separate .cnt file that you should distribute along with the .hlp file. The HTML Help compiler will generate a .chm file.

If the help file you are writing is the online help system for an application written in C or C++, you may want to check this option to make HelpScribble generate an include file that contains a constant defining the Topic IDs for all topics. Each constant's name is the context string of the topic and the value is the topic ID. To change a topic's context string, pick Topic|Properties from the menu or right-click in the Topic Grid and pick Properties from the context menu.

If you have specified Build Tag properties for certain topics, you will be able to select which help topics should be included in the compiled help file. If you select all build tags, or none of the build tags, all topics will be included. If you select some but not all build tags, only topics that have this build tag in their build tag property and topics for which you left the build tag property blank, will be included in the compiled help file.

Project Options: WinHelp

Select Options from the Project menu to edit project-wide settings.

For “kind of help file”, you will almost always select “regular help file”. The other options are for creating help files documenting VCL or CLX components that can be integrated with the Delphi or C++Builder help systems. For applications developed in Delphi or C++Builder, select “regular help file”.

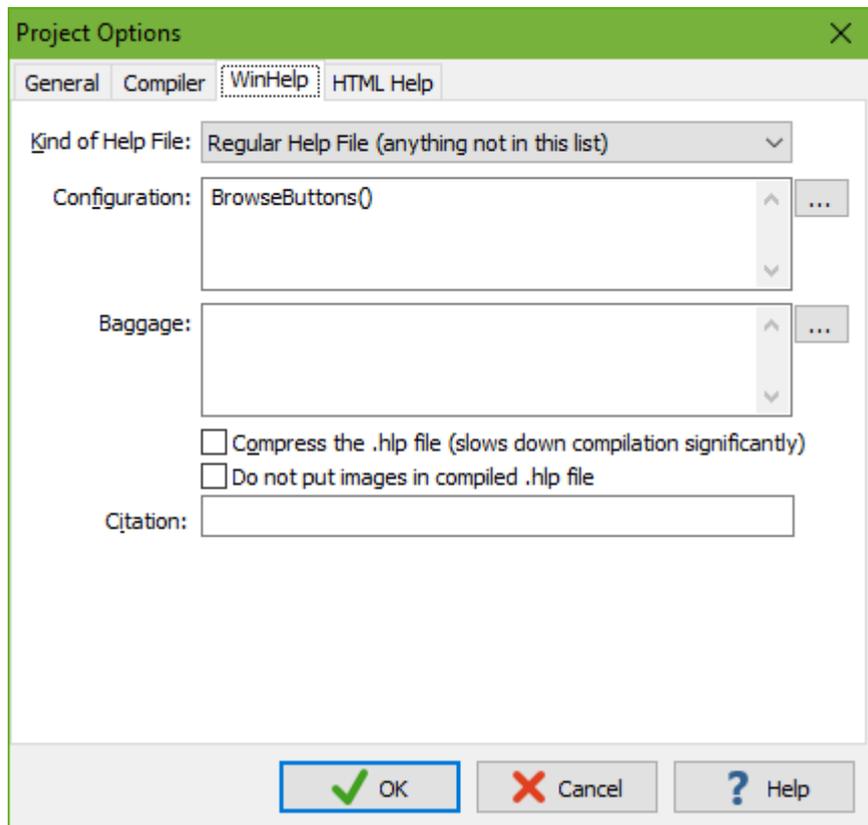
In the Configuration section, you can specify WinHelp macros that should be run when the .hlp file is opened. Click the (...) button to open the WinHelp Macro Editor. By default, HelpScribble will put the BrowseButtons() macro in the Configuration section. This macro makes sure that the <<< and >>> browse buttons appear in WinHelp.

In the Baggage section, you can specify files to be copied into the .hlp file. Certain advanced usages of WinHelp macros may need this.

The help compiler can compress the .hlp file. This will make the help file quite a bit smaller, but it increases the compile time by several orders of magnitude, certainly if your help file contains many images. Also, tools like WinZIP or installers like DeployMaster are able to compress an uncompressed .hlp file better than a compressed .hlp file. So you should not ask the compiler to compress your .hlp file unless you are compiling the final .hlp file and you will not be distributing it inside a .zip archive or a self-extracting setup.

If you select “do not put images in the compiled .hlp file”, the image tags will be visible in the compiled help file. This can be useful if your help file references many missing images (because the graphics artist did not deliver them yet), and you do not want the “cracked walls” to appear while testing the compiled help file.

If you specify a citation, it is appended to the end of any information (except in a context-sensitive pop-up window) that is copied or printed from the Help file. This option is commonly used to attach a copyright notice to any information in your Help file that is copied or printed. During our tests, we found that WinHelp does not always append the citation as it should do according to Microsoft's documentation.



Project Options: HTML Help

Select Options from the Project menu to edit project-wide settings.

You can specify the colors used by the topics in HTML Help and Web Help. Note that the text color specified in Project Options is only used wherever the topic text has the color "auto".

If you mark "compile full-text search information", the compiled .chm file will contain a full-text index. The HTML Help viewer will display an additional tab labeled "Search". The reader can search for any word in the help file on the Search tab. Note that the Search tab is not a substitute for the Index tab. The Index tab contains the keyword index that you can build using HelpScribble's Index Editor. On the Index

tab, the reader can select a keyword from a list and see which topics are relevant to the keyword. On the Search tab, the reader has to type in the word to search for to get a list of all topics containing that word.

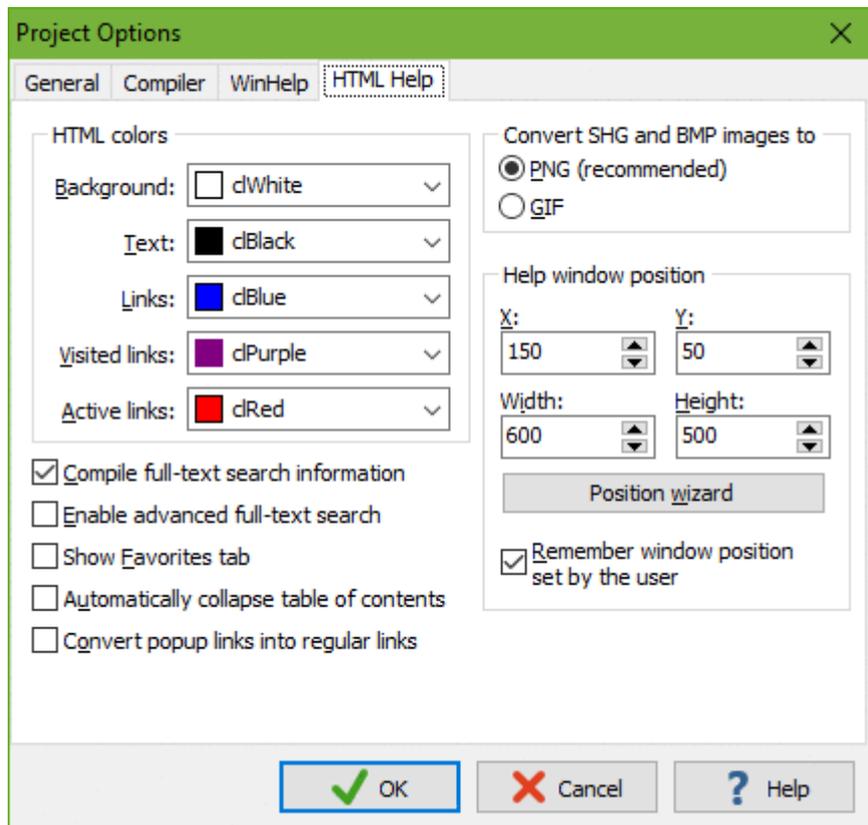
When compiling the full-text search information, you can also enable advanced search. This allows for more complex searches on the Search tab.

You can also make the HTML Help viewer show a "Favorites" tab. On this tab, the reader can keep a list of topics in your help file which he or she often refers to.

The table of contents in an HTML Help file can behave in two ways. By default, it behaves like a regular tree view, just like in WinHelp. The tree starts out as fully collapsed. "Book" nodes can be expanded and collapsed by clicking on the + and - buttons next to them. Nodes stay expanded until the user collapses them. This allows the reader to get a complete view of the table of contents.

