The EditHelp program

The EditHelp program creates Help Files for use with the Windows help system. It can also create files in HyperText Markup Language (HTML)   format.

The Help File may contain links, popup-links, bitmaps and keywords. As you develop your Help File, EditHelp helps you keep track of the status of each topic.

EditHelp can operate in two modes: 'Windows Help' and 'WWW'. This Help file describes how EditHelp behaves in the 'Windows Help' mode. To change mode, select the File|WWW Mode command.

The Examples topic shows many of the features available from EditHelp.

It is recommended that you use EditHelp with a Help Compiler such as Microsoft's HC31. You may already have HC31.EXE (or HC30) on your computer. EditHelp also has an internal Help Compiler.

To find out more about using EditHelp to create Help Files, see the following topics:

Files
Topics
Text Layout
Text Commands
Make HelpFile
Keywords
Projects

If you have already developed Help files using other tools, you can use the File|Open HLP command to convert them into EditHelp format.

The source file for this help is supplied with this program as

edithelp.edh

See also
Menu Commands

File|Open HLP

The File|Open HLP command reads an existing HLP file and converts it into the source format used by EditHelp.

The Open HLP command does not neccessarily create an EditHelp source which can be re-compiled to give exactly the same HLP file - but it should be close.

If you have already developed Help files using other tools, you can use the Open HLP command to convert them into EditHelp format.

If the source file name is

    program.hlp

then the new source file will be

    program.edh

The topics are numbered sequentially from 1.

See also
- File|New
- File|Open
- File|Save
- File|Save As
- File|Project|New
- File|Project|Open
- File|Project|Save
- File|Project|SaveAs
- File|Project|Files
- File|Project|Move Topic
- File|Save As TXT
- File|Save As RTF
- File|Save As WWW
- File|WWW Mode
- File|Open EXE
- File|Exit

Topic|Entry Command

If the Entry Command string is set to a Help Macro command, then the command will be exectuted when the topic is displayed in the main or secondary Help window.

Entry Commands are not run if the topic is displayed in a Popup.

Multiple commands can be separated by semicolons ';'.

The command may be any Help Macro. You can call a DLL by executing the RegisterRoutine macro then calling the routine, for instance:

{link=!RegisterRoutine("sample.dll","FUNC1","i"); FUNC1(1234)!,Call a DLL}

This topic has its Entry Command string set to:

PopupId(qchPath,"PG117")

which displays topic 117 as a popup whenever the topic is displayed. The **qchPath** Predefined Variable passes the name of the current help file to the Help Macro.

See also
- Topic|Next topic
- Topic|Prev topic
- Topic|Back
- Topic|New topic
- Topic|Goto
- Topic|Goto Unfinished
- Topic|Delete
- Topic|Caption
- Topic|Contents
- Topic|Browse
- Topic|Status
- Topic|Preview

Internal Compiler

The Internal Help Compiler converts the current source into a Help file.

It is recommended that you use the HelpFile|Make command which call an external Help Compiler such as Microsoft's HC31. You may already have HC31.EXE on your computer.

The internal Help Compiler is quicker and more convenient than HC31.EXE but the it is still under development. Microsoft do not publish the specification of HLP files so the internal compiler is "experimental". (Please report any bugs to Analogue Information Systems.)

Whether the External or Internal Compiler is used can be selected via the HelpFile|Directories dialog.

The internal Compiler does not support the following features:

Literal

ALIAS.EPJ

Each topic must have a unique Topic number. The number is used when the host program calls the help system to display a help topic.

If you have written a program which must use context strings rather than context numbers, you can use the ALIAS.EPJ file to add an [ALIAS] section to the Help Project File.

The [ALIAS] section assigns one or more context strings to the same topic. Normally, the [ALIAS] section of the Help Project File is empty. If the file

alias.epj

exists in the same directory as the EditHelp source file, then the text of ALIAS.EPJ will be copied to the [ALIAS] section of the Help Project file.

The ALIAS.EPJ file should contain one or more lines with the format:

context_string = alias

Where:

context_string	Specifies the context string that identifies a particular within the HLP file topic.
alias	Specifies the alternative string or alias name.

An alias string is not case-sensitive and may contain the alphabetic characters A through Z, the numeric characters 0 through 9, and the period and underscore characters.

The context string used within the HLP file is the Topic number preceded by 'PG'. For instance, if the topic number is 261 then the context string is

PG261

So if your program displays topic 261 with the context string 'Shortcuts' then ALIAS.EPJ should contain the line

PG261 = Shortcuts

See also
- topics
- Topic number
- Help Project File

File|Save Project As

The File|Save Project As command saves the current Project to a new file. The Save Project As dialog allows you to choose the name of the file.

Project files have the extension '.EHP'.

If any on the source files in the Project have been changed, you will be given the opportunity to save them.

See also
- File|New
- File|Open
- File|Save
- File|Save As
- File|Project|New
- File|Project|Open
- File|Project|Save
- File|Project|Files
- File|Project|Move Topic
- File|Save As TXT
- File|Save As RTF
- File|Save As WWW
- File|Open EXE
- File|WWW Mode
- File|Exit

File|Save Project

The File|Save Project command saves the current Project. If the current Project is unnamed, you will be asked to choose the name of the file.

Project files have the extension '.EHP'.

All of the source files in the Project will also be saved.

See also
- File|New
- File|Open
- File|Save
- File|Save As
- File|Project|New
- File|Project|Open
- File|Project|SaveAs
- File|Project|Files
- File|Project|Move Topic
- File|Save As TXT
- File|Save As RTF
- File|Save As WWW
- File|Open EXE
- File|WWW Mode
- File|Exit

File|New Project

The File|New Project command starts a new Project.

A "Project" consists of a Project file and a set of source files.

To create a new project, open a source file, then select the File|New Project command. The source file will be the first file in the new project.

You can then use the File|Project Files command to add or remove other source files.

See also
- File|New
- File|Open
- File|Save
- File|Save As
- File|Project|Open
- File|Project|Save
- File|Project|SaveAs
- File|Project|Files
- File|Project|Move Topic
- File|Save As TXT
- File|Save As RTF
- File|Save As WWW
- File|Open EXE
- File|WWW Mode
- File|Exit

Projects

A "Project" consists of a Project file and a set of source files.

When the HelpFile|Make command is executed, all of the source files are linked together to form a single Help file.

For small Help files, you will not need to use Projects. Use a Project when you want several source files to be shared by several Help files: perhaps when you have different versions of a program with different combinations of features.

To create a new project, open a source file, then select the File|New Project command. The source file will be the first file in the new project.

You can then use the File|Project Files command to add or remove other source files. You may not have a Project which contains no source files nor include the same source file twice.

The source file which the current Topic belongs to is shown at the bottom of the main window.

You can move Topics from one Project source file to another with the File|Move Topic command. You may not move all the topics out of a source file; it is illegal to have a source file with no Topics.

Save your project with the File|Save Project or File|Save Project As command. You may open it again later with the File|Open Project command.

Some of EditHelp options refer to individual Topics while some refer to the whole Help file. When you open a Project file, the Help file options are read from that file. If you add other source files to the project, the Help file options in them are ignored.

If you are working with a Project, then when you save a source file, the Help file options of the Project are saved with it.

The options which refer to the whole Help file include:

Title
Compress flag
Directories
Font list
Default fonts, sizes, etc.
Secondary Window list
Copyright notice
Icon

The options which refer to an individual Topic include:

Caption
Topic number
Keywords
Contents flag
Status
Browse number

File|Move Topic

The File|Move Topic command moves the current Topic to a different source file within the Project.

The source file which the current Topic belongs to is shown at the bottom of the main window.

You may not move all the topics out of a source file; it is illegal to have a source file with no Topics.

See also
- File|New
- File|Open
- File|Save
- File|Save As
- File|Project|New
- File|Project|Open
- File|Project|Save
- File|Project|SaveAs
- File|Project|Files
- File|Save As TXT
- File|Save As RTF
- File|Save As WWW
- File|Open EXE
- File|WWW Mode
- File|Exit

File|Project Files

The Project Files command allows the user to add or remove help source files from the current Project file.

The Add button opens an existing help source file that was created using EditHelp and adds it to the list of files in the current Project.

The added file may contain topics with numbers which duplicate topic numbers in the current files. If so,you are given the opportunity to renumber the new topics either by hand or automatically (to arbitrary unique values).

See also
- File|New
- File|Open
- File|Save
- File|Save As
- File|Project|New
- File|Project|Open
- File|Project|Save
- File|Project|SaveAs
- File|Project|Move Topic
- File|Save As TXT
- File|Save As RTF
- File|Save As WWW
- File|Open EXE
- File|WWW Mode
- File|Exit

File|Open Project

The Open Project command opens an existing Project file that was created using EditHelp. Project files have the extension '.EHP'.

The help source files specified by the Project file will also be opened.

See also

- File|New
- File|Open
- File|Save
- File|Save As
- File|Project|New
- File|Project|Save
- File|Project|SaveAs
- File|Project|Files
- File|Project|Move Topic
- File|Save As TXT
- File|Save As RTF
- File|Save As WWW
- File|Open EXE
- File|WWW Mode
- File|Exit

Bulleted List

A List command specifies that the following text should be shown as a bulleted list:

```
{list}
```

The following paragraphs are indented and each is preceded by a '-' character. The List is terminated by:

```
{list=0}
```

For instance:

```
{list}a Bulleted List command specifies that the following text
should be shown in list format

each paragraph is preceded by a bullet

in a Definition List, each paragraph is divided into two columns separated by
'n' character widths. The first column shows the list item name, while the second
shows the item text

{list=0}
```

is displayed as:

- a Bulleted List command specifies that the following text should be shown in list format
- each paragraph is preceded by a bullet
- in a Definition List, each paragraph is divided into two columns separated by 'n' character widths. The first column shows the list item name, while the second shows the item text

The List command may only appear at the start of a paragraph; it may be indented.

