

Doctor Help

(or DOC to HLP)

A Macro for translating Word files into Windows Help files

[Click here for help in reading this Help file.](#)

For New Users

Tired of trying to build Windows Help files?

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

This macro is for you!

Introduction

Encapsulated in a Template file (HELPPFILE.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 Windows Help file (.HLP).

Cross references can also be defined, and keywords are automatically defined. Complete Features are described later. Both registered and shareware versions include a Help version of this file, so you can have a look at the product of the registered version.

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 a **copy** of your DOC file and creating a HLP file, but you never know with computers! The macro is careful to save your file, and to make a backup copy of it when the macro starts.

This macro is 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 intention is that you can maintain **one source file** which can be printed or transformed into on-line Help. This is extremely useful for me, and, I suspect, many others, who want to build hyperdocuments without learning a more complex tool.

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 the registered version of this macro will still be useful to you, but you will need to do more work in defining all the cross references.

The evaluation (shareware) version of the macro is limited to 30 topics. The registered version has a limit of 1,000. I don't know of anyone who has come up against this limit yet. You're more likely to run out of memory I suspect! Hence, the new evaluation version is fully functional except for this limitation. Both versions are execute only, which means you can't modify the code.

Getting Started

The following instructions will get you started. More detailed instructions come later under Features Explained.

1. Copy the HELPFILE.DOT file into your **WinWord** directory.
2. **Start** Word for Windows.
3. 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. The macro will then be available to you.
4. Select Win.Ini Options from the **Help** menu. Specify the directory where you have installed Dr Help. This directory should at least contain **DEFAULT.HPJ**, **DEFAULT.ICO** and **DRHELP.INI**. (If these files are in the current directory, this is not necessary).
5. Make sure that you have created a suitable PIF file for the Help Compiler.
6. Run the macro by clicking on the **Doctor Help** button.
7. The macro will create a project file automatically if one is not present (before any processing occurs), and it will allow you to modify the parameters by popping up the Options Dialog. You will also be asked to complete the Win.ini options dialog box. In it, you must specify the directory where DrHelp is installed (eg. c:\windows\drhelp).
8. Wait for the macro to finish. The Help compiler will then be called. At its conclusion, click on the box to indicate its completion, and then you will be asked if you wish to delete the .NEW and .RTF files. Normally, you should answer "yes". Your new help file will then be displayed. That's how easy it is!

PIF file for the Help Compiler

There are actually several macros to do the complete job. The second stage of the main procedure, called Run Help Compiler, requires that the Help Compiler be accessible. It looks for HC.PIF, so create a PIF file for the Help Compiler and put it (the PIF file) in your PATH. (An example PIF file is included). Make sure that the PIF file points to the location of the Help Compiler. You can also run the second macro independently of the first. (I do this a fair bit during development if something goes wrong in the second stage and I want to try it again). You may not need to do this.

To use all the features, you should have version 3.10 of the Help Compiler. It is available from FTP.CICA.INDIANA.EDU in the WHAT.ZIP file which also contains Microsoft tools for building Help files.

What the macro does

The macro skips through your document one paragraph at a time. It checks to see if each 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, using the first line of that paragraph as a title, and (optionally) splits the title into separate words to make a keywords section. A **topic** is one page in the help viewer, and is designated by a hard page break before it (which Dr Help inserts). It generates the context name automatically. (Context names are unique identifiers for the topics). Context names are used by the Help Compiler in constructing the links between topics. The first topic has a context name of "**contents**". (Context names can also be user-defined).


Having made a complete pass through the document, Dr Help starts again, and processes all the cross references, including creating intermediate tables for ambiguous cross references, and then it 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 disables any "Page Break Before" attributes you've set for any of your heading styles. It also removes any hard page breaks you've inserted, and also removes Section breaks, Index entries and some other field types.

Try it out!

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

If you use a **non-English version of Word**, you will first have to define the Language setting in WIN.INI.

Click on the  button, sit back and enjoy. DRHLPUSR.DOC is small enough to be converted by the evaluation version, but it still includes a number of interesting features.

Summary of Main Features

1. Dr Help **saves your document** to disk before it tries to change it, saving the document to its original name, and also under a name of filename.NEW. Only the .NEW version is modified.
2. It creates a **hierarchical structure** for your Help file automatically. Most help files have a basic hierarchical structure, so this makes this part of the job really easy. Your challenge is then to define other cross references, which are also very easy, because you can use ambiguous cross references.
3. It defines **keywords** using the words in the **Title** (first line) of each **heading** section. You can also have user-defined keywords.
4. It creates a **context name** for each topic. This is used for building the links. You can also define your own context name (see defining context names) so that you can handcraft some links (eg. links into other help files, or using hotspots in graphics).
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. This gives the document the same feel as a book, with the added benefit of the partial tables of contents at each branch point.
6. It calls the Help compiler (assuming you have a **HC.PIF** file available).
7. Dr Help does many other things, documented more carefully under "Features explained".

Project File

The first time you run Dr Help on a file, it will create a project file, prompting you for a title. **It must be able to find DEFAULT.HPJ in the current directory, or on the PATH defined in WIN.INI.** If you are creating help files in several directories (as I am), then you can keep one copy of DEFAULT.HPJ in your main DRHELP directory. Of course,

you could keep several such files if you use different settings for different projects.

Once created, you can modify many of the parameters in the Project File using the Options Dialog macro, and a few in the WIN.INI Options dialog.

Limitation of Evaluation Version

The evaluation version is limited to 30 topics. This includes any ambiguous cross reference tables that are built. The registered version can be obtained as described later.

In the registered version there is a limitation of 1,000 topics at the moment. I'm not sure how large you can make this without running out of memory. I guess it depends on how long your title names are.

