Doctor Help

A Macro for translating Word files into Windows Help files

<u>Click here for help in reading this Help file</u>.

For New Users - Installation etc

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 <u>Template</u> file (HELPFILE.DOT) for Word for Windows, the macro uses the document's outline structure (defined by the **heading 1, heading 2,** etc <u>styles</u> to build a hierarchically structured Windows Help file (.HLP).

<u>Cross references</u> can also be defined, and <u>keywords</u> are automatically defined. Complete <u>Features</u> 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 <u>registered version</u>.

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.

Printed vs On-line documents

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.

Make sure you give plenty of thought to the **structure** of your document. You will find that any existing printed document will not be in an ideal form as an on-line document. You will find that you will want to rearrange the material, and add new material to make an effective help file. You will find that this will have a useful feedback on your printed document.

The evaluation (shareware) version of the macro is limited to 30 topics. The <u>registered version</u> has a limit of 2,000. 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 evaluation version is fully functional except for this limitation. Both versions are execute only, which means you can't modify the code, although there are a few minor macros that have been left as editable.

Getting Started

The following instructions will get you started. More detailed instructions on using the macro come later under <u>Features Explained</u>.

1.   **Unzip** Doctor Help into its own directory, eg. C:\WINDOWS\DRHELP. (It doesn't matter which directory you use).
2.   Copy the **HELPFILE.DOT** file into your **WinWord** directory.
3.   Copy **DRHELP.DLL** into your **Windows** directory (or **System** directory, or any other directory on your <u>PATH</u>).
4.   **Start** Word for Windows.
5.   Open **EXAMPLES.DOC** and click on the **Doctor Help**   button.
6.   Since Doctor Help is not yet setup, you will be presented with the <u>Win.Ini Options</u> dialog box. This will only happen once.
     Choose the **language** that you are using. Only English, German and Swedish are supported at the moment. If you have need to use Dr Help with another language, please let me know.
     Specify the **directory** where you have installed Dr Help. (The current one is the default). This directory should at least contain **DEFAULT.HPJ, DEFAULT.ICO** and **DRHELP.INI.** (These files can also be kept in the document directory so that you can keep different versions of them).
     Type the **copyright** string that you want to add to all your help files.
     Click on **OK.**
7.   Make sure that you have created a suitable <u>PIF file for the Help Compiler</u>. You should be using HC31.EXE or later. A sample HC.PIF is included.
8.   The macro will create a <u>project file</u> automatically if one is not present (before any processing occurs), and it will allow you to modify the parameters by popping up the <u>Options Dialog</u>. You should just press **OK** for this first test.
9.   Wait for the macro to finish. The <u>Help compiler</u> will then be called. At its conclusion, click on the box to indicate its completion. Your new help file will then be displayed. That's how easy it is!

PIF file for the Help Compiler

Dr Help runs the Help Compiler by calling HC.PIF, which must be in the current directory, or a directory on your PATH. An example PIF file is included. Make sure that the PIF file points to the location of the Help Compiler.

What the macro does

The macro skips through your document one paragraph at a time. It checks to see if each paragraph <u>style</u> is a <u>heading</u> type, and if so, what **level** it is (1, 2, 3, etc). It records the level. It defines each heading paragraph as a <u>topic</u>, using the first line of that paragraph as a <u>title</u>, and (optionally) splits the title into separate words to make a <u>keywords</u> 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 as described later).

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.

Summary of Main Features

1. Dr Help **saves your document** to disk under 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 it makes this part of the job very easy. Your challenge is then to define other cross references, which are also very easy by simply underlining a word or phrase.
3. The macro 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 directory defined in the [Doctor Help] section 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 Doctor Help 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.

Limitation of Evaluation Version

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

In the <u>registered version</u> there is a limitation of 2,000 topics at the moment.

The Registered Version

The registered version allows up to 2,000 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$30 cash or its equivalent in your local currency
- US$35 cheque (or equivalent in your currency, to cover currency conversion costs at this end), or
- AUD$40 cheque

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

- Transmission by e-mail (no charge). **E-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.