See also
  Definition List

Definition List

A List=... command specifies that the following text should be shown as a Definition list:

{list=n}

where n is a non-zero number.

Each of the following paragraphs is divided into two columns separated by 'n' character widths. The first column shows the list item name, while the second shows the item text. The columns are separated by a Tab character.

The List is terminated by:

{list=0}

For instance:

{list=13}{bold}Key<tab>Function

F1<tab>Calls the help system

Alt<tab>Selects the menu

{list=0}

(where <tab> is a Tab character). This is displayed as:

Key Function

F1Calls the help system

AltSelects the menu

The List command may only appear at the start of a paragraph; it may be indented. If the item text is forced to wrap, it will be aligned under the item text column.

See also
Bulleted List
Line command

Text|Defaults

The Default Text command sets the default Font, Size and Colour of the Caption and text of the Topic.

If no default is specified, the text will be displayed as follows:

Property	**Default**
Text font	0 (Arial}
Text colour	0 (black}
Text size	20 points
Caption font	0 (Arial}
Caption colour	0 (black}
Caption size	40 points

See also

Text|Font Command
Text|Size Command
Text|Bold Command
Text|Italic Command
Text|Colour Command

Topic|Preview

The Preview command shows the Topic as it will be displayed by Windows Help.

Certain features cannnot be displayed, such as Embedded Panes, Literal text and Segmented Graphics Files.

The layout of the text and graphics may be a few pixels different from that displayed by Windows Help. The HelpFile|Make This Topic and HelpFile|Test commands will use Windows Help to display the topic.

See also
- Topic|Next topic
- Topic|Prev topic
- Topic|Back
- Topic|New topic
- Topic|Goto
- Topic|Goto Unfinished
- Topic|Delete
- Topic|Caption
- Topic|Contents
- Topic|Browse
- Topic|Status
- Topic|Entry Command

Edit|DLL Call

A hotspot which calls a DLL is specified in the source text by the DLL command:

    {dll=<dllname>,<funccall>,<text>)

where <dllname> is the name of the DLL file, <funccall> is a call to a function exported by the DLL and <text> is the hotspot text to be displayed.

The <funccall> string has the following format:

    <funcname>(<parameter>,<parameter>, ...)

<funcname> is the name of the function and each parameter has the format:

    <kind>:<value>

<kind> is a single character which represents the following <value> types:

Character	**Description**
u	unsigned short (WORD,word)
U	unsigned long (DWORD,longint)
i	int (INT,integer)
l	long (LONG,longint)
s	near char * (PSTR)
S	far char * (LPSTR,^string)
v	void

Windows Help makes sure the <value> part of the parameter matches the <kind>.

For instance,

    {dll=sample.dll,FUNC1(i:1234),Call a DLL}

specifies that the function FUNC1 is exported by the DLL sample.dll. It has a single Integer parameter which should be passed the value 1234. The command is displayed as:

Call a DLL

A DLL can also be specified in a Bitmap Link. For instance:

    {dll=sample.dll,FUNC1(i:5678),bmp=sample1.bmp}

which is displayed as:

The DLL filename should not contain any path information. The DLL file should be in the same directory as the EDH source file.

Windows Help allows certain Predefined Variables as parameter values.

See also

- Edit|Undo
- Edit|Cut
- Edit|Copy
- Edit|Paste
- Edit|Delete
- Edit|Link
- Edit|Popup
- Edit|Bitmap
- Edit|Embedded Pane

Edit|Embedded Pane

An Embedded Pane is specified in the source text by the Embed command:

{embed=<dllname>,<windowClass>,<text>)

where <dllname> is the name of the DLL file, <windowClass> is the name of a Window Class registered by the DLL and <text> is the "author data" passed to the DLL during window creation.

For instance,

{embed=test.dll,TestWnd,2)

specifies that the code for an embedded pane will be found in the DLL test.dll. Windows Help should display a window of class "TestWnd" and pass it the string "2".

The DLL filename should not contain any path information. The DLL file should be in the same directory as the EDH source file.

Embedded panes can display custom information controlled by a DLL. For instance, an embedded pane can show animation or play sound files. The code for embedded panes is described in

Microsoft Multimedia Viewer: Technical Reference
MM39685-0393

The embed command can take two other forms:

{embed=left,test.dll,TestWnd,2)
{embed=right,test.dll,TestWnd,2)

An {embed=left,...} command means the paragraph text is wrapped to the right of the embedded pane. An {embed=right,...} command means the text is wrapped to the left of the pane.

See also
- Edit|Undo
- Edit|Cut
- Edit|Copy
- Edit|Paste
- Edit|Delete
- Edit|Link
- Edit|Popup
- Edit|Bitmap
- Edit|DLL Call

Shareware

If this is an unregistered version of EditHelp then certain features will have been disabled.

EditHelp is Shareware, please copy and distribute it freely. Details of how to register it can be found under the Help|Shareware command.

When you register, you will receive the latest version of EditHelp plus examples of and DLLs for embedded-pane animation. If other multimedia DLLs have been developed available they will be included as well. Multimedia authoring tools are under development but they will cost extra.

You can optionally purchase a printed manual which contains the same information as this Help file.

Please send your registration fee and any reports of bugs or suggestions for improvements to:

Analogue Information Systems Ltd.
1 Warrender Park Crescent
Edinburgh EH9 1DX
Scotland, UK

Send a formatted floppy and a stamped-addressed-envelope (whether you are registered or not) for other Analogue shareware and freeware games, utilities and screensavers which you may find of interest.

Secondary Window

If the format of a Link command is

{link=num>winName,text}

where 'winName' is the specification of a Secondary Window, then the destination Topic will be displayed in a separate window. ('num' is the Topic number of the destination and 'text' is the link-text to be displayed.)

The specification of a Secondary Window has the format:

winName = "caption", (x,y,w,h,flag), state, (cR,cG,cB), (nR,nG,nB), fTop

This defines the size, location, and colors for the secondary window:

Parameter	Description
winName	the name of the window type; it may be any unique name of up to 8 characters
caption	the title for a secondary window
x,y,w,h	position and size of the window
flag	0: x,y,w,h are in 'help units' (Windows Help assumes the screen is always 1024x1024 'help units')
	1: x,y,w,h are in pixels
state	0: set the window to the size specified by x,y,w,h
	1: maximize the window
	2: minimize the window
cR,cG,cB	the background color of the window
nR,nG,nB	the background color of the non-scrolling region
fTop	0: the secondary window behaves normally
	1: the secondary window is displayed on top of all other windows

Only the 'name', 'caption' and '(x,y,w,h,flag)' fields are mandatory.

For instance

MyWindow="My Window",(0,0,500,500,0),,(255,255,0),,1

{link=249>MyWindow,Secondary Window}

specifies that if the 'Secondary Window' link is selected, the Secondary Window should appear in the top left corner of the screen, have a yellow background and remain on top. Try it:

Secondary Window

All of the specifications of Secondary Windows in the Secondary Window dialog listbox will be copied to the [WINDOWS] section of the Help Project File.

A Bitmap Link can include a Secondary Window clause. A Popup Link command should not include a Secondary Window clause.

If a Link command specifies both a Secondary Window and a New File, the clauses should be in the order:

{link=num@newFile>winName,text}

File|Save As WWW

The File|Save As WWW command saves the current source as a HyperText Markup Language (HTML) file. HTML can be used to represent Hypertext news, mail, online documentation, etc. on the World Wide Web using a standard Internet protocol.

Each topic will be stored in a separate file. The topic with the Contents flag set (if any) will be stored in the file

    <filename>.htm

where <filename> is the name of the current help source file.

The remaining topics will be stored the files with the name

    <fil><nnnnn>.htm

where <fil> is the first three characters of the source file name and <nnnnn> is the topic number padded with '0' to 5 digits. For instance, Topic 117 of the source 'edithelp.edh' will be written as:

    edi00117.htm

HTML does not support several features available in Windows Help files:

Fonts	the command is ignored
Text size	the command is ignored
Text colour	the command is ignored
Help Macros	treated as a Link
Popup Links	treated as a Link
Boxed text	a Line is drawn
Centred text	the command is ignored
Keep	the command is ignored
Tab=...	the command is ignored

If your source file uses any of these features, a warning message will be displayed but the HTML file will be written anyway.

A Tab character will be translated into a single blank.

All Lists will be displayed as simple bulleted lists.

If a Bitmap or Bitmap Link specifies a BMP (or SHG) file then the 'bmp' (or 'shg') extension will be translated into 'gif'. So, for instance, the command:

    {bitmap=sample.bmp}

would be translated into an inline graphics element specifying

    sample.gif

You should make the GIF file available in the same directory as the HTML files. Some HTML browsers will not be able to display inline graphics. If the graphic is essential, it may be wiser to make a link to it (by editing the HTML file) rather than to put it inline.