For Existing Users

Changes in release 4 include:

User-defined Keywords & Split titles

1. Ability to define **additional keywords**. Do this by adding words or phrases after the title, and separated by semi-colons. These words must be **hidden**. A title might look like:

Only the first part will be used for the title, but all words will be entered as keywords. These additional keywords are **not** used in resolving cross-references. That's a possible future development.

25. Titles can now optionally be **split** into separate words to make keywords. This is selectable in the Options Dialog.

Context name identification changed

2. A different technique is used for identifying pre-defined context names. This was a result of development 1. This shouldn't have any effect on you, but if you have any problems, let me know asap.

Index entries removed

3. **Index entries** are now removed from the file. I am considering creating a macro to insert index entries in the same way that keywords are generated. This would eliminate most of the need to add index entries and would create an index more or less identical to the help file's Search list.

Saving during execution

4. The macro now **saves** while it's working. This helps to get around memory usage problems created by too much searching. The default save rate is every 50 topics, but this can be reduced or increased via the Options Dialog.

Screen update problem

5. There is a known problem when screen update is off (the default) that if you have a very long title line, Word will crash. You can tell Word to ignore the error and it

will continue successfully. It has something to do with turning off the screen updating and Word wanting to scroll the footnote pane. If this occurs, either turn screen update **on**, or shorten your title line. This may now occur more often if you define many additional keywords.

Separators and Keywords

6. Title strings are scanned and special characters removed when the keyword list is created. Characters removed are: ,():&-?! You can add to or subtract from this list within the Options Dialog. Naturally, adding to the list slows processing.

Context names not scanned

7. I have removed a scan of the **context strings** when cross references are being resolved. This should make things go a bit faster. I haven't used this feature, and I assume no-one else has either. It seemed like a good idea at the time.

Options Dialog changed

8. The Options Dialog no longer prompts for a filename. It assumes a project file of the same root filename as the currently open file, just like Run Help Compiler. If there is no existing project file, it will prompt for one. Some additional fields have been added to the dialog as explained later.
15. The Options Dialog now includes an option for turning **compression** on and off.

Run Program Macro expanded

9. The RunProgramMacro now lets you test the executable string that you type and makes sure that there is an executable object specified (an .exe, .com or .pif file). It returns and prompts again if either of these things fail. This is not foolproof because Word has slightly different rules than WinHelp, and you may be developing a Help file for a different environment (eg. a network), but it's better than doing nothing!

Case-insensitive matches are used

10. The **Match** routine now ignores case even in exact matches. This was important when resolving ambiguous cross references so that separate (identical) tables weren't generated for "cross references" and "Cross references" for example.

Default icon file

11. The default project file now references a **default icon** called **default.ico** which is a slight twist on the standard Help icon (that's a stethoscope behind the question mark). You can now specify the icon filename in the Options Dialog. Further, the Win.ini Options dialog allows a PATH which points to the directory where the default.ico file and default.hpj files are stored.

Top level heading

12. The top level heading can have a **heading** style if you wish. This allows you to easily fragment an existing large file into smaller ones if you wish.

INVOICE.DOC for the bean counters

13. Administratively, I have added INVOICE.DOC so you can give it to your bean

counters so they can me some money.

InsertTab fixed

14. There was a **bug** in **Release 3** that prevented the **InsertTab** option from being changed through the Options Dialog. That has been fixed. However, InsertTab is an obsolete feature. See the [Menu Styles](#).

Current Directory is correct

16. The macro is now careful to change the current directory to the one that contains the source file. Previously, it was possible to be in the wrong directory, and the HLP and other files would be generated there.

DEFAULT.HPJ expanded

17. Added a simple [MAP] section to the default.hpj file, and various new options. **Please update your own default.hpj file.** It also includes a new item in the Help menu to call up drhlpusr.hlp.

Processing of Fields

18. Modification of several fields has now been included. The Index fields are now removed, as are the AUTONUM and TableOfContents fields. The SYMBOL fields are translated into their corresponding characters. For some reason, the WingDings font doesn't show on my system in Help files. If it works for anyone else, please let me know.

Font size

19. When a fixed font size is chosen, a default size of 12 is assumed unless specified. This is now an obsolete feature. See the [Menu styles](#) for an update.

Default Title & First Topic

20. The default title is "FILENAME.HLP - first line of DOC file"
24. The Title defined in [Options Dialog](#) is checked to make sure it does not exceed 50 characters. If it does, an error is displayed and the dialog box is redisplayed.
21. The title of the first page is used as the title for the first topic, instead of "Contents" as before. "Contents" is still used if the first line of the file is blank. (Try using "lines before" instead of blank lines).

Display of Run Time

22. The macro calculates its run time - for the duration of the macro, and then the duration of the Help Compiler.

Buttons on the Toolbar

23. There are now 4 buttons in the Toolbar:
These run (i) [RunProgramMacro](#), (ii) [JumpToAnotherFile](#), (iii) [Options Dialog](#), and (iv) DoctorHelp.

UserActions Macro

26. There is now a [UserActions](#) macro which is called immediately before the Help Compiler. Some users like to fiddle with the RTF file before the Help Compiler

gets to it. Simply open it up and add whatever you want to it. Thanks to George Mills for this idea.

If there are items that you include here that you think might be of general interest, please let me know.

New Menu Styles

27. Of all the new features in release 4, this is probably the one to cause the most difficulties. As far as I know, everything else is compatible with previous versions. This time, you may find that your recompiled help files are slightly different from before. Let me explain the changes first, and the consequences.

Two new **styles** have been added to the template file, HELPFILE.DOT: "Menu Dr Help" and "Menu Header Dr Help". These give you control over how you want these menus and menu headers to appear. "Menu" items are the many tables of contents that Dr Help writes. A "Menu Header" is the header written for all ambiguous cross references. Edit these styles in HELPFILE.DOT to change them permanently. It was decided to introduce these 2 styles to give you complete control over the appearance of these items. Previously, you could only choose the size and bolding and optionally insert a TAB. Now, you can choose any style you like for them!

