#$+K!

Manual for Tex2RTF 2.0

by Julian Smart

{bmc tex2rtf.wmf}

Contents

[C]ontents
[C]ontents
[b]rowse00001
[K] Contents
[D]isableButton("Up")

$#+K!Copyright notice

Copyright notice
topic0
browse00002
K Copyright notice
DisableButton("Up")

$#+K!**Introduction**

This document describes a utility for converting LaTeX files into several other formats.

Only a subset of LaTeX can be processed by this utility, especially since the target document language will never perfectly match LaTeX. Whether the quality of the results is good enough will depend upon the application and your own expectations. *This caveat is worth emphasizing*, because many people assume that any old LaTeX document will go through without modification: it might, but the chances are you'll need to modify it a bit for Tex2RTF. Tex2RTF was written with portable document maintenance and generation in mind, with less emphasis on accepting all LaTeX syntax. You have been warned!

Tex2RTF is heavily biased towards making on-line, hypertext versions of LaTeX documents, but the RTF converter can be used to generate linear, paper-based documents too.

The latest version of Tex2RTF, plus source code, can be accessedfrom:

```
http://web.ukonline.co.uk/julian.smart/tex2rtf
ftp://www.remstar.com/pub/wxwin/tex2rtf
```

It is available in Sun Open Look, Motif, Windows 3.1, Windows 95/NT, and non-GUI UNIX versions.

Tex2RTF was developed using the free Open Look, Motif and Windows 3.1 C++ class library wxWidgets.

---

lntroduction
topic1
browse00003
K Introduction
DisableButton("Up")

$#+KKKKKK!**Running Tex2RTF**

Tex2RTF may be run in a number of ways: with or without command line arguments, interactively or in batch mode, and with an optional initialisation file for specifying LaTeX macros and detailed options.

Tex2RTF accepts two arguments (input and output filenames) and trailing (optional) switches. If both filenames are given, the utility will work in batch mode. Otherwise, if Tex2RTF has been compiled for GUI operation, a main window will be shown, with appropriate menu items for selecting input and output filenames, starting off the conversion process, and so on.

Note that if the file `bullet.bmp` is found by Tex2RTF, this bitmap will be used as the bullet for items in `\itemize` lists, for WinHelp output. Otherwise, a symbol will be inserted (linear RTF) or bold 'o' will be used instead (all other formats).

Syntax error reporting is fairly minimal. Unrecognised macro errors may actually be produced by an unbalanced brace or passing the wrong number of arguments to a command, so look in the vicinity of the error for the real cause.

Some of the syntax that is OK for true LaTeX but which trips up Tex2RTF, may be detected by the TCHECK program included in the tools directory of the Tex2RTF distribution. Some LaTeX errors may be picked up by the LACHECK program, also found in the tools directory.

It is recommended that you run Tex2RTF twice in order to be sure of resolving all references and including an up-to-date contents page.

If importing RTF files into Word for Windows, you may need to reformat the document. The easiest way to do this is to select all text with CTRL-A, then reformat with F9. Reformat again to ensure all references are resolved. For the second format, respond with *Update Entire Table* to prompts.

Runningtex2RTF
topic5
browse00007
K Running Tex2RTF
K running Tex2RTF
K bullets
K TCHECK
K LACHECK
K Microsoft Word
DisableButton("Up")

$#+KK!**Writing documents with Tex2RTF**

Writing documents with Tex2RTF
topic12
browse00022
K Writing documents with Tex2RTF
K LaTeX
DisableButton("Up")

$#+KK!**Command reference**

The following lists commands which are recognised by the converters. The reader can assume that commands not mentioned here are unrecognised or ignored.

Each command is listed with its name, the number of arguments it takes (excluding optional arguments), and a description. Note that if the command is used as an environment (using `\begin` and `\end`) then the number of arguments must be either one or two. For example, the `\tabular` environment takes two arguments: a first argument for specifying the formatting, and the second argument for the body of the environment.

```
\begin{tabular}{|l|l|}
\row{One&Two}
\row{Three&Four}
\end{tabular}
```

Command reference
topic34
browse00048
K Command reference
K command reference
DisableButton("Up")

$#+KKKK!**Bugs and troubleshooting**

Bugs
Troubleshooting

Bugs and troubleshooting
errors
browse00254
K Bugs and troubleshooting
K bugs
K errors
K troubleshooting
DisableButton("Up")

$#+K!**References**

[1] **Boggan, Scott and Fakas, David and Welinske, Joe**. 1993. *Developing on-line help for Windows.* Sams Publishing. 11711 North College, Carmel, Indiana 46032, USA.

[2] **Smart, Julian**. 1993. *wxWindows 1.50 User Manual.* University of Edinburgh. Artificial Intelligence Applications Institute. 80 South Bridge, Edinburgh, EH1 1HN.

[3] **Kopka, Helmut and Daly, Patrick W.** 1993. *A Guide to LaTeX.* Addison-Wesley.

[4] **Pfeiffer, Katherine Shelly**. 1994. *Word for Windows Design Companion.* Ventana Press.

References
bibliography
browse00266
K References
DisableButton("Up")

$#+K!**Glossary**

GUI
HTML
LaTeX
RTF
wxHelp
wxWidgets

Glossary
topic60
browse00267
K Glossary
DisableButton("Up")

$#+KK!**Status of Tex2RTF**

Windows HTML help, and wxWidgets 2 wxHTML help, are now catered for using the htmlWorkshopFiles setting.