See also
- File|New
- File|Open
- File|Save
- File|Save As
- File|Project|New
- File|Project|Open

File|Project|Save
File|Project|SaveAs
File|Project|Files
File|Project|Move Topic
File|Save As TXT
File|Save As RTF
File|Open EXE
File|WWW Mode
File|Exit

Segmented Graphics Files

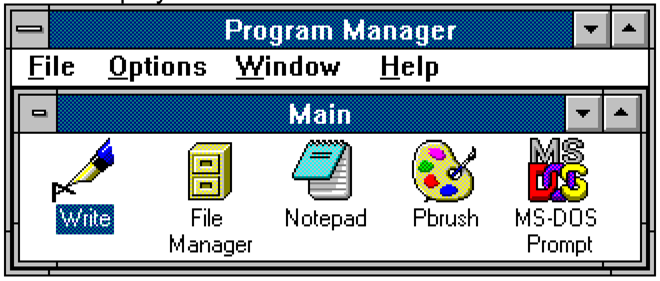
A Bitmap command may specify a Segmented Graphics File (SHG) instead of a BMP file:

{bitmap=<SHG filename>}

A Segmented Graphics File contains a bitmap that includes one or more hotspots. For instance,

{bitmap=sample.shg}

will be displayed as



The hotspots can be links to other topics or Help Macro commands. In the example above, each icon is a hotspot. The Write, FileManager, Pbrush and MS-DOS icons specify a popup link while the Notepad icon specifies a Help Macro which executes the Notepad program.

The SHG file should be in the same directory as the help source file.   The <SHG filename> should not contain any DOS path information.

You can create SHG files with the Microsoft Hotspot Editor

SHED.EXE

SHED may have been supplied with your copy of Windows, or with your Microsoft or Borland programming tools.

If you use SHED to create links, the topic context string for topic number 'nnn' is 'PGnnn'. For instance, this is topic number 244, so the SHED context string for a link to this topic is:

PG244

Topic|Browse

The Topic|Browse command sets the browse group name and number of the current topic.

A browse sequence typically consists of two or more related topics that are intended to be read sequentially. Each group has a browse Group Name and each topic in the group has a different Sequence Number.

Windows Help adds topics with the same Group Name to the same browse group and determines the order of topics in the group by sorting the Sequence Numbers. If two topics in a group have the same number, Windows Help assumes that the topic that was compiled first is to be displayed first.

If any topic belongs to a browse group, EditHelp will add browse buttons to the button bar of Windows Help.

If the Group Name of a topic is empty then the topic does not belong to any browse group.

The browse name and number of the current topic are shown in the top right corner of the main window.

For instance, the Group Name of this topic and all the other topics describing the 'Topic' popup menu have been set to

TopicMenu

and the Sequence Numbers have been set to 1,2,3...

See also
- Topic|Next topic
- Topic|Prev topic
- Topic|Back
- Topic|New topic
- Topic|Goto
- Topic|Goto Unfinished
- Topic|Delete
- Topic|Caption
- Topic|Contents
- Topic|Status
- Topic|Entry Command
- Topic|Preview

HelpFile|Icon

The HelpFile|Icon command allows you to specify the name of an icon (.ICO) file.

The icon will be displayed when the Help file is minimised.

The filename should not contain any path information. The icon file should be in the same directory as the EDH source file.

For instance, if the Icon filename is set to 'sample.ico' then the following icon will be displayed when the Help file is minimised:

If the Icon filename is set blank then the default Windows Help icon will be displayed.

See also
- HelpFile|Make
- HelpFile|Test
- HelpFile|Make This Topic
- HelpFile|Directories
- HelpFile|Title
- HelpFile|Compress
- HelpFile|Copyright
- HelpFile|Status
- HelpFile|Keywords

HelpFile|Copyright

The HelpFile|Copyright command allows you to specify an additional copyright notice which will appear in the Windows Help About box.

See also
- HelpFile|Make
- HelpFile|Test
- HelpFile|Make This Topic
- HelpFile|Directories
- HelpFile|Title
- HelpFile|Compress
- HelpFile|Icon
- HelpFile|Status
- HelpFile|Keywords

Help Macros

Help Macros are commands sent to to the Windows Help system. For instance:

{link='About()',Display About Box}

calls the About macro when the hotspot is selected:

Display About Box

A {link=...} may contain several macros separated by semi-colons.

You may add new commands contained in your own DLLs by calling the RegisterRoutine macro. For instance, this command contains two macros, the first registers the FUNC1 routine in the sample.dll DLL and the second calls it and passes the integer value 1234:

{link=!RegisterRoutine("sample.dll","FUNC1","i");FUNC1(1234)!,Call a DLL}

Selecting the hotspot calls the DLL:

Call a DLL

A macro can also be specified in a Bitmap Link. For instance:

{link='ExecProgram("calc.exe",0)',bmp=sample2.bmp}

selecting this bitmap will execute the Calculator program supplied with Windows:

The Windows Help system supports the following macros:

Macro	Function
About	Displays the About dialog box
AddAccelerator	Assigns a macro to an accelerator key
Annotate	Displays Annotation dialog box
AppendItem	Appends a menu item
Back	Displays previous topic in the history list
BookmarkDefine	Displays the Define dialog box
BookmarkMore	Displays the More dialog box
BrowseButtons	Adds browse buttons
ChangeButtonBinding	Assigns a macro to a button
ChangeItemBinding	Assigns a macro to a menu item
CheckItem	Checks a menu item
CloseWindow	Closes a window
Contents	Displays the Contents topic
CopyDialog	Displays the Copy dialog box
CopyTopic	Copies current topic to the clipboard
CreateButton	Adds a new button to the button bar
DeleteItem	Removes a menu item
DeleteMark	Deletes a text marker
DestroyButton	Removes a button from the button bar
DisableButton	Disables a button
DisableItem	Disables a menu item
EnableButton	Enables a button
EnableItem	Enables a menu item
ExecProgram	Executes a program
Exit	Exits WinHelp

FileOpen	Displays the Open dialog box
FocusWindow	Changes the focus window
GoToMark	Jumps to a marker
HelpOn	Displays the Help on Using topic
HelpOnTop	Toggles on-top state of help
History	Displays the history list
IfThen	Executes macro if marker exists
IfThenElse	Executes one of two macros if marker exists
InsertItem	Inserts a menu item
InsertMenu	Inserts a new menu
IsMark	Tests if a marker is set
JumpContents	Jumps to the Contents topic
JumpContext	Jumps to the specified context
JumpHelpOn	Jumps to Using Help file
JumpId	Jumps to the specified topic
JumpKeyword	Jumps to the topic containing the keyword
Next	Displays the next topic in the browse sequence
Not	Reverses the IsMark macro
PopupContext	Displays a topic in a popup window
PopupId	Displays topic in a popup window
PositionWindow	Sets the size and position of a window
Prev	Displays previous topic in browse sequence
Print	Prints the current topic
PrinterSetup	Displays the Printer Setup dialog box
RegisterRoutine	Registers a DLL function
RemoveAccelerator	Assigns a macro to an accelerator key
SaveMark	Saves a marker
Search	Displays the Search dialog box
SetContents	Sets the Contents topic
SetHelpOnFile	Sets the Using Help help file
UncheckItem	Unchecks a menu item

Text|Colour

The Colour command specifies that subsequent text is shown in the specified colour:

{colour=n}

where 'n' is the colour number.

The following colours are available on a 16-colour display:

{colour=1}	blue
{colour=2}	cyan
{colour=3}	green
{colour=4}	magenta
{colour=5}	red
{colour=6}	yellow

{colour=8}	dark blue
{colour=9}	dark cyan
{colour=10}	dark green
{colour=11}	dark magenta
{colour=12}	dark red
{colour=13}	dark yellow
{colour=14}	dark grey
{colour=15}	grey

See also

- Text|Font Command
- Text|Size Command
- Text|Italic Command
- Text|Bold Command
- Text|Defaults

Layout|Centre

A Centre command specifies that the remaining text of the paragraph should be centered in the screen. For instance:

    {centre}This text is centred

is shown as:

This text is centred

The Centre command must appear before any of the text of a paragraph.

The Centre command may be combined with the Box command:

{centre}{box}This is boxed and centred

is displayed as

This is boxed and centred

See also
- Layout|Heading
- Layout|List
- Layout|Line
- Layout|Box
- Layout|Tab
- Layout|Keep

Layout|Box

A Box command specifies that a box should be drawn around the remaining text of the paragraph. For instance:

```
{box}This text is in a box
```

is shown as:

$\boxed{\text{This text is in a box}}$

The box command must appear before any of the text of a paragraph. It may be indented, as in the above example.

See also

- Layout|Heading
- Layout|List
- Layout|Line
- Layout|Centre
- Layout|Tab
- Layout|Keep

Layout|Line

A Line command specifies that a line should be drawn across the screen. For instance:

{line}

is shown as:

If the Line command is followed by text, the text is displayed immediately above the line:

{line}This is text

is shown as:

This is text

The Line command may only appear before the text of a paragraph. The Line command may be indented.

Both Line and List commands can be used together. For instance:

{list=15}{line}{bold}Key<tab>Function{bold}

F1<tab>Calls the help system

Alt<tab>Selects the menu

(where <tab> is a Tab character). This is displayed as:

Key Function

F1Calls the help system

AltSelects the menu

See also
- Layout|Heading
- Layout|List
- Layout|Box
- Layout|Centre
- Layout|Tab
- Layout|Keep

Layout|List

A List command specifies that the following text should be shown in list format.