User's name and key

In return for your registration fee, you will be provided with a "key" to go with

your name. Enter these 2 items as <u>described later</u>.

Updates

Registration includes **free updates for 12 months** if an e-mail address is supplied. Updates can be purchased at US$10 each or US$20 per year. For those who cannot provide an e-mail address, an additional US$5 per update is payable, or US$10 per year.

Distribution

If you wish to pass the **shareware macro** to others for evaluation, please include **all** the files. The boldface ones are essential. The others you can delete from your disk if you wish.

HELPFILE.DOT	**template file**
DRHELP.INI	**language specifics are stored here**
DRHELP.DLL	**extra macros**
DEFAULT.ICO	**default icon**
DEFAULT.HPJ	**default project file**
DRHELP.DOC	main documentation file (this file)
DRHELP.HLP	**equivalent Help file**
DRHLPUSR.DOC	user documentation file (for readers)
DRHLPUSR.HLP	equivalent help file
HLPMACRO.DOC	information on help macros
HLPMACRO.HLP	equivalent help file
EXAMPLES.DOC	some useful examples
EXAMPLES.HLP	equivalent help file
INVOICE.DOC	invoice for bean counters
HC.PIF	**sample PIF file**

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. Alternatively, the source code can be purchased for US$200 for in-house use only.

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.

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! The items on my list of things to do are described in RELEASE5.HLP.

Changes in Release 5

The changes made from release 4 to 5 are documented in RELEASE5.HLP. If you wish, you can later delete it from your disk.

See RELEASE5.HLP

Working with Doctor Help

To really understand Doctor Help, you need to read the section, Features Explained. It tries to walk you past all the features of the program. This section will get you started quickly, but without an adequate understanding of what is really happening.

New Documents

- Select **File | New** from the menu bar, and then the **HELPFILE** template. If you use the new document button,   you will get a NORMAL document instead. You must use the menu option (unless you have customised the toolbar).
- Create your text and graphics, remembering to use the heading styles.
- Click on the   button on the Toolbar, then click on **OK**.

Existing Documents

- Select **File | Template** from the menu bar, and then the **HELPFILE** template.
- Click on the Doctor Help options button   and choose
- Click on the   button on the Toolbar, and then click on **OK** after you've made any adjustments to the settings.

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.

Unique hotword

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 xref 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.

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 get out of hand. It is worth adopting a convention to always refer to a term in its singular or plural, whichever you prefer.

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 Jumps

Cross references are defined above. Doctor Help makes it very easy to define them. However, there are two instances when you will want explicit cross references or jumps:

1.	jumps to another Help file, and
2.	explicit jumps to topics in the same file.
Such jumps are made very easy by using the Jump to another file button.

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 document Glossary with Word's own Glossary.

Glossary Tabs

Thanks to Hajo ??-Schmitt, included are 2 .SHG files for glossaries, TABS.SHG and BUTTONS.SHG. Each is constructed to expect topics named like glossaryABC, glossary DEF, ... glossary YZ. Include one at the start of your glossary and divide your glossary items into sub-sections. An example is included in EXAMPLES.HLP and .DOC.

Examples of Special Effects

Some special effects and examples are documented in EXAMPLES.DOC and EXAMPLES.HLP. The examples are in a separate file to better illustrate these special effects.

See EXAMPLES.HLP

Hotspot Editor example

DRHLPUSR.HLP now includes a graphic which has defined hotspots using the hotspot editor, SHED.EXE.

Keep with Next

One item only briefly explained in the Help Compiler manual is the use of the "Keep with Next" option for headings. I usually use this for written documents because it prevents stray headings on the last line of a page. In a Help context, "Keep with Next" defines a **non-scrolling region**, containing the title of the topic,

so that it will not scroll off the screen, which is usually a good idea.

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

Keep Lines Together

Keep Lines Together makes for **non-wrapping text**, so for long titles, **do not** use Keep Lines Together. HELPFILE.DOT has now been modified to remove Keep Lines Together from the **heading** styles.