Tex2RTF is very rarely updated these days: it would be nice to rewrite the parser (and indeed the rest of it) at some point, to improve error reporting, space handling and ability to handle more advanced Tex/Latex commands.

Status of Tex2RTF
topic2
browse00004
K Status of Tex2RTF
K status of Tex2RTF
EnableButton("Up");ChangeButtonBinding("Up", "JumpId(`tex2rtf.hlp', `topic1')")

$#+KK!**Acknowledgements**

Thanks are due to the many people in AIAI and on the Internet at large who have pointed out bugs or shortcomings in Tex2RTF. Michel Lavaud has been a great help in giving advice for improvements to the manual.

Acknowledgements
topic3
browse00005
K Acknowledgements
K acknowledgements
EnableButton("Up");ChangeButtonBinding("Up", "JumpId(`tex2rtf.hlp', `topic1')")

$#+KK!**Change log**

Version 2.0, August 24th 1999

{bmc bullet.bmp} Added htmlWorkshopFiles setting, to output .hpp, .hhc and .hhk (HTML Workshop) files, for generating MS HTML Help or wxHTML Help.

Version 1.64, October 20th 1998

{bmc bullet.bmp} Added `\insertatlevel` command.

Version 1.63, October 21st 1997

{bmc bullet.bmp} Debugged problem with Word bookmarks not being inserted for unnumbered sections.

Version 1.62, August 18th 1997

{bmc bullet.bmp} Added contributed changes by Andreas Münzenmaier to support German accents by allowing the characters to be placed in input files, and also converting them back to character codes in the WinHelp `.cnt` file.

{bmc bullet.bmp} Now `\helpref` causes page references to be inserted in linear RTF, or section references if not on Word mode.

{bmc bullet.bmp} WinHelp table caption bug fixed.

Version 1.61, June 11th 1997

{bmc bullet.bmp} `\fcol` now works in HTML using the FONT tag.

{bmc bullet.bmp} `\twocollist` works in indented paragraphs, and is now implemented properly using tables in HTML.

{bmc bullet.bmp} New boolean option **combineSubSections** added, which switches off the generation of separate HTML files below section level. This can reduce the number of HTML files substantially.

Version 1.60, February 18th 1997

{bmc bullet.bmp} The index command now allows complex LaTeX instead of inserting the first argument verbatim.

Version 1.59, February 14th 1997

[C]hange log
[t]opic4
[b]rowse00006
[K] Change log
[K] change log
[E]nableButton("Up");ChangeButtonBinding("Up", "JumpId(`tex2rtf.hlp', `topic1')")

{bmc bullet.bmp} Added special processing for a chapter called Popups.

Version 1.58, August 1st 1996

{bmc bullet.bmp} Added HTML settings: backgroundImage, backgroundColour, textColour, linkColour, followedLinkColour.

{bmc bullet.bmp} Added `\backgroundimage`, `\backgroundcolour`, `\linkcolour`,`followedLinkColour`. `\background` now obsolete (but behaviour is backward compatible).

{bmc bullet.bmp} The default background colour is now white.

{bmc bullet.bmp} Debugged HTML `\ss` (put in wrong place in code).

Version 1.57, July 27th 1996

{bmc bullet.bmp} Added upperCaseNames setting; now all links in HTML files are in lower case unless specified otherwise.

Version 1.56, May 25th 1996

{bmc bullet.bmp} Debugged `\special` processing for HTML (escaped characters such ampersand).

{bmc bullet.bmp} Added contentsDepth for Word RTF contents page.

{bmc bullet.bmp} Removed overlapping href in HTML pages.

Version 1.55, May 6th 1996

{bmc bullet.bmp} `\verb` support corrected for HTML.

{bmc bullet.bmp} Added *abstractName* setting.

{bmc bullet.bmp} Debugged incorrect centring for HTML buttons.

Version 1.54, Feburary 28th 1996

{bmc bullet.bmp} Bug fix for 24-bit bitmap inclusion when generating RTF: caused a floating point error.

{bmc bullet.bmp} Added htmlIndex setting, to generate an `.htx` index file of an HTML document for use in wxHelp version 2 or other programs.

{bmc bullet.bmp} Fixed header/footer bug.

{bmc bullet.bmp} Change colons to spaces for WinHelp RTF keywords, since the colon has a specific meaning in WinHelp.

Version 1.53, January 1995

{bmc bullet.bmp} Now stores paths from file inclusions, so that if you include a file A from a separate directory, which then includes a file B relative to that directory, Tex2RTF will search in the path of A to find file B.

Version 1.52, December 1995

{bmc bullet.bmp} `\helpref` and related commands now generate italicized instead of bold 'anchor' text for linear formats.

{bmc bullet.bmp} Cured bug where Tex2RTF could hang on start up, while reading the `tex2rtf.ini` file. This occurred when a comment finished with the end of file.

{bmc bullet.bmp} Split the commands reference in two (LaTeX and Tex2RTF commands), and added a *Commands by category* section.

{bmc bullet.bmp} Removed a bug that caused HTML output to be garbled on the second pass.

Version 1.51: Windows 95 enhancements.

{bmc bullet.bmp} Added settings winHelpContents (for generating `.cnt` file), winHelpVersion (for specifying target version of WinHelp).

{bmc bullet.bmp} Added space to non-scrolling region of topic.