These styles will automatically be included in each new file. Existing files should be processed as before (I hope!).

There is a new menu option "Add Menu Styles" to create these two macros in old files. This will work if you are already using a **fixed font size** for the menu items. The default font is **Arial**. You can modify the styles as you prefer. The old settings (Menu Bold, Menu Font Size and Insert Tab) have been removed from the Options Dialog.

If you have used the (very) **old option** of making the menus use the same style as the equivalent heading, then **don't** add the styles to your file. If you do choose this option, you will be asked to confirm that this is what you want to do. If in doubt, answer no.

<p>To modify existing files, first add the new styles, then customise them to your heart's content.</p>
--

<p>It is possible that other styles will be added to new releases.</p>
--

The Registered Version

The registered version allows up to 1,000 topics, compared to the 30 topics of the evaluation version. That is the only difference between the versions.

Registration

Registration may be paid by any of the following options:

- US\$20 cash or its equivalent in your local currency
- US\$25 cheque (or equivalent in your currency, to cover currency conversion costs at this end), or
- AUD\$28 cheque

I also need to know how to send the registered version to you. E-mail is preferred.

- Transmission by e-mail (no charge). **Mail address needed.**
- Transmission by disk. Please **add the equivalent of US\$5**, and your snail mail

address.

Send cheques/cash to:

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

Unfortunately, credit card facilities are not possible at this stage.

A typical invoice (if you need it) is included as **INVOICE.DOC**.

Report bugs/problems/suggestions to:

**e-mail: roger.hadgraft@eng.monash.edu.au (preferred),
or by fax to: +61 3 565 4944.**

Licence conditions

Payment of the above registration entitles the licensee to **use Doctor Help on one machine at any one time**. Generous multiple licence arrangements can be negotiated.

Updates

Registration includes updates free for 12 months if an e-mail address is supplied.

Features Explained

This section describes the features of Dr Help in some detail. The information on the use of Help macros has been moved to a separate file (HLPMACRO.DOC and .HLP).

Automatic DOC to HLP

The original point of Dr Help was to take a formatted Word file and with no extra effort, produce a Windows Help file. If you have used the standard heading styles, you should have no problems. Simply assign the template HELPFILE.DOT to your document by choosing **File/Template**, then run Doctor Help by clicking on the button.

For example, in one of my courses, I have 2 Word files which contains the course notes. One is about 90 pages, and the other is about 130 pages. I have translated each of these using Dr Help, and now my students can read them on our Novell network as Windows help files.

When you want to go further, the next improvement in your file is the addition of cross references. You can also add various macros for special effects.

Cross references

Cross references are defined by underlining - single underlining (Ctrl-U) for popups, and double underlining (Ctrl-D) for jumps to other topics. Dr Help scans the file for each underlined phrase, and determines its target by looking for a match in its list of titles. If

there are several matches, 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. This can be redirected to an error file.

Predefined jumps (eg. to a topic in another file) and calls to macros (eg. to launch a program) are bypassed.

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 contains "Office". It is entirely up to you how much care you take with these titles.

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

Cross references are **not** case sensitive, and the word may match anywhere within a title. However, the cross reference cannot be longer than the target title. For example, if the target is "cross reference", you cannot use "cross references" as a phrase on which the reader may click. (You could, of course, just omit the last "s").

If you have a title such as "3.6<TAB>Windows 3.1", it can be matched by "3.6" or "Windows 3.1" or some other sub-string - whichever you prefer.

If an ambiguous cross reference has already been defined, then another similar reference will use that in preference to an existing title. For example, if there is already an ambiguous cross reference for "Windows", and "Windows" is used again, then the same cross reference table will be used. Such a match must be exact (ignoring case differences).

It is possible to force exact matching between hotspots and targets. Check the relevant box in the Options Dialog. This is useful if you have many similar titles, and you don't want the reader bothered by ambiguous references.

Popups

Popups are treated in the same way as cross references, except that instead of using double underlining, **single underlining** is used (Ctrl-U). Again, these can be unique or ambiguous.

The only problem with popups is that the window for them doesn't have a scroll bar, so the amount of text you can present is limited. They are really designed for definitions and short explanations. As a result, the number of target references in an ambiguous cross reference is limited to the resolution of your screen. This may be 20 to 40 depending on the resolution. The macro allows up to 50 to be recorded, but I would suggest 10 to 15 is a practical maximum.

Defining context names

In most cases, the types of cross references described above will be adequate. They are easy to define, and the macro will automatically allow you to have ambiguous ones. Sometimes, however, you will want to craft your own links, eg.

- a link to a specific topic in another help file (see JumpContext which is itself a

- topic in another help file), or
- if you want a **graphic** as a button, or
 - if you are using the hot spot editor to define **multiple hot spots** on a graphic.
- A standard link is defined by having a first part (the button part) indicated by single or double underlining. The button component can be text or a graphic. The second part is the context name of the topic. Normally, you don't need to worry about this, because the macro searches for a match between your button text and one or more of the titles. However, if the button is a graphic, this is not possible.

Hence, **you** need to provide the context name. Since context names are assigned automatically by position within your file, I had to provide a mechanism for you to define a context name of your own for topics which are to be accessed in this way. It's very easy.

Simply type the name (no blank characters) at the **start** of the title, and make it **hidden** (Ctrl-H). It will be removed during the scanning process, and that context name will replace the automatically generated one. See DRHLPUSR.DOC for an example (Browse Buttons).

You can use a graphic as a button by inserting it where you want it, selecting it (clicking on it), then giving it a single or double underline. Insert after the graphic a **hidden** version of the context name as you inserted it before the title of the destination topic. Make sure you turn off the underlining after the button graphic, and there must be no space between the button and the context name. (Also be careful not to insert paragraph breaks with hidden style. These can be a real nuisance to find!).