There are three formats for the list command:

Command	Description
{list}	start a Bulleted List
{list=n}	start a Definition List
{list=0}	end a list

See also

Layout|Heading
Layout|Line
Layout|Box
Layout|Centre
Layout|Tab
Layout|Keep

Layout|Heading

A sub-heading is specified by the Heading command:

    {heading=<text>}

where <text> is the text of the heading.

For instance, the following command:

{heading=Sub-heading}

will be displayed as

Sub-heading

The Layout|Heading command inserts a Heading into the text of the current topic at the current insertion point.

You can also type in a Heading command "by hand".

See also
- Text commands
- Layout|List
- Layout|Line
- Layout|Box
- Layout|Centre
- Layout|Tab
- Layout|Keep

Edit|Bitmap Link

A bitmap link is specified in the source text by the Link command:

    {link=<num>,bmp=<filename>}

where <num> is the Topic number of the destination and <filename> is a file containing the bitmap to be displayed.

For instance, the Edit|Link topic is 2207 so a bitmap-link to that topic could be specified as

    {link=2207,bmp=sample1.bmp}

which would display as

When the user selects the bitmap, the destination topic is displayed.

As in this example:

a bitmap-link can also appear in the middle of text.
A bitmap-link can specify that the destination topic is to be displayed as a popup. For instance

    {popup=117,bmp=sample1.bmp}

would be displayed as

The bitmap filename should not contain any path information. The bitmap file should be in the same directory as the EDH source file.

A link can specify that the destination topic should be displayed in a Secondary Window or that a New File should be loaded.

See also

- Edit|Undo
- Edit|Cut
- Edit|Copy
- Edit|Paste
- Edit|Delete
- Edit|Link
- Edit|Popup
- Edit|Bitmap
- Edit|DLL Call
- Edit|Embedded Pane

File|Open EXE

The File|Open EXE command reads an exe file and examines the menu and dialog resources of the file. From these, it creates a corresponding skeleton EDH source file with one topic for each menu item and dialog box.

If the source file name is

program.exe

then the new source file will be

program.edh

The topics corresponding with menu items are numbered

10000 + id

where 'id' is the id-number of the menu item.

The topics corresponding with dialogs are numbered sequentially from 1 except that if the dialog has a button labelled 'Help' then the topic is given the same topic number as the button id-number.

See also

- File|New
- File|Open
- File|Save
- File|Save As
- File|Project|New
- File|Project|Open
- File|Project|Save
- File|Project|SaveAs
- File|Project|Files
- File|Project|Move Topic
- File|Save As TXT
- File|Save As RTF
- File|Save As WWW
- File|WWW Mode
- File|Exit

Text|Italic

The Text|Italic command sets whether subsequent text will be displayed italic or normal.

    {italic}

The {italic} command toggles between italic and normal characters. For instance:

    Ordinary text and {italic}italic text and back to {italic}ordinary text

is displayed as:

Ordinary text and *italic text and back to* ordinary text

Every paragraph starts with the text normal (not italic).

See also
- Text|Font Command
- Text|Size Command
- Text|Bold Command
- Text|Colour Command
- Text|Defaults

Text|Bold

The Text|Bold command sets whether subsequent text will be displayed bold or normal width.

```
{bold}
```

The {bold} command toggles between bold and normal width characters. For instance:

```
Ordinary text and {bold}bold text and back to {bold}ordinary text
```

is displayed as:

> Ordinary text and **bold text and back to** ordinary text

Every paragraph starts with the text normal width.

See also
- Text|Font Command
- Text|Size Command
- Text|Italic Command
- Text|Colour Command
- Text|Defaults

Text|Size

The Text|Size command sets the size of the font to be used for subsequent text.

```
{fontsize=n}
```

where 'n' is the character size; the default characted size is 20. For instance:

```
This is {fontsize=50}very large{fontsize=20} text
```

is displayed as:

This is very large text

Not all character sizes are displayed well with all fonts

See also

- Text|Font Command
- Text|Bold Command
- Text|Italic Command
- Text|Colour Command
- Text|Defaults

Text|Font

The Text|Font command sets the font to be used for subsequent text.

{font=n}

where 'n' is the font number. For instance:

{font=6}}This is script font{font=0}

is displayed as:

This is script font

The Text|Font command provides the following font options as standard:

{font=0}	Arial: modern
{font=1}	Courier New: modern
{font=2}	Modern: modern
{font=3}	MS Sans Serif: swiss
{font=4}	MS Serif: swiss
{font=5}	Roman: roman
{font=6}	Brush Script: script
{font=7}	Small Fonts: decor
{font=8}	Σψμβολ: ρομαν (Symbol)
{font=9}	Times New Roman: roman

Not all these fonts may be present on your computer.

You may add other font options by typing into the font Combo-box. The format of the font string should be:

<font name>:<font family>

If the specified font is not present, Windows Help will select another font from the same family.

There may not be more than 100 fonts.

See also

Text|Size Command
Text|Bold Command
Text|Italic Command
Text|Colour Command Text|Defaults

Search|Replace with Link

The Search|Replace with Link command searches for and changes text in into a Link.

The Replace with Link command will not replace text which is inside a {link=...} or any other command. Nor will it replace text which is in the same topic as the destination of the link.

During the search, you can match uppercase and lowercase letters or ignore case.

If the 'all occurrences' box is checked then all the occurrences of the search string will be replaced in the current topic.

If you select the 'All Topics' button then the search will start with the first topic in the topic list and will be extended to all the topics in the list. If the 'all occurrences' box is checked then all the occurrences of the search string will be replaced in all topics. If the 'all occurrences' box is not checked then only the first occurrence to be found will be replaced.

See also
- Search|Find
- Search|Replace
- Search|Next
- Search|Find Error

Topic|Back

The Topic|Back command re-displays the last topic to be shown.

The Topic|Back command can be called with the Ctrl-B key.

See also

- Topic|Next topic
- Topic|Prev topic
- Topic|New topic
- Topic|Goto
- Topic|Goto Unfinished
- Topic|Delete
- Topic|Caption
- Topic|Contents
- Topic|Browse
- Topic|Status
- Topic|Entry Command
- Topic|Preview

Bitmaps

A Bitmap is specified by the Bitmap command:

    {bitmap=<filename>}

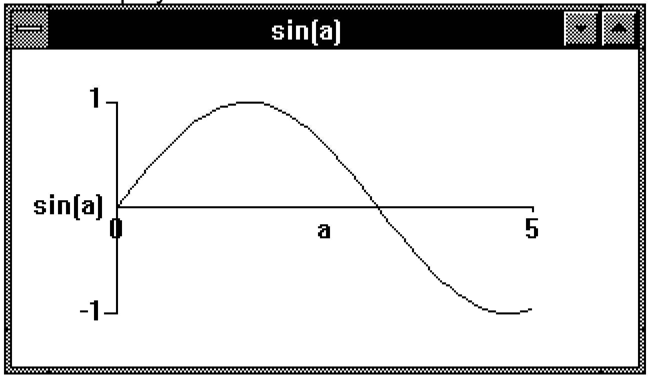
where <filename> is the name of a BMP file to be displayed.

The bitmap should be a standard windows .BMP file in the same directory as the help source file. The <filename> should not contain any DOS path information.

For instance, the following command:

    {bitmap=sample.bmp}

will be displayed as



The help system will display coloured bitmaps but you may find it best to use monochrome bitmaps. They are smaller and can be displayed on any hardware.

You may also specify a Segmented Graphics File (SHG) instead of a BMP bitmap. A Segmented Graphics File contains a bitmap that includes one or more hotspots. The hotspots can be links to other topics or Help Macro commands.

A small selection of bitmaps are supplied with EditHelp.

Bitmaps can be used in a variety of ways:

A bitmap can be embedded in the text ➔ as with this arrow

You can use bitmaps to highlight paragraphs of text

Or you can simulate a Browse sequence with the 'Previous' and 'Next' buttons:

The bitmap command can take two other forms:

    {bitmap=left,sample3.bmp}
    {bitmap=right,sample3.bmp}

A {bitmap=left,...} command means the paragraph text is wrapped to the right of the bitmap.

A {bitmap=right,...} command means the paragraph text is wrapped to the left of the bitmap.

See also

Text Commands

Edit|Bitmap

This is a popup topic. It will remain on the screen until you click the mouse or press a key.

The caption of a popup can be left blank.

Popup Link

A popup link is specified in the source text by the Popup command:

    {popup=<num>,<text>}

where <num> is the Topic number of the destination and <text> is the link-text to be displayed.

For instance, the popup topic is 117 so a link to the popup topic would be specified as

    {popup=117,Popup Topic}

which would display as

    Popup Topic

When the user selects the popup link, the destination topic is displayed in a separate window.

A bitmap can be used as a button which displays the new popup topic.

