

TRUEMTS001yesyesyesThis file _ Edward Guy, CDP
1993/4yesnononoyesyesyesyesHelp for HELLLP!yesyesyeshellp!

Table of Contents

[Main Title Screen](#)
[Getting Started With **HELLLP!**](#)
[Using **HELLLP!** to make a HELP file](#)
[The **HELLLP!** Tool Bar](#)
[To compile your Help file](#)
[To try out your newly compiled help file](#)
[Tables of Contents](#)
[The structure of a Help File](#)
[Colours you can use](#)
[Changing Topic Title Style Colours and Fonts](#)
[Title Bar Text Setting](#)
[User Copyright Line in WinHelp ABOUT box](#)
[Using **HELLLP!** and Winhelp to produce corporate manuals](#)
[Add Help screens to your programs, templates and macros](#)

HELLLP! for the Expert User

[Hotspot graphics and the SHED editor](#)
[Compression of Help File](#)
[HELLLP! from a network server](#)
[Error Messages and problems](#)
[Multiple Project File Builds](#)
[External Jumps](#)
[Mapping Context Strings to Context Numbers and HelpContextID for Applications](#)
[Adding Sound by Playing *.WAV files](#)
[Spawning an External Process from a Help File](#)
[Setting Help Window Size and Screen Position](#)
[Choosing an Icon for the help file](#)
[Choosing the Capitalization \(Case\) of Winhelp Search Keywords](#)
[Button Selection and Control](#)
[Smart Quotes](#)
[Non-Scrolling Region](#)
[Secondary Windows](#)
[Background Colours](#)
[Using non-English versions of Microsoft Word](#)
[Use of WinHelp Macros](#)
[WinHelp Help Compiler Limits](#)
[Launching Other Help Files](#)
[Project Accounting](#)
[Bullets and Symbols](#)
[HotSpot Colours](#)
[Heading styles to pick topics](#)
[Mid-Topic Jumps](#)
[Compuserve Registration Instructions](#)
[Using 256 colour bitmaps](#)
[Glossary Button Box](#)
[Importing files](#)
[Accelerator Keys](#)
[Adding Menu Items](#)
[Fancy Macro Demonstration.](#)
[The SHAZAM! effect](#)
[Customising lines in the HPJ file \(for experts\)](#)
[The symbols +#!K\\$](#)

About Shareware

[Using a copy of someone else's registered copy](#)
[Registration and Payment](#)
[**HELLLP!** order form](#)
[Credit Card Orders](#)
[Registration and Payment via CompuServe](#)
[Upgrade Policy](#)
[Version History \(What's New\)](#)
[RACONTEX Full Text Search Product](#)

The items below are intended as PopUp boxes.

The Up Arrow button

The Down Arrow button

The Curved Arrow button
The Spectacles button

Creating a Topic The T button

Setting the first screen to display The I button
--

Browse Sequence The B button
--

Creating a Hypertext Jump The J button
--

Creating a Pop Up Box ("Definition") The P button

Context Labels (Context Strings) The C button

DISCLAIMER

[The size refresh button](#)

[Acknowledgments](#)

[Hypertext](#)

[Bienvenue a HELLLP!](#)

[Opening Window Size](#)

[Topic Window Size](#)

[SHED incorporation PopUp](#)

[Pop Up Box for Opening Title](#)

[Question Mark Button](#)

[Test Sound](#)

[Include Sound](#)

[Exit Sound](#)

[Change Topic Title 1](#)

[Change Topic Title 2](#)

[Change Topic Title 3](#)

[Change Topic Title 4](#)

[Change Topic Title 6](#)

[Welcome to the world of HELLLP!](#)

[size of secondary windows](#)

[Secondary Windows \(example\)](#)

[Customising lines in the HPJ file \(for experts\)](#)

[Secondary Windows](#)

[Strange Coloured Window](#)

[Background Colours](#)

[SideBar Table of Contents](#)

[Choosing an Icon for the help file](#)

[Upgrade Policy](#)

[SelectPictureToInclude](#)

[Title Bar Text Setting](#)

[Using non-English versions of Microsoft Word](#)

[Jump to RRS](#)

[Glossary Alpha Buttons](#)

[Glossary External File](#)

[Glossary Make Skeleton](#)

[Glossary Insert Box](#)

[Glossary Settings Only](#)

[Glossary Cancel](#)

[Glossary Help](#)

[About HELLLP!](#)

[Menu PopUp](#)

[Pop Up Box](#)

[Macro Demo Popup](#)

Help file produced by **HELLP!**, a product of Guy Software, on 14/04/95



SideBar Table of Contents

(An example of a secondary window)

[Main Window Title Screen](#)
[Main Window Table of Contents](#)
[Getting Started With **HELLLP!**](#)
[Using **HELLLP!** to make a HELP file](#)
[The **HELLLP!** Tool Bar](#)
[To compile your Help file](#)
[To try out your newly compiled help file](#)
[Tables of Contents](#)
[The structure of a Help File](#)
[Colours you can use](#)
[Changing Topic Title Style Colours and Fonts](#)
[Title Bar Text Setting](#)
[User Copyright Line in WinHelp ABOUT box](#)
[Using **HELLLP!** and Winhelp to produce corporate manuals](#)
[Add Help screens to your programs, templates and macros](#)

[HELLLP! for the Expert User](#)

[Accelerator Keys](#)
[Background Colours](#)
[Browse Sequences](#)
[Bullets and Symbols](#)
[Button Selection and Control](#)
[Compression of Help File](#)
[Error Messages and Problems](#)
[External Jumps](#)
[Glossary Button Box](#)
[Heading styles to pick topics](#)
[HotSpot Colours](#)
[Hotspot graphics and the SHED editor](#)
[HPJ file Customisation](#)
[Icon choice for the help file](#)
[Importing files](#)
[Keyword Capitalization](#)
[Launching Other Help Files](#)
[Mapping Context Strings to Context Numbers](#)
[Menu Support](#)
[Mid-Topic Jumps](#)
[Multiple Project File Builds](#)
[Network server use of **HELLLP!**](#)
[Non-Scrolling Region](#)
[Project Accounting](#)
[Secondary Windows](#)
[Smart Quotes](#)
[Sound by Playing *.WAV files](#)
[Spawning an External Process from a Help File](#)
[Symbols +#!K\\$](#)
[Using 256 colour bitmaps](#)
[Using non-English versions of Microsoft Word](#)
[Window Size and Screen Position](#)
[WinHelp Help Compiler Limits](#)
[WinHelp Macros](#)

[About Shareware](#)

[Using a copy of someone else's registered copy](#)
[Registration and Payment](#)

[HELLLP! order form](#)

[Credit Card Orders](#)

[Registration and Payment via CompuServe](#)

[Upgrade Policy](#)

[Version History \(What's New\)](#)

[RACONTEX Full Text Search Product](#)

HELLLP!



The Windows™ Help Authoring Tool

Takes care of all the drudgery and makes authoring Windows Help Files an enjoyable and rewarding experience.

© Edward Guy, CDP 1993/4 All Rights Reserved

Allows you to use color (or colour) to make sure that text you wish to emphasize gets read by the users.

Different sizes and fonts may be used to give additional visual interest.

To get help on the following topics, just click the mouse pointer on the text that is [THIS COLOUR](#).

[Getting started with HELLLP!](#)

[Using HELLLP! to make a HELP file](#)

[The structure of a Help File](#)

[Adding Sound to a Help file](#)

[Main Table of Contents for HELLLP!](#)

[SideBar Table of Contents](#)

[Using HELLLP! to produce on-line corporate manuals](#)

[Add Help screens to your programs, templates and macros](#)

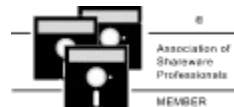
[Registration and Payment for HELLLP!](#)

[If this is a copy registered to someone else](#)

[About the SHAREWARE marketing concept.](#)

[Disclaimer](#)

[Acknowledgments](#)





Getting Started With HELLLP!

The setup sequence is critical for successful use of the system. You might want to print this topic for reference (click the Print key on the button bar).

1. Set up a subdirectory for your project files, we suggest C:\HELPCOMP\PROJECT.
2. Install the Microsoft Help Compiler - for a source of this see below. Note its fully qualified pathname (e.g. C:\HELPCOMP\COMPILER\HCP.EXE).
3. Install the template file HELLLP!.DOT and the help file HELLLP!.HLP in the same subdirectory as your other Word for Windows templates.
4. Within Word for Windows select File, New and choose HELLLP! as the template.
5. Respond to the Setup questions on first use.
6. When prompted either type your help text or import it with Insert File.
7. Ensure that you can see Word 2.0 pattern Toolbars (VIEW TOOLBARS check against Word for Windows 2.0).
8. Follow the instructions in the topic "[Using HELLLP! to make a Help File](#)"

HELLLP! is designed to be as easy to use as possible, while still allowing the expert user to use it to "jump start" the creation of complicated help file projects. For a simple project you will probably only need to accept its suggestions in the choices it offers you in order to achieve success.

HELLLP! is all contained in the template **HELLLP!.DOT** which you copied to the directory in which you keep your templates, and it has its own HELP file which you are reading now, and which may be invoked from the Help menu item, the Help button on the initial setup screen and the Question Mark button on the toolbar.

If you are storing your templates on a shared network drive, please review the topic "[HELLLP! from a Network Server](#)".

If you have the environment variables **TEMP** or **TMP** defined, they must point to directories in which you have write privileges. They should not point to RAM disks unless these are very large since Word and the Microsoft Help Compiler write some large temporary files to them.

In order to do its job, though **HELLLP!** does need to find out a few things about your system and your preferences.

During initial setup, **HELLLP!** asks for a directory name in which to save the files it creates during its operation.

To actually produce a HELP (HLP) file, **HELLLP!** needs access to one of the Microsoft™ help compilers. These have names like HC.EXE or HCP.EXE, or HC30.EXE or HC31.EXE or something similar.

One of these compilers is included in most tool kits sold for producing Windows programs. Updated versions are also available from the Microsoft software library on CompuServe. The HC300B version of the compiler (which writes HELP files for Windows 3.0) was available as the file S13290.ZIP and the HCP version (uses extended memory for speed, and writes for Windows 3.1) was file S13436.ZIP in August 1993.

If using version 6.0 of Word for Windows you will need a version of the help compiler which is compatible with RTF files produced by that version. As of January 1994, a suitable version of the compiler was available for download from the Microsoft Windows SDK forum on CompuServe (GO WINSKD) as HCP505.ZIP in the WinHelp section. It is available as HCP505.ZIP or HCP505.EXE (self extracting) on various other services. For Internet users it is on ftp.microsoft.com (and its mirrors) in ../softlib/mslfiles. Should a later version of Word require a different version of the compiler, do a filename search of your favourite service for the name fragment **HCP**.

During the setup dialog, you are asked for the name and location of the compiler you wish to use. Please provide the full path name e.g. **C:\HELPCOMP\COMPILER\HCP.EXE** if that is what you are using. If you do not have a copy of the compiler, **HELLLP!** will still work, but only to the point of producing the input files that the compiler needs. You may take these to a computer that has the compiler on it.

If you later wish to change any of your initial settings, for example to use a different version of the Microsoft Help Compiler, you may do so with the "Change HELLLP! Setup" option on the Format menu. If you change your project directory but wish to work on an existing project you should move **ALL** files associated with it (they all have the same filename but different extensions) to the new project directory.

There will be files with various extensions in your project directory, all with the same name as your project but with different extensions. They are all maintained by **HELLLP!** and you should not edit them unless you are an expert. You can do everything that you need from within **HELLLP!** using the *.DOC files.

If you are using an unregistered (that is **not paid for**) version of **HELLLP!** you will get a registration reminder popping up when you open a file produced by **HELLLP!**. You are entitled to 30 days free trial before either paying for it or ceasing to use it. For details of how to do this see Registration and Payment for **HELLLP!**.

The help file you are reading was built using HELLLP!, so anything it does, you can do! In fact some of the things in it were done to demonstrate the effects you can achieve. You probably should not include all of them in your help file - it could detract from your application.



HELLLP!

from a network server

You may use **HELLLP!** from a network server provided you take the same steps as you would with any other template located on the server and provided you pay for the correct number of users after your 30 day free trial period.

You will not be able to make any changes to the template while someone else is using a document attached to it.

This means that you should only do the following when no-one else can access the template:

- Change registration information
- Change the Help compiler name or location
- Change the project directory path

since these all require the template to be re-saved and this cannot be done when two users have it open in shared mode. Of course they may be performed only from a user ID which has write access to the template.

The full path name of the compiler must be one that every user can access, as must the project directory path. The users do not need to access the same directory, just one with the same name. e.g. if the project directory path name is C:\HELPPROJ then every user machine would need to have a \HELPPROJ subdirectory on its C: drive.

If you find that you get error messages when compiling on a network drive, the Help Compiler is probably trying to use the root directory of the network drive for its temporary files and you do not have write privileges to the root. Edit your AUTOEXEC.BAT file to SET a TMP variable just like the TEMP variable you probably already have. The Help Compiler will then use the directory TMP points to for its temporary files. Make sure the directory exists and the drive where it exists has sufficient free space.



Using HELLP! to make a HELP file

Windows Help presents "TOPICS" (or Screens) to the user based on searches, the browse buttons (<< and >>) and jumps. To create these screens, various codes have to be inserted into the file containing the text. **HELLP!** creates and inserts these codes for you. If you are an expert, you may use **HELLP!** to do a "first cut" and then fine tune the result yourself before compiling. For most people, **HELLP!** will do it all for you with no hassle.

The process is controlled by the lettered buttons on the toolbar.

To ensure that everything is set up properly, it is necessary to create a file by doing a FILE NEW and choosing **HELLP! as the template. Once that file is created and saved, you may open it like any other file. If you are working with an existing document file, you will be prompted to insert it at the appropriate place. If you just attach your document to **HELLP!.DOT** using FILE TEMPLATE, it will not work properly.**

If you are **converting a formatted manual into a help file**, it will probably be more effective to have the manual DOC file open in another window and carry the text for the topics via the clipboard from it to the file you are creating in **HELLP!** since the manual may well contain footnotes, footers, headers etc. which will not fit well into a help file.

To turn a section of text in your file into a "topic" which will become a screen, just place the cursor (the "insertion point" as the MSWord documentation calls it) in the first line you want included and press the T button. You will be asked for a title for the topic, and will also indicate via a check-box whether you wish the title to appear emphasized at the top of the topic screen. If there are words in that first line which you would like to use as the topic title, select (highlight) them before you press the T button and **HELLP!** will present them as a proposed title, which you may then edit (e.g. to capitalize letters).

If you intend to set non-scrolling topic titles, do not ask HELLP! to put the emphasized title on any topic which you intend to use as a PopUp (if you do, only the title will pop up).

Notice that **HELLP!** inserts a page break ahead of the topic. This is the only circumstance in which there should be a page break, since a page break is what defines a new topic. Text following a page break that is not set up as a topic will never be seen by your users.