Unambiguous jumps are best handled by making your own link as described above. Handcrafted links are ignored by the macro, and look like this:

Project file

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**. The default project file is called DEFAULT.HPJ. You can edit this to suit your own needs.

DEFAULT.HPJ is a Word document, while all the HPJ files that are created are text files. I hope this is not too confusing, but unless I do it this way, Word insists on interrogating you about converting it from text every time that it's opened, and I figured you'd probably prefer that this was not the case.

Finding Default Project File

You only need one copy of DEFAULT.HPJ. Simply define the path to the file via the Win.ini Options. You can, of course, keep multiple copies if you use different defaults for different projects. Dr Help always looks in the document's directory for the DEFAULT.HPJ file, and only retrieves the one from the WIN.INI PATH if one is not present.

Version of Help Compiler

The default Project File is included with the package. It includes several new options since release 3. 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. This Help Compiler also comes with Borland products such as Turbo Pascal and C++ for Windows, and is also available via ftp from FTP.CICA.INDIANA.EDU as **WHAT.ZIP** (which also contains some tools for working with project files). The version I am currently using is 3.10.504. This version number appears whenever you run the compiler. This version uses extended memory when run in a DOS window.

Parameters

You can change the **icon**, remove **compression**, alter the **title** and the **copyright** notice. The [FILES] section contains the name of the file (which 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.

For most purposes, you will never need to edit a project file yourself. All of the relevant features are accessible through the Options Dialog. However, there are many other options that you may choose to use, and these are defined in the **Help Compiler manual**.

Default project file

The default project file is divided into several sections and looks like a .INI file. (The only exception is the [FILES] section.

Comments start with a ;

[OPTIONS]

```
compress=on
warning=1
report=on
;Try creating an icon for your new Help file
icon=default.ico
;change the next line to suit your file
title=default title
;change the copyright to suit yourself
copyright=© Roger Hadgraft, 1993
```

[FILES]

```
xyz.rtf
```

[CONFIG]

```
CreateButton("btn_up", "&Up", "IfThenElse(IsMark(`L 4'),`GotoMark(`L 3')',`IfThenElse(IsMark(`L 3'),`GotoMark(`L 2')',`IfThenElse(IsMark(`L 2'),`GotoMark(`L 1')',`IfThen(IsMark(`L 1'),`Contents()'))'))")
BrowseButtons()
```

```
CreateButton("btn_annotate", "&Annotate", "Annotate()")
;CreateButton("btn_print", "&Print", "Print()")
InsertItem("mnu_help", "mnu_drhlpusr", "&Doctor Help",
"JumpContents(`drhlpusr.hlp')", 2)
SaveMark("PF")
```

[Map]

Contents 0

[Doctor Help]

```
MaxHeader = 9
ExactMatch = 0
UseOldPhraseTable = 1
RevisionDates = 0
ErrorFile = 0
ScreenUpdate = 0
NewStyles=1
SplitTitle=1
```

Extra Buttons - Up and Annotate

Dr Help defines an **Up** button that allows a user to go up one level in the hierarchy. This is a **really** useful way of moving back through a file, and is a feature not available in other Windows Help files. It is implemented by a special macro in the DEFAULT.HPJ file, and by the insertion of a short macro for every topic. The **Up** button will sometimes get confused. In this situation, the reader may get a "Macro was not found" message. The reader can always fall back to using the **Back** and **History** buttons under these circumstances.

One such case is where you define a sequence of headings like this:

heading 1

heading 2

heading 4

Although Dr Help can cope with this, the Up macro cannot. (It assumes that if it is at level 4, it should be able to go to level 3, which in this case does not exist).

If you already have an old .HPJ file, you may wish to copy the relevant lines from the DEFAULT.HPJ file into your own. Otherwise, simply delete your old HPJ file and a new one will be created for you.

By default, Dr Help adds an **Annotate** button to the Help toolbar to encourage users to annotate the Help file. This just triggers the standard Annotate feature, and is similar to scribbling in the margins. Annotations are stored in a .ANN file in the reader's Windows directory. For example, for this Help file, the annotation file would be DRHELP.ANN. An annotation is indicated by a paper clip symbol on the top left corner of the viewing window.

Keywords

The Title is (optionally) split into separate words and each is added as a keyword. Checking for trivial words (and, the, is, etc) in the list is **not** yet implemented, and probably never will be. If you don't want this to happen, remove the check against Split Title in the Options Dialog.

You can define **additional keywords**. Do this by adding words or phrases after the title, and separated by semi-colons. These words must be **hidden**. A title might look like:

Only the first part will be used for the title, but all words will be entered as keywords. These additional keywords are **not** used in resolving cross-references. That's a possible future development.

Error File

When cross references are being processed Dr Help will often find cross references that cannot be resolved. Its default behaviour is to popup a dialog box to inform you of this fact. However, for long files this is a real nuisance, so you can turn on the Error File feature through the Options Dialog so that all such occurrences are written into filename.ERR. You can browse the file after Dr Help has finished its work. The default is no error file, but if you edit DEFAULT.HPJ, you can define your own default.

Unresolved cross references do not cause problems during compilation. The reader, however, will not be able to select them, so you will want to correct the fault before the final version.

Processing of Fields

Word uses many Fields for special effects. Some of these are not correctly translated into the RTF file, so I have included a macro to process some of them. If there are others that you think that Dr Help should process, let me know.

Index entries

Index entries are simply deleted. I am still rather keen to use them as keywords, but this seems rather difficult at the moment.

Table of Contents

Any Table of Contents is removed. Recently I have thought of generating a global Table of Contents to replace the printed version. This will not be included in release 4, but could occur soon after. Let me know if you're interested.

Autonumbering

Word uses three different Fields for autonumbering of chapters, sections, etc. These are removed.

Symbols

Word uses a **Symbol** field to insert bullets and other special characters. Dr Help translates these into the equivalent character from the appropriate font. (But for some reason, the Wingdings font doesn't work in help files).