{bmc bullet.bmp} If winHelpVersion is 4, makes non-scrolling region grey and the rest yellow.

{bmc bullet.bmp} Added `\settransparency` command for WinHelp 4 transparent bitmaps.

Version 1.50:

{bmc bullet.bmp} Tidied up HTML generation (headers and bodies in the right places).

{bmc bullet.bmp} Eliminated extra space after verbatim in HTML.

{bmc bullet.bmp} Added support for simple tables in HTML.

{bmc bullet.bmp} Added `\textcolour`, `\background` for colouring text and background in HTML.

{bmc bullet.bmp} Added `\copyright`, `\registered` symbols in HTML.

{bmc bullet.bmp} Added `\imagel`, `\imager` for left and right aligned images in HTML.

{bmc bullet.bmp} Added `\brclear` for clearing image alignment in HTML.

{bmc bullet.bmp} Added LaTeX font size support in HTML (`\small`, `\large` etc.) using Netscape font extensions.

{bmc bullet.bmp} HTML button-bar change: always shows the same buttons, but may make one or more insensitive. Changing button positions could be very annoying.

{bmc bullet.bmp} Tidied up RTF generation for non-Word viewers (*useWord* set to *false*). Will now look reasonable using Windows 95 Quick View and WordPad:

WordPad doesn't do tables but does bitmaps, and QuickView does tables but not bitmaps. Such is life.

Version 1.49:

{bmc bullet.bmp} Cured some bugs (char used for fgetc instead of int) so now compiles for WIN32s.

Version 1.48:

{bmc bullet.bmp} Added some LaTeX2e fonts commands such as `\rmfamily`, `\textrm`, `\emph`. Most of these are aliases for other commands.

Up to version 1.47:

{bmc bullet.bmp} Added `\backslashraw`, `\rbraceraw` and `\lbraceraw` commands to help output arbitrary RTF.

{bmc bullet.bmp} Added `\sethotspotcolour`, `\sethotspotunderline` commands for controlling WinHelp hotspot appearance.

{bmc bullet.bmp} Added truncateFilenames option.

{bmc bullet.bmp} Improved HTML inline image handling.

Up to version 1.46:

{bmc bullet.bmp} Added `\urlref` command for specifying HTML URLs.

{bmc bullet.bmp} Started support for translating .SHG files to HTML .map files (this works if compiled under Borland, not MS VC++ for some reason!)

{bmc bullet.bmp} Fixed nasty memory bug in HTML code (thanks Petr).

Version 1.40:

{bmc bullet.bmp} Added *generateHPJ* option for generating the .HPJ WinHelp project file

{bmc bullet.bmp} Added support for DDE via a small command set

Version 1.39:

{bmc bullet.bmp} Option for using Word's INCLUDEPICTURE or IMPORT field, since the method that works for Works, doesn't work for Word! See *bitmapMethod* in the settings section.

Version 1.37-1.38:

{bmc bullet.bmp} Improved bibliography reading and cured some minor bugs

{bmc bullet.bmp} Added `\ss` German sharp s

{bmc bullet.bmp} Added rudimentary `\special` command (simply copies the argument to the output)

{bmc bullet.bmp} Added missing '.' in subsubsection reference

{bmc bullet.bmp} Added primitive internationalisation support with contentsName, tablesName etc.

Version 1.36:

{bmc bullet.bmp} All HTML special characters now correctly delimited by a semicolon.

{bmc bullet.bmp} Cured HTML section-duplicating bug I introduced in 1.35.

{bmc bullet.bmp} Cured too much spacing after sections in RTF, introduced in 1.35.

Version 1.35:

{bmc bullet.bmp} Added TCHECK tool, to help track down common Tex2RTF syntax problems.

{bmc bullet.bmp} Included Kresten Thorup's LACHECK LaTeX checking tool with DOS executable.

{bmc bullet.bmp} Now ignores `\@` command.

{bmc bullet.bmp} Table of contents now includes numbered subsubsections.

Version 1.34:

{bmc bullet.bmp} Added `\multicolumn` 'support' to stop RTF readers crashing.

{bmc bullet.bmp} Added *useWord, defaultColumnWidth, compatibility* options to `.ini` file.

{bmc bullet.bmp} `\comment` environment now doesn't complain about unknown syntax.

{bmc bullet.bmp} Added `\toocomplex` environment that treats its contents as verbatim in output, treated as normal output in true LaTeX.

{bmc bullet.bmp} End-of-line comments allowed in in `.ini` files, using semicolon, percent or hash characters to denote a comment.

{bmc bullet.bmp} For linear RTF, Word for Windows support for `\printindex`, `\index`, `\pageref`, `\listoftables`, `\listoffigures`, contents page.

{bmc bullet.bmp} Added RTF support for various symbols.

{bmc bullet.bmp} Added colour support, with `\definecolour`, `\fcol` and `\bcol` commands.

{bmc bullet.bmp} Fixed some bugs: page numbering problems, macros deleted after first pass.

Version 1.33:

{bmc bullet.bmp} Added -charset command-line switch.

{bmc bullet.bmp} Added `\itemsep`, `\twocolumn`, `\onecolumn`, `\setfooter`, `\setheader`, `\pagestyle`,`\pagenumbering`, `\thechapter`, `\thesection`, `\thepage`, `\thebibliography`, `\bibitem` commands.

{bmc bullet.bmp} New environment called `\twocollist` for making two-column lists, with formatting optimized for target file format.