HELLP! will also take the words of more than one letter in the title and present them as proposed "keywords" to be used in WinHelp searches. You may edit these or add or delete words.

HELLP! also takes care of "Context Strings", the means by which the Help Compiler identifies topics. **HELLP!** will create a unique context string for each topic and keep track of it. You do not need to remember it because when you set up your Hypertext jumps, you will be selecting from a list of topic titles.

You will always be given the opportunity to override the automatic choices which **HELLP!** suggests.

Go through your text in this way until you have marked all of your topics. At this stage, you may compile your help file and it will work. It will have a single table of contents with all of your titles listed and you may jump to them as with any other help file. You may test your newly compiled HELP file by clicking on the "Spectacles" button on the toolbar.

However, to make the file really useful you will probably want to go through it again and add Hypertext jumps and PopUp boxes using the J and P buttons. Although the same topic can be reached by both PopUp and Hypertext is unwise to do this, since there may be side effects. In particular, if you have **HELLP!** incorporate a title at the head of a topic to be used in a PopUp, and then set non-scrolling topic titles only the title will pop up!

When HELLP! opens a file it turns on the SHOWALL setting (the paragraph mark in the ribbon). Please do not turn it off since HELLP! needs it in order to find the context strings used in jumps and popups.

You have the option to set up "Browse Sequences". This just means that the << and >> buttons will be able to page your help file from topic to topic.

For more sophisticated applications, you may create your Help file in several sections, you may assign context numbers for use in other applications (called HelpContextIDs in Visual Basic), you may add sound effects and you may start other processes from your Help file to give your users a true **Multi-Media Help Experience**.

If the file you imported has headings set up using Microsoft Word heading styles, you may instruct **HELLP!** to use these as topic titles automatically.

If you imported a file which already has context strings defined in it, you may use them as explained in the topic Importing Files.

In every process, sometimes things go wrong. Some of the possible settings in Word for Windows may conflict with what you want to do in your help file. The topic [Error Messages and Problems](#) will give some pointers to what to do if something strange happens.

Tables of Contents

HELLLP! generates a table of contents in the sequence in which topics are identified. It uses the names used during topic definition and inserts the Hypertext Jumps to them automatically. This table is used by **HELLLP!** to keep track of all of the context labels (known in the Microsoft manuals as context strings). You are taken to this table during the generation of hypertext jumps to pick the place you want to jump to, or the text you want in a PopUp box.

It is recommended that you minimize any editing changes you make to this table of contents. You may move items around, change the indentation, the size, the font color but do not change the text as to underlining or hidden/not hidden because these will prevent the compiler from working. You must also move entire lines or groups of lines, not parts.

The standard table of contents will appear in your help file unless you override it with the I button. You may include your own table of contents, just like a regular page and have it displayed when Windows Help is invoked.

You may set up your own table of contents in a secondary window, just like any other topic.

For smaller projects you may choose to use the Set Button Selection function on the **Format** menu to generate *automatically* a copy of the automated table of contents and display it in a secondary window (which you must define first) or in a PopUp window. This window may have **Close** and **Exit** buttons included in it if you set up a topic button bar in the automated table of contents. If you choose to use a secondary window for this purpose, you may have it display beside or instead of the main window when WinHelp starts your help file using the function provided by the [I] button. The use of a PopUp windows is advisable only for small projects, since a PopUp window will not scroll.

You may replace the default heading **Table of Contents** by changing the text in the template glossary using instructions contained in the topic Using Non-English Versions of Word.

Context Labels (Context Strings) are the means by which topics are identified. **HELLLP!** generates them automatically, guaranteeing that they are unique. However a user has the opportunity to override the automatically defined label.

HELLLP! uses a three letter/three numeral scheme taking the initials of the first three words in the title (padding with underscores if there are less than three words) and adding a three digit number to identify topics with the same three initials. The number starts at 001.

The automatically generated table of contents is TOC000.

To ensure that context strings are unique for multiple project file builds, the project filename (less any characters not legal in a context string) and a period are prepended to the string (for example this topic would have a context of **HELLLP.CLC001**).

Although not needed by the HELP compiler, **HELLLP!** inserts the contexts as hidden text in dark red ahead of the topic title (if any) or text for the convenience of those users who need to know their values.

The automatic table of contents generated by **HELLLP!** provides a ready cross reference list of topic names and their context strings.

Another list of the context strings is maintained in an INI file, for use by those who need to ensure that particular values of context string are not used with the current file (some applications are produced from multiple files). This INI file provides the means to "busy" the context strings to avoid their re-use and is also used by the External Jump feature of **HELLLP!**

The symbols `+#!K$` appearing at the start of each topic are codes used by the help compiler and should not be edited unless you **really** know what you're doing.

The HELLLP! Button (Tool) Bar



You may click on one of these buttons for help about its function.

The **HELLLP!** button bar is a customized version of the regular Word For Windows 2.0 toolbar with some buttons added in place of those you are not so likely to use in making a help file. The actions commanded by the ones omitted may still be performed by the pull down menus. **If you cannot see these buttons on your toolbar, please ensure that you have Word for Windows 2.0 selected as your toolbar.**

Where the user has options associated with the buttons on the button bar (tool bar) they are presented after the button has been clicked.

The Up Arrow button and The Down Arrow button move you quickly to the table of contents, or to the last place in which you added a topic, jump or PopUp.

T Will create a "Topic".

J Will create a hypertext jump from the highlighted text to a topic of the user's choice.

P Will create a PopUp box containing the text of the desired topic.

C Used only in conjunction with the P and J buttons. Used to identify the objective of a hypertext jump or PopUp box.

The Speaker Button incorporates sound into the topic.

The Screen Button permits an external process to be started by the topic (e.g. multimedia effects). This button also permits you to call Winhelp macros.

I Used to determine which topic will be displayed first when the file is opened by Windows help.

B Will create automatic browse sequences.

The Curved Arrow button ("Give it a whirl") prepares the files for the compiler, then calls the compiler to produce a help file that you can test.

The Spectacles button lets you "take a look" at your newly compiled help file.

? Brings up the Help screens for **HELLLP!** (the ones you are looking at). An alternative to using the menus.

T Will create a "Topic", which involves generating a topic title from the text currently highlighted (or entered from the keyboard if nothing is highlighted). The title will appear as a result of a Windows help search whenever any of the words of more than one letter in it are entered as keywords. The title will also be used in **HELLLP!**'s automated table of contents. The user is given the option to edit the automatically generated keywords, adding new ones or deleting any not desired. The user has the option to have the title typed or selected included at the head of the topic in the help screen, in addition to the table of contents and search screens. When the box selecting this option is checked, the title will appear in the style selected as Topic Title in the style box. The topic title style may be changed using the **Change Topic Title Style** function on the **Format** menu, which includes an option to make the topic title non-scrolling.

J Will create a hypertext jump from the highlighted text to a topic of the user's choice. After highlighting some text and pressing the J button, the user is taken to the **HELLLP!** automatic table of contents to identify the other end of the JUMP. The user places the cursor in the table of contents entry for the desired end of the jump and presses the C button. All work is done automatically by **HELLLP!**

(The cursor may in fact be placed in any double-underlined text which is followed immediately by dark red, dotted underlined, hidden text, or directly in any dark red, dotted underlined, hidden text which directly follows underlined text - you are taken to the automatic table of contents for your convenience because all topics are listed there).

The J Button also provides the option to Jump to a topic in another project file using the External Jump button.

You may also jump to a named MidTopic Target if you wish to enter a long topic other than at the top.

P Will create a PopUp box containing the text of the desired topic, instead of a hypertext jump to it. After highlighting some text and pressing the P button, the user is taken to the **HELLLP!** automatic table of contents to identify the topic to put in the PopUp box. The user places the cursor in the table of contents entry for the desired end of the jump and presses the C button. All work is done automatically by **HELLLP!**
(The cursor may in fact be placed in any double-underlined text which is followed immediately by dark red, dotted underlined, hidden text, or directly in any dark red, dotted underlined, hidden text which directly follows underlined text - you are taken to the automatic table of contents for your convenience because all topics are listed there)
The J Button also provides the option to Jump to a topic in another project file using the External Jump button.

Text to go in a PopUp box may also be accessed via a hypertext jump, because it is a topic like any other. It is a good idea to put text which is intended only as PopUp text at the end of the file since it may also be accessed by the browse buttons. Alternatively, text for PopUps may be placed in separate file and compiled without browse buttons. The text may then be incorporated using the External Jump feature offered as an option when the **[P]** button is clicked..

C Normally used only in conjunction with the P and J buttons to identify the objective of a hypertext jump or PopUp box. When you have placed the cursor (the "insertion point") in the double underlined title of the topic you wish to jump to or pop up, you click the **C** button to allow **HELLLP!** to process the context string. Instead of putting the cursor in the title, you may put it in the dark red dotted underlined text ahead of a topic title indicating the context string. If you wish to include a jump to the automatic table of contents you must use this method, since it does not appear in its own list. For your convenience, the **P** and **J** buttons position the screen to the automated table of contents, and the **C** button restores it to where you were.

What the button actually does: If the insertion point is in hidden text, it backs out of it. Then it looks for the next hidden text (which might be the piece it backed out of) and copies it to the location the cursor was at when one of the J or P buttons was pressed. It therefore has copied the context string for the topic to a new location after the highlighted text which became underlined or double underlined as appropriate when P or J was pressed with it highlighted. If the J or P button had not been pressed, the C button assigns the topic for context string it finds as the topic to be displayed initially.

I Used to determine the initial actions when WinHelp calls the file. Principally used to determine which topic will be displayed first when the file is opened by Windows help. When you click the **I** button you are given the option of selecting the initial topic for display. In that case the **HELLLP!** automatic table of contents will be displayed only if you insert a hypertext jump to it. This way you can hide it from the users, replacing it with your own in your final HLP file, but retain it for your own use in keeping the file you produced it from updated. If you change your mind you may repeat the process to select another starting page.

You may also display your own title screen by creating it as a topic, setting it with the **I** button to display first, and supplying just one hypertext jump from it to the table of contents, your own or the **HELLLP!** automated one.

The **I** button was used in this way to make the Windows Help application start with the title screen you saw when you started it this time.

If you do not make a selection with the **I** button, the **HELLLP! automated table of contents** will be the first screen to display when Windows Help calls the file.

A secondary window may also be selected for initial display. If this option is chosen, you are given the choice of having it display alone, or beside the main window topic chosen for initial display. If the former, the main window will display only when a jump from the secondary window to the main window occurs.

More sophisticated users may specify up to 4 WinHelp macros to run on opening.

B Used to generate browse sequences automatically. Use of this button gives the option to delete all previously defined browse sequences and to generate them for all topics except the automated table of contents. These browse sequences will enable the << and >> buttons on the Help Screens so that the user will be able to read through the document in the same sequence as it is in the file. For any topic which you do not wish to be accessed by browsing (for example any topic you wish to appear **only** as a PopUp box), select (highlight) a piece of text in the topic before clicking the B button and then click the command to delete the selection from the browse sequence..

The Up Arrow button will take you to the last place in which **HELLLP!** added an automated table of contents entry.

The Down Arrow button will take you the last place in which **HELLLP!** did work in your text, adding a topic heading, a PopUp or a hypertext jump.

The Browse Sequence

The "Browse Sequence" is the order in which topics will appear when the user presses the << and >> buttons at the top of a windows help screen. In help files for Windows version 3.1, the browse buttons are not always displayed. In **HELLLP!** you can control whether they will be displayed using the **Set Button Selection** function on the **Format** menu. If you use the facilities of **HELLLP!** to generate browse sequences, it will turn on the browse buttons. However you may turn them off again if desired.

If a browse sequence is not defined for a topic, the << and >> buttons are disabled (greyed out) when that topic is displayed.

To remove a topic from the browse sequence, simply select a piece of text in it, click on the **B** toolbar button and delete the selection from the browse sequence.

With the use of the **B** toolbar button, **HELLLP!** will insert browse sequence numbers so that the >> button presents the screens in the same sequence as the topics in the file, or in the selection if text is selected. The automatic **HELLLP!** table of contents is not included in the browse sequence so that a user who wishes to use a different format table of contents, selected with the **I** button can prevent its display to the users, but still preserve it in the file as a means of keeping track of HELP context strings. If you wish to include the automated TOC in the browse sequence, highlight a piece of text in it before clicking on the toolbar **B** button, and give it the sequence number 1 (the automated numbers go up in tens).

Browse sequences are created in the form *group:sequence*. Browsing may only occur between topics in the same browse group.

If no text is highlighted (selected) when the B button is clicked, **HELLLP!** creates (or removes) a single browse group for each project file, with the same name as the project, and appends a five digit number advancing in tens. Every topic in the file will thus be presented in sequence by the browse buttons.

If any text is selected, **HELLLP!** processes those topics with selected text (either for addition or deletion of browse sequences). For the addition of browse sequences in this case, **HELLLP!** prompts for a browse group name and starting number. In this manner, several browse sequences of discontinuous topics may be created in a single file. To assist in this process, **HELLLP!** keeps track of the active browse groups and the highest sequence number assigned in each.

You may delete a single topic from a browse sequence or add it to a different one by highlighting a section of text in that topic alone.

If you wish to analyse a list of existing browse sequences, click on the "Analyse" button on the browse dialog box, or select the **List Browse Sequence** option on the **File** menu. This process is slow, but will present you with a list of your topic titles sorted by browse groups in browse sequence and of those topics not in a browse sequence, sorted by page number.

By default the WinHelp compiler enables the < and > keys as accelerator keys for the << and >> buttons. **HELLLP!** also allows you to enable the left and right arrow keys for the same purpose, since the up arrow and down arrow keys are already enabled as scroll keys.

The Microsoft Help Compiler permits a help file to be built up from several files, and allows Hypertext Jumps and PopUps between them. Some people like to keep their "Jump" topics in one file and their "PopUp" topics in another one. Some projects are large or decentralized so that sections of the help file are built separately and linked later. The External Jump feature of **HELLLP!** is designed to assist in the use of these features.

When you click on **[J]** or **[P]** to create a jump or a PopUp you are offered the option of selecting an External Jump. If you select this, you are presented with a list of other help files in the project directory. When you select one, you are then presented with a list of topics and their context strings. You select one and the jump or PopUp is set to reference that topic in that file. **HELLLP!** also sets a flag to ensure that the file referenced will be included when a help file is created. The initial screen displayed by WinHelp will be that for the file from which the compile was run.

If you prefer to select topics for jumps or PopUps from a list box, instead of by using "point and shoot" at the table of contents, you may use the [External Jump] button to pick topics from the current file. Just choose the current file instead of a different one from the list of help file project names.

The other files may still be worked on and updated, provided that the context strings for the externally referenced topics are not deleted or changed.