Menu Styles

Two new **styles** have been added to the template file, HELPFILE.DOT: "Menu Dr Help" and "Menu Header Dr Help". These give you control over how you want these menus and menu headers to appear. "Menu" items are the many tables of contents that Dr Help writes. A "Menu Header" is the header written for all ambiguous cross references. Edit these styles in HELPFILE.DOT to change them permanently. It was decided to introduce these 2 styles to give you complete control over the appearance of these items. Previously, you could only choose the size and bolding and optionally insert a TAB. Now, you can choose any style you like for them!

These styles will automatically be included in each new file. Existing files should be processed as before (I hope!).


There is a new menu option "Add Menu Styles" to create these two macros in old files. This will work if you are already using a **fixed font size** for the menu items. The default font is **Arial**. You can modify the styles as you prefer. The old settings (Menu Bold, Menu Font Size and Insert Tab) have been removed from the Options Dialog.

If you have used the (very) **old option** of making the menus use the same style as the equivalent heading, then **don't** add the styles to your file. If you do choose this option, you will be asked to confirm that this is what you want to do. If in doubt, answer no.

<p>To modify existing files, first add the new styles, then customise them to your heart's content.</p>
--

<p>It is possible that other styles will be added to new releases.</p>
--

Buttons in the Toolbar

The Doctor Help macro is assigned to the  button at the right hand end of the Toolbar. The other 3 buttons in the Toolbar are (i) RunProgramMacro, (ii) JumpToAnotherFile and (iii) Options Dialog respectively.

RunProgramMacro

This adds the appropriate macro string which will execute a program from within your Help file. It prompts you for the button text and the program name, and allows you to check whether the executable exists by attempting to execute it. (This is not foolproof since Word and Help have different capabilities when it comes to executing programs, but it's better than doing nothing). See also Execute a Program

JumpToAnotherFile

This macro inserts the macro text for either of two types of jumps. It prompts for the

button text and then the target file and the context name of the target topic. To jump to the contents topic, simply press **Enter**. The file **must** exist. See also [Return to my original Help file](#).

Project File options dialog

Several Dr Help options are set in the [Project File](#) because they relate to each document. Some global options are settable in [WIN.INI](#). Doctor Help checks for a project file when it starts. If one doesn't exist, then one is created, and the options displayed in a dialog box for the user to verify. The dialog can be called at any time by clicking on the button on the Toolbar.

Example Settings

Here is an example of the [Doctor Help] section added to each project file. The Options Dialog also includes some non-Doctor Help options, eg. compression, title, copyright, etc.

Please note that the Help compiler will tell you that it does not recognise this section. Please ignore the warning.

[Doctor Help]

```
MaxHeader = 9
ExactMatch = 0
UseOldPhraseTable = 1
RevisionDates = 0
ErrorFile = 0
ScreenUpdate = 0
SaveRate=50
SplitTitle=1
NewStyles=1
Separators=, -():&
```

Title

The default title is "filename.HLP - title of first topic". You can make this whatever you like, but it must not exceed 50 characters. If it does, you will be warned, and returned to the dialog box.

Copyright

The copyright string is added to the **About** box in WinHelp. It is stored in WIN.INI once defined, so you won't have to type the same thing every time.

Compress

Help files are compressed if this option is on. This makes them a good deal smaller, but, of course, it takes longer. **The default is on**. One of my files was reduced from 450 kbytes for the DOC file to 220 kbytes for the HLP file, so it seems to be worth the wait. See also [Use Old Phrase Table](#).

Use Old Phrase Table

The Help Compiler creates a phrase table file to aid with compression. It will use the old one unless you delete it. If you set this option to 0, the old file is ignored each time it runs, and the help compiler creates a new one. **The default is to always use the old file.** For files that change regularly, it is a good idea to **not** use the old file, since compression will be better if a new phrase table is generated each time.

Icon

WinHelp files are allowed to have their own icons. Unfortunately, such an icon is not detected by Program Manager, but is displayed when the user iconifies the WinHelp application while reading your Help file.

A DEFAULT.ICO file is included in the Dr Help package. It is a twist on the standard help icon, made up of the familiar question mark sitting above a Doctor's desk containing a stethoscope. You can, if you wish, create your own default icon file using one of the many icon editors. You can even create a new icon for each one of your Help files (although this is usually very tedious).

Exact Match

This relates to how cross references are detected. By default, ExactMatch is **off**. Turning it on has the effect that cross references must match **exactly** with titles of topics in order to work. (Case differences are ignored however). It is still possible for a cross reference to have several targets, but this should resolve problems with spurious matches. Be careful with topics numbered like:

3.6<TAB>Mating Wombats

because a cross reference now requires an exact match of all characters. Perhaps I could use a third underlining style (words only) for exact matches? Anyone have an opinion about this?

Screen Update

By default, screen updating is off. This speeds up the processing of your file, but doesn't allow you to see changes made except for footnotes and the status line. It is accomplished by calling a macro called **Echo.EchoOff**. If, for some reason, the macro crashes, screen updating (for the current document window) may not be turned on again.

You can turn screen updating on manually by choosing **Tools/Macro.../Echo/Run**

There is a **known problem** when screen update is off (the default) that if you have a very long title line, Word may crash. You can tell Word to ignore the error and it will continue successfully. It has something to do with turning off the screen updating and Word wanting to scroll the footnote pane. If this occurs, either turn screen update **on**, or shorten your title line. This may now occur more often if you define many additional keywords.

Error File

When cross references are not resolved (eg. due to a mis-spelling) a dialog box pops up to inform you of the fact. This is okay for small files, but for large files, this is a nuisance. By checking the **Error File** option, a filename.ERR file will be generated with all the errors. This will allow you to start up Doctor Help before you go to bed on the assumption that all such errors will simply be logged.

