

Quick and Dirty Help

Welcome to the Quick and Dirty way to build help files for Windows 3.x. With Quick and Dirty Help (QDHELP) you use any ASCII editor to generate help files for Windows 3.x. The ASCII file you generate is input to the QDHELP program which outputs a file in rich text format (RTF). This is the format needed by the Microsoft Help Compiler. What this means is that you do not need to purchase Microsoft WORD or some other word processor that generates RTF to be able to make help files for your Windows 3.x applications. The remainder of this help file contains information on the syntax used by the QDHELP program input file along with some hints and tips on using QDHELP.

Pick an Index, any Index!

[.HPJ File Section Commands](#)
[Quick and Dirty Help Commands](#)
[Running Quick and Dirty Help](#)
[Hints and Tips](#)

HPJ file commands

QDHELP now generates an HPJ file for you. To allow maximum freedom you can set any of the data in the different sections of the .HPJ file from within your .QDH file. This means you will not have to edit your .HPJ file manually. The sections and the equivalent QDHELP command are found below.

[OPTIONS]

[FILES]

[BUILDTAGS]

[CONFIG]

[BITMAPS]

[MAP]

[ALIAS]

[WINDOWS]

[BAGGAGE]

The /hpjoptions command

The /hpjoptions command has the following syntax:

/hpjoptions OPTION, value

The /hpjoptions command is used to set one of the option variables available in the .hpj file used by HC .
The valid options are the following:

BMROOT	Root directory for finding bitmaps.
BUILD	Define build criteria.
COMPRESS	Select type of Compression used.
CONTENTS	Select context of contents screen.
COPYRIGHT	Add copyright string to About dialog box.
ERRORLOG	File to write HC compilation messages to.
FORCEFONT	Force use of specific fonts.
ICON	Specify help minimized icon.
LANGUAGE	Sort order for Scandanavian language.
MAPFONTSIZE	Map fonts to different sizes.
MULTIKEY	Select alternate keyword mapping for topics.
OLDKEYPHRASE	Use old keyphrase table.
OPTCDROM	Optimize for CDROM use.
REPORT	Select display of build messages.
ROOT	Root directory to find topic and data files.
TITLE	Specify help window title bar text.
WARNING	Select level of warning messages.

The /hpjfiles command

The /hpjfiles command has the following syntax:

/hpjfiles filename

The /hpjfiles command is used to include files into the .hpj file used by HC . Normally QDHELP will insert the file name into the .hpj file for you. But if you have other .rtf files that you want included into a compile with a file that is being generated by QDHELP just include a line like the following:

/hpjfiles myfile.rtf

and that should do the trick.

The /hpjbuildtags command

The /hpjbuildtags command has the following syntax:

/hpjbuildtags BUILDTAG

The /hpjbuildtags command is used to specify valid build tags for the current help file. Normally QDHELP will insert all the build tags that it finds in your .QDH file into this section for you. This command is here so if you have other build tags that you need defined (let's say of other .RTF files you are including in this help file which were not generated by QDHELP) you can do so.

The /hpjconfig command (Win 3.1)

The /hpjconfig command has the following syntax:

/hpjconfig CONFIG DATA

The /hpjconfig command is used to set data in the [CONFIG] section of the .HPJ file. The config section is used to hold macros that will be executed when the help file is first started. For example, if you want to have left and right browse buttons in your help you would add the following command to your .QDH file:

```
/hpjconfig BrowseButtons()
```

This will cause the macro BrowseButtons() to be run on start up and your help will have browse buttons on the button bar. This is also the place where you register other DLL (Dynamic Link Library) Routines that you wish to call from within your help file.

The /hpjbitmaps command

The /hpjbitmaps command has the following syntax:

hpjbitmaps c:\path\file.bmp

This command specifies bitmap files to be included in the build. Bitmap files need only be specified if they can not be found in the directories in BMROOT or ROOT.