If you turn on the option "automatically collapse table of contents", then "book" nodes can be expanded by clicking on their text labels. When expanding a node, all other nodes are automatically collapsed. This style requires fewer clicks to navigate through a large table of contents.

WinHelp allows you to create popup links to display topics in small, dynamic popup windows that disappear when the reader clicks anywhere. HTML Help does not support such popups. By default, HelpScribble will make popup links open the target topic into a small browser window. Such windows need to be clicked explicitly with the X button. Alternatively, you can turn on the option to convert popup links into regular links. Then popup links will open the target topic in the main help window, replacing the previous topic, just like regular links. This option only affects HTML Help compilation.



HTML Help does not support the BMP and SHG image file formats. Therefore, HelpScribble will convert them into GIF or PNG format. PNG is definitely recommended. PNG images are smaller than the equivalent GIF images. PNG images can contain any number of colors. GIF images are limited to 256 colors.

Click the Position Wizard button to set the default size and position of the HTML Help window. Positions are absolute. Make sure the window will fit on screen when the reader of your help file uses a low resolution monitor. It's best to keep the sum on X and Width under 800, and the sum of Y and Height under 600. Then your help file will appear nicely, even when people use a low resolution 800 by 600 screen. If you set X to 600 and Width to 600 to fit nicely on your 1280 by 1024 screen, then two thirds of the help screen will be invisible if your help file is viewed on a 800 by 600 screen.

Since you can specify only one default window size, it'll never be perfect for everybody's screen resolution. Turn on the option to remember the window position set by the user. The second time your help file is opened on the same computer, it will be displayed at the size and position it had when it was closed last time. You can turn this off if your help file will only be viewed in a setting with fixed screen resolutions. You can also turn it off to test the default help window position on your own computer. Recompiling your help file does not reset the window size and position the help viewer remembered for it.

Contents Editor

Select Contents Editor from the Editors menu to create a table of contents for your help file. Whether the table of contents is displayed in a popup window or side by side with the help text is determined by the format you are compiling the help file into. WinHelp uses the popup window, while HTML Help displays contents and text side by side. You can change the format in Project Options: Compiler. See “How to Choose Between WinHelp and HTML Help” for more information about this choice.

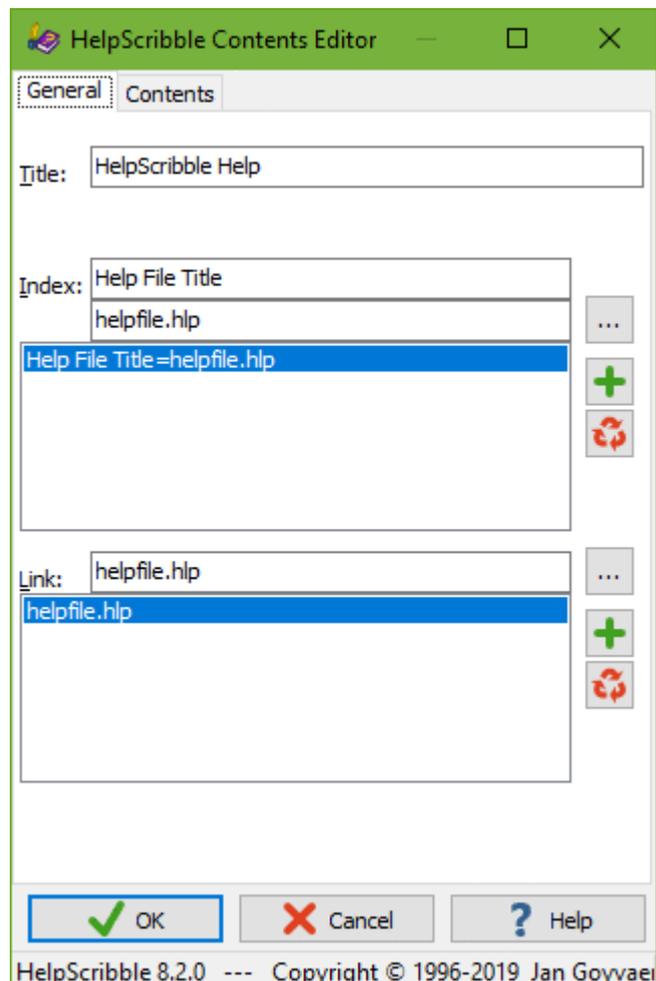
WinHelp stores the table of contents in a separate .cnt file. When moving or copying the .hlp file, make sure to move or copy the .cnt file along with it. HTML Help stores everything into a single .chm file.

Contents Editor: General

On the General page of the Contents Editor, you can specify that the keywords of other help files should be added to the index of this help file. This only works when using the 32-bit WinHelp compiler. This is useful for very large documentation projects where the complete documentation consists of several .hlp files which should appear as one to the end user.

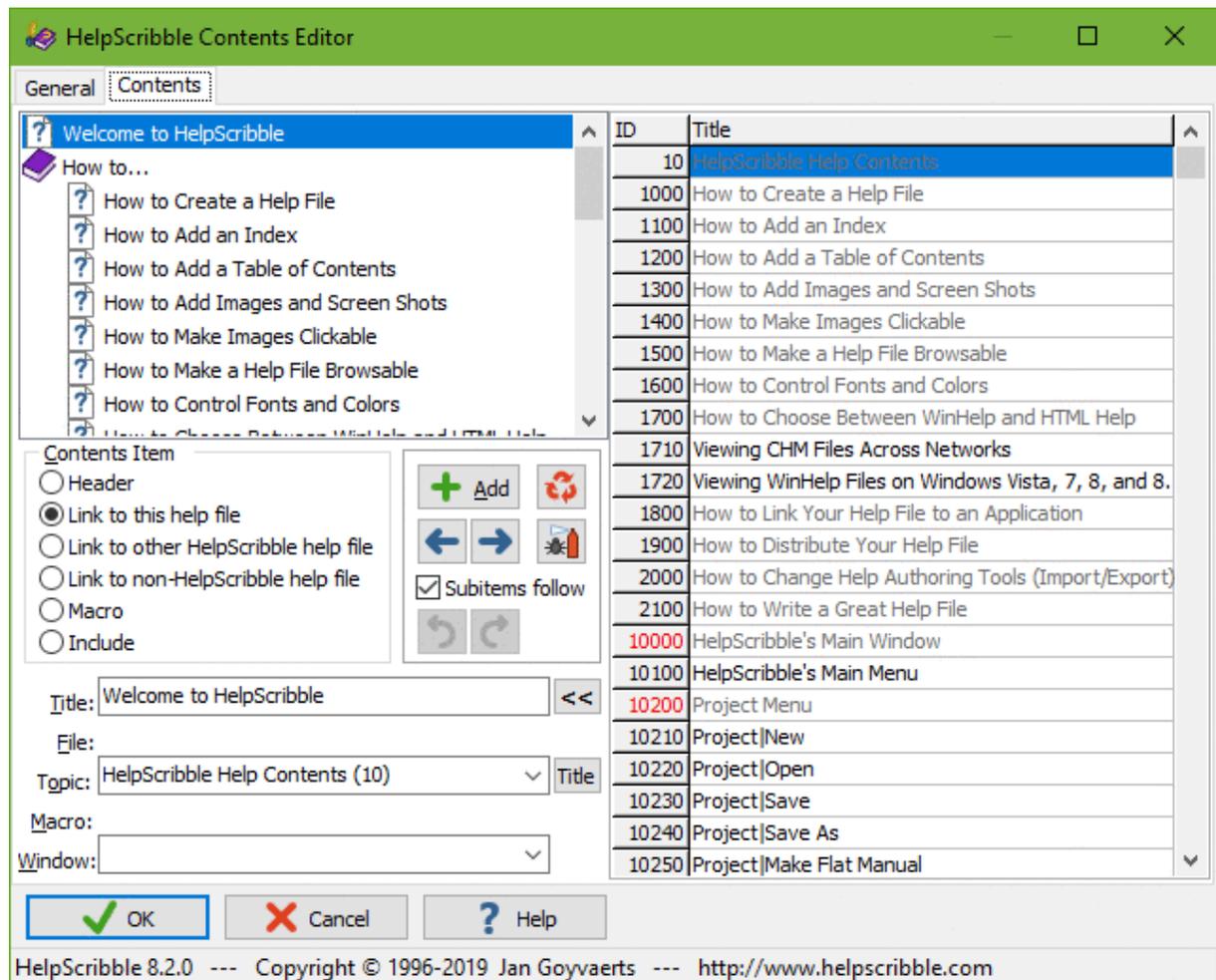
You can merge the keywords of other help files in two ways. The first way is labeled “index”. The keywords of the other help files will be visible to the user in the index. To add a help file to the list, click the Add button. Then type in the title of the other help file in the first field, and the filename of the help file, without the path, in the second field. When the user selects a keyword from the index, the title of the help file as you type it in here will be shown after the topic title.