{bmc bullet.bmp} New `\indented` environment for controlling indentation.

{bmc bullet.bmp} List indentation and bulleting improved.

{bmc bullet.bmp} Added commands `\normalbox`, `\normalboxd` for putting borders around text.

{bmc bullet.bmp} Many options can now be specified in the `.ini` file along with custom macros.

{bmc bullet.bmp} Cured bug that put too much vertical space after some commands.

{bmc bullet.bmp} Improved table formatting.

{bmc bullet.bmp} Optional 'Up' button in WinHelp files for easier navigation.

{bmc bullet.bmp} Verbatim lines followed by `\par` in RTF, to improve WinHelp wrapping.

{bmc bullet.bmp} Conversion may now be aborted under Windows by attempting to close the application.

{bmc bullet.bmp} Added conditional output for all formats: `\latexignore`, `\latexonly`, `\rtfignore`, `\rtfonly`,`\winhelpignore`, `\winhelponly`, `\htmlignore`, `\htmlonly`, `\xlpignore`, `\xlponly`.

{bmc bullet.bmp} HTML generator can now add Contents, Up, << and >> buttons (text or bitmap) to each page except titlepage.

Version 1.32:

{bmc bullet.bmp} `\footnote` command now supported in WinHelp RTF, and `\footnotepopup` added.

Version 1.31:

{bmc bullet.bmp} `\footnote` command now supported, in linear RTF only.

{bmc bullet.bmp} Added `-bufsize` option, for converting large documents.

Version 1.30:

{bmc bullet.bmp} `\image` command now scales metafiles (but not bitmaps).

{bmc bullet.bmp} Fixed macro loading bug, now informs the user of the found macro

filename.

{bmc bullet.bmp} Now supports paragraph and subparagraph commands.

{bmc bullet.bmp} Support for some accents added.

{bmc bullet.bmp} `\verb` command now supported.

{bmc bullet.bmp} Bug in subsubsection handling fixed.

{bmc bullet.bmp} Can save conversion log in a text file.

Version 1.22:

{bmc bullet.bmp} More informative, warns against use of some commands.

{bmc bullet.bmp} Added compile-time support for non-GUI environments (such as plain UNIX).

{bmc bullet.bmp} Improved HTML support.

$#+K!**Tex2RTF Interface**

This is the Tex2RTF interface under Windows. Click on an area of the picture for more information.

{bmc screen.shg}

Menu bar
Message area
Status line
Mode indicator

Tex2RTF Interface
topic6
browse00008
K Tex2RTF Interface
EnableButton("Up");ChangeButtonBinding("Up", "JumpId(`tex2rtf.hlp', `topic5')")

$#+KK!Command line arguments

These are the optional arguments you may give Tex2RTF on the command line.

-bufsize	Specifies buffer size in K (default 60 under Windows, 500 under UNIX). Large files (particularly large verbatim environments) may require a large buffer size, equal to the largest argument of a LaTeX command. Note that this value may not be larger than 64 under Windows.
-html	Specifies HTML (World Wide Web) output.
-interactive	Forces interactive mode even if both filenames are given.
-charset charset	Specifies a character set for RTF production. This can be one of ansi, mac, pc, and pca. The default is ansi.
-macros filename	Specifies a file for the custom macro file -- see Macro not found error.
-rtf	Specifies linear RTF output.
-sync	Forces synchronous mode (no yielding to other processes) -- usually use this in non-interactive mode.
-twice	Tells Tex2RTF to run the conversion twice to ensure all references and citations are resolved and the contents page included.
-winhelp	Specifies Windows Help RTF output.

Command line arguments
topic7
browse00013
K Command line arguments
K command line arguments
EnableButton("Up");ChangeButtonBinding("Up", "JumpId(`tex2rtf.hlp', `topic5')")

$#+KKK!**Initialisation file syntax**

The initialisation file contains further detailed options for customising Tex2RTF's behaviour. A file may be specified with the `-macros` command line switch, otherwise Tex2RTF looks for the file `tex2rtf.ini` in the working directory or input file directory.

The file may comprise macro (command) definitions or option settings.

The syntax for a macro definition is:

```
\name [number of args] {...LaTeX code...}
```

For example:

```
\crazy      [2]{{\bf #2} is crazy but #1 is not}
\something  [0]{}
\julian     [0]{Julian Smart}
```

The syntax for an option setting is:

```
name = value
```

or

```
name = "value"
```

For example:

```
conversionMode = RTF
runTwice = true
titleFontSize = 12
authorFontSize = 10
headerRule = yes
footerRule = yes
```

Options expecting boolean values accept *1, 0, true, false, yes, no* in any combination of upper or lower case.

End-of-line comments are allowed in an initialisation file, using the hash, semicolon or percent signs to denote the start of a comment, which runs until the end of the line.

Tex2RTF options

[I]nitialisation file syntax
[i]nifile
[b]rowse00014
[K] Initialisation file syntax
[K] initialisation file
[K] macros
[E]nableButton("Up");ChangeButtonBinding("Up", "JumpId(`tex2rtf.hlp', `topic5')")

$#+KKKKKKKKK!**DDE commands**

A Windows program can hold a conversation with Tex2RTF using DDE. The Tex2RTF server name is "TEX2RTF", and the topic name to use is also "TEX2RTF".

