

QDHELP

Version 1.0

The Quick And Dirty Help Writers Assistant

Copyright (c) Phil Allen 1991-1992
All Rights Reserved

Notice

Phil Allen reserves the right to make improvements in the product described in this manual whenever without notice.

I have made every effort to be sure that QDHELP is an effective tool for generating help files for Windows 3.0. With that said, the following mandatory legal mumbo jumbo is necessary.

QDHELP is distributed "as is" without any warranty of any kind.

IN NO EVENT SHALL PHIL ALLEN BE LIABLE OR RESPONSIBLE FOR ANY LOSS DAMAGE OR OTHER PROBLEM CAUSED BY USING QDHELP. THE ENTIRE RISK OF USING QDHELP IS YOURS.

QDHELP

3

1. Intro

Welcome to QDHELP the quick and dirty help writers assistant for MS Windows 3.0. QDHELP has a noble purpose, to save you money! Currently, you can purchase the Microsoft help compiler for around \$50.00. It also comes as standard equipment with the MS Windows SDK and Borland C++ for Windows. 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 \$20.00 (you do plan to register don't you?) and Microsoft Word retails for \$495.00 you can save yourself a cool \$475.00 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 present detailed information on all of them. Try QDHELP and I'm sure you will be more productive. Generating MS Windows help files the Quick and Dirty way!

QDHELP

4

2. Registration

2.1 How To Register

QDHELP is shareware. Users can use an unregistered copy of QDHELP for sixty (60) days. After that time users must either register their version of QDHELP or delete QDHELP.

To register send \$20.00 to the following address.

Phil Allen
1185 Lowell Dr. Apt #5
Oconomowoc, Wi 53066

Registered users receive one (1) year of free updates also technical support via phone.

2.2 Bug Reports/Enhancement Requests

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

Compuserve 72047,2134
Genie P.ALLEN8
internet allen@mr.med.ge.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.

QDHELP

5

3. Getting Started

3.1 Running QDHELP

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

```
qdhhelp 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 out.rtf will be used.

To begin you will need to use an ascii editor to generate an input file for qdhhelp. Any editor or word processor that can output straight ascii text can be used. Microsoft Windows 3.0 notebook 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

```
qdhhelp FILENAME.QDH
```

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 OUT.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 and 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.

3.3 Hint and Tips

Read the README file in your distribution. This file contains the latest information that may have not made it into this manual.

QDHELP **is** case sensitive. This means you must enter the commands in lower case except where noted in this manual.

Do **not** put tab characters into your QDHELP text. You can tab to where the text will start for indentation purposes. If you want to put tabs into your text use the RTF command \tab. This will

QDHELP

6

insert a tab into the help file.

QDHELP comes with extensive on-line documentation in the file QDHELP.HLP. You can load this file into the MS Windows help for viewing. See Section 3.3 Testing Your Help Files. The QDHELP file that generated this help is included in your distribution its name is QDHELP.QDH. This is a good place to start looking to see how to lay out a QDHELP file. To generate QDHELP.HLP from source do the following

```
QDHELP QDHELP.QDH
```

This should output a file named OUT.RTF.

```
HC QDHELP.HPJ
```

This runs the help compiler and generates the file QDHELP.HLP. This of course is the file that you can load and run under MS Windows.

QDHELP

7

4. The HYPE

QDHELP has several features built in to allow you to generate help files for MS Windows 3.0 quickly. The following list while not complete will give you an idea of the benefits of using QDHELP.

QDHELP lets you . . .

- o Use your favorite programming (ascii) editor to generate your help files.
(Why should you have to bother learning a Word Processor).
- o Include RTF commands directly in your help.
- o Use templates to get the same look and feel for all your help topics easily.
- o Automatically number your browse sequences
- o include multiple files in your input. Allowing you to better segment your help file writing.
- o generate help files with a syntax that makes sense, and is readable.
- o Get done with those help files fast! (So you can get back to what you really want to do, cut some code!)

QDHELP

8

5. QDHELP Language Commands

Commands must be the first non white space characters on an input line. In QDHELP white space is 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

/include	includes another file in the current help file
/topic	start the definition of a help topic
/endtopic	end the definition of a help topic
/title	set the title of a help topic
/keyword	set the search keywords for a help topic
/browse topic	set the browse category and position of a help topic
/para	start a paragraph
/endpara	end a paragraph
/text	format text in a special way (i.e., bold, underline etc.)
/link	make a hypertext link
/popumlink	make a popup hypertext link
/bitmaplink	make a bitmap picture hypertext link
/bitmap	add a bitmap to the current help topic
//	comment ignore rest of line

QDHELP

9

5.1 /include Command

Syntax:

/include [FILENAME | FILENAME.EXT]

It 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 and 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 than six levels of include files find an easier subject for which to write a help file!

Example:

```
/include FILE2.QDH
```

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

QDHELP

10

5.2 /topic /endtopic Commands

Syntax:

/topic TOPIC_NAME

...

/endtopic

The topic command signals the beginning of the definition of a new topic. There are several commands that are only valid within a topic definition. The following commands can only be found inside a topic definition /title, /keyword, /browse, /para and /endpara. You also will find that there are commands which are only legal inside of the /para .. /endpara commands.

Placement:

The /topic commands can be placed anywhere in an input file except inside another /topic command.

Limitations:

Topics definitions cannot be nested.

Example:

More information about the commands used inside the topic command can be found later in this manual.

For information of command placement see section 6.1 Command Placement.

```
/topic TEST_TOPIC
```

```
    /title Test Topic Search Title
    /keyword test;topic;search;title
    /browse test:AUTO
```

```
    /para
    This is the first and last paragraph in the test topic help topic.
    /endpara
```

```
/endpara
```

QDHELP

11

5.3 /title Command

Syntax:

/title search title text

Use the title command to designate the search title for the topic. In the above example the string "search title text" would become the search title for this topic. The search title is display when a search is done on a keyword defined for the topic.

Placement:

The title command is only valid within a topic definition before any paragraphs have been defined.

Limitations:

None.

Example:

See the example for the /keyword command.

QDHELP

13

Figure 1.

3 Topics Found

|Edit Menu Commands |
|Copying and Pasting Text |
Pasting,Copying, and Cutting Pictures

QDHELP

14

5.5 /browse Command

Syntax:

/browse browse-category-name,position

The browse category name is the name of the browse category that this topic will belong to. Inside each browse category each topic has a position. The position is the value that tells the help engine which topic to go to when the user presses the browse forward and browse backward keys.

In QDHELP position can take on a special value AUTO (It must be in upper case). When position has AUTO in it, QDHELP will generate the position values for the browse sequence. They will be in the same order as the topics are found in the input file. This allows you not to have to worry about renumbering the position of your browse sequences. Just use AUTO and put the topics into your input file in the same order as you want your browse sequence. To change the sequence just cut and paste the topics into the correct order.

Placement:

The browse command is only valid in a topic definition before any paragraphs have been defined.

Limitations:

If you use the AUTO feature to auto number your browse sequences all positions in the category also must use AUTO.

Example:

```
/topic topic1  
/browse category1,1  
/endtopic
```

```
/topic topic2  
/browse category1,2  
/endtopic
```

```
/topic topic3  
/browse category1,3  
/endtopic
```

```
/topic topic4  
/browse category2,AUTO  
/endtopic
```

```
/topic5  
/browse category2,AUTO  
/endtopic
```

```
/topic6
```

QDHELP

15

```
/browse category2,AUTO  
/endtopic
```

Using the above defined topics, while topic1 was displayed by the help engine the browse backward key would be off, the browse forward key would 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 since 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 except the AUTO statement they will act just like topics 1,2 and 3. The benefit of using AUTO is that to add a new topic I do not have to know what the last position value used was. All I have to do is figure out where is the browse sequence I want the new topic to fall and add the text at that point in the input file.

QDHELP

16

5.6 /para /endpara Commands

Syntax:

/para [paragraph format commands]

...

/endpara

The /para /endpara commands mark the start and end of a paragraph respectively. There are several commands that are only valid inside a paragraph definition. They are /text, /link, /popuplink, /bitmap and /bitmaplink. For more information see section 6.1 Command Placement.

The paragraph format commands are RTF commands that are passed through to the help compiler unchanged. Below is a list of the most useful RTF commands for paragraph formatting and their meanings. Many commands deal with the unit called twips. A twip is 1/1440 of an inch (that is not very much!).

NOTE: that RTF commands begin with a backslash whereas QDHELP commands begin with a slash.

RTF Command	Category	Description

\ql	Justification	Paragraph left justified.
\qr	Justification	Paragraph right justified.
\qj	Justification	Paragraph plain justified.
\qc	Justification	Paragraph centered.
\fiXXX	Indentation	First line indented XXX twips
\liXXX	Indentation	Left margin all lines indented XXX twips
\riXXX	Indentation	Right margin all lines indented XXX twips
\saXXX	Spacing	XXX twips after the last line of the
paragraph		
\sbXXX	Spacing	XXX twips before the first line of
the paragraph.		
\slXXX	Spacing	XXX twips space between lines of
the paragraph		
\brdt	Border Placement	Border on top of the paragraph
\brdb	Border Placement	Border on bottom of the paragraph
\brdl	Border Placement	Border on left edge of the
paragraph		
\brdr	Border Placement	Border on right edge of the
paragraph		
\box	Border Placement	Border an all sides of the
paragraph		
\brds	Border Style	Single line border
\brdrth	Border Style	Single thick line border
\brdrsh	Border Style	Shadow border (try it you'll like it!)
\brdrdb	Border Style	Double line border

QDHELP

17

The paragraph format can contain any combination of these. Of course if you give it two (2) different border styles for one paragraph the results will be unpredictable.

Placement:

The para command is only valid in a topic after the definition of the title, keyword, and browse sequences. Of course a topic does not need to have a title, keyword, or browse command in it. But if it does the para must come after them.

Limitations:

None.

Example:

```
/topic PARA_FORMAT_EXAMPLE
```

```
    /title Paragraph Format Example
```

```
    /para \sa200 \box
This text would have a border drawn around every side and would have 200
twips of
blank space placed after it.
    /endpara
```

```
    /para \qc
This paragraph would be centered. It would appear 200 twips after the last
line of the
previous paragraph. This is due to the previous paragraphs \sa200 command.
    /endpara
```

```
/endtopic
```

QDHELP

18

5.7 /text Command

Syntax:

/text format commands , text string

The /text command allows special inline formatting of text in your paragraphs.

The text format commands are RTF commands that are passed through to the help compiler unchanged. Below is a list of the most useful RTF commands for text formatting and their meanings. Some commands deal with the unit called twips. A twip is 1/1440 of an inch (that is not very much!).

NOTE: that RTF commands begin with a backslash whereas QDHELP commands begin with a slash.

RTF Command	Category	Description

\b	Text Formatting	Bold Text
\i	Text Formatting	Italic Text
\strike	Text Formatting	Strike Thru Text
\ul	Text Formatting	Underline Text
\fsXXX	Text Formatting	Change font size to XXX twips
\fXXX	Text Formatting	Change font where XXX is 0 = Tms

Rmn,

2 = Helv and 4 = Courier.

Placement:

This command can only appear inside a /para /endpara group. If found outside an error will be generated.

Limitations:

None.

Example:

```
/topic TEXT_FORMAT_EXAMPLE  
  
  /title Text Format Example  
  
  /para  
  This is some non affected text  
  /text \b,This text would be printed bold  
  /endpara  
  
/endtopic
```

QDHELP

19

5.8 /link Command

Syntax:

/link TOPIC_NAME, link text

The link command is the basic way that you make hyperlinks between help topics. This command will cause a link from the topic named TOPIC_NAME . The link will be activated by pressing the mouse button while over the "link text" that will be underlined and printed in green (standard look for help hyperlink in MS Windows).

Placement:

This command can only appear inside a /para /endpara group. If found outside an error will be generated.

Limitations:

None.

Example:

```
/topic LINK_EXAMPLE

    /title Link Example

    /para
    This is some text that says nothing
    /link TOPIC_2, Topic 2 Link
    /endpara

/endtopic

/topic TOPIC_2

    /title Topic 2

    /para
    This is some more text that says nothing
    /link LINK_EXAMPLE, Back to the first one
    /endpara

/endtopic
```

In the above example each topic is linked to the other. In the first topic there will be an underlined green text "Topic 2 Link" that if clicked on with the left mouse button will take us to TOPIC_2. In topic 2 there is a link string "Back to the first one" that will take us back to the LINK_EXAMPLE topic.

QDHELP

20

5.9 /popuplink Command

Syntax:

/popuplink TOPIC_NAME, link text

The popuplink command is the way that you see a help topic without actually moving to it. This command will cause a link from the current topic to the TOPIC_NAME topic. The link will be activated by pressing the mouse button while over the "link text" that will be dashed underlined and printed in green (standard look for help popup hyperlink in MS Windows).

When activated the topic linked to will be displayed in a popup window for as long as the left mouse button is held down.

Placement:

This command can only appear inside a /para /endpara group. If found outside an error will be generated.

Limitations:

None.

Example:

```
/topic POPUP_LINK_EXAMPLE
```

```
    /title Popup Link Example
```

```
    /para
```

```
    This is some more text that is meaningless.
```

```
    /popuplink POPUP_TEXT, Show popup text
```

```
    /endpara
```

```
/endtopic
```

```
/topic POPUP_TEXT
```

```
    /title Popup Text
```

```
    /para
```

```
    This text will pop up in a window while the user holds down the left mouse  
button
```

```
    on the "Show popup text" link in the POPUP_LINK_EXAMPLE topic.
```

```
    /endpara
```

```
/endtopic.
```

The popup links work just like the regular links except when chosen instead of moving to the new topic the topic is shown in a popup window.

QDHELP

21

5.10 /bitmaplink Command

Syntax:

/bitmaplink TOPIC_NAME, FILENAME.BMP

The bitmaplink command is another way that you make hyperlinks between help topics. This command will cause a link from the current topic to the TOPIC_NAME topic. The link will be activated by pressing the mouse button while over the bitmap that is contained in the file name specified. The bitmap is stored as a standard windows .BMP file. The file name should **not** contain any DOS path information. The path information about where the bitmap file is located on your disk drive should be placed in the .HPJ file used by HC. Your help compiler manual should answer any questions you have regarding the .HPJ files contents.

Placement:

This command can only appear inside a /para /endpara group. If found outside an error will be generated.

Limitations:

None.

Example:

```
/topic BITMAPLINK_EXAMPLE
```

```
    /title Bitmaplink Example
```

```
    /para
```

```
    This is some text that says nothing
```

```
    /bitmaplink TOPIC_2, SOMEFILE.BMP
```

```
    /endpara
```

```
/endtopic
```

```
/topic TOPIC_2
```

```
    /title Topic 2
```

```
    /para
```

```
    This is some more text that says nothing
```

```
    /bitmaplink BITMAPLINK_EXAMPLE, MYFILE.BMP
```

```
    /endpara
```

```
/endtopic
```

In the above example each topic is linked to the other. In the first topic there will be the bitmap found in SOMEFILE.BMP. When clicked on with the left mouse button it will take us to TOPIC_2. In topic 2 there is a bitmap found in MYFILE.BMP that will take us back to the BITMAPLINK_EXAMPLE topic.

QDHELP

22

QDHELP

23

5.11 /bitmap Command

Syntax:

/bitmap position , FILENAME.BMP

The /bitmap command places a bitmap into the help file at a position. The bitmap is in the file named FILENAME.BMP. The file name should **not** contain any DOS path information. The path information about where the bitmap file is located on your disk drive should be placed in the .HPJ file used by HC (MS Help Compiler). Your help compiler manual should answer any questions you have regarding the .HPJ files contents. The position of the bitmap is determined by the value placed into the position parameter. The valid values are shown in Table 1.

Position Value	Description
l	Left, place the bitmap at the left margin of the current text.
r	Right, place the bitmap at the right margin of the current text.
c	Character, place the bitmap at the location that the next character in the sentence would have been placed.

Table 1.

Placement:

This command can only appear inside a /para /endpara group. If found outside an error will be generated.

Limitations:

Text does not always align the way you would expect when using this command. You must try the command and judge the results on a case by case basis.

Example:

```
/topic BITMAP_EXAMPLE

    /title Bitmap Example

    /para
    Left sided bitmap
    /bitmap l,MYBITS.BMP
    To the left!
    /endpara

    /para
    Right sided bitmap
    /bitmap r,MYBITS.BMP
    To the right!
    /endpara
```

QDHELP

24

```
/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 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 if this is a limitation of the help engine or QDHELP.

QDHELP

25

5.12 // Command

Syntax:

// comment text

The // is the comment character sequence in QDHELP. Any text found after a // will be ignored.

The // must be the first 2 characters on the line.

Example:

```
/topic COMMENT_EXAMPLE
```

```
//
```

```
// this is the title comment
```

```
//
```

```
    /title Comment Example
```

```
/endtopic
```

This is just a way to add some info to your QDHELP input file that may be worth remembering.

QDHELP

26

6. Language Structure

6.1 Command Placement Example

Some commands in QDHELP must be positioned in certain places. This section graphically describes the relationship of commands and placement in QDHELP.