The other way to merge keywords is labeled “link”. This will make the keywords “visible” to the KLink WinHelp Macro, rather than the user. This allows you to do KLink keyword lookups across multiple .hlp files.



Contents Editor: Contents

On the Contents page of the Contents Editor, you can build the table of contents of your help file. 16-bit WinHelp files cannot have a table of contents. They use the first topic in your help file, which is the one with the lowest Topic ID, as the table of contents. Creating a table of contents is recommended, but not required. If you do not create one, make sure the first topic in your help file contains links to all important topics in your help file.



A table of contents consists of headers and links. Headers define sections, and can be expanded and collapsed by the user. Links will display a topic when clicked by the user.

To add a header, click Add button, select the "header" radio button and type in its title. To add a topic, click the Add button, select the "link to this help file" radio button, type in the title it should be listed with in the table of contents, and select the topic from the drop-down list. A faster way to add a topic is to expand the Contents Editor to show a topic grid, by clicking the >> button next to the title field. Double-click a topic in the grid to add it to the table of contents.

To modify an item, click on it in the list, and change the values you want to change such as the title and topic fields. The changes will be automatically applied to the item in the table of contents when you click on another item.

To remove an item, click on it, and click the Delete button (red recycle icon).

If you make a mistake, correct it with the Undo button. You can undo the last 100 changes since you last started editing the table of contents. If you undo too many changes, click the Redo button. You can only redo changes immediately after undoing them.

You can change the indentation of items (i.e. the header they belong to) by clicking the left and right arrow buttons. You can rearrange the order of items by dragging and dropping them with the mouse. The option "subitems follow" determines whether, when moving a header, its subitems are also moved or not. Click the Debug button (bug spray icon) to automatically fix any combination of indentation that is not allowed.

Note that WinHelp has a limitation that causes topics to always be included under the preceding book, even if you specify otherwise.

If you specify:

```
Header
  Subitem
Main item
```

The result in the compiled help file will be:

```
Header
  Subitem
  Main item
```

HTML Help does not have this limitation and will show the table of contents as you specified it in HelpScribble.

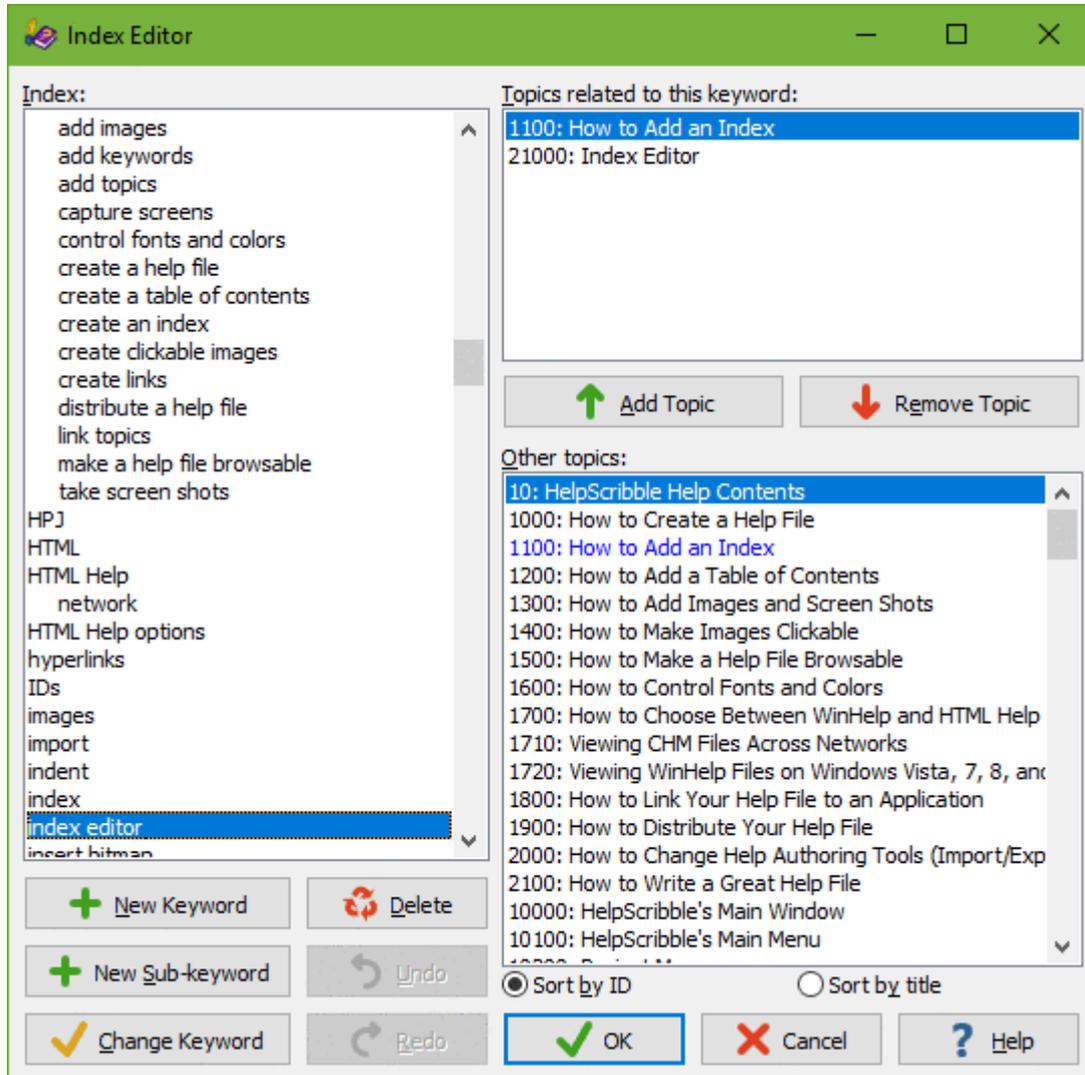
The Contents Editor also allows you to add some advanced items, which only work with WinHelp:

- Link to other HelpScribble help file: Instead of selecting the topic from the drop-down list, type in the filename of the .hlp file that contains the topic you want to link to, and type in its Topic ID.
- Link to non-HelpScribble help file: Type in the filename of the .hlp file that contains the topic, and type in the context string of the topic you want to link to.
- Macro: Run a WinHelp Macro when the item is selected by the user
- Include: Include the table of contents of another help file at a certain point in the table of contents. Select the .cnt file you want to include.

When using WinHelp, you can also specify if a topic should be shown in a secondary window. Select the window you defined with the Windows Editor from the drop-down list.

Index Editor

The Index Editor allows you to easily build the index of your help file.



You can see the index of your help project at the left hand side in the Index Editor.

To add a new keyword, click the “New Keyword” button and type in the new keyword. The new keyword will not have any topics assigned to it. Unless you add topics to it as explained below, the keyword will be removed when you close the Index Editor. If you add a keyword that already exists, the Index Editor will simply select the existing keyword. No duplicate keywords will be added. Keywords that differ only in case (e.g. “HelpScribble” and “helpscribble”) are considered identical.

To add a sub-keyword to a keyword, first select the keyword or one of its sub-keywords in the index. Then click the “New Sub-keyword” button and type in the new sub-keyword. You cannot add sub-keywords below other sub-keywords. If you try, the sub-keyword will be added to the selected sub-keyword’s main keyword.

To change a keyword, select it from the index and click the “Change Keyword” button and type in the new keyword. If the new keyword already exists, the current keyword will be deleted and the topics assigned to the current keyword will be added to the other keyword. In effect, the changed keyword is

merged with the other keyword. If the new keyword does not yet exist, it will be created with the same topics as the current keyword, which will be deleted. In effect, the keyword is renamed.