To compile your HELP file



You may compile your help file by clicking the curved arrow, "lets give it a whirl" or by selecting "Compile the Help File" from the Tools menu. This will save the file as Rich Text Format and will also save the .DOC file. Before proceeding, you may wish to convert any "[Smart Quotes](#)" and to update the [browse sequence](#) so that the << and >> keys page you through the text in the same sequence as the file. If you have installed one of the Microsoft Help compilers and have used **HELLLP!** setup to tell **HELLLP!** where it is, then **HELLLP!** will prepare a Windows Help (.HLP) file which may be used on any machine running Windows. If you do not have the compiler installed, simply take the files with the same names as your project to someone who does, and they will be able to compile it for you.

Compiler error messages will appear. The various versions of the HELP compiler require different commands, and will sometimes reject the command for a different version. Because **HELLLP!** cannot predict the behaviour of the particular version you are running (the name does not identify the compiler, it could have been renamed), it issues several versions of the command. For example, some versions expect the command INDEX to tell them which screen to start with, some expect the command CONTENTS for the same thing. **HELLLP!** issues both, because one will be right. However the other will trigger an error message. The error messages do not indicate a problem with your file, which you may test by clicking the "spectacles" button or by selecting "Test Project Help File" on the Help menu.

IF YOU RECEIVE A COMPILER ERROR WHICH STOPS THE COMPILE YOU MAY NEED TO OBTAIN THE CURRENT VERSION OF THE COMPILER, as explained in [Error Messages](#).

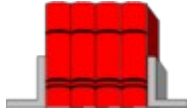
To try out your newly compiled help file

You may try out your newly compiled help file by clicking on the "Spectacles" button to "take a look" at it. As an alternative you may select the item "Test Project Help File" from the Help menu.

User Copyright Line in WinHelp ABOUT box

Help files produced by registered copies of **HELLLP!** will "brand" the help files they produce with a copyright title to appear in the WinHelp ABOUT box. By default they will use the name of the registered user for this. The default may be changed with the **Insert User Copyright** function on the **Format** menu.

Unregistered evaluation copies will include identification to that effect in the ABOUT box of WinHelp.



Using **HELLP!** and Winhelp to produce corporate manuals

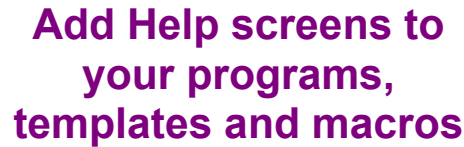
The Windows HELP application is an excellent tool for providing access to corporate manuals, provided you have word processor text for them. The Hypertext and PopUp definition boxes, and the easy incorporation of colour make them more valuable in this form than on paper, and if you keep the HLP files on a server you can be **sure** that everyone is using the same version and that it is up to date.

The users do not need to have a copy of Word For Windows™, since to display help files all you need is the WinHelp application which is included free with every copy of Microsoft Windows™

All you have to do to make this possible is to import the file into **HELLP!** and go through it marking up topics and jumps, then compile. **HELLP!** keeps track of things like context strings. It doesn't even matter if you don't know what they are, just uncheck the box that asks if you want to continue confirming them and forget about them.

There are very few restrictions. You should stay away from using dark green in your text (although you can use any other colours, even Dark Cyan gives enough contrast), because it will confuse your users who will not know where to click for hypertext jumps. You should also stick to fonts that you know all of your users have. Graphics are tricky and best avoided by the beginner, but if you are knowledgeable enough to include bitmaps (including those produced by Microsoft's "SHED.EXE" hot spot graphics editor), they will work.

Once you have produced the HLP file, just set up an item in Windows with the command line `WINHELP FILE` where FILE is the name of the HLP file that the compiler produces, including the path to find it if necessary. Your users will now only have to double-click on the icon to access your manual. You don't need to train them to use another hypertext tool, because they all know how to use Windows Help.



WinHelp needs to know where to find the .HLP file, so it is easiest to keep it with the template it refers to and use the following macro in **Word for Windows**:

'Substitute your own help file name for HELLLP!.HLP

Copy

The SHELL command can be used in **Visual Basic** from version 1.0 onwards for a simple way to display a help file like this:

- 'Invokes WINHELP.EXE

Multiple Project File Builds

It is frequently more convenient to build up a large help file in sections. **HELLLP!** is able to provide assistance with this feature. First, create and test help files for the individual sections to ensure that the topics are set up and tested. Then create a "top level" help file linking into the individual files by using **HELLLP's** "External Jump" Feature.

Please note that using the [J] or [P] button to access a topic in another file requires that you have created that other file with **HELLLP!** and that its DOC file, and the INI and RTF files that **HELLLP!** created for it, are in your project subdirectory. This procedure results in a single help file with no external links.

You may also include buttons or hotspots in your help file which will launch other help files, either produced by you or someone else. In this case, you need to ensure that the other help file to be launched will also exist on the eventual user's machine. If you wish to jump directly to a topic in that other file, you will need to know the topic number to which that topic was mapped when the file was compiled.

When a topic from another file is referenced by a Jump or PopUp, **HELLLP!** sets a flag to ensure that the file referenced will be included in future "Builds" of a help file using the file FROM which the reference is made (the "top level" file). Builds based on the referenced file may be performed without the need to link back to the top level file, unless references go both ways.

A second file referenced in this way is not affected in any way and may be independently updated and compiled. After update **and compilation** of such a file, a new build of a file referencing it will incorporate the updated version.

It should be remembered that if even a single topic of a secondary file is referenced as a jump or popup, then all topics it contains will be included in the Help file - even if they cannot be reached by jumps, because the Microsoft Help Compiler will incorporate the whole file.

Note that if there are references back **from** the secondary file, there may be compiler warning messages generated since some topics in the primary file may need references to other secondary files which the current one knows nothing about.

External Jumps are accessed using the [J] and [P] buttons on the tool bar.

The easiest way to link a multiple file build is to make External Jumps from the "top-level" file to the tables of contents of the other files, although any topic of any file may be referenced from any other file.

To ensure that **HELLLP!** includes all of the needed files in a "build", there must be reference to each of them at least once in the "top level" file. For example, if FILE1 includes a jump to a topic in FILE2 and FILE2 includes a jump to a topic in FILE3, then there must be at least one jump from FILE1 to FILE3 or **HELLLP!** will not know to include FILE3 in the build. That required jump can be made from the non-displaying topic with the yellow text immediately following the automated table of contents if it desired that the user does not see a direct jump from FILE1 to FILE3.

"Boiler Plate" files of definitions etc. may be created and automatically linked in when referenced.

It should be noted that many features of WinHelp files must be consistent within a given help file. Settings of things like background colours, for example, must be done in the top level file and such settings will be used instead of any settings used in the initial testing of the individual lower level files.

Launching Other Help Files

You may launch other help files from within your help file by using the [Spawn External Process](#) menu item or button and selecting the option to invoke a [WinHelp macro](#).

Several of the macro functions available perform this for you. You may start the other help file at its defined contents topic, or at any numbered topic. For the latter you need to know the [HelpContextId numbers](#) which were mapped to the topic when that file was compiled.

Please note that you must ensure that the other help files will be present on the user's machine. Help looks for the other Help file in the following locations:

1. The directory of the Help file that initiated the interfile jump (Help's current directory).
2. The MS-DOS current directory.
3. The user's Windows directory.
4. The Windows SYSTEM directory.
5. The directory containing WINHELP.EXE.
6. The directories listed in the user's PATH environment variable.
7. The directories specified in WINHELP.INI.

If Help cannot find the Help file after searching in all these locations, it displays a "Help file not found" error message.

If you have developed a help project in sections using HELLLP!, you may build a single help file incorporating them all (see [Multiple File Projects](#)). This will make distribution simpler and will guarantee the avoidance of "Help file not found" messages when the user who has only some of the files clicks on the hotspot concerned. Additionally, more efficient [compression](#) may be performed, resulting in a saving in disk space.

Compression of Help File

The Windows Help Compiler is able to compress the help file it produces, which both saves disk space and speeds loading when HELP is invoked. However, compilation is slower when compression is turned on. **HELLLP!** provides an option permitting compression to be turned on and off. It is invoked by selecting **HELLLP! Compression Options** from the Format menu. For a new file being created by **HELLLP!** compression is initially turned off.

Differences can be significant. Without compression, one version of the help file for **HELLLP!** compiled in about half a minute, but resulted in a file size of 376 kbytes. With compression it took 9 minutes to compile, but the file size was reduced to 213 Kbytes



HELLLP!

for the

Expert User

HELLLP! is designed to be easy to use, and to produce help files rapidly. However, even if you are an expert user of the Help compiler, **HELLLP!** can still help you. It produces a DOC file with all of the codes in that you need to compile a help file, and you can edit it to customize it further. Since the expert user may wish to produce a help file from several source files, **HELLLP!** provides the means to "Busy" context strings to ensure that they are unique across several files.

Alternatively, the expert user may choose to employ **HELLLP!**'s automated multiple project file build feature.

The expert user may also be interested in **HELLLP!**'s ability to create the association between Context Numbers for applications and Context Strings by the Map Topic For Application feature.

The following topics may be of particular interest to the expert user:

- Hotspot graphics and the SHED editor
- Compression of Help File
- HELLLP! from a network server
- Error Messages and problems
- Multiple Project File Builds
- External Jumps
- Mapping Context Strings to Context Numbers and HelpContextID for Applications
- Adding Sound by Playing *.WAV files
- Spawning an External Process from a Help File
- Setting Help Window Size and Screen Position
- Choosing an Icon for the help file
- Choosing the Capitalization (Case) of Winhelp Search Keywords
- Button Selection and Control
- Smart Quotes
- Non-Scrolling Region
- Secondary Windows
- Background Colours
- Using non-English versions of Microsoft Word
- Use of WinHelp Macros
- WinHelp Help Compiler Limits
- Launching Other Help Files
- Project Accounting
- Bullets and Symbols
- HotSpot Colours
- Heading styles to pick topics
- Mid-Topic Jumps
- Compuserve Registration Instructions
- Using 256 colour bitmaps
- Glossary Button Box
- Importing files
- Accelerator Keys
- Adding Menu Items
- Fancy Macro Demonstration
- Customising lines in the HPJ file (for experts)
- The symbols +#!K\$

Hotspot graphics and the SHED editor



It doesn't have to be like this!

You may embed graphics by capturing them on the clipboard from another application (like PAINTBRUSH) and then using EDIT, PASTE SPECIAL to paste them in as bitmaps. You may still make them HotSpots by selecting the graphic before clicking the **[J]**, **[P]** or other action button.

HINT: If you want your graphics to appear beside your text, create a table and insert the graphic in one of the cells. If you display the topic as a PopUp it will be automatically set to the correct size and shape. You can even add a pictorial business card this way. For another example of this effect, see the topic Setting Help Window Size and Position.

You can produce a graphical background to your text by putting a graphic in a very thin left hand column of a table and the text in a wider column at the right. The graphic can be either a bitmap or a {bmc ... type statement referencing one. Click here to see the effect which can be achieved. (This has several effects combined, but they were all produced by **HELLLP!**).

A good but frequently overlooked **source of graphics** for use in Help files is the clip-art library supplied with Microsoft power point.

Open PowerPoint, under **File** select **Open Clip Art**, select and display the graphic you are interested in; using the mouse, select the area of the graphic which you want and use Edit Copy to place it on the clipboard. Return to Word and paste it to your file. You will save memory and file size if you use PASTE SPECIAL and paste as a bitmap, then press CTL-SHIFT-F9 (all at once) to unlink from the application which generated the graphic.



<< This one was pasted into a table cell as explained above. It was then selected again, and the Add Sound to Topic function on the Insert menu was used to associate it with the SystemAsterisk sound, which you will hear if you click it.

Graphics with "hotspots" produced by the SHED editor may be incorporated into help files produced by **HELLLP!**



The SHED editor requires some expertise to use. An up-to-date version may be downloaded free from CompuServe's Windows SDK forum section WinHelp, file SHED.ZIP. It has an extensive HELP system (click the SHED icon here if you have SHED installed in your path).

Since the SHED editor must be told the context strings to use for particular hotspots, the topics to be accessed by the graphical hotspots should be created first. The context strings that **HELLLP!** creates will be found adjacent to the topic titles, both at the head of the topic and in the automated table of contents. They are also listed in the project's INI file.

Once the topics are created, the graphic may be created and edited with SHED, then saved as a file (with a .SHG extension). Note that these files usually take much less space than PCX files.


HINT: Run the SHED editor in the left of your screen, run Winword in a window on the right, displaying the automated table of contents at the top of your file.

To incorporate the graphic **by reference** into your help file, include a line like this at the location where you want the graphic to go:

```
{bmc c:\helpcomp\project\hellptb.shg}
```

Remember the curly brackets. These are the curly brackets **on the keyboard**, not the field code delimiters in Word. This line was used to create the hotspot graphic of the **HELLLP! toolbar** in this file. To place the graphic at the left, use bml instead of bmc (which treats the bitmap like a character for placement). To place it at the right, use bmr.

Incorporation by reference is the preferred method for including graphics since it requires less memory at compile time.

Clicking on the  button (or selecting **SelectPictureToInsert** on the **Insert** menu will perform an incorporation by reference. If you wish to have the whole graphic used as a hotspot (other than individual areas set up by SHED) check the appropriate box in the dialog and you will be prompted for the action to be taken when the picture is clicked.

The same command may be used to incorporate graphics which are in **.BMP** file format.

Even if you do not wish to use HotSpots within the graphic, using graphics created by the SHED editor is efficient since SHED's *.SHG files are smaller than most. **If you incorporate graphics directly in the DOC file, it becomes very large and may reach the limit of the compiler.** You may make the whole graphic a hotspot by highlighting (selecting) the command which incorporates it (including the curly brackets) and using the **[T]** or **[P]** toolbar buttons.

If you wish to embed graphics and you do not have access to the SHED editor, you may embed regular bitmap files (*.BMP) in the same manner as *.SHG files. This is frequently the only way to embed large graphics without running out of memory in the compile. To create a BMP file, paste the graphic from the clipboard into Windows Paintbrush, then, using the scissors tool, select the area you wish to use, then use the **Copy To** option from the paintbrush **Edit** menu to save the BMP file. Save it as a 16 colour bitmap.

To do a screen capture of an individual window to the clipboard, use ALT-PRTSRNR (the PRTSRNR key alone copies the whole screen to the clipboard).

The WinHelp application can display only 16 colours. However, with the support of customized DLLs [256 colour pictures](#) may be included in embedded widows.

Using 256 colour bitmaps

The WinHelp application itself is able to handle only a 16 colour palette. To handle 256 colour bitmaps, an embedded window must be employed and controlled by a Dynamic Link Library routine (a **DLL**). **HELLLP!** provides support for the EWDLL.DLL available through the auspices of the Free Software Foundation. The DLL may be obtained from many sources including Compuserve (**GO WINSDK**, Section 16 library) and the Internet (including **ftp.harding.edu** in the directory **/pub/pacheco**). To find it do a partial filename search for EWDLL.

In some cases, depending on your setup, you may not be able to preview the picture within Word by clicking on the spectacles button. Instead a box saying "unable to display picture" will appear. In those cases, double click on the .HLP filename in the Windows File Manager to see the file fully displayed.