The /hpjmap command

The /hpjmap command has the following syntax:

/hpjmap TOPIC_NAME Value

The /hpjmap command allows you to map a value to a topic name for calling context sensitive help. If you're having QDHELP automatically generate your .HPJ file and generate a .HHH file (Header information on context sensitive help) QDHELP will put something like the following into the .HPJ file:

[MAP]

#include <helpname.hhh>

What this does is automatically draw in the .hhh file being generated by QDHELP. If you are going to use context sensitive help then you might as well let QDHELP do all the work for you!

If you want to manually place a context sensitive value into the .HPJ file use the /hpjmap command.

As an example, if you want to be able to call help and get to a topic named EXAMPLE_TOPIC add the following to your .QDH file:

/hpjmap EXAMPLE_TOPIC 0x0001

If you call WinHelp for context sensitive help and pass a value of 0x0001 for the value WinHelp will bring up EXAMPLE_TOPIC.

The /hpjalias command

The /hpjalias command has the following syntax:

hpjalias TOPIC_NAME=ALIAS

This makes ALIAS a second keyword which can be used for TOPIC_NAME.

The /hpjwindows command (Win 3.1)

The /hpjwindows command has the following syntax:

hpjwindows win_name="caption",(h_pos,v_pos,width,height),sizing,(RGB client),(RGB nonscroll)

This command tells the help compiler the size, location, and colors for a secondary help window. If the win_name = "main" these properties are applied to the main help window.

The /hpjbaggage command (Win 3.1)

The /hpjbaggage command has the following syntax:

hpjbaggage file

The /hpjbaggage command allows you to tell the help compiler to store files in its internal file system and not on the DOS file system. This allows better access time for multimedia data.

Quick and Dirty Help Commands

Below you will find a list of the commands found in the QDHELP system. Commands must be the first non white space characters found on a line.

Command Index

.HPJ file Commands

[/hpjoptions](#)
[/hpjfiles](#)
[/hpjbuildtags](#)
[/hpjconfig](#)
[/hpjbitmaps](#)
[/hpjmap](#)
[/hpjalias](#)
[/hpjwindows](#)
[/hpjbaggage](#)

Document Commands

[/include](#)
[/defformat](#)
[/topic ... /endtopic](#)
[/glossarytopic ... /endtopic](#)
[/pragma option](#)

Topic Commands

[/title](#)
[/keywords](#)
[/browse](#)
[/defformat](#)
[/para ... /endpara](#)
[/topicmacro](#)
[/helpid](#)

Paragraph Commands

[/text](#)
[/link](#)
[/bitmaplink](#)
[/popuplink](#)
[/macrolink](#)
[/bitmap](#)

The /include command

The /include command has the following syntax:

/include FILE-NAME

The include command is used to include files into a .QDH file. The file to include is the only parameter on the line following the include command. Any file name is ok. If the file can not be found and does not have an extension then the FILENAME.QDH will be tried. If both fail an error is reported and the program terminates.

Example:

/include filename.qdh

The /defformat command

The /defformat command has the following syntax:

/defformat format commands

The /defformat command allows you to set the default format for the entire document. The format commands can be text formatting commands. This format will be applied to all text in the document.

If the /defformat command is used inside of a /topic then the format will only be used for that topic. If there is a document wide format a topic format will override the document format. You will notice that this topic is a different color from the rest of the topics. This was caused using a /defformat command in this topic to change the text color. The actual command is given below.

Example:

/defformat lcf9

Once a /defformat is in place all text in the document will have those formatting attributes applied to it. You can, however, override the /defformat commands. For example, if you set the defformat to the following:

```
/defformat \fs20\cf5
```

This sets the font size to 20 and the foreground color to color number 5 for the whole document. Now, if inside a topic you use another /defformat command

```
/topic
```

```
/defformat \fs40
```

```
/endtopic
```

You will override the document font size of 20 with a font size of 40, but since you did not change the foreground color it will remain color 5.

