□□□

1. Intro

Welcome to QDHELP, the Quick and Dirty Help writer's assistant for MS Windows. QDHELP has a noble purpose, to save you money! Currently, you can get the Microsoft help compiler with a
number of software development products. It comes standard equipment with the MS Visual C++, Borland C++ for Windows, and Visual Basic Professional. The problem is that the help compiler expects its input to be in rich text format (RTF). One of the few ways to generate RTF is by using Microsoft Word. Coincidence? You can be the judge of that. Since QDHELP costs $49.00 (you do plan to register , don't you?) and Microsoft Word retails for $495.00 you can save yourself a cool $456.00 by using QDHELP to generate your help files. Of course if you already own Microsoft Word you save nothing except time. QDHELP makes it simple to generate a help file. Using any ASCII editor you can have MS Windows help with topics, standard links, popup links, bitmap links, browse sequences and keyword searches in no time!

QDHELP has many features that make your task of writing help files easier than ever before. This document presents detailed information on all of them. Try QDHELP and I'm sure you will be more productive as you generate MS Windows help files the Quick and Dirty way!

2. Registration

2.1 How To Register

QDHELP is shareware. Users can use an unregistered copy of QDHELP for thirty (30) days. After that time users must either register their version of QDHELP or delete QDHELP. Help files created with
an unregistered version of QDHELP may not be distributed in any form.

To register send $49.00 plus $3.00 S&H to the following address. Overseas orders add $2.00 for shipping and handling. Checks must be drawn on an American bank.

Phil Allen
2161 Fawnwood Dr. S.E.
Kentwood, MI 49508

Registered users receive technical support via Compserve and phone.

2.2 Bug Reports/Enhancement Requests

Please send all bug reports and enhancement requests to the above address. If you prefer to contact me via electronic means I can be reached at:

Compuserve	72047,2134
internet	allen@isgtec.com

2.3 Distribution

Since QDHELP is shareware you are encouraged to distribute it to friends, bulletin boards, information services , etc. as long as all the files in the original are distributed.

3. Getting Started

3.1 Running QDHELP

QDHELP is run from the DOS command line. The command has the following syntax:

qdhelp INPUT.QDH [OUTPUT.RTF]

The name of the input file is the first argument to the command. The second argument is the output file name and is optional. If the output file name is omitted the default file name input.rtf will be used, where input is the name of the input file (e.g. qdhelp TEST.QDH will produce an output file named TEST.RTF).

To begin you will need to use an ASCII editor to generate an input file for QDHELP. Any editor or word processor that can output straight ASCII text can be used. MS Windows Notepad is a good choice for starting. The default file extensions that QDHELP uses are .QDH for input files and .RTF for output files.

Once you have an input file that you want to generate a help file from simply type:

qdhelp FILENAME.QDH [OUTNAME.RTF]

where FILENAME.QDH is the file name and extension of your input file. QDHELP will process the input file. If no errors are found a file named OUTNAME.RTF will be produced. This is the file that you will run the Microsoft help compiler (HC) on. Your documentation on HC will explain the steps necessary to run HC.

3.2 Testing Your Help Files

To test your help file after compiling it with HC do the following:
-Enter MS Windows and from any application bring up the MS help.
- In help choose the File menu and the Open option. A dialog box will come up. Move to the appropriate
directory and choose the help file that you just compiled. This will load your help file for viewing.

If you are going to be working on a help file in one directory for a while do the following:
-Open one of your window groups, select the New option from the File menu in the Program Manager.
-Select the Program Item from the dialog box that comes up. Now in the input dialog input line labeled
Command Line enter the winhelp command followed by the directory and name of your help file.

For example, if you have a help file in a directory C:\MYHELP and the name of the help file is TEST.HLP you would enter the following in the Command Line input line:

winhelp C:\MYHELP\TEST.HLP

□□□

4. The HYPE

QDHELP has several features built in to allow you to generate help files for MS Windows 3.x quickly. The following list, while not complete, will give you an idea of the benefits of using QDHELP.