The DLL may be placed in any of the locations that Windows looks for DLLs, but the safest place is probably in **WINDOWS** or **WINDOWS\SYSTEM**.

Support for EWDLL is provided on the "Screen" button or the **Spawn External Process** item on the **Insert** menu as well as via the **SelectPictureToInsert** button and selection.

Choosing an Icon for the help file

The default icon used when a help window is minimised is the yellow question mark. You may choose a different icon by using the **Choose Icon** function on the **Format** menu. You will be presented with a dialog box asking for the full file name of the icon file you wish to use.



Mapping Context Strings to Context Numbers and *HelpContextID* for Applications

For access by context-sensitive help in applications, a topic must be identified by a context number (called the `HelpContextID` property in Visual Basic) instead of by the context string. The association of a context number with a context string is made by the Help Compiler. **HELLLP!** has the ability to insert the necessary instructions to the Help Compiler via a procedure accessed by **Map Topic For Application** on the **Insert** Menu.

This allows you to associate a context number with any Help topic which you have set up in the current file or in any file which you have created and compiled in the same project subdirectory. If the topic is in a different file, then that file will be included in future builds of a *.HLP file from the current file. The other file is not affected in any way, it is just copied into the *.HLP file.

You may test the assignment of context numbers after compilation by using the **Test Numbered Help Topic** item on the **Help** menu.

If you are using an application generator (e.g. Visual C++) which insists on choosing the context numbers for itself, you may still use **HELLLP!** to produce a help file which will respond to it by using the process above and assigning the numbers it provides.

Alternatively you may make use of the **HELLLP!** feature provided to Customise the HPJ File. Create a file with an ASCII text editor (or save as TEXT ONLY from Word) with the same name as your **DOC** file, but with the extension **HPK** (if using multiple files, use the the name of the top level one). Include a [MAP] section:

```
[MAP]
contextstring1 contextidnumber1
contextstring2 contextidnumber2
etc
```

Where:

The contextstrings are the ones assigned by HELLLP! and included as red hidden text both in the table of contents and at the beginning of the topic titles
and

The contextidnumbers are the numbers (in either decimal or hex) which the application assigned for you to use when you want it to call the corresponding topic.

You will be asked to confirm that you want the HPK file included in the build (answer YES).



Adding Sound by Playing *.WAV files



Click on the bell. The Windows help system is able to play sound (*.WAV) files, provided they are in the WINDOWS subdirectory and provided that the machine is able to play them. A machine without a sound card can play *.WAV files through the built in speaker with the aid of the Microsoft Speaker Driver, available as the file SPEAK.EXE from the Microsoft Software Library (GO MSL on CompuServe).



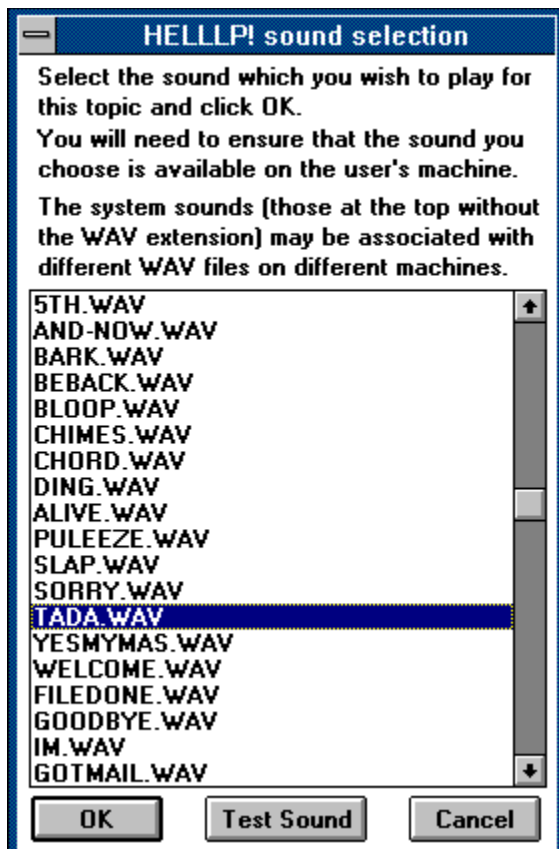
HELLLP! implements the ability to play a WAV file when a topic is displayed, or when HotSpot text is clicked. To invoke the feature, use the item **Add Sound To Topic** on the **Insert** menu or click the "Speaker" icon on the toolbar. If text is selected at the time, that text will become HotSpot text (looking like a hypertext jump) and the sound file will be invoked when it is clicked (this one plays your "SystemHand" WAV file). If no text is selected the sound file will be played when the topic is displayed, as it did when this topic displayed. If you select this item several times, each sound will be played in turn in the order of selection.

Additionally, if graphics are included in the file, they may be linked to WAV files just like text. Highlight the graphic or the statement in curly brackets (including the brackets) which incorporates the graphic and then choose Add Sound To Topic. The graphic then becomes a HotSpot and the WAV file will play when it is clicked.



The "speaker" graphics were incorporated that way and linked to the System Sound called "SystemExclamation". That sound event is associated with a WAV file in the **[sounds]** section of **WIN.INI**, and that WAV file will play when it is clicked.

Sounds are selected using the dialog box below, which appears when the **Add Sound To Topic** function is selected or the speaker button on the toolbar is clicked.



Of course these buttons are "live" - click one and see!

The HotSpot buttons here were set up using [the SHED editor](#).

To remove sounds from the opening display of a topic, use the function **Clear Topic Opening Effects** on the **Edit** menu. Please note that it will also clear any window size change set up as an opening effect for this topic.

If you merely wish to use sounds to gain attention, it is probably wise to use the standard SystemSound names since most machines will have them defined and you do not need to distribute WAV files. If, however, you need specific sounds you should use WAV files and distribute them with your application. Windows 3.1 came with the files CHIMES.WAV, CHORD.WAV, DING.WAV and TADA.WAV. Other WAV files are available widely from bulletin boards.

Since setup code must be generated when compiling to include sound, there must be at least one sound call invoked in the top level file of any multi-file compile if lower level files include sound calls. That required call may be from the yellow type, non-displaying topic immediately following the automated table of contents.



Spawning an External Process from a Help File

The Windows help system is able to start external processes, provided that the machine is able to find them. In this way, various multi-media applications may be controlled from a help file.



As an example this graphic will start the Calculator application when you click it. [So will this text.](#)

HELLP! implements the ability to spawn an external process when a topic is displayed, or when HotSpot text is clicked. To invoke the feature, use the item **Spawn External Process** on the **Insert** menu or click the "screen" button on the toolbar. If text is selected at the time, that text will become HotSpot text (looking like a hypertext jump) and the external process will be invoked when it is clicked. If no text is selected the external process will be invoked when the topic is displayed. You may also invoke a [WinHelp macro](#) from this same button or menu item. This function is particularly useful if you wish to call another application's help file from within yours. Use the `JumpContents()` macro for this purpose.

Additionally, if graphics are included in the file, created by [the SHED editor](#) or directly embedded, they may be linked to external processes just like text. Highlight the graphic of the statement in curly brackets (including the brackets) which incorporates the graphic and then choose **Spawn External Process**. The graphic then becomes a HotSpot and the process will start when it is clicked.

Spawn External Process will display a dialog box asking for the command line which you wish to be invoked. You should enter the command line exactly as you would enter on the Windows Program Manager **File Run** menu.



This graphic, when clicked, will start up the Sound Recorder acting on the file `\\WINDOWS\\TADA.WAV` (if the current machine does not have that file, the sound recorder will give its normal error message). The "Command Line" entered in the **Spawn External Process** dialog box was **SOUNDREC.EXE C:\\WINDOWS\\TADA.WAV**.

Setting Help Window Size and Screen Position

You may specify the size and position which your Help window will use when it first displays (your users can move it and resize it as usual afterwards). You may also define up to five secondary windows using this function.

The function is found on the **Format** menu as **Set Help Window Size**. When invoked, it presents a dialog box giving you the ability to set the left, right, top and bottom margins around your help windows as a percentage of screen size, which are easier to visualize than pixel counts. It will not allow you to set impossible sizes and will override your choice if you attempt to. The values pre-set in the dialog box are those which are set for the initial display of the current topic, or if none are set, for the initial display of help. This makes it easy for a topic to force a restore to the initial window size if reached after the size has been changed. It is a good idea to set the initial display screen topic to this size, as well as the topics in the browse sequence before and after one which sets a different size. To pick up the size options for any other window, select that window with the radio buttons and click **Refresh**.



**YOU MAY
CLICK ON THIS GRAPHIC
TO OBTAIN AN
EXPLANATION OF
ACTIONS.**

This function was used to cause this size change. To restore to the size and position which this help file had on opening, click on CONTENTS then Main Window Title or one of the browse buttons (those topics also have a size and position setting).

This graphic is a hypergraphic, also to demonstrate an effect. This interesting effect was created by setting up a three column table and including the bmc statement explained in the topic "SHED EDITOR" inside the left hand cell(which is very narrow). This text was placed in the right hand cell.

The shadowed overlay was created by including that text in the centre cell.

You could use this to "Watermark" example graphics in your help file to prevent copyright violation by screen capture.

You may choose to fill the screen by using zero margins all round if you wish (your user will still be able to resize the window as usual).

You may elect to have the window change to a specific size and position when a topic is displayed. To do this, ensure that the cursor is in the topic concerned before selecting **Set Help Window Size** and then click the lower option button in the dialog box. Please note that a Help window size, once set, remains in effect until it is explicitly changed either by the user or by a topic with a different setting. If windows sizes are critical to your desired effect, you might want to do an explicit window size setting for every topic - but this may annoy a user who keeps altering it, only to have it switch back the next time the topic is displayed.

To remove window size changes from the opening display of a topic, use the function **Clear Topic Opening Effects** on the **Edit** menu. Please note that it will also clear any sound effects set up as an opening effect for this topic.



Button Selection and Control

The Windows help system can have a variety of control buttons just below its title bar. The selection of which buttons will appear there is made at the time of help file compilation. **HELLLP!** supports the selection and placing of several of these. The function is found as **Set Button Selection** on the **Format** menu. **HELLLP!** also provides the means to include buttons in the text of topics, and at the head of individual topics. This also provides a means of placing button bars at the top of secondary windows.

You may also define a "[Glossary Button Box](#)" with alphabetic buttons, which is particularly useful for a glossary employing [Mid-Topic Jumps](#) to access initial letters.

The text on the tops of the buttons created may be edited, for example to use non-English text.

To change the buttons on the main window button bar, choose **Button Bar for Main Window** in the initial dialog.

Options are provided to add:

- Browse Buttons (automatically turned on if **[B]** button is used)
- Print Button
- Copy Button
- Close Button (Closes the main help window)
- Exit Button (Closes all help windows)
- About Button (Displays the WinHelp About Box)

An option is also provided to change the topic accessed when the **Contents** button is clicked and the window (main, secondary or PopUp) in which it is displayed. (If a secondary window is to be used, it must first be defined using the [window size and position](#) option. In this way, it is possible to set one topic for initial display (using [the I button](#)) and a different one to be displayed when the **Contents** button is clicked. This help file is set so that the title screen is initially displayed, but the contents button invokes the SideBar Table of Contents.

If the **automated table of contents** is selected for display in a secondary window, a SideBar Table of Contents is created with the jumps going to the main window (otherwise everything would display in the secondary window).

Up to three additional buttons with author-chosen labels may be added to the main window button bar, and connected to any topic of the author's choice, or to any WinHelp macro (**Only use the macro option if you know what you are doing in depth**). The options are invoked by adding button label text to the appropriate edit box in the button selection dialog for the main window. Once a special button has label text defined, you are prompted for the action you wish it to cause.

The macros presented in the macro selection dialog include those to disable or remove a button. If selected, they present a list of already defined buttons which may be disabled or removed (some buttons may not be removed). To create a "**One-Shot**" button, which will operate only once, you may use the macro to disable or remove the button as the first of a chained sequence. However, if the button is not already yet defined it will not be presented in the list available for disabling or removing. The solution is to define the button first, calling an innocuous macro like **About()**, then redefine it to disable itself and to do the job you intend by calling the **SetButtonSelection** function again and picking a new action for this button. Note that this ability is available only with buttons on the main button bar.

Topic Button bars and secondary window button bars.

HELLLP! provides an automated means to insert a horizontal button bar in a non-scrolling region at the head of a topic in main or secondary windows. This is the only way in which a secondary window may have a button bar. Select the option **Button Bar For Current Topic** from the initial dialog for **Format, Set Button Selection** in order to achieve this. To delete a topic button bar previously defined, simply uncheck all of the boxes in the dialog box.

The topic button bar is initially set up in a small non-scrolling region at the head of the topic, even if the topic title is scrolling. If the **Change Topic Title Style** function is used to change the topic title to non-scrolling, then the button bar is merged with it. If the topic title is subsequently change back to scrolling, then the button bar will scroll with it.

This option was used to insert the extra **Contents, Copy and Close** buttons at the head of this topic. Notice how the extra **Contents** button is connected to the main window table of contents, while the main one brings up the SideBar Table of Contents. For another example of **Close** and **Exit** buttons used this way, see the [SideBar Table of Contents](#) of this help file.

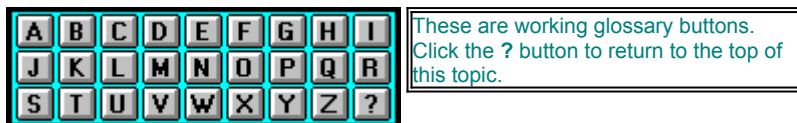
The topic button bar option may also be used to place a button bar (which could be a single button) anywhere in the topic. It was used with the **Put Button Bar Here** option selected to add this extra copy button.

Copy

Buttons in a Topic Button Bar for **Contents, Help, Back** and **Browse** permit (and require) the help file author to determine which action will occur (usually the action will be a jump to a topic) when they are clicked. This is necessary, both to make them topic-sensitive and because the regular functions will not work if called from a secondary window. WinHelp macros may also be invoked from topic button bars, but caution is required because some macros will not work if called from a secondary window.

Please note that you will not actually see the buttons in the text of your DOC file, although you will see the text which will appear on them. The buttons will, however, appear in the completed help file.

If you are placing a Topic Button Bar in a topic which will be called as a **PopUp**, you should use the PlaceHere option in the dialog box, otherwise the remainder of the PopUp text may not be displayed in the PopUp window since if any of a topic is set to be non-scrolling then only it will display in a PopUp window.



Glossary Button Box

You may include a block of alphabetic buttons like that above. These may be set to jump to any topic or mid topic target. The control which sets them up can also generate automatically a set of mid-topic targets like those below. Notice how the appropriate letter surfaces when you click it in the box. If you intend to include a glossary, this will assist you by setting up a skeleton with alphabetic headings in the **Glossary Title** style (initially this is identical to the **Topic Title** style, but it may be changed by **Format Style**).

To use this facility to create a glossary skeleton, first create an empty glossary topic using the **T** button and giving the topic an appropriate title.