Bitmaps and Graphics

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.

Metafiles (Graphics) in Titles

Dr Help now eliminates metafiles (eg. graphics or equations) from titles. It substitutes a space instead. In release 4, this caused unpredictable results from the <u>help compiler</u>.

Bullets

Bullets (as inserted by Word) don't show up in a Help file. However, with the processing of the SYMBOL field, bullets now work.

Superscripts and Subscripts

WinHelp seems to mess up **superscripts**, by translating them into subscripts (even though the RTF file seems correct). It is suggested that you use the **equation editor** instead. This is a little inconvenient, but does, at least, provide a solution.

Subscripts, however, seem to work fine.

Borders

Boxed borders around paragraphs show up as expected 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.

Colours and shading for borders do **not** work.

Coloured text

Coloured text is supported, and is an excellent way of highlighting important headings and paragraphs. Simply choose the colour you want in the **Format | Character** dialog box in Word.

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, which is available from ftp.cica.indiana.edu in the file WHAG.ZIP.

See HLPMACRO.HLP.

Printing a Doctor Help file

A macro has been added to translate some of the Help specific things into a form more suitable for printing. For example, Double underlining is translated into ***bold italics***, and Single underlining becomes *italics*. This macro (called ReadyForPrinting) has been added to the **print icon** in the toolbar . The macro prompts first to make sure that you want it to modify the Help specific items in the document. Otherwise, it just runs **File | Print** as usual. You can try it out with this document to see how it works.

Features processed are:

Bitmaps

Bitmaps defined with {bm??? ...} instructions are imported. Hotspot files (.SHG) are handled by checking for either a .BMP or .WMF file. Files with other extensions are assumed to be correct. If Word cannot interpret a given file, then an **Error! Cannot read or display file.**§ message is inserted.

Create Index Entries

It also includes the insertion of all defined keywords as **index entries** to match the hypertext document. That is, the same words that are added to the hypertext document to work with the **Search** button, are added to the Index. **UpdateFields** is switched on for printing so that the index will be automatically updated.

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). Another section describes Options, Toolbar buttons, Dialogs, Menu items.

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 <u>heading</u> styles, you should have no problems.

For an <u>existing document</u>, 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 <u>cross references</u>. You can also add various <u>macros</u> for special effects.

WinWord Version 2.0a or later required

If you are still running version 2.0 of WinWord, you **must** upgrade to at least version 2.0a. Version 2.0 has a memory bug that results in an **Out of memory** error on some files. Microsoft will provide such an upgrade at little or no cost.

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 <u>Win.Ini Options</u>. 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 <u>WIN.INI</u> PATH if one is not present.

DEFAULT.HPJ - New Version

DEFAULT.HPJ has been modified to **include** the new **Menu** button, and to make it a HELPFILE type file. This was necessary so that it has access to all the macros attached to HELPFILE.DOT. If you have your own version of this file, make sure you update this version, rather than vice versa.

Version of Help Compiler

The default Project File is included with the package. It includes several new

options since release 4. 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 <u>Help Compiler</u> also comes with Borland products such as Turbo Pascal and C++ for Windows. 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

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

Extra Buttons - Up, Menu and Annotate

Up Button

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.

Menu button

A **Menu** button now allows a reader to browse up through the hierarchy. This is a little like the <u>Up button</u>, except that **Up** always backtracks along the route that you have taken, while **Menu** follows the hierarchy strictly. The Menu button is not used by default, but is selectable within the <u>Project File Options dialog</u>.

Annotate Button

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.

Optional Buttons

With the addition of the Menu button it was necessary to preserve compatibility with old files, and also allow extra control over which buttons would be inserted in the Help file's button bar. A check box for each of the three buttons (Up, Menu, Annotate) has been added to the Project File options dialog. This macro writes the relevant macro strings into the project file. All of this takes a bit of time, (5 to 10 seconds), so be patient.

Keywords

The Title is (optionally) split into separate words and each (except the first) 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 Project File Options Dialog.

User-defined keywords

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.