Using this information it is possible to lay out your /defformat so as to make your help writing as easy as possible. If most of your paragraphs in a topic need a space of 100 after them but some need as space of 0, set your /defformat in the topic as follows:

```
/defformat \sa100
```

Now on the paragraphs that need 0 space do the following:

```
/para \sa000
```

```
/endpara
```

This will cause this single paragraph to have no spacing after it, which is just what we wanted.

The /topic command

The /topic command has the following syntax:

/topic TOPIC_NAME [,BUILDTAG;BUILDTAG...]

topic commands...

/endtopic

The topic command is used along with the /endtopic command to mark the beginning and end of a help topic. Most other commands are placed inside the /topic .. /endtopic pair. The TOPIC_NAME is the internal name of the topic. The BUILDTAGs are optional. Buildtags allow you to select which portions of a help file to build by specifying only topics with certain buildtags that are to be built. You specify the buildtags wanted in the [OPTIONS] portion of the .hbj file by setting the BUILD option.

The /glossarytopic command

The /glossarytopic command has the following syntax:

/glossarytopic TOPIC_NAME [,BUILDTAG;BUILDTAG...]

topic commands...

/endtopic

The glossarytopic command is used along with the /endtopic command to mark the beginning and end of a help topic. Most other commands are placed inside the /topic .. /endtopic pair. The TOPIC_NAME is the internal name of the topic. The BUILDTAGs are optional. Buildtags allow you to select which portions of a help file to build by specifying only topics with certain buildtags that are to be built. You specify the buildtags wanted in the [OPTIONS] portion of the .hpf file by setting the BUILD option.

The only thing different about a glossarytopic and a regular topic is that the glossary topic will automatically be put into the glossary generated by QDHELP. This means if you have a number of topics you want to be in a glossary just put them into the QDH file as glossarytopics instead of regular topics and QDHELP will do the rest for you.

The /pragma command

The /pragma command has the following syntax:

/pragma option

The /pragma command is used to set internal QDHELP processor options. The following options are supported:

debug
noheader
nohpj
vbheader
nodatabase

The options cause the following actions:

debug

This option causes QDHELP to put special RTF commands into the resultant help file. The commands put a new button on the WinHelp button bar named "Source". This button will bring up the file which contains the source for the help page which you are currently viewing. The default editor used is Windows Notepad.

If you do not wish to use NOTEPAD you must set the environment variable QDHELP_DEBUG_EDITOR in your DOS environment prior to running QDHELP...

set QDHELP_DEBUG_EDITOR=b.exe

NOTE: you must be in the directory with the source files for the debug source browse to work. If you run WinHelp by calling it with a full path to the help file the source browse will not be able to find the source files.

noheader

This option suppresses the output of the .HHH file. The QDHELP default is to generate the .HHH file.

nohpj

This option suppresses the output of the .HPJ file. The QDHELP default is to generate the .HPJ file. If you suppress the .HPJ file you must place the correct information in the .HPJ file for the /debug option to work. Look at a .HPJ file generated by QDHELP with debug on and off to see what special things QDHELP adds to the .HPJ file for debug.

vbheader

This option causes the output of the .VB file. The QDHELP default is to not generate the .VB file. The .VB file contains the global defs for the context sensitive help ids you need.

nodatabase

This option suppresses the output of the .DB file. The QDHELP default is to generate the .DB file.

The /title command

The /title command has the following syntax:

/title title string

The title command is used to set the text string which will be displayed as the title of the topic during a successful keyword search.

Command Placement

This command must be placed after the /topic command but before any /para command.

The /keywords command

The /keywords command has the following syntax:

/keywords keyword1[;keyword2...]

The keywords command is used to set the keywords that will find this topic in a help keyword search. An example might look like

/keywords Gates;Microsoft;Money;Power
for a topic that dealt with Microsoft!

Command Placement

This command must be placed after the /topic command but before any /para command.

The /browse command

The /browse command has the following syntax:

/browse browsename,browsenum or AUTO

The browse command is used to set the browse sequence numbers used by the Windows 3.0 help window to determine what order to browse through the help topics in.