QDHELP lets you . . .

- Use your favorite programming (ASCII) editor to generate your help files. (Why should you have to bother learning a word processor?).
- Automatically generate the .HPJ file needed by the help compiler.
- Take advantage of all the new Windows 3.1 WinHelp features, such as running macros from hyperlinks.
- Automatically generate an .HHH file which contains the values needed to do context sensitive help.
- Automatically generate a glossary topic in your help file. You define which topics you want included in the glossary by using the /glossarytopic keyword to define those topics.
- Include RTF commands directly in your help.
- Use templates to get the same look and feel for all your help topics easily.
- Automatically number your browse sequences
- Include multiple files in your input, allowing you to better segment your help file writing.
- Generate help files with a syntax that make sense and is readable.
- Get done with those help files fast! (So you can get back to what you really want to do, cut some code!)

5. QDHELP Language Commands

Commands must be the first non white space characters on an input line. In QDHELP white space is a combination of space or tab characters.

This section will discuss the commands that the QDHELP language understands. Don't be intimidated, there are not that many. We will cover each command in some detail. The following is a list of all
the commands that QDHELP understands. They are sorted in order of likely usage in a help file. The
detailed explanations are in the same order.

Command	Description
/hpjoptions	pass data into the OPTIONS section of the .HPJ file
/hpjfiles	pass data into the FILES section of the .HPJ file
/hpjbuildtags	pass data into the BUILDTAGS section of the .HPJ file
/hpjconfig	pass data into the CONFIG section of the .HPJ file
/hpjbitmaps	pass data into the BITMAPS section of the .HPJ file
/hpjmap	pass data into the MAP section of the .HPJ file
/hpjalias	pass data into the ALIAS section of the .HPJ file
/hpjwindows	pass data into the WINDOWS section of the .HPJ file
/hpjbaggage	pass data into the BAGGAGE section of the .HPJ file
/include	includes another file in the current help file
/pragma	set option to influence the operation of the QDHELP parser
/defformat	set the default format for the entire document or a single topic
/topic	start the definition of a help topic
/glossarytopic	start the definition of a help topic and put the topic into the glossary
/endtopic	end the definition of a help topic
/title	set the title of a help topic
/keywords	set the search keywords for a help topic
/browse	set the browse category and position of a help topic
/defformat	set the default format for the document or topic
/para	start a paragraph
/endpara	end a paragraph
/topicmacro	assign a macro to execute when a topic is selected
/helpid	assign the context sensitive help value to this topic
/text	format text in a special way (i.e., bold, underline etc.)
/link	make a hypertext link
/popuplink	make a popup hypertext link
/bitmaplink	make a bitmap picture hypertext link
/macrolink	make a macro execution hypertext link
/bitmap	add a bitmap to the current help topic
//	comment, ignore rest of line

5.1 /hpjoption command

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 critera.
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 Scandinavian 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.

□□□

5.3 /hpjbuildtags Command

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 that if you have other build tags that you need defined (lets say of other .RTF files you are including in this help file which were not generated by QDHELP) you can do so.

□□□

5.5 /hpjbitmaps Command

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.

5.6 /hpjmap Command

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 are having QDHELP automatically generate your .HPJ file and generate an .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 pull 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.

□□□

5.8 /hpjwindows Command (Win 3.1)

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.

□□□

5.10 /include Command

Syntax:

/include [FILENAME | FILENAME.EXT]

This command inserts the text of another file into the input file at that point. The only parameter to the /include command is a file name. If the file name does not have an extension and the file cannot be found in the current directory, then the extension .QDH is added to the file name and an attempt to open the new file name is made. If the file cannot be found an error is displayed.

Placement:
The /include command is valid anywhere inside a QDHELP input file.

Limitations.
The include files can only be nested six (6) deep. This means you cannot have any more than six layers of include files including other include files. If you need more that six levels of include files find an easier subject for which to write a help file!