See also
    Links
    Edit|Popup

Menu Commands

The following menu commands are available:

File

- New
- Open
- Save
- Save As
- Project
  - New
  - Open
  - Save
  - SaveAs
  - Files
  - Move Topic
- Save As TXT
- Save As RTF
- Save As WWW
- Open EXE
- Open HLP
- WWW Mode
- Exit

Edit

- Undo
- Cut
- Copy
- Paste
- Delete
- Link
- Popup Link
- Bitmap
- Bitmap Link
- DLL Call
- Embedded Pane

Search

- Find
- Replace
- Replace with Link
- Next
- Find Error

Layout

- Heading
- List
- Line
- Box
- Centre
- Tab

Keep

Text

Font
Size
Bold
Italic
Colour

Topic

Next topic
Prev topic
Back
New topic
Goto
Goto Unfinished
Delete
Caption
Contents
Browse
Status
Entry Command
Preview

HelpFile

Make
Test
Make This Topic
Directories
Title
Compress
Copyright
Icon
Status
Keywords

HelpFile|Status

The HelpFile|Status command displays the Project Status dialog. This shows the Topic number, Caption, Keywords and Status of each topic.

The Project Status dialog helps you keep track of how far you've got in developing your Help File.

You may add a short note to each topic to remind you what to do next.

See also
- Topic|Goto Unfinished
- Topic|Status

- HelpFile|Make
- HelpFile|Test
- HelpFile|Make This Topic
- HelpFile|Directories
- HelpFile|Title
- HelpFile|Compress
- HelpFile|Copyright
- HelpFile|Icon
- HelpFile|Keywords
- Topic|Preview

HelpFile|Compress

The HelpFile|Compress command toggles whether the Help Compiler should compress your Help File.

The Help Compiler takes longer to compile your file if compression is switched on but the resulting Help File is smaller.

The Internal Compiler does not produce a compressed file. (Certain of the compression techniques have been patented by Microsoft.)

See also
- HelpFile|Make
- HelpFile|Test
- HelpFile|Make This Topic
- HelpFile|Directories
- HelpFile|Title
- HelpFile|Status
- HelpFile|Copyright
- HelpFile|Icon
- HelpFile|Keywords

HelpFile|Title

The HelpFile|Title command allows you to specify the title of the Help File.

The title is shown on the caption bar of the main window when the Help System displays your help file.

See also
- HelpFile|Make
- HelpFile|Test
- HelpFile|Make This Topic
- HelpFile|Directories
- HelpFile|Compress
- HelpFile|Copyright
- HelpFile|Icon
- HelpFile|Status
- HelpFile|Keywords

HelpFile|Directories

The HelpFile|Directories command allows you to specify the directory into which the Help (.HLP) File will be copied and to specify whether the Internal Compiler or an External Help Compiler should be used.

The help source file is always in the current directory but the Help File may be placed in any directory. Usually, the help source file will be in same directory as the source of the program you are writing while the Help File will be in same directory as your EXE file.

The External Compiler text box should contain the full path name and program EXE name of your help compiler.

See also

- HelpFile|Make
- HelpFile|Test
- HelpFile|Make This Topic
- HelpFile|Title
- HelpFile|Compress
- HelpFile|Copyright
- HelpFile|Icon
- HelpFile|Status
- HelpFile|Keywords

HelpFile|Test

The HelpFile|Test command calls the Help System to display the current topic of the current Help File.

You should have compiled the current source by calling the HelpFile|Make command before calling the Test command.

The Test command can be called with the Ctrl-F1 key.

See also
- HelpFile|Make
- HelpFile|Make This Topic
- HelpFile|Directories
- HelpFile|Title
- HelpFile|Compress
- HelpFile|Copyright
- HelpFile|Icon
- HelpFile|Status
- HelpFile|Keywords

HelpFile|Make This Topic

The HelpFile|Make command compiles the current topic (not the whole source) into a Help File.

If the "Internal Compiler " button has been selected in the HelpFile|Directories dialog EditHelp uses the Internal Compiler.

Otherwise, EditHelp saves the current topic as a Rich Text Format file and a Help Project file.

It then executes the Help Compiler which compiles these into the Help File:

```
<directory><filename>.hlp
```

where <filename> is the name of the current help source file and <directory> is the 'Help File Directory' specified by the HelpFile|Directories command.

You can then test the current topic by calling the HelpFile|Test command.

The HelpFile|Make command creates a Help File for all the topics. However, the Help Compiler is very slow when compiling large projects. So if you want to test a single topic, you should use the HelpFile|Make This Topic command.

See also
HelpFile|Make
HelpFile|Test
HelpFile|Directories
HelpFile|Title
HelpFile|Compress
HelpFile|Copyright
HelpFile|Icon
HelpFile|Status
HelpFile|Keywords

HelpFile|Make

The HelpFile|Make command compiles the help source into a Help File.

If the "Internal Compiler " button has been selected in the HelpFile|Directories dialog EditHelp uses the Internal Compiler.

Otherwise, EditHelp saves the help source as the Rich Text Format file

    ~out.rtf

and creates a Help Project file:

    ~out.hpj

EditHelp then executes the Help Compiler which compiles ~out.rtf into the help file

    ~out.hlp

This Help File is copied to

    <directory><filename>.hlp

where <filename> is the name of the current help source file and <directory> is the 'Help File Directory' specified by the HelpFile|Directories command.

All the ~out files are then deleted.

The external Help Compiler is a DOS program so EditHelp creates a DOS batch file. The batch file pauses after the Help Compiler has executed so you can see whether the compiler generated any error messages. Press any (printable) key to continue execution.

The Make command can be called with the F9 key.

See also
- HelpFile|Test
- HelpFile|Make This Topic
- HelpFile|Directories
- HelpFile|Title
- HelpFile|Compress
- HelpFile|Copyright
- HelpFile|Icon
- HelpFile|Status
- HelpFile|Keywords

Topic|Status

The Topic|Status command cycles through the status values for the current topic. The status value can be one of:

Incomplete
Test
Finished

The Status value has no effect on the operation of EditHelp but allows you to keep track of which parts of the help source have been completed.

You can also call the Topic|Status command by clicking on the status area of the main window.

See also
HelpFile|Status

Topic|Next topic
Topic|Prev topic
Topic|Back
Topic|New topic
Topic|Goto
Topic|Goto Unfinished
Topic|Delete
Topic|Caption
Topic|Contents
Topic|Browse
Topic|Entry Command
Topic|Preview

Topic|Contents

The Topic|Contents command toggles whether the current topic is the Contents topic. The Help System displays the Contents topic when the user selects the Contents command.

Only one topic can have the Contents switch set to True. If you toggle the Contents command, all the other topics will have their Contents switch set to False.

See also
- Topic|Next topic
- Topic|Prev topic
- Topic|Back
- Topic|New topic
- Topic|Goto
- Topic|Goto Unfinished
- Topic|Delete
- Topic|Caption
- Topic|Browse
- Topic|Status
- Topic|Entry Command
- Topic|Preview

Topic|Caption

The Topic|Caption command changes the topic number, caption and keywords of the current topic.

You can also call the Topic|Caption command by clicking on the caption area of the main window.

See also

- Topic|Next topic
- Topic|Prev topic
- Topic|Back
- Topic|New topic
- Topic|Goto
- Topic|Goto Unfinished
- Topic|Delete
- Topic|Contents
- Topic|Browse
- Topic|Status
- Topic|Entry Command
- Topic|Preview

Topic|Delete

The Topic|Delete command deletes the current topic from the help source.

See also
- Topic|Next topic
- Topic|Prev topic
- Topic|Back
- Topic|New topic
- Topic|Goto
- Topic|Goto Unfinished
- Topic|Caption
- Topic|Contents
- Topic|Browse
- Topic|Status
- Topic|Entry Command
- Topic|Preview

Topic|Goto

The Topic|Goto command allows you to select a new topic from the topic list and load it into the Edit window.

You may select the topic by topic number, caption or Keywords.

See also
- Topic|Next topic
- Topic|Prev topic
- Topic|Back
- Topic|New topic
- Topic|Goto Unfinished
- Topic|Delete
- Topic|Caption
- Topic|Contents
- Topic|Browse
- Topic|Status
- Topic|Entry Command
- Topic|Preview

Topic|New topic

The Topic|New Topic command creates a new topic. Each topic should have a unique topic number.

See also

- Topic|Next topic
- Topic|Prev topic
- Topic|Back
- Topic|Goto
- Topic|Goto Unfinished
- Topic|Delete
- Topic|Caption
- Topic|Contents
- Topic|Browse
- Topic|Status
- Topic|Entry Command
- Topic|Preview

Topic|Prev topic

The Topic|Prev Topic command displays the previous topic in the topic list in the Edit window.

The Prev topic command can be called with the Shift-F6 key.

See also
- Topic|Next topic
- Topic|Back
- Topic|New topic
- Topic|Goto
- Topic|Goto Unfinished
- Topic|Delete
- Topic|Caption
- Topic|Contents
- Topic|Browse
- Topic|Status
- Topic|Entry Command
- Topic|Preview

Topic|Next topic

The Topic|Next Topic command displays the next topic in the topic list in the Edit window.

The Next topic command can be called with the F6 key.

See also
- Topic|Prev topic
- Topic|Back
- Topic|New topic
- Topic|Goto
- Topic|Goto Unfinished
- Topic|Delete
- Topic|Caption
- Topic|Contents
- Topic|Browse
- Topic|Status
- Topic|Entry Command
- Topic|Preview

Search|Next

The Search|Next command repeats the last search or replace operation without opening the Find dialog box.

The Next command can be called with the F3 key.

See also
- Search|Find
- Search|Replace
- Search|Replace with Link
- Search|Find Error

Search|Replace

The Search|Replace command searches for and changes text in the current topic. During the search, you can match uppercase and lowercase letters or ignore case.

If the 'all occurrences' box is checked then all the occurrences of the search string will be replaced in the current topic.

