

Help Use Outline

A Macro for translating Word files into Windows Help files

Disclaimer

You use this macro at your own risk! **You** are responsible for making suitable **backups** in case it screws up. As far as I know, it does not have any side-effects except modifying your DOC file and creating a HLP file, but you never know with computers!

This macro is also not intended to be an all singing, all dancing, help development system. It is good at what it does, namely, **transforming existing written documentation into on-line Windows Help files**. The registered version will also allow you to define cross references in a very natural way, and does various other things. The intention is that you can maintain **one source file** which can be printed or transformed into on-line Help. If you want to create a hypertext that has no hierarchical structure and is more akin to an adventure game (not a clever idea in my humble opinion), then this macro may not help you too much.

Introduction

Tired of trying to build Windows Help files?

Trying to maintain printed documentation as well as on-line help files?

This macro is for you!

Encapsulated in a Template file, HELPFILE.DOT for Word for Windows, the macro uses the document's outline structure (defined by the **heading 1**, **heading 2**, etc styles to build a hierarchically structured HLP file. Cross references can also be defined (but are only processed by the registered version - US\$20 for registration - a mere pittance).

Getting Started

Copy the HELPFILE.DOT file into your **WinWord** directory.

Create a new file with **File, New** and choose **HELPFILE** as the **template**. If you already have a document that you wish to translate, use the **File, Template** command to assign this new template to it, so that the macro will be available to you.

For the unregistered version, you must manually create a **.HPJ** (Help ProJect) file. A sample is included, which you can view to see how it works. The registered version will create a project file automatically if one is not present.

Run the macro by choosing **Help Use Outline** from the **Help** menu. If you don't like it on the Help menu, you can move it elsewhere, or assign a keystroke to it using the **Tools, Options** command.

There are actually two macros. The second one, called **RunHelpCompiler**, requires that the Help Compiler be accessible. It looks for HC.PIF, so create a PIF file for the Help Compiler and put it in your PATH. Make sure that the PIF file points to the location of the Help Compiler.

What it Does

The macro reads your document one paragraph at a time. It checks to see if the paragraph **style** is a **heading** type, and if so, what **level** it is (1, 2, 3, etc). It records the level. It defines each **heading** paragraph as a **topic**, including the first line of that paragraph as a **title**, and also as the **keywords** section. It generates the **context** name automatically. A future version might take more care with the keywords, but this is a fairly easy way to handle things, and moderately effective. The first topic is called "**contents**".

The title for this section should **not** have a **heading** style. Create the top level header with a style of **Normal** or something like that.

Having made a complete pass through the document, it starts again, and inserts all the menus required for a user to select sub-topics within any particular topic.

Since the macro inserts hard page breaks before topics, it's a good idea to make sure that you haven't used "Page Break Before" for any of your **header** styles.

Try it out!

Assuming that you have a HC.PIF file already (if not, go and do that now!), and assuming that you've copied HELPFILE.DOT to your WinWord directory, then you should be able to convert **this** file into a Help file right now.

Although this document uses the Template HELPFILE, **your** version of WinWord may not yet know that it exists. Select **File, Template** to assign it again.

If you now pull down the **Help** menu, you should see that the last option is **Help Use Outline**. Select it, sit back and enjoy.

Features

1. It saves your document to disk before it tries to change it (but beware of automatic savers - a problem fixed in the registered version).
2. It creates a **hierarchical structure** for your Help file automatically.
3. It defines keywords using the **Title** (first line) of each **heading** section.
4. It creates **context names** and **keywords** for each topic.
5. It creates a **Browse** sequence from start to finish so that the document can be read in a linear fashion, just like the printed version.
6. It calls the Help compiler (assuming you have a **HC.PIF** file available).
7. The registered version does many other things. These are documented later, together with a list of planned enhancements.

Project File

An example Project File is shown below. It can be modified to suit yourself. Note that you

should be using the Help Compiler which comes with the **SDK for Windows 3.1** in order for some of these options to work.

You can change the **icon**, remove **compression**, alter the **title** and the **copyright** notice. The [FILES] section contains the name of the file (must end with RTF and be the same name as your DOC file). The **BrowseButtons** option makes sure that the browse buttons are included. The macro creates a browse sequence from the start to the finish, so the document can be read in linear fashion if desired.

[OPTIONS]

compress=on

icon=helpoutl.ico

title=Help file generated from the Outline

copyright=© Roger Hadgraft, 1992

[FILES]

helpoutl.rtf

[CONFIG]

BrowseButtons()

Limitations

This version is for shareware release only.

It does not include cross referencing, nor does it have a default HPJ file. These are available in the registered version. It is also explicitly limited to 4 levels of outlining, although there's no reason why that can't be changed.

There is a limitation of 200 topics at the moment. You can change that by modifying the first DIM line. I'm not sure how large you can make it without running out of memory. I guess it depends on how long your title names are.