To delete a keyword, select it from the index and click the “Delete” button. If you delete a main keyword that has sub-keywords, all the sub-keywords will be deleted as well.

To add topics to the selected keyword, select them from the list of topics in the lower right corner, and click the “Add Topic” button. You can also double-click the topics in that list to add them.

To remove topics from the selected keyword, select them from the list of topics in the upper right corner, and click the “Remove Topic” button. You can also double-click topics in that list to remove them.

If you make a mistake, click the Undo button to undo the last action. You can undo all actions since you last opened the Index Editor. You can also redo the last undone action with the Redo button. You can redo everything, until you take an action other than undo or redo.

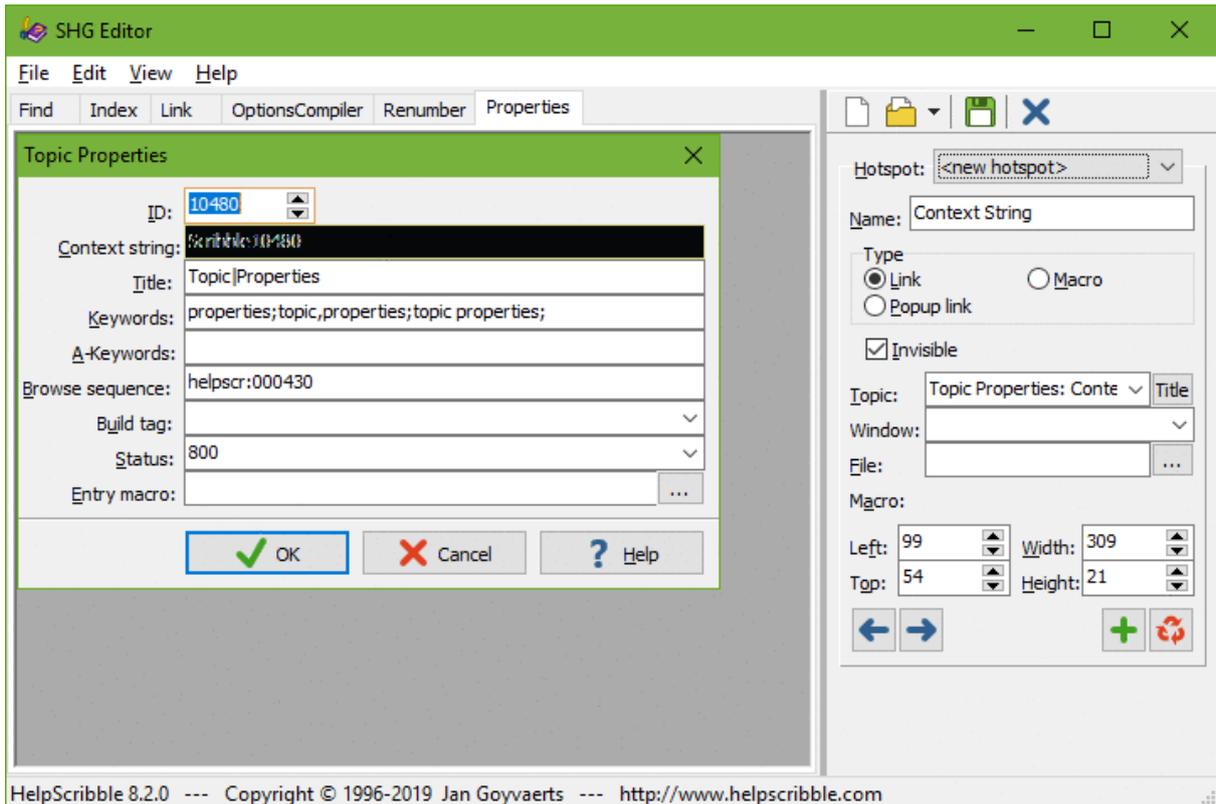
To undo or redo many actions at the same time, right-click the Undo or Redo button. A pop-up menu will list up to 16 actions that you can undo or redo. The topmost action is the most recent one. If you select an action, that action, and all the actions listed above it in the menu, will be undone. Note that while the menu is limited to 16 items, the actual undo and redo lists are unlimited. If you want to undo or redo more than 16 actions, simply right-click the Undo or Redo button again, and the next available actions will be shown.

Click the OK button to update the keywords of all topics in your project to build the new index.

If you click Cancel, all changes made in the Index editor will be forgotten.

SHG Editor

With HelpScribble's SHG Editor you can easily create Segmented HyperGraphics or SHG files. An SHG is a bitmap that has regions the user can click on to get more information about the clicked region. A common use of SHG graphics, is to include a screen shot of the window you are documenting and putting hotspots over the controls on it. SHG files can be included directly into WinHelp files. HTML Help does not support them, but HelpScribble will automatically convert SHGs to .gif or .png files plus an HTML image map. So the functionality of the SHG will remain the same.



SHG Editor: Taking Screen Shots

The SHG Editor can also save .bmp, .gif and .png files. This is useful for taking screen shots. The screen shots in this help file were created with the SHG Editor. Press Print Screen to capture the entire screen, or Alt+Print Screen to capture only the active window. Then select Edit|Paste Bitmap in the SHG Editor to paste the screen shot into the SHG Editor. If you want, you can add hotspots and save as an .shg file. Or, you can save the screen shots as a regular image in .bmp, .gif or .png format. Just select File|Save As from the menu in the SHG Editor.

SHG Editor: Adding Hotspots

To create a SHG or hypergraphic, you need to supply the bitmap first. You can do this by putting it on the Clipboard and picking Edit|Paste Bitmap from the menu. You can also open a .bmp, .gif or .png file created with another application and use it as the bitmap for the SHG.

You can add hotspots or clickable regions by drawing rectangles on the bitmap with the mouse, like you would do in a paint program. For each hotspot, you can specify the following options:

Name: Naming hotspots makes it easier to edit SHG files later. The names will appear in the Hotspot drop-down list for selecting which hotspot you want to edit.

Type: A hotspot can be a regular link, a popup link or execute a WinHelp macro.

Invisible: Determines if the hotspot is visible on the bitmap in the compiled help file, or not.

Topic: Select the topic you want the hotspot to link to from the drop-down list

Window: When using WinHelp, you can select in which secondary window the topic should appear.

File: If the hotspot should link to another help file, specify the file here. Instead of selecting a topic from the drop-down list, you will need to type in the context string of the topic you want to link to.

Macro: If the hotspot should execute a macro, specify it here. Click the (...) button to open the WinHelp Macro Editor.

SHG Editor: Sorting Hotspots

In addition to clicking on hotspots, the user can also activate them by pressing tab on the keyboard until the desired hotspot has been highlighted, and then pressing Enter. Therefore, the hotspots should be added in a logical order to the SHG. You can sort the hotspots by selecting Edit|Sort Hotspots from the menu in the SHG Editor.

To move a hotspot in the list, drag-and-drop it with the mouse, or click on it and then click the Up or Down button.

SHG Editor: Saving SHGs

When saving the SHG file in the SHG Editor, you need to select whether you want to save it with 16 colors or millions of colors (true color). 16 color files are many times smaller than true color ones, but they look awful if the original bitmap was color-rich. On the other hand, true color images will look awful if the system your help file will be viewed on does not support high or true color. 64000 or more colors must be supported for true color SHGs to be displayed properly. Often, they are displayed as solid, black rectangles. 16 color images look the same on all systems.

Conclusion: if compatibility with systems configured to use few colors is important, use 16-color SHGs. If your images are photographs or otherwise color-rich, save using true color to prevent image distortion and advise your users to configure their systems to use at least 64,000 colors.

In addition to SHGs, the SHG Editor can also save bitmaps in .bmp, .gif and .png format. These files will only contain the bitmap, and not any hotspots that you may have added. If you did add hotspots, HelpScribble will ask you if you are sure you want to save only the bitmap and not the hotspots.