Tex2RTF functionality is accessed using the DDE *Execute* message. The *Execute* data should consist of a command name and possibly one argument, e.g.

```
INPUT c:\docs\mine.tex
```

If the command is not recognised, a standard TEX2RTF.INI option is assumed.

The *Request* DDE message can be used to query the return status of an *Execute*command, and will be one of *OK* (no error), *CONVERSION ERROR*, or a more specific error string.

The following DDE commands may be used:

Command	Description
EXIT	Takes no argument, and exits Tex2RTF.
GO	Takes no argument, and initiates the conversion.
INPUT	Takes a file name as the argument, and sets the input file to be this name.
MINIMIZE	Takes no argument, and minimizes Tex2RTF.
OUTPUT	Takes a file name as the argument, and sets the input file to be this name.
RESTORE	The same as SHOW.
SHOW	Takes no argument, and unminimizes Tex2RTF.

[D]DE commands
[t]opic10
[b]rowse00020
[K] DDE commands
[K] DDE
[K] EXIT
[K] GO
[K] INPUT
[K] MINIMIZE
[K] OUTPUT
[K] RESTORE
[K] SHOW
[E]nableButton("Up");ChangeButtonBinding("Up", "JumpId(`tex2rtf.hlp', `topic5')")

$#+KK!**Performance issues**

Since Tex2RTF reads the whole file into memory, a lot of memory is needed. For very large documents, 16MB of RAM is adviseable.

I tested conversion of the wxWidgets 1.63 manual on both VC++ 1.5 and Watcom WIN32s versions of Tex2RTF, both running under Windows 3.11 on a Gateway P60 with 16MB of RAM and a 2MB disk cache. Two passes were made, with 1.5MB of WinHelp RTF being generated. The unoptimized 16-bit version took 169 seconds. The optimized WIN32s version took 126 seconds, a significant improvement. Systems with faster disk subsystems should see an even better relative performance of the 32-bit version.

Performance issues
topic11
browse00021
K Performance issues
K performance
EnableButton("Up");ChangeButtonBinding("Up", "JumpId(`tex2rtf.hlp', `topic5')")

$#+K!**Why use LaTeX?**

LaTeX happens to be a very convenient format if you need to produce documents (such as manuals, help facilities, up-to-date information) in both printed and on-line media. Being a language rather than a WYSIWYG system, it allows explicit specification of layout and document structure, lending itself well to hypertext applications and automatic document generation. Many people also prefer to use LaTeX for ordinary use since it encourages a logical document structure and the user is not distracted by having to perfect the appearance; many layout decisions are taken by LaTeX automatically.

Although LaTeX is not as fancy as modern word processors and desk-top publishing packages, it is for many purposes quite adequate, and sometimes more flexible than its modern counterparts.

The conversion utility gives LaTeX a new lease of life by allowing virtually all other wordprocessor formats to be generated from documents containing a reasonable subset of LaTeX syntax. From the same LaTeX sources, we can now generate printed manuals, Windows Help files, wxHelp files, RTF-compatible word processor formats such as MS Word, and HTML files for use in the World Wide Web. Since the conversion tool is free, as are LaTeX, HTML viewers, wxHelp and (effectively) Windows Help, there are no financial or time penalties for providing documentation in a wide range of printed and hypertext formats.

---

why use LaTeX?
topic13
browse00023
K Why use LaTeX?
EnableButton("Up");ChangeButtonBinding("Up", "JumpId(`tex2rtf.hlp', `topic12')")

$#+KK!**Help versus the printed page**

The purist may argue, quite rightly, that on-line help systems and printed manuals have different characteristics; help windows tend to be much smaller than pages, help topics should be more stand-alone than pages in a manual, navigation methods are very different, etc. Therefore, help systems should be *based* on printed documentation but separately hand-crafted into hypertext help, preferably by an independent person or team.

This might be the ideal, but many organisations or individuals simply do not have the time: on-line help wouldn't get done if the documentation effort had to be doubled. However, Tex2RTF does provide some commands to allow tailoring the documentation to printed or on-line form, such as `\helponly` and `\helpignore`. An awareness of the design issues should go a long way to making the compromise a good one, so a book such as *Developing On-line Help for Windows* [1] is highly recommended.

Help versus the printed page
topic14
browse00024
K Help versus the printed page
K on-line help
EnableButton("Up");ChangeButtonBinding("Up", "JumpId(`tex2rtf.hlp', `topic12')")

$#+KKKKK!**Output Formats**

At present the following output formats are supported:

{bmc bullet.bmp} RTF (Rich Text Format). This is the most well developed converter. RTF is commonly used as a document exchange format amongst Windows-based applications, and is the input for the Windows Help Compiler. Tex2RTF supports both linear documents and Windows Help hypertext format.

{bmc bullet.bmp} HTML (Hypertext Markup Language). This an SGML-based format commonly used by documents in the World Wide Web distributed hypertext system, and formats text dynamically rather like Windows Help.

{bmc bullet.bmp} wxHelp. This is the platform-independent help system for the class library wxWidgets (see the wxWidgets User Manual [2]). It can display ASCII files with embedded codes for changing font styles, but no formatting is done by wxHelp.

Output Formats
topic15
browse00025
K Output Formats
K output formats
K RTF
K HTML
K wxHelp
EnableButton("Up");ChangeButtonBinding("Up", "JumpId(`tex2rtf.hlp', `topic12')")

$#+KKKKKKKKKKK!**What compromises must I make?**

As a LaTeX user, you need to be aware that some commands or facilities don't transfer to other formats, either because they are not supported by the target format or because the converter does not support them. Maths formatting is a good example of an unsupported feature.