The browsename is the name of the browse category that the browsenum is part of. For example, you may have one category called commands and one called examples. In this case the order of the browsenum values is only valid in reference to the browsename that number is associated with. The browsename is just any name you find suitable for your purposes. When in help and while viewing a topic that is in a browsename category, if the user hits the forward browse key he/she will be moved to the topic which is in the same browsename category and has the next highest number for browsenum. The reverse browse key will move to the same browsename category with the next lowest browsenum.

READ THIS... If instead of a number you put the string AUTO in the browsenum place QDHELP will automatically increment the browse count for you. This is a GREAT feature that I use in all the help files I write. You no longer have to number the sequences on your own. Just put AUTO in browsenum and you will browse through the topics in the same order that you have them in the file. If you use AUTO do NOT use any numbers for browsenum in the same browsename. This will cause unpredictable results since the number you choose may have already been used by the AUTO command.

Command Placement

This command must be placed after the /topic command but before any /para command.

The /topicmacro command (Win 3.1)

The /topicmacro command has the following syntax:

/topicmacro macro1[;macro2...]

The /topicmacro command is used to cause a macro to be run when a topic is activated via a hotspot, etc. In this way you can cause a macro to be run when a topic is selected. Select this to see an [Example](#)

Command Placement

This command must be placed after the /topic command but before any /para command.

This is an example. Notice how the About box came up when you chose this topic! That was a /topicmacro running

The /helpid command

The /helpid command has the following syntax:

/helpid contextNumber

The helpid command is used to set the context sensitive help id of a help topic. It is optional. It is provided in case the context numbers are predetermined and QDHELP must generate the numbers already defined for a topic.

Command Placement

This command must be placed inside a topic.

The /para command

The /para command has the following syntax:

/para paragraph specific format strings
...text strings...
/endpara

The /para command is used to let the QDHELP system know that this is the start of a new paragraph. Inside of the /para command there are a number of commands that can be used to set up hyperlinks and popu links to other parts of the help system.

Command Placement

This command must be placed after the /topic,/title,/keyword,/browse commands.

Commands used inside /para /endpara pair.

Internal to the /para command there are a number of commands that can be used to create links to other parts of the help system. These commands are listed below.

/text command
/link command
/popup command
/bitmaplink command
/bitmap command
/macrolink command

Paragraph format strings

Here you will find a list of formatting commands that can be placed after the /para command. If the reader is familiar with the rich text format, they may find the look of these commands very similar. This is because these are RTF commands!

Paragraph Justification.

\ql left justified paragraph
\qr right justified paragraph
\qj justified paragraph
\qc centered paragraph
[Paragraph Justify Examples](#)

Paragraph Indentation

\fi000 first line of paragraph indent.
\li000 left indent.
\ri000 right indent
[Paragraph Indent Examples](#)

Paragraph Spacing

\sb000 space before paragraph.
\sa000 space after paragraph.
\sl000 space between lines.
[Paragraph Spacing Examples](#)

Paragraph Border Placement

\brdrt border on top of paragraph.
\brdrb border on bottom of paragraph.
\brdrl border on left of paragraph.
\brdrr border on right of paragraph
\box border around entire paragraph
[Border Placement Examples](#)

Paragraph Border Style

\brdrs single thick border
\brdrth thick border
\brdrsh shadow border
\brdrdb double border
[Border Style Examples](#)

Paragraph Text Wrap/Scroll Control

\keep non scrolling region (Win 3.1)
\keep non word wrapping region (Win 3.1)
[Scroll Wrap Examples \(Win 3.1\)](#)

Paragraph Justification Examples.

