

Contents

Overview

General Editing Considerations

RTFGEN Defaults

Steps for Building a Help File

ASCII Input File Layout

Commands

References

Copyright

Commands

Command Syntax

RTFGEN Commands

Topic Header Commands

Cross Reference Commands

Definitions

Bitmap Commands

Block Formatting Commands

RTF Commands

Font Commands

Paragraph Commands

Tabbing Commands

Overview

Normally, to create a help file with the Windows help compiler, you must first write a help topic file using a Rich Text Format (RTF) word processor. For many, there are two problems here:

1. RTF word processors are expensive.
2. Learning to use a new word processor is a nuisance and not very productive.

RTFGEN solves both these problems by allowing you to use your favorite ASCII text editor to produce RTF formatted files acceptable to the help compiler.

General Editing Considerations

Almost any ASCII editor may be used. I use Borland's TURBO or TPW. It's probably best to turn off options like 'optimal fill' which can leave invisible and possibly unknown tab characters in the document.

The use of tabs is required to obtain column alignment with proportional fonts. Either the tab character or the RTF command, \tab, may be used with the same effect. The \tab has the advantage that it's visible in your ASCII file.

RTFGEN and/or the RTF language have special use for the following characters:

`	(grave accent) Used to mark end of paragraphs.
[,]	Used to mark jumps and definitions.
{, }	Used to mark areas where font characteristics will change.
\	Used as a command start character.

When these characters are to be used in the text, they should be preceded by a '\'. Thus, '\[' will cause the left bracket to be inserted in text and '\\ will result in a single backslash being inserted in text.

(* . . *) may be used to insert comments within the text.

RTFGEN Defaults

Helv, 10 point font.

Paragraphs are left justified. The first line is not indented.

Paragraph indentation is set by indentation of the first line of the paragraph.

Tabs are spaced every 720 twips (about 6 default characters).

Any of these defaults may be changed globally for the whole document, or locally at the topic, paragraph, or even character level.

Steps for Building a Help File

The steps required to create a Windows help file using RTFGEN are:

1. Using an ASCII editor, create one or more input text files. [See input file layout](#)
2. Use RTFGEN to convert these files to RTF files.
3. Create a Help Project File (.HPJ) for input to the help compiler. This file should reference the RTF files.
4. Run the help compiler to make the help file.

Steps 3 and 4 are described in the help compiler documentation. This documentation deals with the first two steps only.

Once the ASCII files have been written, RTFGEN is used to convert them to suitable RTF format. The file, HEADING, must be present on the default drive. From the DOS prompt:

```
RTFGEN [Infile [Outfile]]
```

The optional filenames specify the names of the ASCII input file and the RTF file to be generated. If no extensions are given, .RTF is used by default for the output file. If the output filename is missing from the command line, an RTF file will be created using the same name as the source file. When neither filename is specified, names will be requested once execution starts.

```
RTFGEN MYHELP
```

will cause RTFGEN to look for a source file, MYHELP and create an object file, MYHELP.RTF.

The file, HEADING, is copied to the start of each RTF file generated. Along with some boilerplate RTF commands, it contains the font table defining fonts by number.

RTFGEN ASCII Input File Layout

document header
\docstart
topic header
=====

topic text

.....
topic header
=====

topic text

\docend

TWIP: A 20th of a point. For our purposes, 1440 twips per inch. One 10 point character is about 120 twips wide.

A grave accent, `', is used to mark paragraph ends.

The command, `\docstart`, marks the end of the document header and the start of the topics.

The command, `\docend`, marks the end of the document.

Start of Topic Text Marker

A row of 5 or more '='s starting in column 1 marks the end of a topic header and the start of the topic text.

End of Topic Marker

A row of 5 or more hyphens starting in column 1 marks the end of a topic.

An optional formfeed or several blank lines may follow the row of hyphens to better format your ASCII file.

Document Header

The optional document header may contain commands to override the RTFGEN defaults. Commands changing the font, font size, or paragraph first line indent might be appropriate here.

The \docstart command follows the document header to mark the start of the topics.

Example:

```
\f0 \fs24 ( * Times Roman font, 12 pitch for entire help file * )  
\docstart
```

Topic Header

The topic header contains commands defining build tags, context strings, title, keywords, and browse sequence number as appropriate for the topic. Topic global formatting commands may also be included. The topic header section is terminated with a row of 5 or more '='s.