If you select the 'All Topics' button then the search will start with the first topic in the topic list and will be extended to all the topics in the list. If the 'all occurrences' box is checked then all the occurrences of the search string will be replaced in all topics. If the 'all occurrences' box is not checked then only the first occurrence to be found will be replaced.

See also
- Search|Find
- Search|Replace with Link
- Search|Next
- Search|Find Error

Search|Find

The Search|Find command searches for characters or words in the text of the current topic. You can match uppercase and lowercase letters or ignore case.

If you select the 'All Topics' button then the search will start with the first topic in the topic list and will be extended to all the topics in the list.

See also
- Search|Replace
- Search|Replace with Link
- Search|Next
- Search|Find Error

Edit|Delete

The Edit|Delete command deletes the selected text from the current topic, but does not place the text onto the Clipboard.

Use Delete when you want to delete text from the current topic but you have text on the Clipboard that you want to keep.

The Delete command can be called with the Del key.

See also
- Edit|Undo
- Edit|Cut
- Edit|Copy
- Edit|Paste
- Edit|Link
- Edit|Popup
- Edit|Bitmap
- Edit|DLL Call
- Edit|Embedded Pane

Edit|Paste

The Edit|Paste command pastes a copy of the Clipboard contents at the insertion point or replaces selected text in the current topic.

The Paste command can be called with the Shift+Ins key.

See also
- Edit|Undo
- Edit|Cut
- Edit|Copy
- Edit|Delete
- Edit|Link
- Edit|Popup
- Edit|Bitmap
- Edit|DLL Call
- Edit|Embedded Pane

Edit|Copy

The Edit|Copy command copies the selected text from the current topic onto the Clipboard, leaving the original intact and replacing the previous Clipboard contents.

The Copy command can be called with the Ctrl+Ins key.

See also
- Edit|Undo
- Edit|Cut
- Edit|Paste
- Edit|Delete
- Edit|Link
- Edit|Popup
- Edit|Bitmap
- Edit|DLL Call
- Edit|Embedded Pane

Edit|Cut

The Edit|Cut command deletes the selected text from the current topic and places it onto the Clipboard, replacing the previous Clipboard contents.

The Cut command can be called with the Shift+Del key.

See also
- Edit|Undo
- Edit|Copy
- Edit|Paste
- Edit|Delete
- Edit|Link
- Edit|Popup
- Edit|Bitmap
- Edit|DLL Call
- Edit|Embedded Pane

Edit|Undo

The Edit|Undo command undoes your last editing action, including cut and paste actions.

Undo only works with actions performed on the help source text in the Edit window.

The Undo command can be called with the Alt+BkSp key.

See also
- Edit|Cut
- Edit|Copy
- Edit|Paste
- Edit|Delete
- Edit|Link
- Edit|Popup
- Edit|Bitmap
- Edit|DLL Call
- Edit|Embedded Pane

File|Exit

The File|Exit command exits from the EditHelp program. If you've made changes to the help source, you will be asked if you want to save them.

See also
- File|New
- File|Open
- File|Save
- File|Save As
- File|Project|New
- File|Project|Open
- File|Project|Save
- File|Project|SaveAs
- File|Project|Files
- File|Project|Move Topic
- File|Save As TXT
- File|Save As RTF
- File|Save As WWW
- File|Open EXE
- File|WWW Mode

File|Save As

The File|Save As command saves the current help source to a file. The Save As dialog allows you to choose the name of the file.

Help source files have the extension '.EDH'.

See also
- File|New
- File|Open
- File|Save
- File|Project|New
- File|Project|Open
- File|Project|Save
- File|Project|SaveAs
- File|Project|Files
- File|Project|Move Topic
- File|Save As TXT
- File|Save As RTF
- File|Save As WWW
- File|Open EXE
- File|WWW Mode
- File|Exit

File|Save

The File|Save command saves the current help source to a file. Help source files have the extension '.EDH'.

If the current help source hasn't been named yet, EditHelp displays the Save File As dialog box.

See also
- File|New
- File|Open
- File|Save As
- File|Project|New
- File|Project|Open
- File|Project|Save
- File|Project|SaveAs
- File|Project|Files
- File|Project|Move Topic
- File|Save As TXT
- File|Save As RTF
- File|Save As WWW
- File|Open EXE
- File|WWW Mode
- File|Exit

File|Open

The File|Open command opens an existing help source file that was created using EditHelp. Help source files have the extension '.EDH'.

The File|Open command displays the File Open dialog box.

See also

- File|New
- File|Save
- File|Save As
- File|Project|New
- File|Project|Open
- File|Project|Save
- File|Project|SaveAs
- File|Project|Files
- File|Project|Move Topic
- File|Save As TXT
- File|Save As RTF
- File|Save As WWW
- File|Open EXE
- File|WWW Mode
- File|Exit

File|New

The File|New command creates a new help source with one empty topic.

See also

- File|Open
- File|Save
- File|Save As
- File|Project|New
- File|Project|Open
- File|Project|Save
- File|Project|SaveAs
- File|Project|Files
- File|Project|Move Topic
- File|Save As TXT
- File|Save As RTF
- File|Save As WWW
- File|Open EXE
- File|WWW Mode
- File|Exit

Caption

Every topic may optionally have a Caption. The Windows help system will display the caption in larger bold text at the head of the topic screen.

Popup topic captions are usually blank.

You can change the caption by selecting the Topic|Caption command.

Files

EditHelp can read or write several different file formats:

Source file: EDH

The input to the EditHelp program is a 'source file'. A source file has the extension '.EDH'.

You may examine the source file using any plain ASCII text editor but be careful not to disturb its format.

Help file: HLP

When you select the HelpFile|Make command, EditHelp writes the contents of the source file as a Rich Text Format file and a Help Project File. It then executes the Help Compiler which compiles these into a Help File.

The Help File file is created as:

<directory><filename>.hlp

where <filename> is the name of the current help source file (or Project file) and <directory> is the 'Help File Directory' specified by the HelpFile|Directories command.

Help Project file: HPJ

The Help Compiler requires a Help Project file to direct the compilation of the Rich Text Format file to a Help File.

You can specify extra text to be inserted into the Help Project file.

(Do not confuse the Help Project File - HPJ - with a Project file - EHP.)

Help Project Extra files: EPJ

A Help Project file consists of various sections: e.g. [OPTIONS], [FILES], [CONFIG], etc. You can specify extra text to be inserted into each of these sections. For instance, if the file

options.epj

exists in the same directory as the EditHelp source file, then the text of the EPJ file will be copied to the [OPTIONS] section of the Help Project file.

Rich Text Format file: RTF

EditHelp creates a temporary Rich Text Format file (~out.rtf) which the Help Compiler compiles to a Help File.

Normally, you will not need to examine the Rich Text Format file.

Rich Text Format files can be read and edited by the Microsoft Word word processor. You can use the File|Save As RTF command to write the current source as a Rich Text Format file.

Project file: EHP

A Project consists of a Project file and a set of EDH source files.

When the HelpFile|Make command is executed, all of the source files are linked together to form a single HLP Help file. Use a Project when you want several source files to be shared by several Help files.

World Wide Web: WWW

The File|Save As WWW command saves the current source as a HyperText Markup Language (HTML) file. HTML can be used to represent Hypertext news, mail, online documentation, etc. on the World Wide Web using a standard Internet protocol.

Each topic will be stored in a separate file:

<filename>.htm

ASCII Text file: TXT

The File|Save As TXT command saves the current source to a file as plain ASCII text. The text file will have the name:

<filename>.TXT

Each topic appears on a separate page. You can use the text file as the basis of your user manual.

Executable file: EXE

The File|Open EXE command reads an exe file and examines the menu and dialog resources of the file. From these, it creates a corresponding skeleton EDH source file.

If the source file name is

program.exe

then the new source file will be

program.edh

Topics

A Help File is divided into 'Topics'. Each topic is a "page" full of information describing a particular subject.

Each topic must have a unique number. The number is used when the host program calls the help system to display a help topic.

The user selects a topic by following Links or by selecting a Keyword from the Search dialog.

Topic number

Each topic must have a unique number. The number is used when the host program calls the help system to display a help topic.

For instance, this topic is topic number 105, so the host program would display this topic by calling:

```
WinHelp(hWindow, HelpFileName, Help_Context, 105)
```

You can change the topic number by selecting the Topic|Caption command.

If you have written a program which must use context strings rather than context numbers, you can use the file ALIAS.EPJ to assign strings to topics.

See also
- topics
- file names

Keywords

Keywords appear in the Windows Help Search dialog box when the user selects the Search command.

The Keywords section may contain several keywords separated by semicolons or newlines. For instance, the keywords for this topic are:

    keywords;search;topic name

If, in Windows Help,   you select the Search command, you will find this topic indexed by each of these keywords.

Several topics may share the same keyword. In which case, when the user selects the keyword, the 'topic names' will appear in the GoTo box of the Search dialog. The 'topic name' is the first keyword in the keyword list.

You can change the keyword list by selecting the Topic|Caption command.

Comments

If first two characters on a line are '//', the line will be ignored. For example

```
// this is a comment
```

Help Compiler

The Help Compiler converts a Rich Text Format file into a Help File.

EditHelp was written for use with the Microsoft HC31 help compiler (copyright (c) Microsoft Corp 1990 - 1992). This reports the version number:

    Version 3.10.445

It may also work with older versions of HC31 or even with the Windows 3.0 version - HC30.