Ensure that the cursor is in that topic and below the title.

Use the **Format**, **SetButtonSelection**, **GlossaryButtonBox** function to create the glossary button box and mid-topic target alphabetized jumps by clicking on the **Make Skeleton.Glossary Topic** button.

Use the function again, but this time click on the **?** button in the glossary button box to choose a topic to jump to for that button. You will probably select the appropriate help topic to deal with the glossary. This time, click on the **Make Setting Changes Only** button, since you do not want to insert another copy of the button box right now.

Insert your definitions following the appropriate heading letter.

Even if you generate the automated glossary skeleton of mid topic jumps, you may override them with jumps to other topics or mid topic targets. In this example, the **S** button has been set to jump to a different topic.

HELLLP! will look in the project directory for the file **alph.shg** and if it does not find it there, it will ask you to edit the pathname to it. If you wish to design your own SHG file you may do so, provided that you set the hotspots to call the context strings **_A** (underscore, A) through **_Z** and **_QRY**. The file **ALPH.SHG** is included in the **HELLLP!** distribution archive.

The command string to incorporate the box will be placed in the left column of a two column table. To include text to the right of it, place that text in the right column, with any character formatting you desire. To border the text, highlight it and use **Format Border** for an effect like that above. For the unadorned box, leave the right column of the table empty.

The **?** button can be set to call any topic you wish (probably the the topic giving help about the topic in whic the box is included). If not re-assigned it defaults to the A topic when you generate an automated glossary skeleton. In this topic, it is set to return you to the head of the topic.

Should you not choose to generate an automated glossary skeleton, **HELLLP!** will check that all buttons have been assigned topics and will warn you if they have not.

You may include copies of the box of buttons in more than one place, but they will all perform identically (i.e. they will jump to the same topics or targets).

Click button to which to assign an acti

A	B	C	D	E	F	G	H	I
J	K	L	M	N	O	P	Q	R
S	T	U	V	W	X	Y	Z	?
Make skeleton glossary topic								
<input type="checkbox"/> External jumps allowed								
Insert button box now								
Make setting changes only								
Help								
Cancel								

For additional help, please click on the appropriate button or box for an explanation of its function.

A

You would put your A definitions in this section.

B

C

D

E

F

G

H

I

J

K

L

M

N

O

P

Q

R

S

T

U

V

W

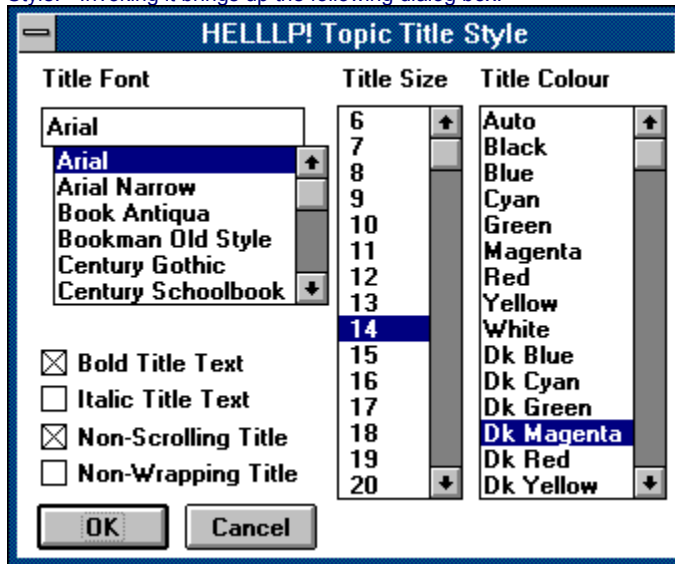
X

Y

Z

Changing Topic Title Style Colours and Fonts

HELLLP! provides the means to change the standard topic titles which it has generated to another font, colour and size. In addition it permits the setting of the topic title as non-scrolling. The function is found on the Format menu as Change Topic Title Style. Invoking it brings up the following dialog box:



The dialog box is titled "HELLLP! Topic Title Style". It contains three main sections: "Title Font", "Title Size", and "Title Colour".

Title Font: A list box showing various fonts. "Arial" is selected. Below the list are four checkboxes: "Bold Title Text" (checked), "Italic Title Text" (unchecked), "Non-Scrolling Title" (checked), and "Non-Wrapping Title" (unchecked).

Title Size: A list box showing font sizes from 6 to 20. "14" is selected.

Title Colour: A list box showing various colors. "Dk Magenta" is selected.

At the bottom are "OK" and "Cancel" buttons.

Title Font	Title Size	Title Colour
Arial	6	Auto
Arial	7	Black
Arial	8	Blue
Arial Narrow	9	Cyan
Book Antiqua	10	Green
Bookman Old Style	11	Magenta
Century Gothic	12	Red
Century Schoolbook	13	Yellow
	14	White
	15	Dk Blue
	16	Dk Cyan
	17	Dk Green
	18	Dk Magenta
	19	Dk Red
	20	Dk Yellow

Bonjour

Using non-English versions of Microsoft Word

HELLLP! is set up in the English language. It may be used with some non-English versions of Microsoft Word to produce help files in those languages. The help file authors will need to understand English to respond to the prompts.

Because **HELLLP!** needs to know how certain commands are translated into the target language, there are some routines in the **FILE NEW HELLLP!** sequence which detect those commands and the target language form of them. For this reason, if a new version of Microsoft Word with a different language base is installed, then the first action with **HELLLP!** must be a **FILE NEW HELLLP!** sequence to permit this to occur.

Some of the features of **HELLLP!** will place text in the resulting help file, and that text is in English unless changed. . For example, the heading **Table of Contents** is automatically generated. **HELLLP!** keeps that text as the item **ContentsTitle** in the glossary. If you are producing help files in German, you might want to use **Inhaltsverzeichnis** instead. To set this as the default, start a help file (a dummy one you will throw away) to access the template. Then type **Inhaltsverzeichnis** and highlight (select) it. Select **Edit Glossary** and select **ContentsTitle** in the dialog box. Then click **Define** then select the option **Template** and confirm that you wish to replace the entry. You will need to save the template to make this permanent. In Word v 6.0, the term **Autotext** may be used instead of the term **Glossary**.

The feature which adds extra buttons to the button bar for the main window, and adds button bars to individual topics and secondary windows has button text which defaults to English. The button-top text may be edited in the **SetButtonSelection** dialog boxes to use different words.

Because **HELLLP!** ships with its defaults set to English, it is especially important that a new version installed to work with a non-English version of Word be started with the equivalent of **FILE NEW HELLLP!** (even if a dummy file is produced and discarded) before being used to edit existing files. The **FILE NEW** process detects that a non-English version of Word is in use and makes changes accordingly. If this is not done, **HELLLP!** will try to create fields for example like **{SET ..** where a Spanish version needs **{ASIGNAR** and a German version needs **{BESTIMMEN..** and the system will fail.



About Shareware

Shareware is copyrighted software that is distributed by authors through bulletin boards, on-line services and disk vendors.

Shareware allows you to try the software for a reasonable limited period. If you decide not to continue using it, you throw it away and forget about it. You only pay for it if you continue to use it. Shareware is a distribution method, not a type of software. You benefit because you get to use the software to determine whether it meets your needs, before you pay for it.

The shareware system and the continued availability of quality shareware products depend on your willingness to register and pay for the shareware you use. It's the registration fees you pay which allow authors to support and continue to develop our products. Please show your support for shareware by registering those programs you actually use.

Even if the person who supplied you with the software tells you that registration is not necessary, be assured that if the copy of the program presents a screen reminding you to register, then the author has not been paid for that copy and registration is necessary.

This program is produced by a member of the Association of Shareware Professionals (ASP). ASP wants to make sure that the shareware principle works for you. If you are unable to resolve a shareware-related problem with an ASP member by contacting the member directly, ASP may be able to help. The ASP Ombudsman can help you resolve a dispute or problem with an ASP member, but does not provide technical support for members' products. Please write to the ASP Ombudsman at 545 Grover Road, Muskegon, MI 49442-9427 USA, FAX 616-788-2765 or send a CompuServe message via CompuServe Mail to ASP Ombudsman 70007,3536.

[Exit

Using a copy that has been made from someone else's registered copy is contrary to copyright law.

We are pleased that you have decided to evaluate **HELLLP!**, and have provided the means to convert a registered copy into a legal evaluation copy.

To make an illegal copy into a legal evaluation copy, select the item "UNREGISTER" on the TOOLS menu, and the registered user's name will be changed to "Unregistered User". The copy may then be used for 30 days of evaluation, and copies may be made for distribution to others for the same purpose. Copies of **HELLLP!.DOT** may only be distributed together with **HELLLP!.HLP**.

After an evaluation period of 30 days, you are required to either pay for the continued use of the product, or to cease its use. "Registration" involves payment, and the receipt of a registration number. The registration number is an encryption of the user's name which will be displayed in the copyright box when **HELLLP!** starts.

Exit



Registration and Payment

Registration supplies the right to the continued use of the product after the 30 day free trial, and the elimination of the reminder screens. It does not rescind the legal disclaimer of liability under which this product is used.

Additionally, Windows HLP files produced by an unregistered version of **HELLLP!** are so identified in the "About Help" box (where registered copies permit a user's copyright notice to be placed) and they have a topic at the very end noting that they were produced by an unregistered version. That topic is seen by the users only if they use the browse buttons to go to the very last topic, or if they go to the search topic UNREGISTERED. It does not appear in the table of contents.

Credit card orders may be submitted to PsL

A cash order form may be obtained by printing the topic INVOICE

For use on a single computer, registration costs **\$30 US**, paid to:

Guy Software,
1752 Duchess Avenue
West Vancouver
British Columbia V7V 1P9
Canada (CompuServe 71750,1036)

For packages mailed outside of Canada and the United States there is an additional shipping and handling charge of **\$10 US**.

If the user provides an electronic means for the passage of the registration number (Electronic mail accessible via CompuServe or Internet or a **Secure** North American Fax number), then the cost is **\$20 US** per copy in recognition of the lower costs involved. Please ensure that any electronic route is secure, since anyone with your registration number can make it appear that you have illegally given them a copy registered to you. This special electronic registration may be performed entirely within CompuServe using their Shareware Registration Database.

If the route chosen permits the transmission of a binary file, an MsWord DOC file will be transmitted which, when opened in Word automatically performs the registration. In other cases it will be necessary to type in the registered user's name and registration number.

For network use, the price is **\$100 US** if no more than six people may access the product at once, and **\$10 US** per additional permitted user for numbers in excess of that. If the user's network has the ability to limit the number of people accessing a file, that number may be used instead of the number of nodes on the network. For use in a **bona fide educational institution** the limit of six users is waived provided all users are in the same classroom.

For the single copy licence, if **HELLLP!** is installed on the internal hard drive (or equivalent) of a computer and one person uses that computer more than 80% of the time it is in use, then that person may use **HELLLP!** on one other computer.



CASH INVOICE (Please send credit card orders via PsL)

Name: _____ Supply of the **HELLLP!** System as below
Special "Electronic Registration" package
Address: _____ for one computer only \$20.00
Mailed Package-for one computer; \$30.00
Mailed Network Package first six users \$100.00
Additional users on network @ \$10.00 each _____
Shipping/handling charge outside USA or Canada \$10.00
Payment Enclosed (US Funds):. \$ _____
For electronic registration,
please give Email or Fax address _____
to send registration code.

Name for identification on
copyright screen (if different) _____

Fold here first

A
F
F
I
X

P
O
S
T
A
G
E

**GUY SOFTWARE
1752 Duchess Avenue
West Vancouver
British Columbia V7V 1P9
CANADA**

Fold Here second and then tape edges

Registration and Payment via CompuServe

CompuServe provides a convenient route for the registration of Shareware, and the special "Electronic Registration" of **HELLLP!** may be performed by that route. Note that the electronic registration price of **HELLLP!** is only dependent on your providing an electronic route for us to send you your registration code number. That price is available whether or not you use CompuServe to contact us.

To access CompuServe's registration database, **GO SWREG** in CompuServe. **HELLLP!** is **Registration ID 1346** in the CompuServe database and its price is **\$20.00(US)**. Only the "Electronic Registration" is available by this route. Once CompuServe informs us of your registration, we will transmit to your CompuServe ID a WordForWindows document called **REGISTER.DOC**, and when you open that in Word your registration will occur automatically. That same document will permit you to register the next few upgrades as you download them (none are planned, but inevitably there will be maintenance releases for bug fixes and minor enhancements).

The following instructions are provided by CompuServe on the use of the Shareware Registration Database.

Register Shareware+

INSTRUCTIONS TO REGISTER SHAREWARE

The Shareware Registration database contains descriptions of hundreds of shareware programs designed for a variety of platforms and purposes. To register a program, you must first find its description in the Shareware Registration database.

To begin, select "Register Shareware" from the previous menu. A list of search criteria will be displayed to you. You may search for your program's description by its TITLE, FILE NAME, AUTHOR'S COMPUSERVE ID, and AUTHOR'S NAME. You may also search for your description by its KEYWORDS (categories). For example, to find the description of a Macintosh utility, you might use the keyword: utility. (For information about registering by your program's REGISTRATION ID, see below.)

After you enter your search criteria, you are usually displayed a list of programs that meet the criteria in the database. Select the program that you wish to register from the list.

Sometimes after you enter search criteria, you are displayed a menu of choices. This occurs when your search yields a large number of "matches" in the database. The menu offers you the ability to NARROW your results further, to DISPLAY the list of "matches," or to BEGIN AGAIN. After you select your program from the list, you are displayed the program's description, with a prompt: "Would You Like to Register? (Y/N)."

Read the description carefully to verify that you wish to register the program. The description details the program, its capabilities, and the benefits of registration. Be sure to note the program's registration fee. When you register any program, the registration fee is charged to your CompuServe account.

Enter "Y" at the prompt to register; enter "N" at the prompt to ensure that you do not register the program, and to return to the previous menu. When you enter "Y" to register, you will be prompted to enter your full name, company name (optional), your complete address, your phone number (optional) and the total number of copies of the program that you wish to register. When you finish entering this information, you can VIEW, CHANGE, SEND or CANCEL it.

Select VIEW to confirm that the information you entered is correct. Select CHANGE to edit any of the information you entered. Select SEND to complete your registration, receive a detail of your total charges, and to send the information to the author. Select CANCEL to cancel registration of the program. When authors submit their descriptions to the Shareware Registration database, their descriptions receive a REGISTRATION ID. If you know the REGISTRATION ID (for **HELLLP!** the registration ID is **1346**) of the program description in the database, you may enter it under option #1 on the search criteria menu. You will then navigate directly to the program's description. This is the easiest and fastest method of registering a program.

If you have any questions or concerns about shareware registration, send a message to the Shareware Administrator by selecting the "Provide Feedback" option at the main SWR menu.