Full Title as a Keyword

In release 4, it became possible to fragment titles as separate keywords. In release 5, the full title is added **as well as** the separate keywords. This allows unambiguous jumps via keyword by using the full title (see Jump to another file). (It is still possible to have only full titles, by choosing the appropriate option in the Project File Options Dialog).

Modified Keyword List

Doctor Help adds the full title as a keyword, as well as (optionally) each individual word from the title. The first word of the title is no longer added (since it would appear in the right place in the **Search** dialog in any case). This includes single word titles.

Table of Contents

I have added a macro to create a Table of Contents to replace Word's Table of Contents. Each item is a hotspot for the relevant topic. This can help a user to browse through the entire file's contents, rather than just levels of it.

The format of this contents page is determined by your own setting of the **toc** styles, which are now predefined in HELPFILE.DOT.

You can use this feature to generate different styles of Contents page for your help file.

Separated Top Level Table of Contents

A scheme has been implemented to allow a set of separated topics at the top level. This builds on the Table of Contents feature. Three steps are required:

1.   Insert a Table of Contents using **Insert | Table of Contents**, and indicate only **two** levels are to be used. (You can use more if you wish). Normally this would be done below a **heading 1** item eg. "Contents", since this is the topic that the user will see when the file is opened, and also when s/he clicks on the **Contents** button.
2.   Define a context name (eg. "toc") for the **heading 1** item. See EXAMPLES.DOC as an example, and Defining Context Names.
3.   Add a line to the project file in the [Options] section: contents=toc

See EXAMPLES.DOC and EXAMPLES.HPJ to see how to make this work.

Note that the **styles** in which the headings are written are determined by the **toc 1** and **toc 2** styles. Make sure that they are what you want in terms of bolding, tabs, and paragraph spacing.

Index

If an Index is found in a document, an instruction is inserted suggesting that the reader use the **Search** facility in its place. This required an extra item in DRHELP.INI and the Win.Ini Options macro was updated accordingly. If you are printing your Word document, the index will be generated in the same way that keywords are defined.

MAP file

Dr Help produces a .MAP file which contains a mapping of the context names to integers. This is useful when calling WinHelp from within applications via a

        WinHelp(hWnd, lpHelpFile, HELP_CONTEXT, dwData)

call or from a dialog box in *Microsoft Excel*. However, beware that the numbers assigned to topics will change if you change the file. A better way to go is to use a HELP_KEY call, and if necessary, define a unique keyword. You can also define your own context names (see Defining context names)

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 option Add Menu Styles to create these two styles in old files.

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

It is possible that other styles will be added to future releases.

New Styles - toc, index

HELPFILE.DOT now contains default styles for Tables of Contents and Indexes. You can add the default styles to existing documents using <u>Add Template Styles</u>.

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 "<u>Keep with Next</u>" and "Keep Together" options. This means that the heading doesn't scroll off the page when the user scrolls the topic. Unfortunately, the <u>Help Compiler</u> won't tolerate this option if the topic is to be a <u>popup</u>. 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.

Run Matching Help File macro

This macro calls up the matching help file. This is sometimes useful while working on a file.

Show Help

Show Help simple displays this file. It is attached to the   button on the toolbar.

Run Dialog Editor macro

This was mostly written for my benefit so I could call up Word's dialog editor when I needed it.

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 Jump to another File 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 Jump to another File 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!

Buttons

The standard Help buttons are now included in the template's glossary. This can be useful if you want to include them in your own files. Next (>>) and Previous (<<) even come with their own macros.

WinHelp Features

Macros

Descriptions of the macros which are used in Help Files are now kept in HLPMACRO.HLP. 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.

Copy Topic menu item

DEFAULT.HPJ now includes a macro to insert a **Copy Topic** item in the **Edit** menu.

Explanatory Help file - DRHLPUSR.HLP

Dr Help is supplied with a help file called DRHLPUSR.HLP which describes 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.

DRHLPUSR.DOC has been restructured into a **tutorial format**. This relies on drawing the reader into browsing through it. At the end is a diagram that allows the reader to click on various items to refresh his/her memory about it.