This paragraph is left justified. This is just some text so that you can see that it is left justified! It was generated with the command **`/para \ql`**

This paragraph is right justified. This is just some text so that you can see that it is right justified! It was generated with the command **`/para \qr`**

This paragraph is justified. This is just some text so that you can see that it is justified! It was generated with the command **`/para \qj`**

This paragraph is centered. This is just some text so that you can see that it is centered! It was generated with the command **`/para \qc`**

Paragraph Indentation Examples

This is a paragraph where the first line is indented. It was generated using the command `/para \fi300` where 300 is the number of twips to indent the first line.

This is a paragraph that is indented on the left. It was generated using the command `/para \li300` where 300 is the number of twips to indent the entire paragraph.

This is a paragraph that is indented on the right. It was generated using the command `/para \ri300` where 300 is the number of twips to indent the entire paragraph

Paragraph Spacing Examples

This is a paragraph with a lot of space before it. It was generated with the command **/para \sb500** where 500 is the number of twips between the previous paragraph and this paragraph.

This is a paragraph with a lot of space after it. It was generated with the command **/para \sa500** where 500 is the number of twips between the previous paragraph and this paragraph.

This is a paragraph with alot of space between the lines. If you only have one line resize the help window to a smaller size to cause more than one line to be used. This paragraph was generated with the command **/para \sl450** where 450 is the number of twips between the lines.

This is a paragraph with the border placed only on top of it. It was generated with the command **`/para
\brdt`**

This is a paragraph with the border placed only on the bottom of it. It was generated with the command **`/para \brdrb`**

This is a paragraph with the border placed only on the left of it. It was generated with the command **`/para
\brdrl`**

This is a paragraph with the border placed only on the right of it. It was generated with the command **`/para \brdrr`**

This is a paragraph with the border placed all the way around it. It was generated with the command **`/para \box`**

This is a paragraph with a border of single thickness around it. It was generated with the command **`/para \box\brdrs`**.

This is a paragraph with a thick border around it. It was generated with the command **`/para \box\brdrth`**.

This is a paragraph with a shadow border around it. It was generated with the command **`/para \box\brdrsh`**.

This is a paragraph with a double border around it. It was generated with the command **`/para \box\brdrdb`**.

This text is non scrolling!

This text does not word wrap no matter how small you make the window.
It just stays the way it is.

This is just some text to scroll so you can see that the top part of the window does not scroll

1
2
3
4
5
6
7
8
9
10
11
12
13
14
15
16
17
18
19
20
21
22
23
24
25
26
27
28
29
30

The /text command

The /text command has the following syntax:

/text text format commands , **the text**

or

{ text format commands **the text** **}**

In the first syntax the format properties will be applied to the text found between the comma (,) and the end of the line which contains the /text command.

In the second syntax the text format commands come first, then the text **NO** comma in between. The second syntax can be used anywhere in the text string. That is to say that it can be used in the middle of a text string **{ \b for bold text }** ya see!

Text format strings

Following are strings that you can add to the /text command to cause the string printed by the text command to take on special properties. The output of these effects will depend on the font in use. Since these are really RTF commands you can put any RTF text formatting command into this position. These are the most common:

Examples

\tab	=	tab character
\b	=	bold text
\i	=	italic text
\strike	=	strikethru text
\f000	=	<u>font number</u> where 000 is the font of choice.
\fs000	=	font size in half points where 000 is the size.
\ul	=	underline text
\cf0	=	<u>foreground color value</u> where 0 is the color number of choice.

The following are examples of different text formatting commands that you can use. It is important to note that the help compiler uses certain sequences of commands to signal links,popups etc. Results of using these commands on text will be unpredictable if you use a sequence that help has defined for its own use. This should be a very rare case. Consult your help compiler manual to understand the sequences used by the help engine.

Input: /text \b, This is bold text

Output: **This is bold text**

Input: /text \i, This is italic text

Output: *This is italic text*

Input: /text \strike,This is strikethrough text

Output: ~~This is strikethrough text~~

Input: /text \fs50,This is font size 50 text

Output: **This is font size 50 text**

Input: /text \ul,This is underlined text

Output: This is underlined text

[Press here if you want to see the fonts](#)

[Press here to see the foreground colors](#)

The following list correlates font name to font number.