Maximum Heading Depth

This specifies the maximum number of levels that you wish to process. In most cases it is not necessary to change it. The default is 9.

Revision Dates

This option might appeal to those of you maintaining Frequently Asked Questions lists, and I am indebted to Tom Haapanen from Waterloo Engineering Software for the implementation code. This option (RevisionDates=1) requires the first line of each topic to be of the form:

title<TAB>date

where **date** is in reverse order (eg. 92-11-27). Dates are entered as **keywords**, so can be searched upon. Dates are propagated back up through the hierarchy, so changes made at leaf nodes are visible at higher level menus.

Split Title

By default, Dr Help splits each title into separate words, and enters each as a keyword. If you don't want this to happen, uncheck this box. Each title then forms a single keyword. You can also define your own keywords to add to this list.

Save Rate

Dr Help performs a save every 50 topics, at the end of scanning the topics, and every 25 cross references. This was found to be necessary to avoid memory overflows in large files. You can change this number to anything you desire.

Separators

When the title of each topic is split into separate keywords (see [Split Title](#)), special characters can be removed. The default ones are ,()-&!?. You can add or subtract from this list. Adding to it slows down the processing of your file. Other candidates are []:{}.

New Styles

The other old options, namely "Menu Bold", "Menu Font Size" and "Insert Tab" have been removed. Although old files should be correctly formatted as before, the definition of the new styles will over-ride the old settings, so it is not necessary to remove them from the Project File.

However, it is suggested that as you recompile old files, you add in the new Menu Styles by choosing Add Menu Styles from the **Help** menu.

Help Menu Options

Some additional macros are available in the **Help** menu in Word.

Run Help Compiler

Run Help Compiler is the second stage of processing of the DOC file. It saves the document in RTF format, calls the UserActions macro, then calls the Help Compiler. When the compiler is finished, it optionally deletes the NEW and RTF files, reloads the DOC file and displays the HLP file. If you retain the NEW file, you can choose Run Help Compiler from the Help menu. This is sometimes useful if you know what you're doing.

Assign Help File Template

I wrote this because I wanted to be able to force documents to use the styles as I'd defined them in HELPFILE.DOT, which is **very** useful for existing documents that may have been created from a different Template. This macro calls **Format/Style** to **force** the styles to comply with HELPFILE.DOT. This is really useful if you're trying to create a large number of help files with standard appearance (as I am). It is also useful for updating old documents to include styles defined in new version of HELPFILE.DOT.

A rather similar, but specialised macro (Add Menu Styles) has been added to the Help menu for adding the Menu Styles to old files.

Add Menu Styles

The **Add Menu Styles** option in the Help menu allows you to add the new menu styles to old Help files. New files will have the styles included automatically.

Win.ini Options

A number of semi-permanent options are settable in WIN.INI. Some of these provide language independence for foreign users. These options are settable using the **Win Ini options** item in the **Help** menu. The options below show the defaults. If your version of Word uses different names for these things, simply insert the new values. Note that the "heading" word must match **exactly** (including upper and lower case) with what your version of Word uses.

Summary

```
[Doctor Help]
directory=c:\windev\drhelp
copyright=© Roger Hadgraft, 1993
heading=heading
contents=contents
pagebreak=^d
IndexEntries=xe
```

symbol=symbol
toc=toc
autonum=autonum

There are also Project File options. These are discussed earlier.

Directory

Dr Help will find its default files (DEFAULT.HPJ, DEFAULT.ICO and DRHELP.INI) provided that you tell it where to find them using this option. Simply enter the directory name here. You can still keep local copies of the first two files if you want.

Copyright

The copyright string is included in the **About...** dialog box within the help program for each of your Help files. If it is not present in win.ini, Dr Help will prompt for it then write it into win.ini when you create your first Dr Help file.

Language

The Win.ini options dialog includes support for English, German and Swedish. Simply select the language, and all the other settings should be correct. If your language is not supported, you will need to set them manually (in WIN.INI). When you have, please mail the settings to me so I can include them in the next version.

Heading

The hierarchical structure of the document is found by looking for paragraphs with a style starting with the word "heading". Change this word to suit your own language version of Word. It must be letter and case perfect.

IndexEntries

All index entries are stripped from the file at the start. In the English version, these are denoted by "XE". Change this to suit your own language version. Open FIELDS.DOC and use **View/Fields** to check your own version of Word. FIELDS.DOC contains all the fields recognised by Dr Help.

Symbol

Symbol fields are not recognised by the Help Compiler. This is a nuisance because Word inserts them for bullets and with the **Insert/Symbol** command. English uses "symbol". Dr Help translates them into normal characters.

TOC

A TOC (Table of Contents) field is removed from the file. (It is not necessary in a hypertext file). I am thinking of replacing it with a global table of contents that a reader could scroll down and click on any topic. It would be indented appropriately.

Autonum

There are 3 different autonumbering fields. They all share some common characters ("autonum" in the English version). Change this to suit your own language.

Other Useful Macros

ModifyFields

This is another internal macro. It deletes some fields as used by Word, and translates Symbol fields into the corresponding characters of the appropriate font. It is editable so you can make your own changes. Please send me copies of any changes.

UserActions

Sometimes, you may find that you want to make some custom changes to the RTF file before the Help compiler is called. There is now a UserActions macro which is called immediately before the Help Compiler. Simply open up the macro, and add whatever you want to it. Thanks to George Mills for this idea.

KeepWithNextOff

I normally setup all my headings to use the "Keep with Next" and "Keep Together" options. This means that the heading doesn't scroll off the page when the user scrolls the topic. Unfortunately, the Help Compiler won't tolerate this option if the topic is to be a popup. Hence, I created this macro and assigned it to Shift-Ctrl-N to make it easy for me to turn it off.