Minor features

View Normal

Dr Help now switches page view to Normal mode. It was found that it sometimes got confused if the document was in Page Layout mode.

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.

Negative indents

Negative indents are removed from all styles before the file is processed. This happens because the Help Compiler doesn't understand negative indents. This can take a few seconds for files like this one with many styles attached.

Special Characters

Some characters with ANSI codes between 128 and 255 are not translated correctly into the Help file. These are now replaced with their ASCII equivalents. The ones processed at the moment are ' ' " " —.

Underlined End-of-line

It is very easy to inadvertently insert an underlined end-of-line character. These are automatically stripped.

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.

Options, Toolbar buttons, Dialogs, Menu items

Buttons in the Toolbar

The Doctor Help buttons added to Word's toolbar are:

 Show Help - shows this file.

 Jumps Button - allows definition of jumps and executing program macro.

µ § Dr Help Options - project file dialog box, win.ini options, etc.

 The main Doctor Help macro that performs the transformation. See Features Explained and Working with Doctor Help.

Dr Help Options Button

The options button on the Toolbar µ § that previously only accessed the Project File options dialog now has an intermediate dialog (shown below) which allows access to several options:

- Project File options dialog
- **Edit Project File** runs Notepad to edit the project file directly. Most of the time, this will not be necessary, but occasionally it is useful. You **must** have Notepad in your PATH. This is the case if you have done a normal installation of Windows. This option is particularly useful for editing the [FILES] and [CONFIG] sections of the project file (if you know what you're doing!).
- the Win.Ini options dialog (previously in the Help menu)
- Add Template Styles (useful for standardising attributes across a range of help files)
- Add Menu Styles adds just the specific Dr Help styles, which is handy when you want to change a normal .DOC file into a Help file, without changing all the other styles.
- a **Help** button that gives options-oriented help from this file.

Add Template Styles (Add 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 existing documents to include styles defined in a new version of HELPFILE.DOT.

A rather similar, but specialised macro (Add Menu Styles) is useful for adding the Menu Styles to old files.

Add Menu Styles

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

The Add Menu Styles macro has been removed from the menu bar and moved to the Options dialog. It now correctly adds the Dr Help styles to a document, including creating a project file if necessary. This is particularly useful when you are about to convert an existing .DOC file.

See also Add Template Styles, which differs by allowing you to add all the Dr Help styles, and not just those peculiar to Dr Help. This is useful if you want to have a large number of documents with uniform appearance.

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 µ § button on the Toolbar, and then **Project File Options**.

{bmc projfile.bmp}

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=,-():&
Monitor=0

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.

Processing options

This first group of options determines **how** the macro works, rather than the appearance of the final product. See also <u>Formatting options</u> and <u>Miscellaneous options</u>.

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 or keyword 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. 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. The execution time is also written to the error file.

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. (This has not happened even in some commercial software!).

Compress HLP file

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 <u>Use Old Phrase Table</u>.

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.

Monitor Usage

There is now a second DLL (dynamic link library) which can be used to monitor usage of your help files. When selected in the <u>Project File Options dialog</u> it adds an extra macro to every topic. As the user reads the file, it writes a text file in the current directory. This is part of my research into use of these hypertext files. Apart from this, letting someone try your help file, and monitoring their usage can be helpful in redesigning the layout of the file.

Formatting options

Use Menu Styles

The old menu options, namely "Menu Bold", "Menu Font Size" and "Insert Tab" were removed in release 4. 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 <u>Menu Styles</u> by choosing **<u>Add Menu Styles</u>** from the **Help** menu.

Split Title for Keywords

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.

Exact Xref 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?

Include 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.

Up Button

Add the **Up** button to the help toolbar (default).

Menu Button

Add the **Menu** button to the help toolbar (**not** the default). The Menu button always goes to the current topic's parent, which is different from the **Up** button.

Annotate Button

Add the **Annotate** button to the help toolbar (default).

Miscellaneous options

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.

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 []:'{}".

Icon filename

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 iconises 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).

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.

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.