Sometimes LaTeX facilities must be accessed in a slightly different way to support the variety of formats, particularly hypertext formats where LaTeX references are often replaced by hypertext jumps (but must still look right in printed documentation). Tables don't transfer well to RTF and HTML (and not at all to wxHelp) but an attempt is made to approximate tables so long as special row commands are used, instead of the usual end of row delimiter.

Bibliographies are handled quite well since the utilities can read in `.bib` files and resolve citations. Numbers are used in citations; the references are not yet sorted alphabetically.

Pictures are handled in a limited way: if the PSBOX macro package is used, an `\image` command can be used to place Encapsulated PostScript files in LaTeX, and Windows RGB-encoded bitmap files or placeable metafiles when converting to RTF.

Nested file inclusion is handled with `\input`, `\include` and `\verbatiminput`, and the comment environment is supported. However, using `\input` to include macro packages is not advisable. If you do this, make sure you add a line in the Tex2RTF initialisation file to ignore this file, unless it's a simple LaTeX file that conforms to Tex2RTF restrictions. The file `psbox.tex` is the only file ignored by Tex2RTF by default.

Because of the way LaTeX is parsed, some syntax has to conform to a few simple rules. Commands such as `\bf` and `\it` need to occur immediately after a left brace, and have a block of their own, since the text within their scope is regarded as its argument. This syntax means the same thing as using `\begin ... \end`, which is usually a one argument

---

[W]hat compromises must I make?
[t]opic16
[b]rowse00026
[K] What compromises must I make?
[K] compromises
[K] LaTeX
[K] pictures
[K] PSBOX
[K] file inclusion
[K] syntax restrictions
[K] on-line help
[K] label
[K] helpref
[K] helprefn
[K] popref
[E]nableButton("Up");ChangeButtonBinding("Up", "JumpId(`tex2rtf.hlp', `topic12')")

command (the argument is the text between the `\begin` and `\end`). See Space.

As a Windows hypertext help writer, you don't have access to all RTF commands but you'll be able to get most of what you want. In particular, any LaTeX document you write will automatically be a hypertext document, because the converter takes advantage of the hierarchy of sections. Further jumps can be placed using the commands `\label`, `\helpref`, `\helprefn`, and `\popref`. Tex2RTF outputs help files that may be read linearly using the << and >> buttons, with an additional Up button for ease of navigation.

When writing HTML, multiple files are generated from one LaTeX file since browsing HTML works best with many small files rather than a few large ones.

wxHelp files are least well supported since there is no formatting support, only font style, sizes and colours. Still, some hypertext help support on UNIX/X platforms is better than none. wxHelp is now being rewritten (March 1996) to use HTML files.

Sometimes you will use a local macro package that is unrecognised by the converters. In this case, you may define a custom macro file where macros are defined in terms of supported LaTeX commands and text. Even if the result is not the same as in LaTeX, you can probably end up with something adequate, and at least avoid undefined macro errors. See Initialisation file syntax for further information.

$#+K!**Changes to LaTeX syntax**

Here are the conventions you need to observe to satisfy the Tex2RTF parser.

Changes to LaTeX syntax
topic17
browse00027
K Changes to LaTeX syntax
EnableButton("Up");ChangeButtonBinding("Up", "JumpId(`tex2rtf.hlp', `topic12')")

$#+KK!Tex2RTF for non-LaTeX users

You don't need to have LaTeX installed to use Tex2RTF. You can still output RTF files to be imported into your favourite word processor, and hypertext files for on-line help.

This chapter gives a very brief introduction to LaTeX. For further information, Kopka and Daly's *A Guide to LaTeX* [3] is recommended.

[T]ex2RTF for non-LaTeX users
[t]opic22
[b]rowse00034
[K] Tex2RTF for non-LaTeX users
[K] LaTeX
[E]nableButton("Up");ChangeButtonBinding("Up", "JumpId(`tex2rtf.hlp', `topic12')")

$#+KKKK!**Hypertext features**

LaTeX is inherently suitable for specifying hypertext documents since it encourages description of the logical structure of a document using section commands. Therefore, a LaTeX document is automatically a hypertext document, without any further editing.

For Windows Help, a single RTF file is generated with topics corresponding to sections. A top level contents page shows each chapter or top-level section, and each chapter or section ends with a list of further sections or subsections. Tex2RTF outputs help files that may be read linearly using the << and >> buttons.

Similarly, a single wxHelp XLP file is generated.

For HTML, a different file is generated for each section, since the XMOSAIC browser works best with a large number of small files. The files are named automatically based on the name of the output file, with the contents page filename being formed from the output filename with `_contents` appended to the name. If the truncateFilenames option is begin used, then the contents page is just the root name, with a .htm suffix. The conversion may result in the generation of several hundred files for a large LaTeX input file.

To specify explicit jumps around a hypertext file, the `\helpref` command is used. The first argument is the text to be displayed at the point of reference, which will be highlighted in a hypertext file to allow jumping to a reference. The second argument is the reference label (there should be a corresponding `\label` command in the file, following a section or figure).

To use extra Tex2RTF features in proper LaTeX, such as `\helpref` and the C++ and CLIPS class reference documentation features, include the style file `texhelp.sty`.