Another option would be to leave off the Keep with Next option on all headings, and only define it for the long ones. The trouble is that you can't always tell when a user is going to scroll the page. More importantly, this is a very useful option in printed documents to prevent headings being left at the bottom of pages.

ProcessHypertextFAQ

This is a simple macro I wrote to process the **alt.hypertext** frequently asked questions file. It will translate any file with a similar structure into a form ready for Doctor Help. It assumes a three level numbering scheme of the form 1.2.3 where the numbers start at the left margin. Many text files, eg. RFCs have this type of format.

The macro could easily be made more general. I have included it to help you convert other text files to Help files. If you create a better version of the macro, send me a copy.

ProcessTextFile

This macro simply prompts for a recognisable string and lets you set all paragraphs containing it to a particular heading style. I've used it with files containing lines like:

Q: How do I translate Word files into help files?

Echo

Echo contains a couple of macros used internally by Dr Help. It allows screen updating to be turned off during processing, which speeds things up by about one third. If the macro crashes for some reason, screen updating may still be disabled. You can re-enable it by choosing **Tools/Macro** then double clicking on **Echo**.

Ini

Ini contains some useful macros for reading and writing INI files. Dr Help uses them for reading and writing HPJ files which have the same structure (except for the [Files] section).

MakeDirectory

This is a little macro that lets you create directories in Word rather than going back to the File Manager. You could move it into your NORMAL.DOT file if you want it available in all your documents. It has nothing to do with Dr Help, and fills what I think is a glaring omission in Word.

DeleteFile

DeleteFile also has nothing to do with Dr Help. I was tired of not being able to delete a file without using the File Manager. You should probably move this macro into NORMAL.DOT so that it becomes global.

Useful Glossaries

Word allows standard text and graphics to be inserted via the **Glossary** feature. Simply select **Edit/Glossary** and choose the item you want, and then press **Insert**.

Return to my original Help file

The [JumpToAnotherFile](#) macro inserts a string that marks the take-off point. Your file can jump back to that location using a special macro which is included as a Glossary item in the Template file:

Simply click on **Edit / Glossary...** and double click on **Return to my original Help file**. The **PF** marker is inserted by the [JumpToAnotherFile](#) macro.

About this Help file

I often insert a small topic describing how the Help file was produced. It also includes a jump to DRHLPUSR.HLP. Simply select **About this Help file** from the **Glossary**, and then **Insert**. You may wish to customise this one since it contains blatant advertising for this product!

Help Macros

Descriptions of the macros which are used in Help Files are now kept in HLPMACRO.HLP. That file also includes some examples. This information was split off because it was fairly self-contained, and was the sort of information that might be referenced fairly frequently, whereas this file may not be.

Explanatory Help file

Dr Help is supplied with a help file called DRHLPUSR.HLP which describes some of the extra features in Dr Help files which are not standard in other Help files. The source is included so that you can customise it. I have made this a separate file so that you don't have to have a copy of it in every one of your Help files, and so that it is separately upgradable. It tries to explain Help files for a relatively inexperienced reader.

Minor features

Removes temporary files

When the Help compiler is finished it asks if you want to delete the NEW and RTF files to save space. If so, it automatically reloads the DOC file. Normally you should answer **yes**, since both of these files will be regenerated when you next run the macro.

Ignores "Page Break Before"

I have included the ability to ignore and remove "Page Break Before" if it has already been inserted for any heading. This removes one more impediment to having a single source document for printed documents (for which a page break before certain headings may be desirable), and help documents, for which it is essential for **every** heading. Hard page breaks are also stripped from the file.

Elapsed Time

If you are using an Error File the elapsed time for the macro, plus the total time (macro time plus help compiler time) are written into the error file. This is only likely to be of interest for large files, so this is not available as an interactive option, thus removing the need to click on yet more OK buttons.

Planned Enhancements

There are a whole lot of things I have in mind. I guess the rate at which these things are done depends on how many people register and request such things!

Cross References to other files

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.

It is already possible to define a jump into another file if you know the context name of the topic to be accessed. See [defining context names](#) to find out how to define a context name for a specific topic.

A spin-off of this approach would be to allow a defined cross reference **within** a file to be easily checked. An additional macro would rebuild the .XRF file without performing the complete compile.

Global Table of Contents

For user orientation, it would be useful if there was one, single Table of Contents which a user could scroll down and choose a topic. This would probably be most useful for medium sized documents with, say, fewer than 100 topics. It would replcae the normal {TOC} field.

Miscellaneous

I would like an option to prevent generation of an initial contents page. This would allow several smaller files to be concatentaed together in a larger project.

I intend to write some macros to allow bitmaps to be easily extracted from an existing file, or to be included in the printed version of the file.

I want to resolve why the Wingdings font doesn't display in Help Files. Any ideas?

The macro should allow optional keywords to also be searched when resolving cross references.

Headings should be prevented from having negative indents.

Helpful Hints

Using a Glossary

I have found it useful in writing several help files to add a "[Glossary](#)" section at the end of the file which is used as a catch-all for all the various popups that I have added to that file. I usually arrange the popup/glossary topics in alphabetical order, and you can see an example in this file. Just remember that the **style** must **not** include the [Keep with Next](#) option. (Don't confuse this with Word's own Glossary).

Naming topics and Cross references

Because of the way that cross references are resolved, you need to be fairly careful about naming them on the one hand, and referring to them on the other. For example, if I name a topic "cross reference" and then want to refer to it as "cross references", the link will be unresolved. Likewise, if I have several topics containing the words "cross references" and refer to them in one place as "cross reference" and in other places as "cross references", then 2 separate [target tables](#) will be built. Now, this is not a real problem in most cases unless the number of target tables got out of hand. It is worth adopting a convention to always refer to a term in its singular or plural, whichever you prefer.

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 you should **not** use a "Keep with Next" on a topic that is to become a popup. Otherwise, you'll only see the Title of the topic. I hope that the next version of the Help Compiler fixes this problem. I assume that this is a bug (in the Help Compiler) and not a feature.