The // command and the /include command can be used anywhere

```
//  
/include SOMEFILE.QDH
```

The /topic command has a specific order in which the commands should be placed in it.

```
/topic TOPIC1
```

```
-----  
| /title This is the title | <-- These commands must be between the  
/topic and                |  
| /keyword one;two;three  | the first /para command. They can  
be in any order.          |  
| /browse category,1      |  
|                          |  
-----
```

```
    /para  
    Some text  
    /endpara  
  
    /para  
-----  
| /text \b,BOLD TEXT      |  
| /link TOPIC2,Go To Topic 2 |  
| /popuplink TOPIC2,Popup Topic 2 | <-- These commands must be used  
inside the                |  
| /bitmaplink TOPIC2, TOPIC2.BMP | /para . . . /endpara  
commands. They can be    | use multiple times in any  
order.                    |  
|                          |  
-----  
    /endpara
```

```
/endtopic
```

QDHELP

27

6.2 Command Nesting

The `/topic /endtopic` and `/para /endpara` commands must occur in matched pairs and cannot be nested.

Incorrect use of `/topic /endtopic` commands

```
/topic TOPIC1
```

```
/topic TOPIC2 <-- ERROR The TOPIC1 definition must be finished before beginning  
TOPIC2
```

```
/endtopic
```

```
/endtopic
```

QDHELP

28

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 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 after the ANNO: /para command are formatting commands. In this specific example \sa150 instructs the help ANNO: 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)

QDHELP

29

and with a font size of 50 twips (\fs50).

```
/text \b\i\fs50,Template #1
```

ANNO: The /endpara command signals that the current paragraph definition is finished.

```
/endpara
```

ANNO: The only thing different about this paragraph is the \fi200 paragraph format command. This ANNO: command tells the help compiler that the first line in this paragraph should be indented 200 twips.

```
/para \fi200 \sa150
This is template #1 for the QDHELP system. This template
has a large title then an introductory paragraph. Follow
that comes bold topic headings and paragraphs for each
topic.
/endpara
```

ANNO: This paragraph contains no new information.

```
/para \sa150
/text \b\fs30,Topic Heading
/endpara
```

```
/para \fi200 \sa150
This is the topic heading paragraph. It will contain
information on the topic heading. Next you will find
a series of links to the subtopics of this topic.
/endpara
```

ANNO: This paragraph contains our first /link command

```
/para
```

ANNO: The /link command signals QDHELP to make a hypertext link between the current topic and the ANNO: topic named as the first parameter (in this case SUBTOPIC1). The second parameter is the text ANNO: that will appear for this link (in this case Sub Topic 1). This is the text the normally appears in ANNO: green with an underline in the windows help engine.

```
/link SUBTOPIC1,Sub Topic 1
/endpara
```

```
/para
/link SUBTOPIC2,Sub Topic 2
/endpara
```

ANNO: The /endtopic command signals QDHELP that the definition of the current

QDHELP

30

topic is finished.

/endtopic

//

// subtopic 1

//

/topic SUBTOPIC1

 /title Sub Topic 1

ANNO: The /keywords command signals QDHELP that the following keywords should be added to the ANNO: help compilers list of search keywords. If the users performs a search on the one of the

ANNO: keywords the title for this topic, along with any other topic that contains this keyword, will be ANNO: displayed.

 /keywords subtopic1;topic

ANNO: The /browse keyword signals QDHELP that this topic belongs to a browse category. The first ANNO: parameter after the /browse command is the browse category. The second is the position of this ANNO: topic in that browse category. In this example the category name is subtopic and this is the first ANNO: item in that category. Note: Once of the features of QDHELP is that instead of numbering each ANNO: item in a browse category you can let QDHELP do it for you. If instead of a numeric value you

ANNO: place the keyword AUTO (must be caps) as the second parameter to the /browse command ANNO: QDHELP will number the items for the browse category in the same order in which they occur in ANNO: the input file.

 /browse subtopic,1

ANNO: You have seen this before.

 /para \sa150

 /text \b\i\fs50,Sub Topic 1 Title

 /endpara

 /para

 This will be the text for the sub topic 1 information.

 /endpara

/endtopic

//

// subtopic 2

//

QDHELP

31

/topic SUBTOPIC2

/title Sub Topic 2
/keywords subtopic2;topic
/browse subtopic,2

/para \sa150
/text \b\i\fs50,Sub Topic 2 Title
/endpara

/para
This will be the text for the sub topic 2 information.
/endpara

/endtopic

The file TPLATE1.QDH is one of the template files distributed with QDHELP. The templates give you a starting place for developing a look and feel for your help files. My suggestion is that you copy one of the template files to new file and play with the paragraph and text formatting commands until you get a look that you like. This will become the template for all your topics in your help file.

QDHELP

32

Glossary

RTF	Rich Text Format
HC	Microsoft Help Compiler
twip	1/1440th of an inch
MS Windows	Microsoft Windows