Example:

/include FILE2.QDH

This command will read in the file named FILE2.QDH. QDHELP will process the file just as if the text had been in the original file.

5.11 /defformat Command

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 is caused by using a /defformat command in this topic to change the text color. The actual command is given below.

Example:

/defformat \cf9

Once a /defformat is in place all text in the document will have the specified 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

the font size is set 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 because you did not change the foreground color it will remain color 5. Using this information it is possible to lay your /defformat out 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 causes the paragraph to have no spacing after it., which is just what we wanted.

□□□

□□□

5.13 /pragma Command

Syntax:

/pragma option

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

debug
noheader
nohpj

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 the special things QDHELP adds to the .HPJ file for debug.

□□□

□□□

□□□

□□□

Using the above defined topics, while topic1 is displayed by the help engine the browse backward key will be off and the browse forward key will be on. Pressing the browse forward key would bring you to
topic2. At this point the browse backward key would be on as well as the browse forward key. Hitting the browse forward key again brings us to topic3. At this point the browse backward key is on and the browse forward key is off because this is the last topic in category1.

For category2 we are using the AUTO browse numbering feature of QDHELP. This means that topic4 will be the first in the browse sequence for category2 (it came first in the input file) and that topic5 will be second and topic6 will be last. In all ways they will act just like topics 1,2 and 3. The benefit of using AUTO is that to add a new topic you do not have to know what the last position value used was. All you have to do is figure out where in the browse sequence you want the new topic to fall and add the text at that point in the input file.

□□□

□□□

□□□

□□□

□□□

□□□

□□□

□□□

□□□

□□□

□□□

/para
before the bitmap
/bitmap c,MYBITS.BMP
after the bitmap
/endpara

/endtopic

In the above topic the same bitmap would be displayed three (3) times. First the bitmap would be on the left margin of the help topic, then on the right margin, and then in mid sentence. The right and left margin justified bitmaps do not always come out before the text that is after them in the paragraph. We are working to find out if this is a limitation of the help engine or QDHELP.

□□□

□□□

□□□

7. An Annotated Example

To practice using QDHELP the following example will guide you step by step through the basics needed to generate a simple help file. As part of your distribution you will find a file named TPLATE1.QDH. Below you will find a line by line annotation of that file.

ANNO: the // characters signal a comment to QDHELP. The text following the // characters are
ANNO: ignored completely.

```
//
// template #1 for QDHELP
//
// This is a sample template for use in building
// help files using the QDHELP program. These
// templates give you the basic building blocks
// to quickly put together a help file for your
// programs.
//
//
```

ANNO: The /topic command is the QDHELP command that starts a help topic. In help files the basic ANNO: unit of information is the topic. The name following the /topic command is the name of the given
ANNO: topic. Here the topic is named INDEX. This is a name usually given to the first topic in the ANNO: help file. The index is the topic that is returned to when the Index button in the help engine is ANNO: pressed.

```
/topic INDEX
```

ANNO: The /title command is the title of the topic. The title of the topic is used when keyword searches
ANNO: are performed in the help engine. When the search is performed a list of keywords is shown. ANNO: When a search is performed on a keyword the title of all topics under which the keyword is found ANNO: are listed.

```
    /title Template #1
```

ANNO: The /para command signals the start of a new paragraph to QDHELP. The RTF commands found ANNO: after the/para command are formatting commands. In this specific example \sal 50 instructs the ANNO: Help Compiler to leave a space of 50 twips after this paragraph and to left justify the paragraph.

```
    /para \sa150
```

ANNO: The /text command signals QDHELP that we want to perform some special text formatting on the ANNO: text that follows on this line. The \b\i\fs50 instruct the Help Compiler that the text Template #1 ANNO: should be printed bold (\b) italic (\i) and with a font size of 50 twips (\fs50).

□□□

□□□

□□□

□□□