Upgrade Policy

Most shareware authors are appreciative of suggestions for product improvement made by their customers. We are no exception and have found that most ideas for improvement have come from users who have said "It would be nice if ...". Our only caveat has been that we do not add features in a way which will harm the user-friendly interface of **HELLLP!**.

This sometimes results in unpredictable releases to add valuable new user-suggested features. Our upgrade policy is that registration for a previous release will be honoured automatically for a new release. **HELLLP!** handles this automatically in most circumstances. The first time a new unregistered release is used on a machine where it replaces a registered release, it will present the registration dialog. The previous User Name and Registration Number will be honoured. In most cases they will be pre-entered in the dialog box, having been stored on the machine in a parameter file (HELLLP!.INI) by the previous version. Upgrades are thus free for the downloading from any source carrying the unregistered version.

Colours you can use

Colours you can use. You probably won't want to use some of them, and if you aren't looking at them on a colour screen you are having to imagine them.

You may change the colours of the topic titles all at once with the Change Topic Title Style item on the **Format** menu.

Black

Blue

Cyan

Green

Magenta

Red

Yellow

Dark Blue

Dark Magenta

Dark Red

Dark Yellow

Dark Grey

Light Grey

Dark Green looks too much like Hypertext or PopUps.

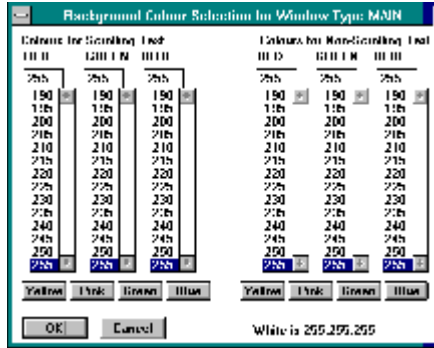
For some really wild effects, you can change the background colour.

Note that you can also change hotspot colours, or prevent your users from doing so.

Background Colours

You may change the colour of the background for the main and secondary windows independently. [Click here](#) for a sample. You may use different colours for the scrolling and non-scrolling areas if you have set non-scrolling titles.

You may create a graphical background behind some text, as in the [main title screen](#) for this help file, by following the instructions in the topic dealing with [graphics](#).



The **Set Background Colours** function is on the **Format** menu, and presents a dialog box allowing you to set colours in terms of RGB (Red Green Blue) values, with four buttons to preset basic pale colours which you can further vary. To find the RGB values for a given colour, you may use the **Color** control of the Windows **Control Centre**.

If you have defined any [secondary windows](#), you will first need to respond to the dialog box determining for which window you are setting the colours.

It is probably a good idea to set up the overall structure of your help file and get it working before dealing with cosmetic matters like screen sizes and colours.

If you do not set a background colour, the WinHelp application will use whatever background colour the user has set as his/her windows default. If you are using colours for your text or graphics, they may not show up against the background the user has set. You might even want to force a white background using this function, to ensure that the help topics look the same to your users as they do to you. This help file has background colours set this way, both for the reason given and (as with many things in this file) to show what can be done using **HELLP!**

If you set a background colour using this function, you may wish to force the hotspot colour to be some other colour than the default dark green, or you may wish to prevent your user from changing it. This may be achieved by the [Set Hotpot Colour](#) option on the **Edit** menu.

Error Messages and Problems

Warning messages will **frequently** occur with the more sophisticated applications. Some of the things **HELLLP!** does to create a particular effect may cause a warning.

Error Messages may sometimes appear. They probably mean that you accidentally deleted a hidden code of some kind. **HELLLP!** uses these MsWord "Bookmarks" to hold values and as place markers. There should be several of them and a page break before the Automatic Table of Contents. There are also two within the text, one called "IwasHere" in the main text - used to keep track of where to come back to after doing something elsewhere, and one called "nextTOC", marking the location for the next entry in the Automatic Table of Contents.

HELLLP! tries very hard to restore any bookmarks that it finds missing, but it will not always succeed.

If you get a "**Bookmark not found**" when **setting up a topic**, you have probably deleted the "nextTOC" bookmark in the table of contents. To restore it, position the insertion point (The cursor) in the last entry of the Automatic Table of Contents and click the UPARROW button on the toolbar.

If your new Help File has lost the title from its title bar, you may re-create it with the **HELLLP! Project Title** selection on the Format menu.

The other bookmarks will be re-created when you perform an appropriate action. The "IwasHere" will be recreated when you create a new topic, jump or PopUp. Those which hold settings from checkboxes will revert to default values, and will be re-created when the values are changed again.

If you see a message "Out of File Handles" during the compile, you will need to increase the number in the **FILES=** line of your CONFIG.SYS file. **FILES=50** seems to work most of the time, but this will depend on what else is loaded (some systems have been reported to require **FILES=60**, which MicroSoft suggests for anyone using Word version 6, or even higher). If you are close to the limit of your number of file handles when you start a compile, strange things may happen depending on exactly how far the compiler gets before it hits the limit. **Increasing the FILES = number is a good thing to try for any strange error.**

The most frequent cause of an **out of memory** error in the compile, apart from the use of the wrong compiler, is the incorporation of too many large graphics. Larger graphics should be incorporated by reference, i.e. by converting the graphic to a BMP or SHG format file, putting that file in the same directory as the project files and including a line in the DOC file referencing it.

Another possible cause of an **out of memory** error in the compile is the use of the Fast Save option in Word. This saves a larger file than necessary and should be turned off. (It is also a security exposure in that deleted text may remain invisible in the file but viewable with a DOS file viewer).

Occasionally a Word error "**out of memory**" may occur during an initial install. This is frequently caused by having the TEMP environment variable pointing at a RAM disk. Word writes temporary files to this directory and reports a memory error when it fills it.

Should you receive error messages from the Microsoft help compiler which stop the compile, you should check the version of the help compiler and possibly download an updated version as described in Getting Started. Each version change in WinWord seems to require an updated compiler to handle the Rich Text File (RTF) form which it produces. To find the compiler, do a filename search on your favourite service for the name fragment **HCP**.

Occasionally, after some aggressive editing, you may do a compile which produces a help file with apparently only one topic full of **yes** and **no** switch values. This means that **HELLLP!** has lost track of which topic is supposed to display first, probably because a context string or a whole topic was deleted, so it displays the first in the file, which is where it keeps a whole bunch of settings (it has to work with Word 2.0 as well as 6.0 and 2.0 doesn't include document variables). The solution is to use the I button to tell **HELLLP!** which topic to use for initial display.

If you are using Word version 6.0, some of its "smart" features may cause problems. You should ensure particularly that its "smart cut and paste" feature is turned off (use the **Tools Options Edit** dialog box) or

some of your jumps may not work.

There may be other errors not listed here, no program is free of bugs including MS Word for Windows (As documented in the [Hacker's Guide to Word for Windows](#) - strongly recommended reading.)

DISCLAIMER: The supplier and author bear no liability to the licensee or any other person or entity for any damage or loss, including special, incidental or consequential damages caused by this product directly or indirectly. The software is supplied "as is" without warranty of any kind, either expressed or implied. Warranties of merchantability or of fitness for any purpose are specifically disclaimed.

HELLP! was inspired by a series of articles by Ray Duncan in [PC Magazine](#) in the spring of 1993, and by Woody Leonhard and Vincent Chen's excellent book the [Hackers Guide to Word For Windows](#).

This is a Pop Up Box. The Microsoft literature calls it a "definition box" because it's useful for definitions of terms, where it may be popped up without taking the user away from the context he or she was looking at. It's useful for lots of other things, though. Your imagination is the only limitation.

That's right, you just clicked on some text that will cause another topic to be displayed. Where it has a dotted underline, a box like this pops up, where it has a solid underline you are taken to the topic. Don't worry, you won't get lost, because pressing the Back button will retrace your steps. If you are using Windows 3.0 the Back button is a pair of footprints.

To make this box pop down again, click again.

Non-Scrolling Region

The top section of a topic may be set as a non-scrolling region - useful for titles and headings. This is done by setting the paragraph formatting for those paragraphs as Keep With Next. For the topic title, this is an option on the **Change Topic Title Style** function on the **Format** menu. The scrolling and non-scrolling regions may have different background colours if desired. Setting a non-scrolling section after a scrolling section may result in a compiler error message. **Note that any topic which may be called as a PopUp must not have any non-scrolling text or text below it will not be displayed.** If you are setting up a topic which may be called as a PopUp, either do not have **HELLLP!** insert a topic title, or manually reset the paragraph formatting afterwards.

The Question Mark Button brings up the Windows Help File for **HELLP!**. That is the file you are looking at now.

Word for Windows is often set up so that it automatically incorporates typesetters "curly quotes" instead of the familiar " and ' .

However, the Help system cannot display the curly quotes. **HELLLP!** therefore provides the means to convert them back to " and ' by using the **Undo Smart Quotes** feature on the **Tools** menu. If using Word 6 you must first turn off the Change "straight-quotes" to "smart-quotes" box in the Tools.Autocorrect dialog and in the Tools.Options.Autoformat dialog or Word will change them back again.

Hypertext is the term used for a method of moving around an on-line document to follow cross references, etc. In Windows Help, whenever a user clicks on an underlined green section of text, the application jumps to the page of the cross reference, but it keeps track of where it was. Users are able to retrace their steps using the Back button. Much better than keeping a thumb in the page, because you quickly run out of thumbs! If the green text has a dotted underline, then the reference "pops up" in a box like this. Think how useful this will be if you have lots of terms that mean things to your experienced staff, but not to newcomers.

This line of text is, of course, a graphic because if the text were used the picture would be here instead of the sample line!

This button lets you check the sound.

This button incorporates the sound into your file.

This button exits.

This box lists all of the fonts on your system. Some may not be suitable for titles, and you should ensure that you choose a font which your users will have on their machines.

This box list the font sizes available for topic titles.

This box lists all the colours available. You probably won't want to use the paler colours and you certainly won't want to use white.

These boxes allow you to choose these attributes.

This button implements the change.

This option sets the window size when WinHelp opens this file.

This option sets the window size when this topic is displayed.

This button obtains the old size settings for the selected window option.

These option buttons set the size of secondary windows, and turn on the options which deal with them.



Credit Card Orders Only

You can order with MC, Visa, Amex or Discover from Public (software) Library by calling **800-2424-PsL** or **713-524-6394** or by FAX to **713-524-6398** or **Internet Email** to **71355.470@compuserve.com** or by **CompuServe Email** to **71355,470**. (If you use another Email network, it probably has a gateway to Compuserve). You can also mail credit card orders to PsL at **P.O.Box 35705, Houston, TX 77235-5705**. In all cases please include the name exactly as on the card and the expiration date, in addition to the card number and your name and address (please include also the card billing address if different - this is a new bank requirement). For **Email orders** please include the word **order** in the subject line. For electronic registration, please remember to include your Email or Fax address for the registration code to be sent to.

The above numbers are for orders only.

Please quote **item/product code # 11177**

Any questions about the status of the shipment of the order, refunds, registration options, product details, technical support, volume discounts, dealer pricing, site licenses etc. must be directed to:



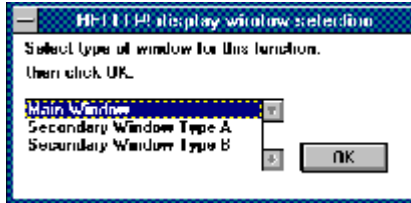
1752 Duchess Avenue
West Vancouver
BC Canada V7V 1P9
(604)926-1370 CIS 71750,1036

To ensure that you get the latest version, PsL will notify us the day of your order and we will ship directly to you.

Secondary Windows

The Windows help system has the ability to display secondary windows, in addition to main windows and popups.

To initiate the use of secondary windows, you use the **Set Help Window Size** function on the **Format** menu, to set the size and position for one or more Secondary Windows Types (identified as **windowA** through **windowE**).



Once you have defined secondary windows, it will be necessary to respond to this dialog box when performing some other functions, in order to identify which windows they apply to.

You may differentiate your secondary windows from your main window by giving them different background colours, using the **Set Background Colour** function on the **Format** menu.

You may choose to have a secondary window (one only), as well as or instead of the main window, displayed when Windows Help opens a help file. To do this use the [I] button and select the secondary window to be employed, and the topic to be displayed in it. **If you use this option you may not change the file name of the help file after compilation.** It is frequently convenient to display a "Sidebar Table of Contents" in a secondary window, and use it to control topic display in the main window.

If you set a secondary window for initial display you are given the option to hide the main window until it is called by a jump from the secondary window.

Exit




Welcome to the world of HELLLP!

© Edward Guy, CDP 1993/4 All Rights Reserved

To get help about the use of the **HELLLP!** system at any time, just click on the question mark button on the toolbar in Word.

To view the **HELLLP!** help table of contents at any time, click on the  button of this window.

The **HELLLP!** system needs to know where you want it to keep its working files and will ask for the name of a subdirectory for that purpose. You should have set up a directory for that purpose separate from any other files, to keep all of your **HELLLP!** files in. **HELLLP!** will create files with the extensions **DOC**, **RTF**, **INI**, **HPJ**, **BAT**, **ERR** and **HLP** for each help project in this directory. **If you have not set up such a directory, you will not be able to complete this setup process and you should click  now, to give yourself the opportunity to create one.**

HELLLP! also needs to know the name and location of the copy of the Microsoft Help Compiler on this machine, and will present a dialog box asking for that information.



This is a secondary window.

Unlike a PopUp, it will not go away when you click again. It must be specifically de-activated.

To use secondary windows in your application, define a size and position for it/them using the [Set Help Window Size](#) function on the **Format** menu. After you have done this, you will be required to tell **HELLLP!** which window type (main or one of the secondaries you have defined) you wish to use for every jump you define and for certain other functions. A dialog box will be presented for this purpose.

You may specify sizes and positions for up to five secondary windows, but only one may be displayed at once. [Click to bring back the sidebar table of contents.](#)

If you are creating a multi-project file build, the window types you are going to use should be defined in each of them. During the compile, the sizes and positions set for the file from which the compile is called will be the ones used.

Customising lines in the HPJ file (for experts)

The expert user may wish to add lines to the **HPJ** file which are not generated by **HELLLP!**

This may be achieved by creating a file in the same format as the **HPJ** file, but with the extension **HPK**. When **HELLLP!** creates the **HPJ** file, it checks for the existence of an **HPK** file and merges the lines from it into the appropriate sections of the **HPJ** file.

These lines are added after any lines which **HELLLP!** puts into each section.

Before using this option, you should look at the **HPJ** file created by **HELLLP!** to ensure that any lines you put into the **HPK** file do not conflict with them.

For example, you might wish your file **BIGFILE.HLP** to be optimized for display from CD-ROM, but **HELLLP!** does not support this directly in order to keep its interface simple. You would create a file **BIGFILE.HPK** using an ASCII editor like Notepad consisting of the lines:

```
[OPTIONS]  
OPTCDROM=YES
```

and include it in your project directory. That option would then be included by **HELLLP!** in **BIGFILE.HPJ** when it runs the compile of **BIGFILE.DOC**.