Useful Macros

It is possible to include a range of macros in your Help files. These access in-built capabilities within the Help engine. The full range of them is described in Microsoft's documentation. Here are a few that you might find useful.

Bitmaps

If you intend to include large bitmaps in a Help file, I suggest you store them in separate files, rather than paste them into your DOC file. I have had some problems with the Help Compiler reporting "out of memory" errors on embedded Help files.

Section Breaks

I ran the macro on one of my papers which had a section break in it. For some reason, the function that returns the filename of the current document returns a blank instead. This means that the Run Help Compiler macro will crash. I can't think of any solution to this problem except for removing section breaks. This is done automatically.

Bullets

Bullets don't show up in a Help file. However, with the processing of the SYMBOL field, bullets now **will** work.

Subscripts & Superscripts

Subscripts and superscripts **don't** show in a Help file. Perhaps Microsoft will fix this? If anyone has a work around, please let me know.

Borders

Boxed borders around paragraphs show up just fine in Help files. They are a useful means of highlighting some concepts. There are some examples in this file. Whereas Word merges adjacent boxed paragraphs, WinHelp does not, so things will look slightly different in the help file.

Distribution

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

HELPPFILE.DOT
DRHELP.DOC
DRHELP.HLP
DRHELP.INI
DEFAULT.ICO
DEFAULT.HPJ
DRHLPUSR.DOC
DRHLPUSR.HLP
HLPMACRO.DOC
HLPMACRO.HLP
NEWMENU.DOC
NEWMENU.HLP
BUTTONS.DOC
BUTTONS.HLP
POPUP.DOC
POPUP.HLP
FIELDS.DOC
INVOICE.DOC
HC.PIF

Note that the .HLP file included in the shareware version was generated with the registered version, so that you can see the advantages of registering.

Please do not pass on the registered version!

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.

Mailing List

I have created a mailing list to discuss enhancements to Dr Help. Send a one line e-mail message like this:

sub drhelp fred bloggs

to: **listserv@eng.monash.edu.au**

and, subsequently, send messages to **drhelp@eng.monash.edu.au**.

Glossary

Here I define all those odd paragraphs that are useful as popups.

Ambiguous cross references

An unambiguous cross reference is one which has several targets. This is possible where the phrase being used is found in the Title of several Topics. This macro builds a table of destinations, and presents this table to the reader as a popup. The reader can choose a destination. Once the list of destinations gets over 10, this will be a bit overwhelming, so make sure that this doesn't happen.

Browse

Browse is the ability to read topics in sequential order. The macro creates just one browse sequence from the first topic to the last, so that one of these help files can be read as if it was a book.

Context names

The context name is the code by which the Help system refers to each topic. This macro generates context names automatically.

Cross References

Most hypertext systems have numerous cross references to other parts of the same document, or even to other documents, so that the user is pointed towards potentially useful information.

field

Word uses fields for many purposes, such as Symbols, Indexes, Table of Contents, Formulas, etc.

hard page breaks

A hard page break is inserted using Ctrl-Enter. All of these are stripped out when Dr Help starts up.

heading

Word uses a set of "heading" styles to represent the hierarchical structure of a document. Heading 1 is a top level heading like a chapter. Heading 2 is a section heading, heading 3 is a sub-section, and so on. Dr Help uses these same styles to determine the hierarchical structure.

Help Compiler

You need the Help compiler to generate Help files. This macro writes out the required input file for the compiler. The compiler is included as part of the Microsoft Windows SDK, and other developers, such as Borland, have licensed it from MS, and have included it with their Windows products.

Hidden text

The Help Compiler uses hidden text to represent the destination for jumps and for the text of macros. You can create hidden text by marking it and pressing Ctrl-H.

jumps

Jumps allow a reader to move to another part of a Help file. It is possible to backtrack along a line of jumps using the **Back** and **History** commands. When a jump is executed, the new topic replaces the existing topic in the main Help window. By contrast, a popup appears in a separate window above the main window. Popups are not included in the History sequence, so a **Back** command will not retrieve them.

Keywords

The Windows Help system allows keywords to be defined by the author. This macro adds each title as a "keyword". A later version will probably add the component word separately.

RFC

Request For Comment. Many of the Internet documents are released in this form. They can be obtained from many FTP sites. I have even converted some of them into Help files using Dr Help.

Styles

Styles are used in Word to format paragraphs of text. There is a style called **Normal** for normal text, and there are a set of **heading** styles for defining headings. **Heading 1** is the top level section heading, then comes **heading 2** and so on. This macro uses the **headings** to define the document's hierarchical structure. Note that the title of the document should have **no heading** style. The major sections should be defined by **heading 1**.

target tables

Target tables are the popups built by Dr Help when it finds multiple targets for a cross reference. They are appended to the end of the file but are **not** included in the Browse sequence.

Template

Word for Windows uses Template files (with a .DOT extension) for storing the look of particular documents. They are editable in the normal way, and you can set styles and all sorts of other things to suit the different types of documents that you produce. Templates can also contain **macros** which is how this macro is distributed. Simply copy the HELPFILE.DOT file into your Word directory.

Title

The title of each topic is assumed to be the line which has the **heading** style. The title is

used by the Help system for backtracking, for bookmarks, and for references in the **Search** capability.

Topic

A **topic** is the basic unit in a Help file. It corresponds to a page. You can only read one topic at any one time, but you can branch to see other topics in a variety of ways. Normal topics will have a scroll bar, while popup topics do not.

underlining

Underlining is used to define two types of cross references. Single underlining initiates a popup like this window. It shows above the main help window. **Make sure that the heading style has "Keep with Next" off!** Otherwise, you will only see the title.

Double underlining is used for a true cross reference where the contents of the viewing window changes. The user can always backtrack to the previous screen using the **Back** button. Cross references can be ambiguous.