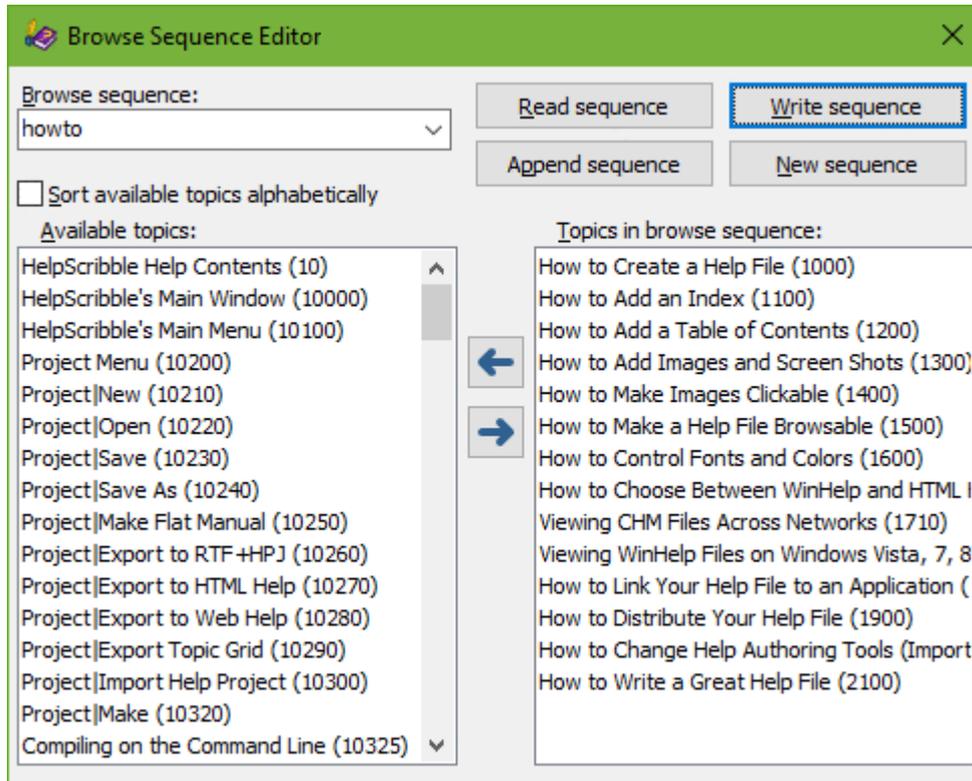
SHG Editor: Replacing the Bitmap of an SHG

There is no need to recreate an SHG completely just because you want to update the underlying bitmap. You can easily replace the bitmap with a new version, while keeping all the hotspots.

First, open the SHG in the SHG Editor. If the new bitmap was saved to a .bmp, .gif or .png file, use File|Open in the SHG editor to open it. If the bitmap is on the clipboard, select Edit|Paste Bitmap from the menu. Either way, the SHG Editor will ask you if you want to use the bitmap as the foundation for a new SHG, or if you want to replace the bitmap of the current SHG. Opt to replace. The bitmap of the SHG will now have been changed. The hotspots will remain unchanged, ready for further editing.

Browse Sequence Editor

WinHelp can display <<< and >>> browse buttons that allow the user to browse through the topics in your help file. The order in which the topics are browsed through is determined by the value of the browse sequence properties of each topic. In HelpScribble, you can easily create browse sequences by selecting Browse Sequence Editor from the Editors menu.



The "Browse sequence" drop-down list contains all the browse sequences currently defined in your help project. Click "Read sequence" after selecting a sequence from the drop-down list in order to read that sequence into the "Topics in browse sequence" list. Click "Append sequence" to read the sequence by adding the items to the end of the list, rather than clearing the list first. Click "New sequence" to clear the list.

Click "Write sequence" after typing in a new name for the browse sequence in the drop-down list to create a new sequence with that name using the topics in "Topics in browse sequence". If you select an existing sequence instead, "Write sequence" will cause that sequence to be overwritten.

The list "Available topics" holds all the topics in your help project that are not in the "Topics in browse sequence list". These topics may or may not be part of a browse sequence. You can add one of the topics to the other list by double-clicking on them. You can also select one or more topics and click the right arrow button, or drag-and-drop them from "available topics" to "topics in browse sequence".

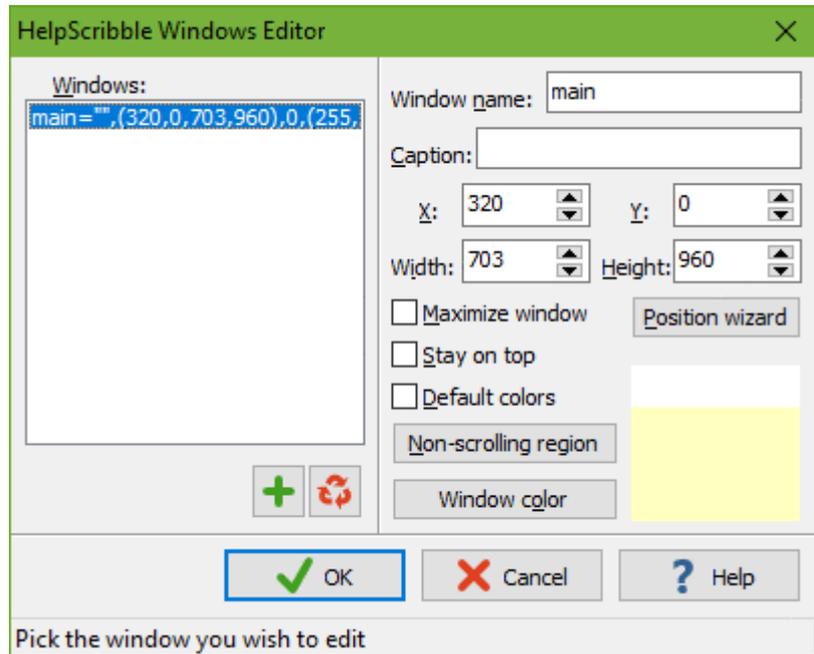
The list "Topics in browse sequence" is where you edit the browse sequence. You can double-click on a topic to remove it from the browse sequence. You can change the order of the topics by dragging and dropping them with the mouse inside the "topics in browse sequence" list. You can remove multiple topics by selecting them and clicking the left arrow button or by dragging them to the "available topics" list.

Changes to the browse sequence will not be saved until you click the "Write sequence" button. Note that each topic can appear in only one browse sequence. If you write a sequence that includes topics that were also part of another sequence, they will be automatically removed from the other sequence.

Windows Editor

If you are creating a WinHelp file, you can use the Windows Editor from the Editors menu to change the appearance of the main help window, and to define secondary windows. After defining one or more windows, you can select in which window a topic should be displayed through HelpTools|Link and in the table of contents.

To change the look (colors, position) of the main help window (the default window), create a new window and set its name to `main`. You can add a new window by clicking the Add window. You can delete a window by clicking the Delete button. For each window, you can specify the following values:



- | | |
|-----------------------|---|
| Window name: | The name of the window used internally in the help file. |
| Caption: | The text that appears on the window's caption bar. |
| Position: | X, Y, Width and Height define the default position of the window. WinHelp uses a special coordinate system. The full screen measures 1024 units wide and 1024 units high. Valid values range for X and Y range from 0 to 1023. Valid values for Width and Height range from 1 to 1024. The easiest way to specify these values is to click the Position wizard button. Resize and move the wizard window to where you want the help window to appear. |
| Maximize window: | Maximize the window by default |
| Stay on top: | Make the help window stay on top of all other windows, even if it's not the active window. |
| Default colors: | When marked, the help window will use the system default color. When not marked, you can specify the colors by clicking the buttons explained below. |
| Non-scrolling region: | The background color of the part of the help text before the keepn tag. |
| Window color: | The background color of the part of the help text after the keepn tag, or of the entire help text, if there is no keepn tag in the topic displayed. |

Note that WinHelp will not use the colors that you specified in the Windows Editor if the default window background color is set to anything but white in the Windows Control Panel.

WinHelp Macro Editor

More advanced users can use WinHelp macros to give their help files more functionality. To assist you in using the macros, HelpScribble has a Macro Editor. You can access it wherever a macro is needed by clicking on the (...) button next to the edit control where you would enter the macro manually.