Win.Ini Options

A number of semi-permanent options are set in WIN.INI. Some of these provide language independence for foreign users. These options are settable using the **Win.Ini options** item in the **Options** dialog. 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.

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 DRHELP.INI for 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.

Index and IndexEntry

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.

If Doctor Help finds an index within your document, it replaces it with a simple reminder to the reader to use the **Search** button to provide the same function.

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.

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

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, and then write it into win.ini when you create your first Dr Help file.

Debug

In WIN.INI you can specify debug mode on or off. When it is on, it will ask if you wish to delete the RTF and NEW files at the end of each run. If it is off, the files will be deleted automatically. Debug=on is the old behaviour.

User's name

When you have registered your copy, you will be provided with a user's name, that you should enter in this box. Make sure you type it **exactly** as it was supplied, including matching case and spaces between words.

User's key

To go with your User's name is a user key, which you enter in this box. Type it exactly as it is supplied to you.

Defaults

[Doctor Help]
directory=.
copyright=© Roger Hadgraft, 1993
heading=heading
contents=contents
pagebreak=^d
index=index
indexEntry=xe
symbol=symbol
toc=toc
autonum=autonum
There are also Project File options. These are discussed elsewhere.

Jumps Button

The third button from the left in the Toolbar   is now used for:

- jumps to other files
- macros to execute programs, and for
- checking the next popup or jump in this file (see Check next Jump and Check next Popup)

An intermediate dialog box allows you to choose between them. Context sensitive Help is available.

{bmc jumpdlg.shg}

Jump to another File

Jump to another File is a new way to define links to other .HLP files (as well as the current help file). It accesses the .MAP file for that .HLP file, allowing you to browse through the topics until you find the one you want, and includes a **Help** button to show this information. Click on **Cancel** to abort the process.

{bmc jumpdlg2}

You are required to define **3 parameters**. You don't have to use the **Choose** buttons. They just make it easier for you. You can just type in the name of the file, and/or the context or title names.

Button text

This text which will be underlined, and which the reader will click on. If you have already entered the button, simply leave blank. If you highlight existing text before you use the Jumps button, that text will be copied into the button field. The text is then copied back into the file when you click on **OK**.

Filename

This is the file containing the topic required. The existing file's name is automatically inserted. Click on the **Choose** button to bring up a dialog box to find another file. The .HLP file must exist, and there **must** be a corresponding MAP file to match.

Destination

The destination is the required topic. Click on **Choose** to search the .MAP file for the file you chose at the last step. It will prompt for a string to search for if you have not entered something for the button text. (You usually have some idea of what the topic is called). You can search up and down through the .MAP file. When you've identified the topic you want, click on **OK**. (Check next Jump is used to do the searching).

Both title and context name are copied back into the Jump dialog box. You can choose which one you will use for the jump. If the context name has been defined by you (ie. it's **not** created by Doctor Help - something like card35), then use the

context. Use the title when you know that it's **unique**. Simply click on the appropriate option button, then on **OK**.

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:

Run a Program

This adds the appropriate macro string, ExecProg, which will execute a program from within your Help file. It prompts you for the button text and the program name (which you can select using the **Choose** button), 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).

Your reader must have a copy of DRHELP.DLL in his/her PATH (or the directory containing the Help file). WinHelp contains its own ExecProgram routine, but it is of limited capability since it understands nothing about file associations.

Execute Programs DLL

I have built a DLL (DRHELP.DLL) to execute programs in the Windows style, ie. you only need to give the document name, and it will look in WIN.INI for the application. This is particularly useful if you are using the same help file on many different systems (with applications installed in different directories). A RegisterRoutine command has been added to DEFAULT.HPJ. For old Project files, you will need to add this line manually to the [CONFIG] section:

RegisterRoutine("drhelp.dll", "ExecProg", "v=Si")

The name of the routine is **ExecProg** (as opposed to the built-in function which is ExecProgram - I tried using the same name but it didn't work):

!ExecProg(drhelp.hlp,1)

The values for the integer parameter are as follows:

    0   hide
    1   normal
    2   minimise
    3   maximise

These match the parameters used by Windows in its ShowWindow function, but are **not** the same as WinHelp's ExecProgram (where 0 indicates a normal window).