Input: \li100\tq\tx4000

text \f000,This is Tms Rmn,\tab value 000
text \f002,This is Helv,\tab value 002
text \f004,This is Courier,\tab value 004
text \f011,This is Helvetica,\tab value 011
text \f067,This is Modern,\tab value 067
text \f068,This is Roman,\tab value 068
text \f069,This is Arial TT,\tab value 069
text \f070,This is Cour TT,\tab value 070
text \f071,This is Symbol TT,\tab value 071
text \f072,This is Times TT,\tab value 072
text \f073,This is WingDing TT,\tab value 073

Output:

This is Tms Rmn,	value 000
This is Helv,	value 002
This is Courier,	value 004
This is Helvetica,	value 011
This is Modern,	value 067
This is Roman,	value 68
This is Arial TT,	value 69
This is Cour TT,	value 70
Τηισ ισ Σψμβολ TT,	παλνε 71
This is Times TT,	value 72
This is WingDing TT,	value 73

The following examples describe the use of tabs to place text in a table type format.

Tab stops by default are set at every half inch.

1 2 3 4 5 6 7 8

To change the tab stops we use the `\txNNN` command which sets the tabs to the number of twips NNN.

Tab stops are now set at every inch

1 2 3 4 5 6

You can also change the alignment of the text which is placed by the tabs.

left centered right

Colors

The following list correlates colors to color numbers:

Input: \fs40 \cf1 This is color 1

Output: This is color 1

Input: \fs40 \cf2 This is color 2

Output: This is color 2

Input: \fs40 \cf3 This is color 3

Output: This is color 3

Input: \fs40 \cf4 This is color 4

Output: This is color 4

Input: \fs40 \cf5 This is color 5

Output: This is color 5

Input: \fs40 \cf6 This is color 6

Output: This is color 6

Input: \fs40 \cf7 This is color 7

Output: This is color 7

Input: \fs40 \cf8 This is color 8

Output: This is color 8

Input: \fs40 \cf9 This is color 9

Output: This is color 9

Input: \fs40 \cf10 This is color 10

Output: This is color 10

Input: \fs40 \cf11 This is color 11

Output: This is color 11

Input: \fs40 \cf12 This is color 12

Output: This is color 12

Input: \fs40 \cf13 This is color 13

Output: This is color 13

Input: \fs40 \cf14 This is color 14

Output: This is color 14

Input: \fs40 \cf15 This is color 15

Output: This is color 15

Input: \fs40 \cf16 This is color 16

Output:

The /link command

The /link command has the following syntax:

/link TOPIC_NAME ,link string [,line]

The TOPIC_NAME is the name of the topic that this link should transfer the user to. The link string is the string that is printed in the help window. Comma line is optional. If it is included then any text following this link will start on a new line. This is the preferred method of placing each link on a line of its own.

The /popuplink command

The /popuplink command has the following syntax:

/popuplink TOPIC_NAME ,link string

The TOPIC_NAME is the name of the topic that this link should transfer the user to. The link string is the string that is printed in the help window.

The /bitmaplink command

The /bitmaplink command has the following syntax:

/bitmaplink TOPIC_NAME ,file.bmp

The TOPIC_NAME is the name of the topic that this link should transfer the user to. The filename is the name of the bitmap file where the image data will be gotten from. The file.bmp file name may contain path information. If there is path information QDHELP will strip it out. If QDHELP is generating the .HPJ file (**the default**) it will put the path information into the .HPJ file where it must go for HC. If QDHELP is not generating the .HPJ file then the path information will be tossed into the bit bucket and not used.

The /macrolink command (Win 3.1)

The /macrolink command has the following syntax:

/macrolink text,macro

The text in the above command is the hot spot text and the macro is the macro to be executed when the text is activated. There are too many macros to go over them all here. They are all documented in the Windows 3.1 SDK and also in Volume 4 of the Microsoft Windows 3.1 Documentation Set.