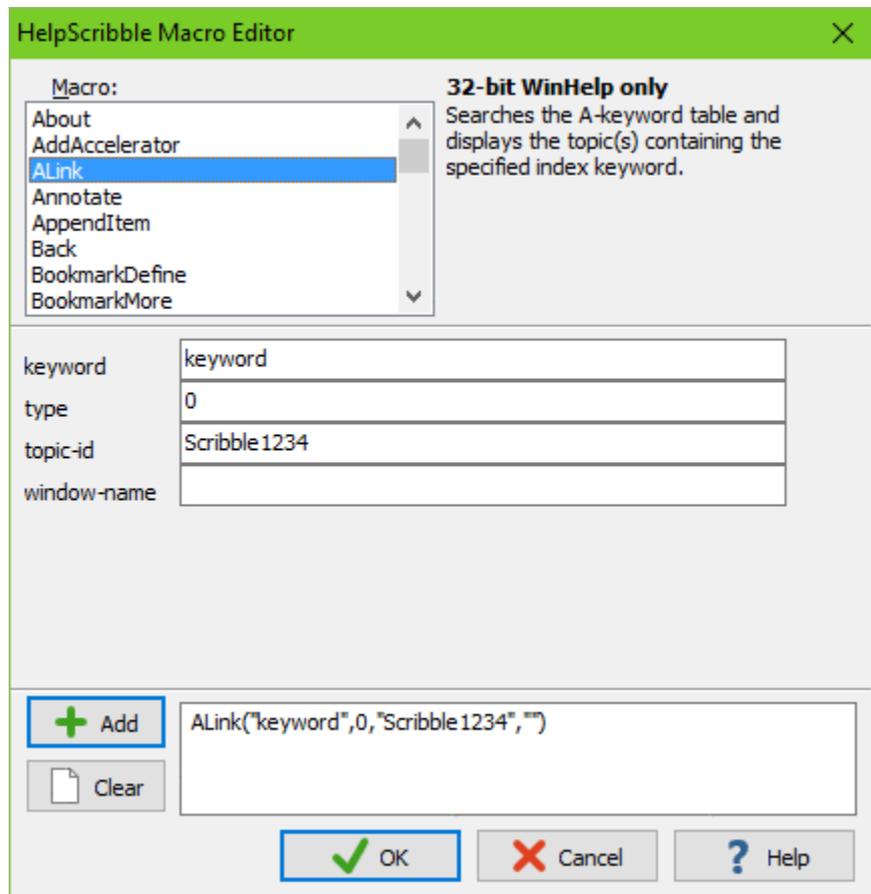
As their name implies, WinHelp macros only work when using the WinHelp compiler. When compiling to HTML Help, HelpScribble will convert the ExecFile() and ExecProgram() macros to HTML links. Other macros will not be converted.

Please do not confuse WinHelp macros with TextMacros. TextMacros are a feature of HelpScribble that store text properties so you can easily apply them to other parts of the help text. WinHelp macros extend the functionality of your help file. You can use them to add items to the menu bar or button bar, display dialog boxes, etc...

The list "macros" lists all WinHelp macros that are available. When you select one, you can see a brief description of the macro to the right of the list. Some macros are marked "32-bit WinHelp only". These macros will only work with the 32-bit WinHelp compiler. The others work with both the 16-bit and 32-bit WinHelp compiler.

Below the list of macros, you will see the parameters that the currently selected macro needs. Move the mouse over the field of a parameter to see a description below the parameter list. If the parameter requires the filename of a .hlp file, you can click the (...) button to select the file. If the parameter requires another WinHelp macro, you can click the (...) button to open another WinHelp Macro Editor to specify the macro for that parameter.

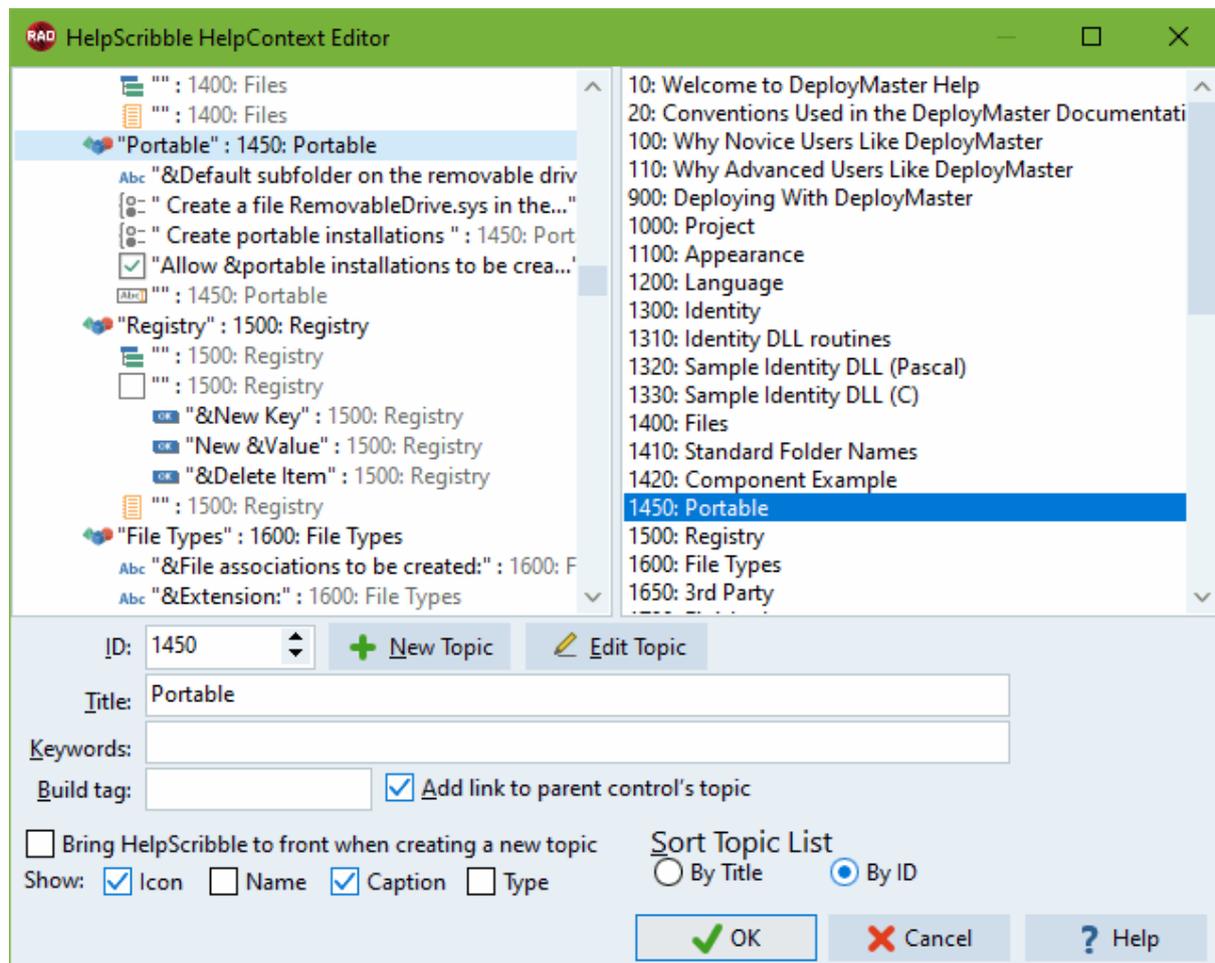
When you're done defining the macro, click the Add button to add it to the list of macros below. You can then specify another macro and add it. If you add more than one macro, they will be executed sequentially. Click the OK button to close the Macro Editor and transfer the macro to the place from which you opened the Macro Editor.



HelpContext Property Editor

Help files created with HelpScribble can be used to provide context-sensitive help for any Windows application, no matter which development tool it was created with. Most development tools will use the Topic IDs from HelpScribble to request help. A few will use the context string instead.

In Borland Delphi and C++Builder, you need to assign the Topic IDs from HelpScribble to the corresponding HelpContext properties in Delphi or C++Builder. To make this task easier, HelpScribble provides a HelpContext property editor. After installing the property editor, you can double-click on any HelpContext property in the Object Inspector to easily assign help topics to forms and controls.



The left hand list displays a tree of all the components on your Delphi or C++Builder form that have a HelpContext property. When you select a component, the properties of its help topic will be shown in the edit controls. Double-click the component or click on the Edit Topic button to edit the topic in HelpScribble. If the selected component has no help topic (help context is zero), the edit controls will show the default properties for the help topic to create for this component. Change them if you want and click the New Topic button to create a new topic for this component. If the ID is set to zero, the Topic ID for the new topic is determined like when you pick Topic|New from the menu in HelpScribble. If you specify the ID in the property editor, the New Topic button will give the new topic that ID.