[H]ypertext features
[t]opic27
[b]rowse00039
[K] Hypertext features
[K] hypertext
[K] helpref
[K] label
[E]nableButton("Up");ChangeButtonBinding("Up", "JumpId(`tex2rtf.hlp', `topic12')")

$#+KK!**Special sections**

The treatment of bibliography, glossary and index are worth special mention.

Bibliography
Glossary
Index

Special sections
topic28
browse00040
K Special sections
K special sections
EnableButton("Up");ChangeButtonBinding("Up", "JumpId(`tex2rtf.hlp', `topic12')")

$#+K!**Authoring HTML documents**

When an HTML document is generated, the suffix '_contents' is appended to the input file root. This will be the contents page for the document. A number of further HTML files will be generated, possibly a large number for a document with a large number of sections. If you are running a 16-bit Windows version of Tex2RTF, you may wish to use the *truncateFilenames* option to generate DOS filenames with appropriately truncated references inside the HTML files.

Tip: to reduce the number of sections generated and make the document more linear, you could define new chapter and section commands. Alias them to the normal commands in real LaTeX (edit `texhelp.sty`), and to appropriate bold/large headings (but not section commands) in the Tex2RTF initialisation file.

Each HTML section file (except for the contents page) is given browse buttons, similar to a Windows Help file: Contents, Up, Down, Back, Forward. You can set *htmlBrowseButtons* to specify whether bitmaps or text should be used for these buttons. On a text-only browser, the buttons will show as text even if images have been specified.

As well as the usual jumps within a document, you can use the `\urlref` command to jump to other documents. 'Advanced features' which are implemented for HTML include:

{bmc bullet.bmp} Simple tables: `\tabular` command

{bmc bullet.bmp} Background colour/bitmap: `\backgroundcolour` and `\backgroundimage`

{bmc bullet.bmp} Text colour: `\textcolour` command

See HTML options for relevant initialisation file switches.

[A]uthoring HTML documents
[t]opic30
[b]rowse00044
[K] Authoring HTML documents
[E]nableButton("Up");ChangeButtonBinding("Up", "JumpId(`tex2rtf.hlp', `topic12')")

$#+KKKK!**Authoring Windows Help documents**

To produce a Windows Help file, you need to generate a WinHelp RTF file with Tex2RTF and then invoke a Windows Help compiler (such as hc505.exe) to translate this to a .hlp file.

WinHelp support has split into two streams, Windows 3.1 help format and Windows 95 (WinHelp 4) format. You control this with the *winHelpVersion* option, setting it to 3 for Windows 3.1, and 4 for Windows 95. In the latter case, you also need the Help Compiler for Windows (hcw.exe and associated components) which are available in the WIN32 SDK and with Windows 95 compilers.

Tex2RTF can produce a Windows 95 `.cnt` file if *winHelpContents* is switched on. This file is used to generate the new-style contents page, allowing hierarchical browsing of the topic contents. In fact this file can be used with ordinary Windows 3.1 files on Windows 95: so to hedge your bets, generate a Windows 3.1 help file along with `.cnt` file.

Tex2RTF also generates (optionally) a `.hpj` (Help Project) file which is fed to the help compiler and specifies the RTF file being used amongst other things. In WinHelp 4 mode, Tex2RTF adds entries to the project to enhance the appearance of the help file. In particular, the non-scrolling (topic title) region is coloured grey, and the rest is coloured a light yellow in keeping with other Windows 95 help files.

Tip: you can maintain two versions of a help file by specifying an alternative `.ini` file on the command line when invoking Tex2RTF, and compiling to a different directory. Tex2RTF instructs the help compiler to use the input file directory to find bitmaps and metafiles, so using a different output directory is not a problem.

There is a slight wrinkle with generation of the `.cnt` file: to work around a 'feature' in the Windows 95 help compiler, Tex2RTF may insert extra book icons in the contents page. So your contents page may not exactly match the structure in your LaTeX file.

'Advanced features' which are implemented for WinHelp include:

{bmc bullet.bmp} Transparency: `\settransparency` command

{bmc bullet.bmp} Colour: `\definecolour`, `\fcol`, `\bcol` commands

{bmc bullet.bmp} Hot spot appearance: `\sethotspotcolour`, `\sethotspotunderline` commands

[A]uthoring Windows Help documents
[t]opic31
[b]rowse00045
[K] Authoring Windows Help documents
[K] WinHelp files
[K] CNT file
[K] HPJ file
[E]nableButton("Up");ChangeButtonBinding("Up", "JumpId(`tex2rtf.hlp', `topic12')")

Tex2RTF automatically generates browse buttons for jumping to the above, previous and next topics.

See RTF/WinHelp options for relevant initialisation file switches.

$#+KK!**Authoring linear RTF documents**

Linear RTF documents come in two main flavours. It can produce simple RTF that can be read by a wide variety of readers, such as Windows 95 WordPad, the Windows 95 viewer, and most word processors. Tex2RTF can also output MS Word compatible RTF which has special fields for contents page and index formatting, headings, and other enhancements.

Use the *useWord* initialisation file flag to switch Word mode on or off. Hypertext links (using `\helpref` and other commands) will be formatted as bold 'anchor' text plus a section or figure number in parentheses.

In Word mode, using an index section generates a proper Word index. Similarly, a Word table of contents, list of figures, list of tables and page reference may be generated.