A rectangular button with a grey gradient and a black border, containing the text "Contents" in a black sans-serif font.A rectangular button with a grey gradient and a black border, containing the text "Close" in a black sans-serif font.A rectangular button with a grey gradient and a black border, containing the text "Exit" in a black sans-serif font.

This is a secondary window to show this rather wild effect. You will have to close it to get rid of it (or else click CONTENTS). For more on secondary windows, [click here.](#)

Title Bar Text Setting

You may set the text to be displayed in the title bar when Windows Help displays your help file. The function is found on the **Format** menu as **HELLLP! Project Title Bar**.

Version History (What's New)

Version 1.0	First distributed version.
Version 2.0	Added sound effects and external process calls.
Version 2.1	Added secondary window functions, background colour control and button bar control.
Version 2.2	Added support for non-English versions of Word for Windows.
Version 2.3	<p>Added enhanced <u>button bar control</u> and <u>WinHelp macro</u> call functions.</p> <p>Additional buttons may now be added to both main and topic button bars, with user-defined text on the button tops for all buttons (may be used to create non-English buttons).</p> <p>Side Bar table of contents may be created automatically for secondary window or popup.</p> <p>WinHelp macros may be set to run on startup or invoked from any hotspot or button and may be chained so that one click causes several actions (e.g. display different topics simultaneously in main, secondary and popup windows).</p> <p>The capitalization pattern for search keywords may be changed.</p> <p>Symbols commonly used for "Bullets" are processed as bitmaps which the Help compiler can handle.</p> <p>A "project accounting" option which provides a count of topics, jumps/popups and macro/special effects calls is included.</p> <p>HotSpot colours may be fixed or changed.</p> <p>Topics may be automatically picked from Winword heading styles.</p>
Version 2.3a	<p>Improved browse sequences (multiple sequences in one file).</p> <p>Added ExecProgram() macro to those connectable to buttons.</p>
Version 2.4	<p>Added feature for jumps to mid-topic targets.</p> <p>Added support for languages where the measure "point" does not translate to "Pt" (e.g. in Spanish it is "Pto").</p> <p>Added support for EWDLL.DLL to allow display of 256 colour bitmaps.</p> <p>Added glossary button box and automatic glossary skeleton.</p> <p>Added feature to import files with context strings etc. already defined. e.g produced by other tools.</p> <p>Added direct support to generate accelerator keys.</p> <p>Added support for menu control.</p>
Version 2.4a	Added function to omit menu bar totally.
Version 2.4b	Fix problem with I button if only one file in directory.
Version 2.5	<p>Re-Release of 2.4b to satisfy needs of automated distribution networks which need a new filename.</p> <p>Permit confirmation box to be turned off in "Pick Topic From Style".</p> <p>Added toolbar button for graphics by reference.</p>
Version 2.5a	<p>Added ability to disable/remove buttons.</p> <p>Fixed problem with external jump into glossary topic.</p> <p>Fixed problem with button-top text in some languages.</p>
Version 2.5b	<p>Amended MapTopicForApplication to take large hex numbers.</p> <p>Increased to 120 the number of external files which can be linked into one compile.</p>

\$HAZAM!

Use of WinHelp Macros

The Windows Help Compiler permits the inclusion of macros in the resulting windows help file. These are pre-defined actions to be taken in defined circumstances, usually by click of a hot spot or button.

Macros may be chained together in some circumstances, permitting for example the simultaneous changing of several windows with one action (click the **SHAZAM!** button at the head of this topic for an example).

Additionally, up to four macros may be set to run when WinHelp is loaded. These macros may be defined by clicking the **[I]** button, which is also used to set the initial screens to display. After macros are defined (you may leave them undefined if you wish), you may click the OK button to avoid setting initial screens, or click the long "Set initial topics" bar to set initial display topics in either main or secondary windows. It takes two clicks of the **[I]** button to set initial topics for both main and secondary windows.

When a function can incorporate macro code (e.g. the **Set Button Selection** option), a dialog box presenting the macro option is presented to you. You may select the appropriate macro from the dialog box, or you may select the last option on the list, **"Manually Enter.."**. The last option means that you are **really** performing without a net, so use it only if there is no other way of achieving your purpose.

Selection of certain macros will cause the display of further dialog boxes to determine the parameters needed for the macro.

The **SHAZAM!** effect was created by selecting **Set Button Selection** from the **Format** menu, then selecting **Topic Button Bar** then selecting **Head of Topic**. One of the JUMP buttons was checked and its button text was changed to **SHAZAM!** before clicking OK. The macro dialog option Jump to Topic was selected and the "chain" box was checked before clicking OK. The topic was chosen and a secondary window selected. The process was repeated except that the main window was selected. The process was repeated except that the PopUp action was selected. The process was repeated except that the **sndPlaySound** macro was selected and the "chain" box was not checked. Thus four macros were chained. When the button is clicked they run in sequence. The **SHAZAM!!** menu item was created in the same manner using the **SetMenuSelection** function from the **Format** menu.

To permit the connection of macros to regular hotspot text or graphics, you may use the "Screen" button (which also permits the launching of external processes) for this purpose.

Please note that some macros will not work when called from secondary windows, and that some combinations of chained macros will not work properly. **Test well.** We're really pushing the envelope here folks!

Just like magic! This is the effect of three chained macros. You may now click the back button to return to your previous screen. Note that the button you clicked had customised text (**All Right!**) on it. It could have been in French or German or anything else you desire, provided that its letters are available in the ARIAL TrueType font.

Back

Who said you can't have buttons in a PopUp window?

Bienvenue à **HELLLP!**

Voici un bouton customisé en Français.

WinHelp Help Compiler Limits

Here is a Microsoft Knowledge Base article on Help limits (© Microsoft).

The following lists the limits of the Windows Help application, version 3.1. These limits are subject to change in future versions.

Paragraph size(1): approximately 32,000 bytes

Topic title string length: 127 characters Topics per RTF source file: approximately 32,700 Topics per help file: 42,000,000

By-reference bitmaps per help file : 32,767

Help file size: approximately 2,157,000,000 bytes (2 gigabytes)

Keywords per help file: limited only by file size

Keyword length: 255 bytes

Topics per keyword(2): 32,767

Browse sequence length: limited only by the number of topics

Bookmarks per help file: limited only by file size Annotations per topic: 1

Annotations per help file: 32,767

History list length: 40 topics

Text that can be copied to clipboard: limited only by machine memory

Macro length: 512 characters

Main windows per topic: 1

Secondary windows per topic: 5

(1) The maximum paragraph size includes graphics pasted into the RTF source. Therefore, each pasted graphic is limited to 32,000 bytes. To work around this limitation, import large bitmaps by reference.

(2) Performance degrades for files with more than several hundred topics. The Search dialog box displays a maximum of 400 topics.

Choosing the Capitalization (Case) of Winhelp Search Keywords

HELLLP! generates keywords for use by the WinHelp search box by picking out words from the topic title assigned when the topic is generated. To maintain a standard capitalization pattern, it automatically converts the words to upper case. To change the capitalization pattern used, you may select the item **Help Keyword Case Choice** from the **Edit** menu. This gives the choice of all uppercase, all lowercase or mixed case (initial letter uppercase, others lowercase). When an option is selected, all keywords already picked for the current file are converted to that pattern and new ones created for that file are created with that pattern.

Project Accounting

HELLLP! can measure the amount of work you did with it for accounting or billing purposes. The item **HELLLP! Project Accounting** on the **File** menu will count the number of topics, jumps/popups, macro/sound/external calls for you.

Bullets and Symbols

The Microsoft Help Compiler will not handle the symbol fields used by Microsoft Word. (Word places fields in the file where symbols are to go - you can see them if you set View Filecodes On.) Since Word 2.0 uses these fields to define bullets, this causes a problem when a Word 2.0 file with bullets is submitted to the Help compiler - the bullets get ignored.

Word 6.0 defines bullets differently, but still has a problem with the Help compiler.



HELLLP! provides its own bulleting routine, called from the



button of the HELLLP! toolbar. This routine provides a choice of the symbols defined below and various indent distances. When called, the routine first removes any existing bullets from the highlighted section of text and then inserts its own bitmap bullets. There is an option to use a special bullet called a "Push Button" (



) which may be used as a hotspot for jumps, popups, macros and other effects.



HELLLP! replaces symbol fields with bitmaps immediately before the compiler is called. The font name and character number are extracted from the symbol field and concatenated to form a glossary entry name. For example, the symbol ♦ appears as {SYMBOL 117 If "Wingdings"}. The glossary name **Wingdings117** is manufactured and a bitmap of that name is sought in the glossary and used to replace the symbol. If an appropriate glossary entry is not found, the symbol Wingdings111 (□) is used.



The following symbols are currently defined:



● (Wingdings # 108)



■ (Wingdings # 110)



□ (Wingdings # 111)



♦ (Wingdings # 117)

HotSpot Colours

WinHelp hotspot text is usually coloured dark green. However this may be changed by the users' choosing some other colour setting using the JUMPCOLOR setting in WIN.INI. If the help file author has chosen a background colour which is not compatible with this color, then the user may have difficulty reading the file.

Sometimes the help file author has some reason for wanting the hotspot colour to be something other than dark green.

The **Set Hotspot Colour** function on the **Edit** menu covers both these eventualities. The colour selected may not be changed by the user to anything other than dark green, and then only by using the COLORS=NONE option in the [WINDOWS HELP] section of WIN.INI.

The colour chosen will be used for hotspots defined after its selection. Another colour may then be selected if desired so that a mixture of hotspot colours will occur in the same file. Selecting AUTO colour turns off the option.

This feature should be used sparingly, and it is advised that it be mainly used to force the hotspots to dark green if the **Set Background Colours** option of the Format menu has been used and there is a concern that the user will set JUMPCOLOR to an incompatible value.

Heading styles to pick topics

If you are importing a document which you created in Microsoft Word, you may have headings in it set up using the Styles feature of Word. If so, **HELLLP!** is able to use these styles to identify topics automatically.

The function **Pick Topic From Style** on the **Insert** menu permits this. When this option is selected, you are presented with a list of the styles used in the document (except the **normal** style and the **Topic Title** style which **HELLLP!** generates). Occurrences of the style selected are automatically highlighted, converted to the **Topic Title** style and then presented as suggested topic titles, just as if you had highlighted them and clicked the **T** button.

Importing files

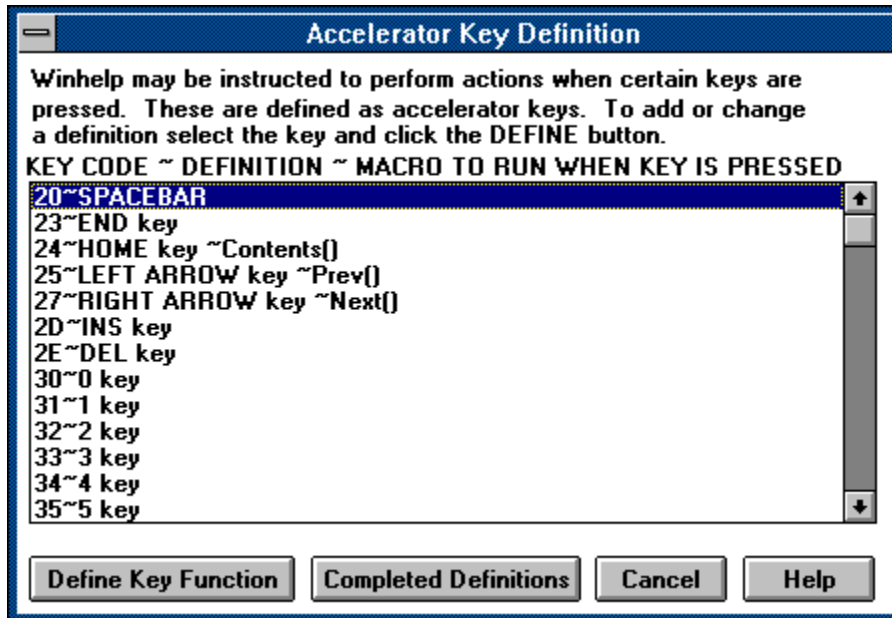
Sometimes you may wish to import a file (either an RTF or a DOC file) which already has some context strings and jumps etc. defined in it. These will work when you compile, but the context strings will not be presented by **HELLLP!** if you wish to add additional jumps to them. To remedy this, the function **Find Foreign Contexts** is provided on the **Insert** menu. It will scan the file for context strings not inserted by **HELLLP!** and will record them in your project INI file. To jump to them, even from inside the file containing them, you will need to use the **External Jump** feature, in whose dialog box they will be identified as "FOREIGN CONTEXT IN FILE". To assist you in assigning more meaningful names, the process will pause as each is displayed and ask for a mnemonic phrase. It will provide as a default choice the next \$ footnote in the file which will usually be the "title" string assigned to this topic.

If the file you import has a topic which you wish to use as an initial topic (e.g. it may have its own table of contents) you may select that topic as the initially displayed topic by using the **[U]** button on the **HELLLP!** toolbar.

Accelerator Keys

In addition to using the mouse to click on buttons and hotspots, you may use the keyboard to cause actions to happen in WinHelp. Keys set up this way are known as **Accelerator Keys**. When you assign text to a button in the main button bar, you may make any letter in the button title an accelerator by preceding it with a **&** character where it appears in the button legend. Most of the keys on the keyboard may be set up this way.

Additionally keys may be defined as accelerators even if they are not used in button legends. In this file the left arrow and right arrow keys have been set up as accelerator keys for the left and right browse buttons and the **HOME** key has been set up to return to the main title screen.



Accelerator keys are defined using the **SetAcceleratorKeys** function on the **Format** menu. Invoking this produces a dialog box showing the keys available for setting up as accelerators and any actions which you have already associated with them. Those keys assigned to the standard buttons are not presented for re-assignment.

You should be careful not to assign the same key as an accelerator for two different functions, since strange things may result.

For experts who wish to do more sophisticated programming: You may insert **AddAccelerator** macro code anywhere that you can define a WinHelp macro in **HELLLP!**. In particular you should refer to the topics concerning the **I** button and the use of an HPK file.

Adding Menu Items

You may add items to the drop down menus which WinHelp displays, but you may not delete the standard items unless you omit the whole menu bar. You may change the actions of two of the items on the **Help** menu item. You may also add one or two more drop down menus to the menu bar.

The item **SetMenuSelections** on the **Format** menu provides the functionality to do this. It presents the following dialog box:

HELLLP! WinHelp Menu Definition

A	B	C	D	E
File	Edit		Help	SpecialHelp
Open	Copy	[]	HowToUse	Adding Me
PrintTopic	Annotate		AlwaysOn	Getting Sta
PrintSetup	[]		AboutHelp	Make A He
Exit			HELLLP! T	Problem Sc
[]			[]	SHAZAM!!
				Creating S

Click on REFRESH after making item selections in boxes above.