The component tree can show an icon for each component, its Name property, its Caption property and its class name. You can choose to see any or all of these in the component tree by marking the corresponding checkboxes.

The right hand list displays a list of all topics in the help project you are currently editing in HelpScribble. Double-click on a topic to assign it to the component selected in the left hand list. You can sort the list by Topic Title or Topic ID by making a different choice in "Sort Topic List".

The option "add link to parent control's topic" determines whether clicking the New Topic button will also add a link to the new topic in the help topic for the parent control. The parent control is the control that contains the component for which you are creating a new help topic. This hierarchy is determined by Delphi or C++Builder, and is reflected in the component tree in the HelpContext property editor.

"Bring HelpScribble to front when creating a new topic" does exactly what it says. It allows you to edit the topic in HelpScribble right after clicking the New Topic button. When this option is off, you will need to click the Edit Topic button to edit the topic in HelpScribble.

Click the OK button to update all HelpContext properties on the form to those you just specified. Click Cancel to cancel all your changes.

Delphi Source Code Parser

If you build VCL components for Borland Delphi, you can let HelpScribble parse your source code and create an outline help file for you. All you have to do is to fill in the spaces to create a help file that will integrate very nicely in the Delphi help system.

If you want to create online help for Delphi applications, you can do so easily using the HelpContext property editor. The Delphi source code parser is only suited for documenting VCL components, not for end-user applications. Note that HelpScribble can also generate a help file for VCL components that integrates into C++Builder, but it will only parse Object Pascal code.

Before you can write the help file, you need the source code of your component. Make sure there are no more bugs in it. Not in the interface section anyway, the implementation part is ignored by HelpScribble. Otherwise HelpScribble will have problems with it. If Delphi can compile it, it's OK. You will also want to have a bitmap of your component to include it in the help file. The bitmap should be the 28*28 bitmap of the button that Delphi displays on the Component Palette.

You may want to start with a fresh project first by picking Project|New from the menu. Otherwise, the topics will be added to the current project. Then pick Project|Open Delphi Unit from the menu. Select the .pas file that is the source for your Delphi component.

Now HelpScribble will parse your source code. It will read all information in the interface section, skipping private parts in object (or class) declarations. Private class members and anything declared in your unit's implementation section is skipped, since the user of your unit has no access to those parts of your unit anyway. Properly designed VCL components are perfectly useful without source code. Any modifications the user might want to implement, should be possible by overriding (protected) members of the component class.

A dialog box will pop up listing all the declarations in your unit. Now you can choose which declarations should be included in the help file and which not. By default, everything is included except for protected members of class declarations. Inherited properties that are published by your component are listed in the help file, but no topics are created for them by default. Instead, they will link to the Delphi help files where they are documented.

Read these topics on how to include or exclude declarations from the help file:

- Including or Excluding Classes and Their Members
- Including or Excluding Class Members
- Including or Excluding Types, Variables, Constants and Routines

HelpScribble can use the comments you provide in your source code as the default descriptions for all the properties, methods, types, etc. Note that the comments regarding a certain declaration should be placed after the declaration, not before it. If you want HelpScribble to use your comments, make sure the Use Comments as Descriptions checkbox is checked.

When you are done, click OK. Sit back while HelpScribble generates your help file.

Different versions of Delphi and C++Builder require different help files to be integrated in the online help system. You can build all help files from the same HelpScribble project. Simply select the target version of Delphi or C++Builder in the Project Options before you compile the project.

The only difference between the Delphi 4-7 and C++Builder 3-6 options is the help file that is linked to in your help file's .cnt file. These are DEL4VCL.HLP, DEL5VCL.HLP, DEL6VCL.HLP, D7VCL.HLP, BCB3VCL.HLP, BCB4VCL.HLP, BCB5VCL.HLP and BCB6VCL.HLP respectively. If you do not link to the VCL documentation, you can remove this line and use the same help file for Delphi 4 through 7 and C++Builder 3 through 6.

Use Comments as Descriptions

If this checkbox is checked in the Delphi Source Code Parser, HelpScribble will use any comments following a declaration as the description for that declaration, instead of writing <<< Description of ... >>>. The comment can be spread over several lines. Excessive spaces, tabs, etc. (used for formatting your source code) will be removed from the description.

Example:

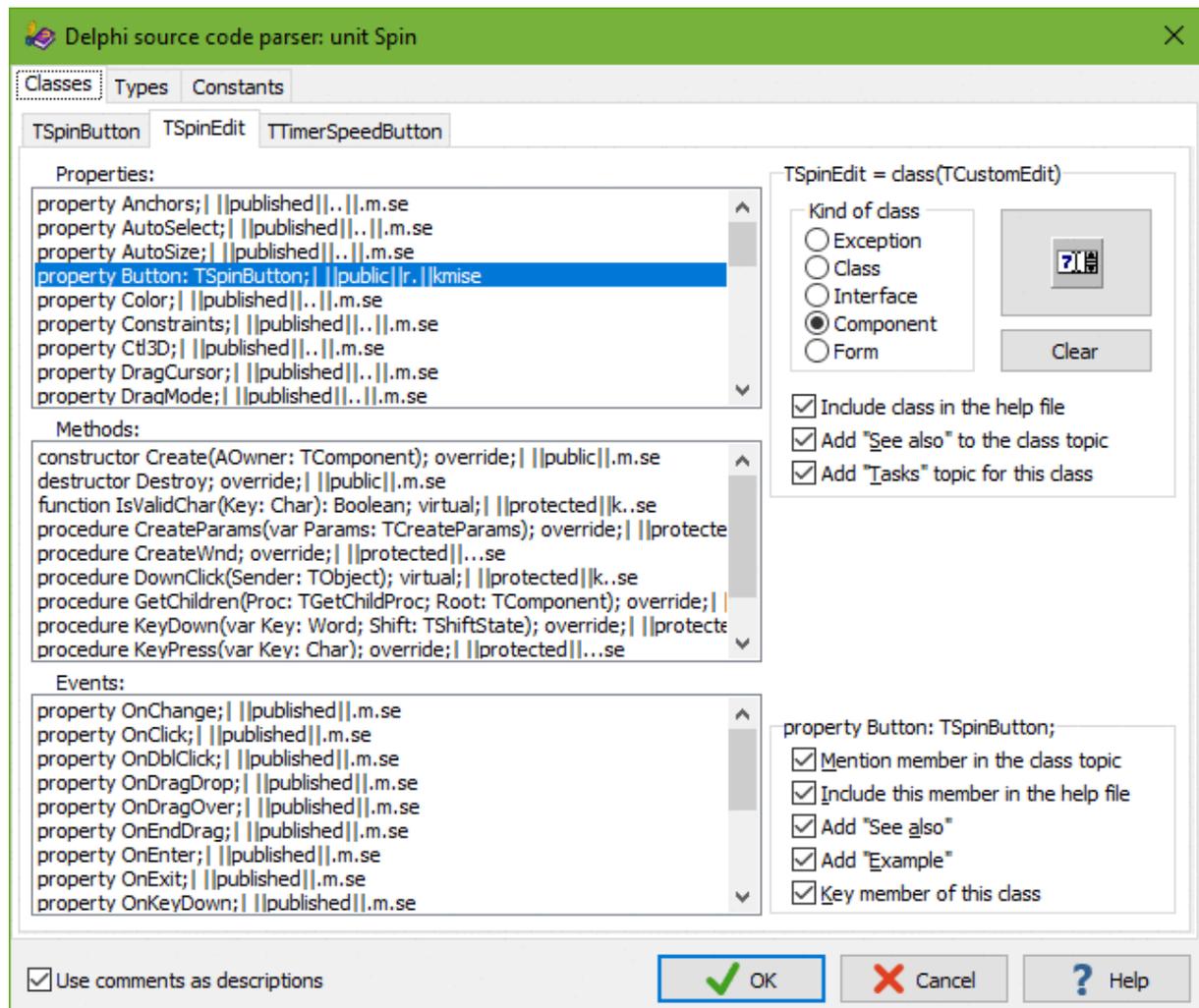
```
{rtf=\keep}class TMyComponent(TComponent)
  private
  ...
  published
    property One: Integer read FFirst; // This is the first property,
(*   which has   *)
    { a
      long
      comment }
    property Two: Integer read FSecond;
    property Three: Integer read FThird; // property #3
  end;
```

The respective descriptions for the three properties will be:

- This is the first property, which has a long comment
- <<< Description of the Two property >>>
- property #3

Including or Excluding Classes

Before HelpScribble actually builds the help file for your VCL component, you need to select in the Delphi Source Code Parser which items you want to have included in the help file. By default, everything is included except for protected members of class declarations. Private class members and anything declared in your unit's implementation section is skipped, since the user of your unit has no access to those parts of your unit anyway.