Example:

```
\title Table of Contents`
\topic contents`
\keyword contents`
\fo \fs24 ( * Times Roman font, 12 pitch for this topic * )
===== ( * end of topic header * )
```

See also : [Topic Header Commands](#)

Topic Text

Text for the topic is entered here. Rules for text entry are:

Paragraph ends are marked with a grave accent, ``

Blank lines will appear as blank lines.

By default, paragraphs are left justified. The indenting of the paragraph is set by the indenting of the first line of the paragraph. Indenting of lines after the first is ignored.

Carriage returns and linefeeds within the paragraph are ignored as far as the final results are concerned.

Commands may be interspersed within the text for special effects. The commands themselves won't appear in the final result.

Examples:

ASCII

Here is a typical paragraph. It will appear in the font and pitch specified in the topic header or document header. It is left justified and indented by 360 twips because the first line is so indented. The paragraph is terminated with a grave accent character.`

Result

Here is a typical paragraph. It will appear in the font and pitch specified in the topic header or document header. It is left justified and indented by 360 twips because the first line is so indented. The paragraph is terminated with a grave accent character.

ASCII

\b\fi480 Here is another paragraph. It will appear in bold font because of the `\b` command. It will also be indented by 360 twips and left justified. The `\fi480` will cause the first line of the paragraph to be indented an additional 480 twips.`

Result

Here is another paragraph. It will appear in bold font because of the `\b` command. It will also be indented by 360 twips and left justified. The `\fi480` will cause the first line of the paragraph to be indented an additional 480 twips.

Command Syntax

Commands for RTFGEN and RTF commands passed thru to the help compiler have a similar syntax. Commands start with a backslash followed by a lowercase alpha symbol. Some RTF commands have a numerical value attached. Some RTFGEN commands have an additional parameter string.

Examples:

```
\box
\topic <topic name>
\fs24      RTF command with number
\fi-720    minus sign may be used if appropriate
```

When commands are mixed with text, a trailing space may be required as a delimiter. The first space following a command is considered part of the command--additional spaces are part of the text.

Topic Header Commands

```
\buildtag <parameter string>`  
\topic <parameter string>`  
\title <parameter string>`  
\keyword <parameter string>`  
\browse <parameter string>`
```

These commands correspond to the *, #, \$, K, and + footnotes described in the help compiler documentation. Each topic header will contain one or more of these commands as appropriate (ordering is unimportant).

The parameter string should--

- have format and contents exactly as described in the help compiler documentation for the equivalent footnote.

- be entirely on one line.

- be terminated by an end of paragraph mark.

More than one \keyword command may be used if the number of keywords would exceed the allowable editor line length.

Examples:

```
\topic proc_deleting_text`  
\title Deleting Text`  
\browse procedures:020`  
\keyword delete;clipboard`
```

Cross References

Cross references may be included in the topic text using the format:

[<green text>:<context>]

where the brackets indicate a cross reference, <green text> will appear highlighted, and <context> is the topic context string for the jump destination.

Example:

To change the color of an existing object, the object is first [selected:selecting]. Its color may then be changed.`

Definitions

Definitions may also be included in topic text by enclosing them in double brackets:

[[<green text>:<context>]]

Example:

Each space is equivalent to 120 [[twips:twip_def]].`

Bitmap Commands

`\bml <filename>`

`\bmr <filename>`

`\bmc <filename>`

These commands are used to insert bitmaps within the topic text.

Example:

ASCII

A bitmap may be placed in a sentence `\bmc glass.bmp` just like any character.`

Result



A bitmap may be placed in a sentence just like any character.

It's also possible to use a bitmap to cause a jump or bring up a definition as:

ASCII

`[[\bml tryme.bmp:bmpPop]]``

Result



Try Me

Illustrates a bitmap popup.

Block Formatting Commands

```
\blockstart <commands>`  
\blockend
```

The blockstart, blockend commands define a block over which a set of RTF commands will apply. These commands are used mainly when laying out tables where formatting is somewhat complex. An example of their use is given in the [tabbing commands section](#).

The block commands may be nested to a level of 4.

Font Commands

Fonts are referred to by number. Number vs font is set by the font table in the HEADING file. For the HEADING file supplied:

- 0 Times Roman
- 1 Symbol
- 2 Helv (default)
- 3 Courier

You can add fonts or change the order by editing the HEADING file.