Select [Examples](#) to view some of the macros available.

Macro Examples (*Win 3.1*)

Here is where we try the macro link ability of WinHelp 3.1. The following is a small sampling of the macros available in the 3.1 help system. You really must get some of the Windows documentation to take full advantage of them. I suggest

Microsoft Windows 3.1
Programmer's Reference
Volume 4
Resources

Chapter 15 of this book covers all of the macros in detail, so if you want to really do the macro thing, pop for the 23 bucks and put a little more cash into Bill Gates retirement fund!

[Tell Me About it!](#)

[Annotate me](#)

[Go Back](#)

[Define a bookmark](#)

[More bookmarks](#)

[Show Browse Buttons](#)

[Add Time Button](#)

[Time now is Notepad](#)

[Destroy Time Button](#)

[Contents...](#)

[Copy Me](#)

[Copy topic to Clipboard](#)

Tab Examples

This will be the text for the tabs example. We will put some tabs here just to show you how to do it.

<u>Left</u>	Right	Centered /newline 1111	2222	3333 /newline 4444	5555
6666 /newline 7777		8888 9999			

The /bitmap command

The /bitmap command has the following syntax:

/bitmap position,file.bmp

The position in the above command is a single character with the following meaning:

l=left,r=right,c=character (which means treat the bitmap just like a character and put it wherever it goes in the sentence). The filename is the name of the bitmap file where the image data will be gotten from. The file.bmp file name may contain path information. If there is path information QDHELP will strip it out. If QDHELP is generating the .HPJ file **(this is the default)** it will put the path information into the .HPJ file where it must go for /popuplink HC,HC. If QDHELP is not generating the .HPJ file then the path information will be tossed into the bit bucket and not used.

Running Quick and Dirty Help

The following items contain information on running Quick and Dirty Help from the DOS command line. The QDHELP program is invoked on the DOS command line. The following syntax is used:

<code>qdhelpt inputfile [outputfile [headerfile]]</code>
--

QDHELP generates a file named inputfile.hhh which contains #defines for context sensitive help. If you include this file in both your .hpt file and .c file you will be all set to do context sensitive help.

To stop the output of the .hhh file use the /pragma noheader command.

QDHELP generates a file named inputfile.hpt which is the file needed by the Microsoft help compiler. When run with the flag the .hpt file will contain the correct .rtf file name which is being generated. If the -hC option is also on the .hpt file will contain the inputfile.hhh include file for context sensitive help.

To stop the output of the .hpt file use the /pragma nohpt command.

The outputfile is optional. If no output file name is given the output will be written to the default file inputfile.rtf.

If you specify an output you may also specify the name of the header file generated. If no name is given inputfile.hhh will be used.



Hints and Tips

The best way to get a feel for using the QDHELP system is to look over the file QUICK.QDH. This is the file that contains the input to QDHELP which was used to generate the help file that you are currently reading.

When developing your help system, first develop a standard look to each of your topics. This should include what the title will look like, the spacing of the paragraphs, the method you will use for subtitles, etc. By putting in a little work up front you will be able to create most of your topics from this template. Given an ASCII editor with cut and paste (the Windows 3.0 Notebook will do) you should be able to crank out your help file in a short amount of time.

To generate a blank line between 2 groups of text use the \sa300 command in the paragraph format string for the first paragraph.

To generate a list of links each on a separate line place a ,line at the end of each link command.

You really don't need to ask, do you?

Whitespace is non printable characters found in a text file such as tab or space. For the purposes of QDHELP the ONLY recognized whitespace characters are space and tab.

HC is the name of the Microsoft help compiler.

A twip is a unit used to measure output to a graphics device. A twip has a value of $1/1440$ of an inch. In other words, there are 1440 twips per inch.



Errata

Below is a list of known problems and work arounds if any are known

H

HC

M

MICROSOFT

T

TWIPS

W

WHITESPACE