On the Classes tab, you will see one tab for each class in the unit. For each class, you can select what the class is used like. HelpScribble tries to determine this information by itself, but you can change it if needed. This information is used to add the proper kind name after any reference to your object, like the title in the object's topic and the references to it in the topics of the properties and methods.

If the class is a component that has a bitmap on the component palette, you will want to place it in the component's help topic like Borland's help files do. Click on the large button and select the file. The button will then display the bitmap for your reference. The bitmap should be a 16-color 28*28 bitmap displaying the components button on Delphi's Component Palette. However, HelpScribble will accept any bitmap. You can even enter a filename that does not exist (yet).

If you uncheck *Include class in the help file*, the class will not be mentioned at all in the help file. If *Add "see also"* is checked, a "See also" topic will be created for the object. If *Add "Tasks"* is checked, a "Tasks" topic will be created. This topic should be a step-by-step guide on how to use the object or component.

These three list boxes contain all non-private properties, methods and events in the help file. (Private members are not included as they are not visible outside the unit anyway.)

Select the property, method or event you want to include or exclude. Then use the checkboxes to change the parameters.

Check *Mention in component topic* to include the member in the Properties, Methods or Events pop-up topics of the object. These three pop-up topics are automatically created when needed.

By default, any public or published member is mentioned.

Note that mentioning a member does not automatically create a topic for it. Members which do not have their own topics, will link to the Delphi help files by default. You can edit these links in the help file when it is generated.

Check *Include in help file* to actually create a topic for the member.

Check *Add "See also"* to create a See also pop-up topics for the member.

Check *Add "Example"* to create a topic where you put an example on how to use the member.

Key determines if the member is a key member. Key members are members that are special for this object or component, as opposed to general members that are usually inherited. Key members will have an image of a key next to them.

If you want to list the properties, methods or events in a different order, drag and drop the items in the lists with the mouse. HelpScribble places the items in alphabetic order, but you may want to place cooperating items together.

Sometimes, it happens that HelpScribble is unable to correctly distinguish whether a class member is a property or an event (events are special kinds of properties). If that would happen, you drag and drop the incorrect item from the properties list to the events list or vice versa.

After specifying the details for the class as a whole, you can determine what should be generated for each property, method or event.

Including or Excluding Class Members

On the Classes tab in the Delphi Source Code Parser, you will see one tab for each class in the unit. The three list boxes on the Classes tab contain all non-private properties, methods and events in the class. Private members are not included as they are not visible outside the unit.

Select the property, method or event you want to include or exclude. Then use the checkboxes to change the parameters.

Check *Mention member in the class topic* to include the member in the Properties, Methods or Events pop-up topics of the object. These three pop-up topics are automatically created when needed. By default, all public and published members are mentioned. Note that mentioning a member does not automatically create a topic for it. Members which do not have their own topics, will link to the Delphi help files by default. You can edit these links in the help file when it is generated.

Check *Include in help file* to actually create a topic for the member. Check *Add "See also"* to create a See also pop-up topics for the member. Check *Add "Example"* to create a topic where you put an example on how to use the member.

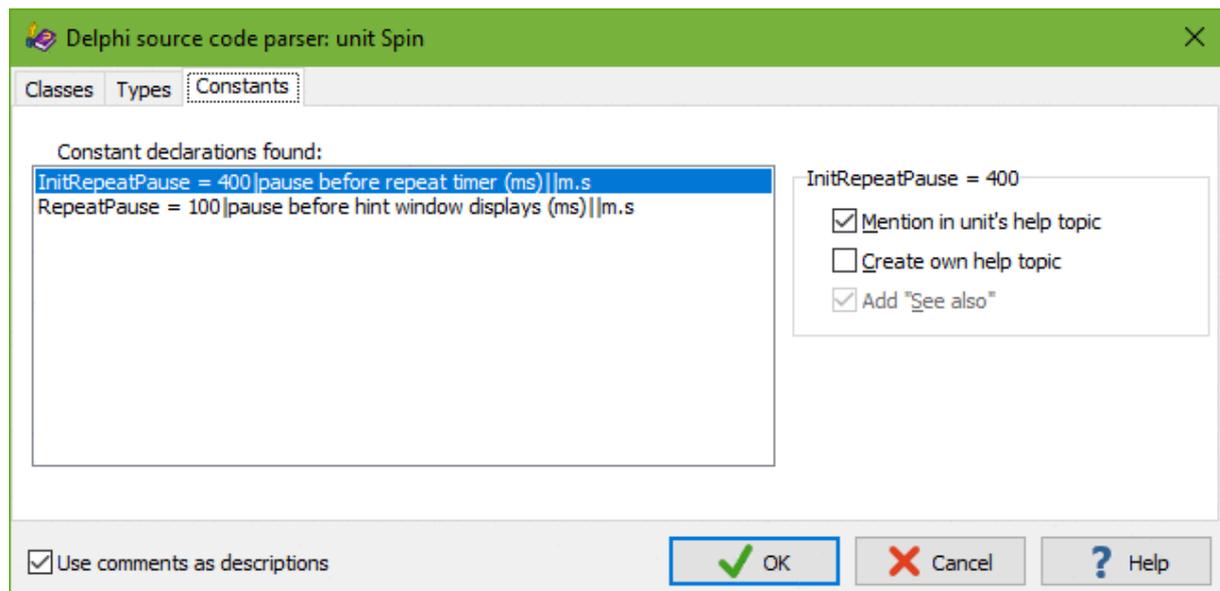
Key determines if the member is a key member. Key members are members that are special for this object or component, as opposed to general members that are usually inherited. Key members will have an image of a key next to them.

If you want to list the properties, methods or events in a different order, drag and drop the items in the lists with the mouse. HelpScribble places the items in alphabetic order, but you may want to place cooperating items together.

Sometimes, it happens that HelpScribble is unable to correctly distinguish whether a class member is a property or an event. If that would happen, you drag and drop the incorrect item from the properties list to the events list or vice versa.

Including or Excluding Types, Variables, Constants and Routines

In the Delphi Source Code Parser, you can specify which types, variables, constants and routines defined in your unit should be included in the help file.



To the left, you see the list of items that can be included in the help file. If you want to include the declarations in a different order than the one proposed by HelpScribble, drag and drop the items in the list with the mouse.

Check *Mention in unit's help topic* if you want the declaration to be included in the unit's help topic. If the declaration has its own topic, a link to it will be added in the unit's topic if it is mentioned. Check *Include in help file* if the declaration needs to have its own help topic. Check *Add "See also"* to create a "See also" topic for the declaration. Check *Add "Example"* to create a topic where you provide an example on how to use the type or routine.

Linking to the Delphi or C++Builder documentation

When you are creating a help file for your Delphi/C++Builder VCL component, you may want to include links to the online documentation that comes with Delphi or C++Builder. That way, you can point the user to more information on properties, methods, etc. which your component inherited from standard VCL components or which are useful in conjunction with your component.

Unfortunately, this is only possible with Delphi 1 through 7 and C++Builder 3 through 6. Newer versions of Delphi and C++Builder no longer use WinHelp. They use a new help system that can't be easily linked to.

When creating a WinHelp file for Delphi 1 through 7 or C++Builder 3 through 6, you can do this easily with a `LinkDelphi` tag, which is very similar to the `LinkID` tag.

To link to a certain class, use:

```
{linkDelphi=classname_Object}  
e.g.: {linkDelphi=TButton_Object}
```

To link to a property, method or event, use:

```
{linkDelphi=membername_[Property|Method|Event]}  
e.g.: {linkDelphi=Caption_Property}, {linkDelphi=Create_Method}, {linkDelphi=OnClick_Event}
```

If you link to the VCL documentation, it is important to specify the proper kind of help file in Project Options: WinHelp