REFRESH detail of menu items below ☐ OMIT ALL MENUS

A: File Item [PrintSetup] = PrinterSetup() [CAN'T CHANGE]
 B: Edit Item [Copy] = CopyDialog() [CAN'T CHANGE]
 C: Item [] = [UNASSIGNED]
 D: Help Item [HowToUseHelp] = JumpHelpOn()
 E: SpecialHelp Item [SHAZAM!!] = JI('hellp!.HLP>windowB', 'HELLLP.SI

Select menu item to process in box above then click button below.

OK **Change Legend** **Change item action** **Delete item** **Cancel**

On initial display only the standard menu items are shown. To add a new item to a drop down menu, type a title in place of the [] at the bottom and click **Refresh**. To see the action associated with an item, select the item and click **Refresh**. The macros associated with all selected items are listed as **A, B, C** etc. To change one of them, highlight it and then click the appropriate **Change Item Action** button and you will be presented with **HELLLP!**'s macro selection dialog box. You may thus associate a macro with a menu item.

To add a new drop down menu, type a title for it in the box in the top line and then click **Refresh**. **HELLLP!** will not add items to the standard **Bookmark** menu, so it is not shown. If a title is typed in the box in the centre of the menu title list, a new drop down menu with that name will be inserted between the **Edit** and **Bookmark** standard menus. The box at the end will add a menu to the right of the **Help** item on the standard menu.

If you change your mind about the legend for one of the items you inserted in a list box, simply ensure that it is one of the legends highlighted in the upper boxes, click **REFRESH**, highlight it in the lower box and click the **Change Legend** button to alter it or the **Delete Item** button to remove it.

Please remember that the item you wish to process must be highlighted in the lower box before you click an action button.

For experts who wish to do more sophisticated programming: You may insert menu manipulation macro code anywhere that you can define a WinHelp macro in **HELLLP!**. In particular you should refer to the topics concerning the I button and the use of an HPK file.

Exit



Version 2.5b

About HELLLP!

© Edward Guy, CDP 1993 All Rights Reserved

HELLLP! is a user-friendly system to aid in the production of Windows help files. It requires Microsoft Word for Windows version 2.0 or higher. It will work with versions of Word for most western languages. Registration codes for the current version will be honoured by the next version but it is the user's responsibility to obtain a copy.

You have just invoked the file **HELLLP!.HLP** in the **WinHelp** application by clicking a button in **Word for Windows**. This topic was specified as the topic to show for that button by mapping it for context sensitivity. You can do the same thing from your own applications. To see this help file at any time, click the **[?]** button on the **HELLLP!** toolbar. The table of contents appears in the sidebar at the left of the screen. Click the **Close** button to close an individual window, click the **Exit** button to close both.

To see an extreme example of the effects you can create, just click the **SHAZAM!!** item on the **SpecialHelp** menu.

HELLLP! is available as shareware from the Microsoft Word for Windows forum (GO WINWORD) and the Windows Software Developers Kit Forum (GO WINSDK) on Compuserve as the file HELLLP.EXE. The author and copyright owner may be contacted at Compuserve ID 71750,1036.

HELLLP! is available as shareware via the Internet from any FTP site which carries the SimTel collection (including OAK.Oakland.Edu in SimTel/msdos/windows3/), as HELLLPnn.ZIP where nn is the version number.

HELLLP! is available as shareware from many major bulletin boards and on many CD-ROM collections.

To assist in finding the latest version of **HELLLP!**, we suggest that you do a partial file name search for **HELLLP**. On the Internet, ARCHIE will find you several copies.

Just like magic! This is the effect of four chained macros. (You may now click the **Back** button to return to your previous screen).

You wouldn't necessarily want to do this often, but it shows what YOU can do with **HELLLP!**.

The background effect was produced using a table as described in the topic on graphics.

Back

Who said you can't have buttons in a PopUp window?

Fancy Macro Demonstration.

Macros are extremely powerful and **HELLLP!** provides a user-friendly interface to allow you to use them. They can be linked to run in sequence in many cases. You will need trial and error to determine whether a particular sequence will run.

This demonstration shows that one action can display three windows to your user.

You may run a macro or macro sequence from any hotspot text or graphic (by using the SpawnExternalProcess item on the Insert menu or by clicking the Screen button on the toolbar), or from a button you insert in your topic or in the main WinHelp button bar or from a menu item which you add to the main WinHelp menu).

To see the PopUp again, click the button.

PopUp

Count 'em. Three different windows and a sound effect (if you have sound enabled). All triggered by one action. Click anywhere to dismiss the PopUp. Click **Back to return to your previous screen.**

ADVERTISEMENT FOR

RACONTEX Full Text Search Product



Inexpensive competition for FolioViews, AskSam, Zyindex etc. and for mainframe products like STAIRS. RACONTEX provides full text search capabilities under both DOS and Windows. This product was used to produce the databases-on-disk in the back of QUE Books "Using the Internet".

Many business uses. In addition to the obvious uses for statutes, legal transcripts, etc., Racontex is valuable for use with collective agreements, policy manuals, office handbooks, library catalogues -- in fact anywhere that someone has to "look something up" and there is a personal computer available. Ideal for online manuals for software!

Contextual Search Tool. Does not lose the context of pages found, can "page up" and "page down" from each occurrence of the search criteria, (even if those pages do not contain the search criteria) and return to it instantly by pressing one key. Moves to next or previous incidence of search criteria by forward arrow or backward arrow keys.

Proximity Search. Supports searches of word alternatives and combinations of alternatives within a given number of words of each other, or on the same page as each other.

Variable Stopword List. Licensees of the loading (indexing) program may specify their own list of stopwords (words not to be included in searches, such as "and" and "the") - this is a tremendous disk space saver.

Variable Delimiter List. Licensees of the loading (indexing) program may specify their own list of delimiters. e.g. They may determine whether a hyphenated word is one word or two for search purposes.

Optional copy prevention. The resulting self searching documents may be loaded onto computers other than the original (subject to the appropriate licence) and then made "non-portable" to prevent unauthorized copying. Useful for significant databases in companies with high staff turnover.

Easy to Add to Files. Additional sections can be added easily to a document in the system. Ideal for daily transcript of an ongoing hearing. The whole record of a hearing can be built up on the hard drive of a laptop, and passages accessed in less than a second.

Windows Interface. Windows interface in same package as DOS version.

Inexpensive. Single copy registration price, \$99.95 includes both indexing and searching modules. Special prices for those wishing to publish in Racontex searchable form.

"Try Before You Buy". Thirty day evaluation copies available from "Shareware" vendors, bulletin boards, CompuServe (as RACONT.ZIP), Internet (SimTel sites) as RCNTXnnn.ZIP. To find Racontex on your favourite BBS, do a filename fragment search for **RCNTX**. These copies display an advertising screen at the start of a search.

Mid-Topic Jumps

You may sometimes wish to jump into the middle of a long topic. In order to do this you must establish a **MidTopic Target** to jump to. These are set up in similar manner to topics, but do not cause a page break and the titles you give them are inserted in hidden bold text for your convenience in reference. These titles appear in the automated table of contents just like topics, and may have keywords associated with them.

To set up a MidTopic Target, use the item **InsertMidTopicTarget** on the **Insert** menu. Any highlighted words will be used as a suggested title, just as with a topic.

As an example of a mid topic jump, this next jump will take you part way into the topic regarding [registration via CompuServe](#).

Notice how you may scroll up from the target as well as down. This feature is convenient for glossaries and indices, where you may jump to the set of words beginning with a given initial. HELLLP! provides an [alphabetic button box](#) and the means to generate a glossary skeleton automatically.

Click on one of these buttons to assign a jump to a topic or a mid topic target in this or another file. If you have assigned a topic to the button already, you will be given details of the assignment and offered the option to change it.

This box, when checked, permits the choice of a jump to a topic in another file. If a jump to another file is selected, the whole of that file will be included in the compile, even though some topics may not be reachable by the user.

This button will insert the glossary button box into the top of the current topic and will also set up headings A, B, C, D etc in the Glossary Title style (initially identical to Topic Title, but you can change it), with mid-topic jumps to each from the corresponding button in the box. Specific button targets may be overridden by jumps assigned by clicking specific buttons. Normally the ? button will be overridden with a jump to the help topic dealing with the glossary.

This button will insert the button box in the top of the current topic without setting up a glossary skeleton. The box may be inserted in more than one location but the jumps will always point to the same targets.

This button will make the changes just specified without inserting a new copy of the button box.

This button will cancel the operation without writing any of the changes just made.

This button calls the help topic you are now viewing.

yesyesyesyesHELLLP! Generatedwindsize

Table of Contents

Top Margin
Bottom Margin
Right Margin
Left Margin
PopUpSection
The Window Size Dialog Box

Help file produced by **HELLP!**, a product of Guy Software, on 01/02/94 for Edward Guy.

The above table of contents will be automatically completed and will also provide an excellent cross-reference for context strings and topic titles. You may leave it as your main table of contents for your help file, or you may create your own and cause it to be displayed instead by using the I button on the toolbar. This page will not be displayed as a topic.

HINT: If you do not wish some of your topics to appear in the table of contents as displayed to your users (you may want them ONLY as PopUps), move the lines with their titles and contexts to below this point. If you do this remember to move the whole line, not part.

PopUpSection
Left
Right
Top
Bottom









The Window Size Dialog Box

It is easier to show how this box works than to explain in words. Therefore the buttons themselves are live to show which one affects which dimension.

FALSEyesyesyesyesHELLLP! Generatedhypertxt

Table of Contents

[Hypertext Document Structure](#)
[automated table of contents](#)
[Author's table of contents](#)
[lower level topics](#)
[intermediate topics](#)
[automatic hypertext jumps](#)
[hotspots in topics](#)
[authors hypertext jumps](#)
[Ed Guy](#)
[Business Card](#)

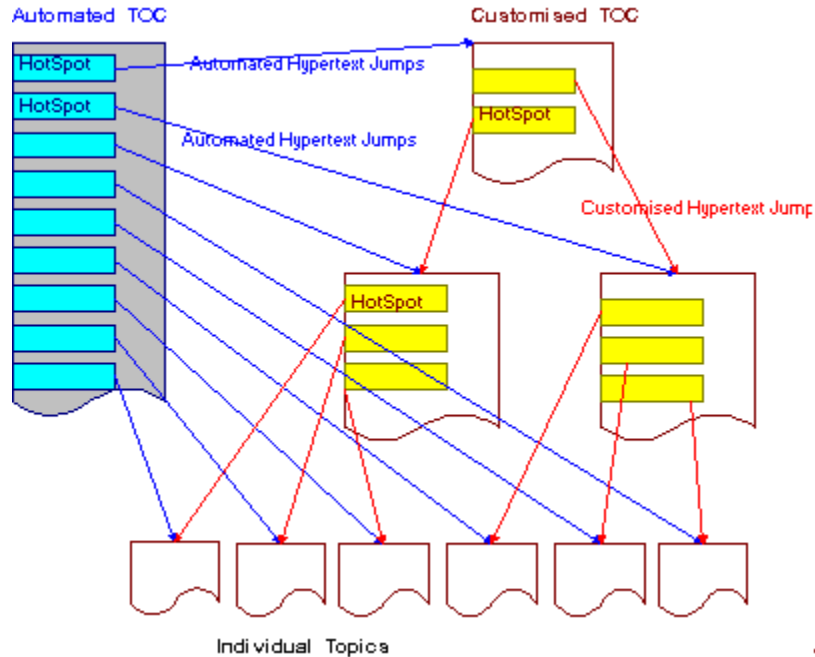
Help file produced by **HELLLP!**, a product of Guy Software, on 04/08/94 for Edward Guy.

The above table of contents will be automatically completed and will also provide an excellent cross-reference for context strings and topic titles. You may leave it as your main table of contents for your help file, or you may create your own and cause it to be displayed instead by using the I button on the toolbar. This page will not be displayed as a topic.

HINT: If you do not wish some of your topics to appear in the table of contents as displayed to your users (you may want them ONLY as PopUps), move the lines with their titles and contexts to below this point. If you do this remember to move the whole line, not part.

Hypertext Document Structure


A windows help file usually contains several levels of hypertext "menus", in which there are jumps from a top-level menu to one or more other levels of menu. **HELLLP!** makes customization of this structure very easy. The cool colour "blue" items on the structure below are automatically created by **HELLLP!**, the hot colour "red" items are inserted or imported by the author. This drawing is a hypertext picture. Just click on an item for an explanation of its place in the scheme of things.



This table of contents is created by HELLLP!, and it has automatically created hypertext jumps ([shown in blue](#)) to every other topic set up by HELLLP!. Unless the Help file author specifies otherwise, this is the table of contents which is first displayed when the help file is invoked. For a simple help file, the author may be satisfied with this. However most authors will want to create their own structures. The automated table of contents is able to assist in this, since it has every topic listed along with its context string, all in one place.

When an author clicks on the [\[J\]](#) or [\[P\]](#) buttons, he/she is taken to this table of contents to select the target of the jump or PopUp, since titles are easier to remember than context strings.

The red jumps are those inserted by the author, the maroon and yellow topics are those inserted or imported by the author.

This is the Author's Table of Contents or other opening screen. Since no automatic hypertext jumps go **to** the automated table of contents and it does not appear in the keyword list or browse sequence, there is no way in which it can appear to the ultimate user if the author uses the  button to set this as the initially displayed topic.

These are the lower level topics, but even they may have hypertext jumps or PopUps leading from them.

These are intermediate levels of "Menu" set up by the author. There may be many of these (in the thousands if needed).

These are hypertext jumps which are automatically set up by HELLLP!

These are HotSpots, either text or graphics, included in the text. The jump occurs when the user clicks the mouse pointer on one of them.

These are hypertext jumps which are set up by the author, using the Point & Shoot techniques provided by HELLLP!.



Edward Guy P. Eng., CDP
Information Technology Consultant

yesyesyesyesyesHELLLP! Generatedpayup08/06/94

Table of Contents

Registration Reminder

Help file produced by **HELLLP!**, a product of Guy Software, on 08/06/94 for Unregistered User.

The above table of contents will be automatically completed and will also provide an excellent cross-reference for context strings and topic titles. You may leave it as your main table of contents for your help file, or you may create your own and cause it to be displayed instead by using the I button on the toolbar. This page will not be displayed as a topic. It is given a context string of __ and a HelpContextID property of 32517, but these are not presented for jump selection.

HINT: If you do not wish some of your topics to appear in the table of contents as displayed to your users (you may want them ONLY as PopUps), move the lines with their titles and contexts to below this point. If you do this remember to move the whole line, not part. As an alternative, you may wish to set up your own table of contents, see Help under The Structure of a Help File.

Do not delete any codes in the area above the Table of Contents title, they are used internally by HELLLP!



EGISTER HERE

HELLLP! has been made available to you on a 30 day free trial based on the [Shareware principle](#). According to our records that period has now expired.

You may legally continue to use the product while your payment and registration is being processed, but otherwise you should cease its use and remove it from your machine.

You may make your payment by any of several methods:

[Credit Card Orders by Telephone](#)

[Cheque or Money Order by Mail](#)

[CompuServe SWREG Service](#)