You may already have HC31.EXE on your computer; it is supplied with many program development systems.

EditHelp also has an internal Help Compiler.

Whether the External or Internal Compiler is used can be selected via the HelpFile|Directories dialog.

Edit|Link

A link is specified in the source text by the Link command:

{link=<num>,<text>}

where <num> is the Topic number of the destination and <text> is the link-text to be displayed.

For instance, the Text Layout topic number is 114 so a link to the Text Layout topic would be specified as

{link=114,Text Layout}

which would display as

Text Layout

When the user selects the link, the destination topic is displayed.

A bitmap can be used as a button which displays the new topic.

A link command can also specify that a Help Macro should be executed. Select the 'Macro' button of the Edit|Link dialog.

The Edit|Link command inserts a Link into the text of the current topic at the current insertion point. You can choose the link topic from a list of all topics.

You can also type in a link "by hand".

A link can specify that the destination topic should be displayed in a Secondary Window or that a New File should be loaded.

See also
- Edit|Undo
- Edit|Cut
- Edit|Copy
- Edit|Paste
- Edit|Delete
- Edit|Popup
- Edit|Bitmap
- Edit|Bitmap Link
- Edit|DLL Call
- Edit|Embedded Pane

Text Layout

The Edit window of the EditHelp program shows the text of the current topic.

The layout of the text affects the layout of the Help File.

Single newlines are ignored in the text so that the source:

> Single newlines are
> ignored in
> the text

is displayed by the Help system as:

> Single newlines are ignored in the text

Two or more newlines start a new paragraph, so that:

> Single newlines
> are ignored.
>
> But two newlines
> start a new paragraph.

is displayed as:

> Single newlines are ignored.
>
> But two newlines start a new paragraph.

A '\' character at the end of a line is displayed as a newline, so:

> Single newlines\
> are ignored.

is displayed as:

> Single newlines
> are ignored.

If the first line of a paragraph has leading blanks then the whole paragraph is indented. Leading blanks on subsequent lines are ignored.

Trailing blanks on a line are deleted; a single trailing blank is then added. This is usually what is required in normal text - if you want a specific number of trailing blanks, use the Literal command.

Lines must have fewer than 255 characters.

Text can be displayed bold or italic and in different fonts and sizes.

See also
  Text Commands

Advanced layout includes
  Heading command
  Line command
  List Command
  Box command
  Centre command
  Keep command

Text Commands

The source text can contain Commands specifying links, sub-headings and bitmaps to be included in the Help File.

A Link is specified by the Link Command:

    {link=<num>,<text>}

where <num> is the Topic number of the destination and <text> is the link-text to be displayed.

A Popup Link is specified by the Popup Command:

    {popup=<num>,<text>}

where <num> is the Topic number of the destination and <text> is the popup link-text to be displayed.

A Bitmap is specified by the Bitmap Command:

    {bitmap=<filename>}

where <filename> is the name of a BMP file to be displayed.

A sub-heading is specified by the Heading Command:

    {heading=<text>}

where <text> is the text of the sub-heading.

The command name for every text command must appear in lower case. For instance, use 'link' not 'Link'.

Every text command starts with a '{' character. The '{' character may be inserted into the text as

    {{}

See also
- Text Layout
- Heading command
- Line command
- List command
- Box command
- Centre command
- Keep command

File|Save As TXT

The File|Save As TXT command saves the current source to a file as plain ASCII text. The text file will have the name:

    <filename>.TXT

where <filename> is the name of the current help source file.

Each topic appears on a separate page; pages are separated by FF (#12) characters.

If you select 'Remove all formatting' then all formatting and link commands are removed. Otherwise, link commands will be replaced with the text of the hotspot but other formatting commands will be written to the TXT file; for instance

    {font=6}

will be written as

    <<font=6>>

You can use the text file as the basis of your user manual.

See also
- File|New
- File|Open
- File|Save
- File|Save As
- File|Project|New
- File|Project|Open
- File|Project|Save
- File|Project|SaveAs
- File|Project|Files
- File|Project|Move Topic
- File|Save As RTF
- File|Save As WWW
- File|Open EXE
- File|WWW Mode
- File|Exit

HelpFile|Keywords

The HelpFile|Keywords command displays the Keywords, Topic number and Caption of each topic. The display is sorted by all of the keywords of each topic.

The Keywords command helps you keep track of which keywords are common to several topics.

See also

- HelpFile|Make
- HelpFile|Test
- HelpFile|Make This Topic
- HelpFile|Directories
- HelpFile|Title
- HelpFile|Compress
- HelpFile|Copyright
- HelpFile|Icon

Topic|Goto Unfinished

The Topic|Goto Unfinished command selects the next unfinished topic from the topic list and load it into the Edit window. An unfinished topic is one whose status is 'Incomplete' or Test'.

The Goto Unfinished command can be called with the Ctrl-F6 key.

See also
Topic|Status
HelpFile|Status

Topic|Next topic
Topic|Prev topic
Topic|Back
Topic|New topic
Topic|Delete
Topic|Caption
Topic|Contents
Topic|Browse
Topic|Entry Command
Topic|Preview

Search|Find Error

The Search|Find Error command searches for an error reported by the Help Compiler.

If the Help Compiler reports an error such as

    Error 4639: Error in file '~out.rtf' at byte offset 0x1BC9.

enter the error address (e.g. 1BC9) into the Find Error dialog box. The cursor will move to a position near the error in the source.

Warning: The Help Compiler seems to report the wrong address for errors if you have linked Baggage into the Help File

See also
- Search|Find
- Search|Replace
- Search|Replace with Link
- Search|Next

Edit|Popup

The Edit|Popup command inserts a Popup Link into the text of the current topic at the current insertion point. You can choose the popup topic from a list of all topics.

You can also type in a popup link "by hand".

See also
- Edit|Undo
- Edit|Cut
- Edit|Copy
- Edit|Paste
- Edit|Delete
- Edit|Bitmap
- Edit|Link
- Edit|DLL Call
- Edit|Embedded Pane

Edit|Bitmap

The Edit|Bitmap command inserts a Bitmap into the text of the current topic at the current insertion point. You can choose the Bitmap from a list of BMP files.

You can also type in a Bitmap command "by hand".

The bitmap filename should not contain any path information. The bitmap file should be in the same directory as the EDH source file.

See also

- Edit|Undo
- Edit|Cut
- Edit|Copy
- Edit|Paste
- Edit|Delete
- Edit|Popup
- Edit|DLL Call
- Edit|Embedded Pane

Make command

Before the HelpFile|Make or HelpFile|Make This Topic command is run, you have the opportunity to save the current help source.

You may select:

Yes:	the source file is saved then the Help File is made
No:	the source file is not saved before the Help File is made
Always:	the source file is always saved before the Help File is made
Never:	the source file is never saved before the Help File is made

If you choose Always or Never, this dialog will not be presented again for this help source file.

If you choose No or Never, you can save the help source file later with the File|Save or File|Save As command.

See also
- HelpFile|Make
- HelpFile|Make This Topic
- Help Compiler
- Files

Help Project File

When you select the HelpFile|Make command, EditHelp writes the contents of the source file as a Rich Text Format file and a Help Project file. These are called:

~out.rtf Rich Text Format file
~out.hpjHelp Project file

It then executes the Help Compiler which compiles these into a Help File called

~out.hlpHelp file

The Help File is then copied to

<directory><filename>.hlp

where <filename> is the name of the current help source file and <directory> is the 'Help File Directory' specified by the HelpFile|Directories command.

A Help Project file consists of the following sections:

[OPTIONS]	Specifies options that control the build process. This section is optional. If this section is used, it should be the first section listed in the project file, so that the options will apply during the entire build process.
[FILES]	Specifies topic files to be included in the build. This section is required.
[BUILDTAGS]	Specifies valid build tags. This section is optional.
[CONFIG]	Specifies Help macros that define nonstandard menus, buttons, and macros used in the help file. This section is required if the help file uses any of these features. This section is new for Windows 3.1.
[BITMAPS]	Specifies bitmap files to be included in the build. This section is not required if the project file lists a path for bitmap files by using the BMROOT or ROOT option.
[MAP]	Associates context strings with context numbers. This section is optional.
[ALIAS]	Assigns one or more context strings to the same topic. This section is optional.
[WINDOWS]	Defines the characteristics of the primary Help window and the secondary-window types used in the help file. This section is required if the help file uses Secondary Windows.
[BAGGAGE]	Lists files that are to be placed within the help file (which contains its own file system). This section is optional.

You can specify extra text to be inserted into each of these sections. For instance, if the file

options.epj

exists in the same directory as the EditHelp source file, then the text of the EPJ file will be copied to the [OPTIONS] section of the Help Project file.

For instance, if the file

config.epj

contains the line:

```
CreateButton("btn_clk", "&Clock", "ExecProgram(`clock.exe',0)")
```

Then the Windows Help viewer will have an extra button labelled "Clock". If the user selects this button, the Windows Clock program will be run.

If you have written a program which must use context strings rather than context numbers, you can use the file ALIAS.EPJ to assign strings to topics.

MS-DOS

This is the MS-DOS program.

Paintbrush

This is the Paintbrush program.

File Manager

This is the File Manager program.

Microsoft Write

This is the Microsoft Write program.

Examples

This text will be kept while the rest of the page can be scrolled.

This Topic shows many of the features of Help files produced by EditHelp.