I expect other routines will be added to this DLL in due course. It **must** be in a directory on your PATH, eg. the Windows or Windows System directory, or in the same directory as the help file.

The DLL will also handle full pathnames, and will change to the correct directory before executing the application, eg.

    ExecProg("c:\\infosys\\ar-r\\winhelp\\chapters.hlp", 1)

In this case, **chapters.hlp** will be opened, and its default directory will be c:\ infosys\arr\winhelp, which is handy when that help file points to other help files which it expects to find in the current directory. This is useful for launching another application and making sure that its home directory is appropriate. **Note that you must double all the backslashes**. Sorry about that, but it seems unavoidable.

Both relative and absolute paths may be used. The following will also work:

    ExecProg("winhelp\\chapters.hlp", 1)

Check next Jump and Check next Popup

These macros scan the matching .MAP file to see how many matches exist for a given defined jump. Just click **before** the defined (underlined) hotspot, and CheckJump will locate the next underlined term. It opens the .MAP file (after you confirm it is a Text file - I really wish Microsoft would fix that), and starts scanning for the phrase given. It reports if no match is found. If you keep clicking on **Next**, it will scan for the next occurrence. Press **Previous** to search backwards, or **Cancel** to stop. After each search, you have the option to redefine the search string.

Use the Jump to another File option in the Jumps Button to choose the destination for a jump.

Help buttons

Doctor Help now has access to the standard WinHelp call in Windows. This has allowed me to build **Help** buttons into its dialog boxes, such as the Project File Options Dialog and Win.Ini Options. It brings up the relevant pages in this document.

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.

Open Text File

This is a useful macro for importing text files (eg. off the Internet), and doing some initial formatting.

Run Hotspot Editor

This option runs Microsoft's Hotspot editor, SHED.EXE. It is used for defining hotspots on graphics.

About Doctor Help ...

This pops up a dialog to inform you what version is being used.

Miscellaneous Dialog Boxes

Minimise Word Window ?

Minimise Word Window ? (It runs more slowly, but you can work on other things.)

This prompt occurs as soon as the macro starts to run. Sometimes you will want to minimise Word, and sometimes not. It is not a setting that can be stored in WIN.INI or the Project File. If you think it should be, let me know.

This lets the macro run in the background, and lets you get on with other things. Of course, it will take longer to run, because it is multitasking, rather than having almost exclusive use of the machine.

Delete RTF and NEW files?

This prompt only comes up if you set debug mode in the Win.Ini Options. It was the default in previous version, but is now suppressed by default.

Cancel Button in "Unresolved Cross References" dialog

A **Cancel** button has been added to this dialog box so a user can quit out of the compilation process. It also optionally reloads the DOC file. The change was made to the generic **ErrorMsg** routine, so all error messages inherit the **Cancel** option.

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 is 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.

Contents page

The contents page is the page which the reader sees when the help file is opened, or when s/he presses the **Contents** button. By default, it is the topmost topic, but with the new Separated Top Level Table of Contents, it is possible to use a

different topic, with a layout different from the default.

Context names

The context name is the code by which the Help system refers to each topic. This macro generates context names automatically, but you can define your own (see defining context names).

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" as well as each word of the title.

Notepad

Notepad is the standard Windows text editor.

PATH

The DOS PATH allows Windows and other applications to find other executable files. Doctor Help looks for HC.PIF on the DOS PATH, for instance, and ExecProg looks on the PATH for executable files.

Popup

A Popup is a separate window which popups up above the main Help window to display small amounts of additional information. Popups are particularly useful for short definitions, like this one.

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 **heading** styles to define the document's hierarchical structure. Note that the title of the document should have either a **non-heading** style (eg. "Top header"), or it should be a heading number one less than the major sections.

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 (like this one) 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. CrossXE "Cross"§ references can be ambiguous.

Index

μC

Cross, 27