The macro modifies your DOC file. Be careful that you don't save the modified version on top of the original document. The registered version saves the modified document as docname.NEW before any changes are made, just in case you use a utility for automatic saves (as I do).

Registration

Registration costs US\$20 or AUD\$28. Send cheques (preferably in Aussie dollars) to:

Roger Hadgraft
Department of Civil Engineering
Monash University
Clayton, Victoria, 3168.
Australia

Please provide an e-mail address if possible to which I can send the registered version.

If credit card facilities would be easier, please contact me by e-mail. Such things can be arranged through the University.

Report bugs/problems/suggestions to:

e-mail: roger.hadgraft@eng.monash.edu.au
fax to: +61 3 565 4944.

Extra Features

Extra features in the registered version are:

1. **Cross references**, including the detection of predefined jumps and calls to macros. Cross references are defined by using all or part of a card title. If this is not unique, a table of possible jumps is built, and appears as a popup for the user. If the cross reference cannot be resolved, a warning is issued.

Cross references are defined by using double underlining (Ctrl-D) on the word or word to be used. If there are multiple targets, the double underlining is automatically changed to single underlining so that a popup, and not a jump appears.

When defining a cross reference, it is only necessary to use enough of the title as is unique, eg. "Office" might be enough to match "Office Procedures" as long as nothing else starts with "Office".

Ambiguity can be used to find several sections starting with the same word, eg. "Windows" will match Window 1, Windows 2, Windows 3.0, Windows 3.1, Windows 3.1. In this case, a popup will appear from which the user may choose one of these or none at all.

At the moment, cross references are case sensitive, and the word must match starting at the start of the Title. Hence "printing" won't match "Workstation printing", but "Workstation" would. This is something for development in the future perhaps.

Also, the start of a title is considered to be the actual start, or after the first TAB. This allows for a title such as "3.6 Windows 3.1" to be matched by "3.6" or "Windows 3.1".

If a multi-cross references has already been defined, then another similar reference will use that in preference to an existing title. For example, if there is already a multi-cross reference for "Windows", and "Windows" is used again, then the same cross reference table will be used (which is probably what you intended). If instead, there was a multi-table for "Windows NT", and you used "Windows" it will match with "Windows NT". Perhaps I should make sure that any multi-table match must be exact.

2. **Popups** are treated in the same way, except that instead of using double underlining, single underlining is used.
3. A **project file** is automatically created if one is not present. This means that given a Word file, it can be translated to a Help file **without** any other activities, assuming that it has a structure defined using **headings**.
4. The modified document is saved with a **NEW** extension to avoid over-writing the original.
5. The **number of topics** has been increased to 500.
6. Other improvements have been made to its **robustness**.

7. The macro asks if you want to keep the old phrase table (used during compression), and also asks if you want to delete the NEW and RTF files to save space. If so, it automatically reloads the DOC file.
8. Registration includes updates free for 12 months if an e-mail address is supplied.

Helpful Hints

Keep with Next

One item not explained in the Help Compiler manual is the use of the "Keep with Next" and "Keep Together" options for headings. I usually use these for written documents because they prevent stray titles on the last line of a page. In a Help context, they produce a page for which the title will not scroll off the screen, which is usually a good idea.

Just beware that because of the way that the menus are built, you may receive some warning messages from the Help Compiler saying that a "non-scrolling region has been defined after the first line of a topic". Just ignore these. They seem to cause no problems.

Planned Extensions

1. I plan to split the Title into separate words and add each as a keyword. To save having trivial words in the list, I will reject all 1, 2 and 3 letter words. (I may also check words against a list of other common words - from, than, etc).
2. I plan to add a dialog box at the start which allows various parameters to be set.
3. I could provide the option to write warnings to an error file rather than have a continuous barrage of popup message boxes.
4. It would be a neat idea to be able to resolve cross references into other Help files. This could probably be done by writing an output file (filename.XRF) which contains titles and context names which could be scanned when cross references from other files are being resolved.
5. Other macros will be written for defining topics in Help files, and are useful if you are building a system of help files launched from a top level menu. They would be used for defining common links such as:
 - calls to execute programs
 - jumps to topics in other help files
6. I would like to be able to define an "Up" button and macro that would allow a user to go up one level. I haven't yet worked out whether it's possible with the macros available within the Help Compiler.

I guess the rate at which these things are done depends on how many people register and request such things!

Distribution

If you wish to pass this macro to others for evaluation, please include **all** the files:

HELPCFILE.DOT
HELPCOUTL.DOC
HELPCOUTL.HPJ
HELPCOUTL.ICO

Please do not remove the copyright notice from the text of the macro.

Customisation

I am happy to customise the macro to suit registered users. If the work involved is likely to be of general use, there will probably be no charge, but if it is specific, then we can negotiate a charge.