<code>\f000</code>	change to font 000
<code>\fs000</code>	set font size in half points
<code>\plain</code>	plain text (as opposed to bold, italic)
<code>\b</code>	bold
<code>\i</code>	italic
<code>\ul</code>	underline
<code>\uldb</code>	double underline
<code>\uld</code>	dotted underline
<code>\strike</code>	striketru text

The scope of font commands may be document global (placed in document header), topic global (placed in topic header), paragraph global (placed at start of paragraph), or character or word global (scope defined with { . . }).

Examples:

ASCII

`\b` This paragraph will appear in bold text.`

Result

This paragraph will appear in bold text.

ASCII

This paragraph is `{\ul mostly}` in plain text with a `{\b bold}` and an `{\i italicized}` word. `{\f0 Fonts and \fs30 pitch}` can be temporarily changed.`

Result

This paragraph is mostly in plain text with a **bold** and an *italicized* word. Fonts and **pitch** can be temporarily changed.

Paragraph Commands

<code>\par</code>	end of paragraph. RTFGEN handles this.
<code>\pard</code>	reset to paragraph defaults. RTFGEN handles this.
<code>\li000</code>	left paragraph indent in <u>twips</u> . RTFGEN handles this.
<code>\fi000</code>	first line indent. May be negative.
<code>\ri000</code>	right paragraph indent.
<code>\sb000</code>	space below paragraph.
<code>\sa000</code>	space above paragraph.
<code>\ql</code>	left justified (default)
<code>\qc</code>	paragraph is centered
<code>\qr</code>	right justified
<code>\box</code>	paragraph is surrounded by a box
<code>\brdrl</code>	left border
<code>\brdrb</code>	bottom border
<code>\brdrr</code>	right border
<code>\brdrt</code>	top border

The scope of paragraph commands may be document global (placed in document header), topic global (placed in topic header), or paragraph global (placed at start of paragraph).

Examples:

ASCII

`\box\qc` This paragraph is surrounded by a box and is centered.`

Result

This paragraph is surrounded by a box and is centered.
--

ASCII

`\fi-360` The first line of this paragraph overhangs by about 3 characters. Note that RTFGEN has already indented the paragraph 720 twips because of its interpretation of the 6 character indentation on the first line.`

Result

The first line of this paragraph overhangs by about 3 characters. Note that RTFGEN has already indented the paragraph 720 twips because of its interpretation of the 6 character indentation on the first line.

Tabbing Commands

In RTFGEN, either the tab character or the RTF command, \tab, may be used to produce a tab in the help file. By default, tabs are every 720 twips (about 6 characters). If neat columns are desired with proportional fonts, tabs are a must.

\tab causes tab
\tx0000 defines tab position in twips. May be used multiply and should be used prior to using tabs.

The following should be used just prior to the \tx000 command to which they are to apply:

\tqr tab is right justified
\tqc tab is centered

The \blockstart..\blockend commands may be used to prepare the format for a table.

Example:

ASCII

```
\blockstart \li1000\tqr\tx2700\tqc\tx4000` ( * tab setup * )
  \ul Left\tab Right\tab Centered` ( * underlined table heading * )
  Data1\tab 1234\tab 12345`
  ABC \tab 12\tab 3333444`
  XYZdef\tab 45678\tab 9`
\blockend
```

Result

<u>Left</u>	<u>Right</u>	<u>Centered</u>
Data1	1234	12345
ABC	12	3333444
XYZdef	45678	9

References

The following files may be found on CompuServe in several locations. To locate them, use the IBM File Finder (GO IBMFF).

RTF.TXT	Info on the RTF syntax. Many more commands are listed than given here.
QDHELP.LZH	Another system to produce help files using an ASCII editor.

(C) Copyright 1992 by L. David Baldwin.
All Rights Reserved

RTFGEN may be copied and distributed freely providing that no fee is charged and it is not part of a package for which a charge is made.

Please report all bugs, suggestions, and problems to Dave Baldwin, CompuServe ID #76327,53.

22 Fox Den Rd., (Summer)
Hollis, NH 03049
(603) 465-7857

144 13th St. East, (Winter)
Tierra Verde, FL 33715
(813) 867-3030