Text can be shown in many styles, colours, fonts and sizes:

AbCdEf AbCdEf *AbCdEf* AbCdEf AbCdEf ΑβΧδΕφ

Layout

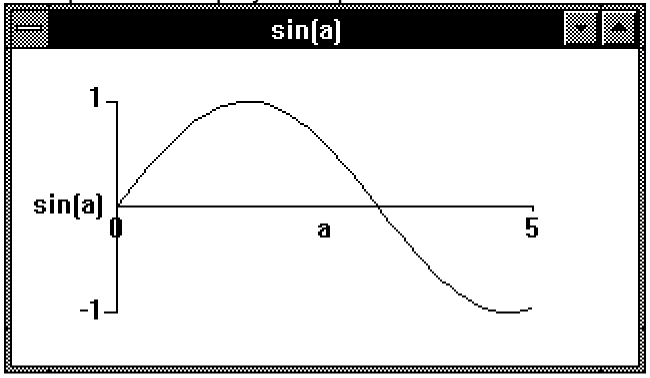
Here is an example of an indented list:

Feature	**Description**
Icon	The Icon command allows you to specify the name of an icon (.ICO) file. The icon will be displayed when the Help file is minimised.
Copyright	The Copyright command allows you to specify an additional copyright notice which will appear in the Windows Help About box.
Browse	The Browse command sets the browse group name and number of the current topic. A browse sequence typically consists of two or more related topics that are intended to be read sequentially.

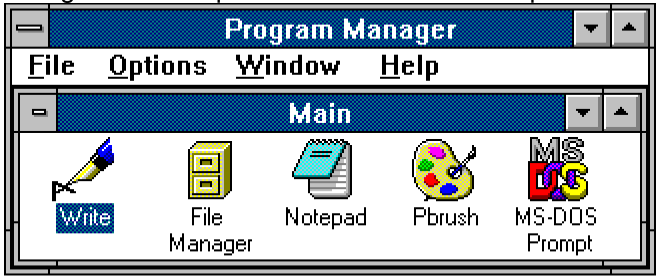
This text is centred in a box.

Pictures

A Help file can display bitmaps:



If you press this Button it will display a popup topic.
A Segmented Graphics File contains a bitmap that includes one or more hotspots. For instance:



The Write, FileManager, Pbrush and MS-DOS icons specify a popup link while selecting the Notepad icon will execute the Notepad program.

Text can also be displayed in a Secondary Window.

File|Save As RTF

The File|Save As RTF command saves the current source to a file in Rich Text Format. The file will have the name:

    <filename>.RTF

where <filename> is the name of the current help source file.

You can edit the RTF file using Microsoft Word. Microsoft Word is the traditional editor for producing Windows Help files. It is a powerful word processor but, unlike EditHelp, lacks many features specific to the creation and maintainance of Windows Help files.

See also
- File|New
- File|Open
- File|Save
- File|Save As
- File|Project|New
- File|Project|Open
- File|Project|Save
- File|Project|SaveAs
- File|Project|Files
- File|Project|Move Topic
- File|Save As TXT
- File|Save As WWW
- File|Open EXE
- File|WWW Mode
- File|Exit

Layout|Keep

This text will be kept while the rest of the page can be scrolled.

The Keep command specifies that the text of the paragraph containing the command and all the preceding text (including the Caption) should remain at the top of the window.

```
{keep}
```

The user can scroll all the text following the Keep command.

A line will be drawn below the non-scrolling region.

If you want just the Caption to appear in the non-scrolling region, insert a line contining nothing but

```
{keep}
```

as the first line of your topic text.

If the window is smaller than the non-scrolling region, the user will be unable to view the rest of the topic.

See also
- Layout|Heading
- Layout|List
- Layout|Line
- Layout|Box
- Layout|Centre
- Layout|Tab

Secondary Window Example

The following definition:

```
MyWindow="My Window",(0,0,500,500,0),,(255,255,0),,1
```

specifies that the Secondary Window should appear in the top left corner of the screen, have a yellow background and remain on top.

New File

If the format of a Link command is

    {link=<num>@<newFile>,<text>}

where '<newFile>' is the name of a Help file, then the new file is opened and the destination Topic is loaded from that file. ('<num>' is the Topic number of the destination and '<text>' is the link-text to be displayed.)

The '<newFile>' string should not contain any DOS path information.

For instance:

    {link=2231@myhelp.hlp,Other File}

will load the Help file

    myhelp.hlp

If a Link command specifies both a Secondary Window and a New File, the clauses should be in the order:

    {link=num@newFile>winName,text}

Literal

If a source line contains the Literal command:

    {literal}

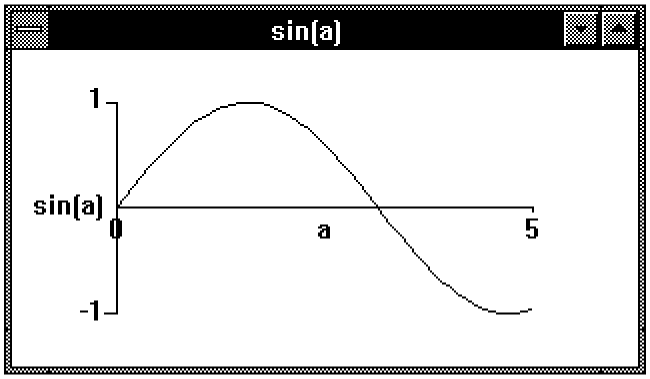
The the remaining text on the line is copied verbatim to the RTF file.

You can use this feature to insert RTF commands which are not otherwise available from EditHelp.

For instance

```
{literal} }{\par}\pard\{bml sample.bmp\}{\fs20 A
{{}bml ...} command before the start of a paragraph means
the paragraph text is wrapped to right of the bitmap.
```

is displayed as:



A {bml ...} command before the start of a paragraph means the paragraph text is wrapped to the right of the bitmap.

To understand why this example works, you will need an RTF viewer manual such as

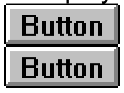
> Microsoft Multimedia Viewer: Technical Reference
> MM39685-0393

and you will need to examine the RTF file written by EditHelp.

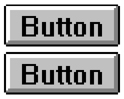
Normally, trailing blanks on a source line are deleted and a single trailing blank is then added. If you want a specific number of trailing blanks, use the Literal command towards the end of the line, followed by as many blanks as you need. For instance:

```
{bitmap=sample1.bmp}{literal}
{bitmap=sample1.bmp}
```

is displayed as:



without the {literal}, a blank would have been inserted:



Predefined Variables

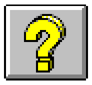
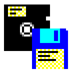
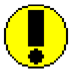
Windows Help allows allows the following Predefined Variables as parameter values for DLL Calls:

Variable	Description
hwndApp	the handle of the main Help window (U, DWORD,longint)
hwndContext	the handle of the current Help window (U, DWORD,longint)
qchPath	the the path of the Help file (S, LPSTR,^string)
lTopicNo	the current topic number (U, DWORD,longint)
coForeground	the foreground colour (U, DWORD,longint)
coBackground	the Background colour (U, DWORD,longint)

N.B. the "current topic number" is an internal number used by WinHelp. It is not the same as the Topic number used by EditHelp.

Sample Bitmaps

The following selection of bitmaps are supplied with EditHelp:

	btnpress.bmp
	btnok.bmp
	btncancl.bmp
	btnhelp.bmp
	btnhelp1.bmp
	disk1.bmp
	disk2.bmp
	exclamat.bmp
	informat.bmp
	prev.bmp
	next.bmp
	arrow.bmp
	point1.bmp
	point2.bmp

File|WWW Mode

The EditHelp program creates hypertext files for use with either the Windows Help system or the World-Wide-Web.

EditHelp can create files in HyperText Markup Language (HTML) format. HTML can be used to represent Hypertext news, mail, online documentation, etc. on the World Wide Web using a standard Internet protocol. To view an HTML file created by EditHelp, you will need an HTML Browser such as 'Mosaic'.

EditHelp can operate in two modes: 'Windows Help' and 'WWW'.

In WWW mode, EditHelp disables several features which are available in Windows Help but not in HTML files.

This Help file describes how EditHelp behaves in the 'Windows Help' mode. To change mode, select the File|WWW Mode command.

A file created in WWW mode can also be compiled as a Windows Help file.

See also
- File|New
- File|Open
- File|Save
- File|Save As
- File|Project|New
- File|Project|Open
- File|Project|Save
- File|Project|SaveAs
- File|Project|Files
- File|Project|Move Topic
- File|Save As TXT
- File|Save As RTF
- File|Save As WWW
- File|Open EXE
- File|Exit

Layout|Tab

Tab characters in the help source file (.EDH) will be translated into tabs in the Help File (.HLP).

Similarly, a tab command:

    {tab}

will be translated into a tab in the Help File.

Alternatively, a tab command can specify the tab position:

    {tab=<pos>}

The tab position, <pos>, is measured from the left edge of the window. The default is 6.

For instance:

```
1{tab}2{tab}3{tab}4\
1{tab=18}2<tab>3<tab>4\
1<tab>2{tab=18}3<tab>4\
1<tab>2<tab>3<tab>4\
```

where <tab> is a tab character, is displayed as:

```
1       2       3       4
1       2               3       4
1       2               3       4
1       2       3       4
```

See also
- Text Layout
- Layout|Heading
- Layout|List
- Layout|Line
- Layout|Box
- Layout|Centre
- Layout|Keep