See RTF/WinHelp options for relevant initialisation file switches.

Authoring linear RTF documents
topic32
browse00046
K Authoring linear RTF documents
K RTF
EnableButton("Up");ChangeButtonBinding("Up", "JumpId(`tex2rtf.hlp', `topic12')")

$#+KK!**Authoring wxHelp documents**

The wxHelp (.xlp) file is the most basic kind of file that Tex2RTF can handle. Since spacing is passed through to the output, you need to format your input document appropriately, with lines of reasonable length.

The generated xlp file is an ASCII file that can be read directly by wxHelp, the generic wxWidgets help viewer.

Authoring wxHelp documents
topic33
browse00047
K Authoring wxHelp documents
K wxHelp
EnableButton("Up");ChangeButtonBinding("Up", "JumpId(`tex2rtf.hlp', `topic12')")

$#+K!**LaTeX Commands**

LaTeX Commands
topic35
browse00049
K LaTeX Commands
EnableButton("Up");ChangeButtonBinding("Up", "JumpId(`tex2rtf.hlp', `topic34')")

itshape:1
label:1
large:1
Large:1
LARGE:1
LaTeX:0
ldots:0
maketitle:0
marginparwidth:1
marginpar:1
marginpareven:1
marginparodd:1
mdseries:1
multicolumn:3
newcommand:3
newpage:0
nocite:1
noindent:0
normalsize:1
onecolumn:0
pageref:1
pagestyle:1
pagenumbering:1
paragraph:0
paragraph*:0
parindent:1
parskip:1
par:0
printindex:0
quote:1
quotation:1
ref:1
rm:1
rmfamily:1
sc:1
scshape:1
section:1
section*:1
sf:1
sffamily:1
shortcite:1
sl:1
slshape:1
small:1
special:1
ss:0
subparagraph:1
subparagraph*:1
subsection:1
subsection*:1
subsubsection:1

subsubsection*:1
tabbing:1
table:1
tableofcontents:0
tabular:2
TeX:0
textbf:1
textit:1
textrm:1
textsf:1
textsc:1
textsl:1
texttt:1
textwidth:1
thebibliography:1
title:1
tiny:1
today:0
tt:1
ttfamily:1
typeout:1
twocolumn:0
underline:1
upshape:1
verbatiminput:1
verbatim:1
verb

$#+K!**Tex2RTF Commands**

Tex2RTF Commands
topic36
browse00169
K Tex2RTF Commands
EnableButton("Up");ChangeButtonBinding("Up", "JumpId(`tex2rtf.hlp', `topic34')")

normalboxd:1
param:1
popref:2
psboxto:2
rbraceraw:0
registered:0
row:1
ruledrow:1
rtfignore:1
rtfonly:1
rtfsp:0
sectionheading:1
setfooter:6
setheader:6
sethotspotcolour:1
sethotspotunderline:1
settransparency:1
textcolour:1
toocomplex:1
twocolitem:2
twocolitemruled:2
twocollist:1
twocolwidtha:1
twocolwidthb:1
urlref:2
winhelpignore:1
winhelponly:1
xlpignore:1
xlponly:1

$#+K!**Accents**

The following LaTeX accents work for RTF and HTML production:

{bmc bullet.bmp} `\'{a}` produces á. Valid for a, e, i, o, u, A, E, I, O, U

{bmc bullet.bmp} `\`{a}` produces à. Valid for a, e, i, o, u, y, A, E, I, O, U, Y

{bmc bullet.bmp} `\^{a}` produces â. Valid for a, e, i, o, u, A, E, I, O, U

{bmc bullet.bmp} `\~{a}` produces ã. Valid for a, n, o, A, N, O

{bmc bullet.bmp} `\"{a}` produces ä. Valid for a, e, i, o, u, y, A, E, I, O, U, Y

{bmc bullet.bmp} `\.{a}` produces å. Valid for a, A

Accents
accents
browse00240
K Accents
EnableButton("Up");ChangeButtonBinding("Up", "JumpId(`tex2rtf.hlp', `topic34')")

$#+KK!**Commands by category**

Below are categories of LaTeX commands, to help you find the right command for a particular purpose.

Commands by category
topic37
browse00241
K Commands by category
K commands
EnableButton("Up");ChangeButtonBinding("Up", "JumpId(`tex2rtf.hlp', `topic34')")

$#+K!**Bugs**

Command parsing. If a command is used followed by inappropriate argument syntax, Tex2RTF can crash. This an occur when a command is used in an asterisk form that is only formed in the non-asterisk variety. The non-asterisk form is assumed, which makes the following asterisk trip up the parser.

Setlength. Using the \setlength command doesn't work, since its first argument looks like a command with the wrong number of arguments. Use an alternative form instead, e.g. `\parindent 0pt` instead of `\setlength{parindent}{0pt}`.

Newcommand bug. Environments in a command definition confuse Tex2RTF. Use the command form instead (e.g. \flushleft{...} instead of \begin{flushleft} ... \end{flushleft}.

Bibliography. There's no flexibility in the way references are output: I expect I'll get round to doing something better, but only if people tell me they need it!

Tables. Tables can't handle all LaTeX syntax, and require the Tex2RTF `\row` commands for decent formatting. Still, it's better than it was (RTF only).

Indexes and glossaries. Not completely supported.

Crashes. Crashes may be due to an input file exceeding the fixed-size buffer used for converting command arguments, especially for the `\verbatim` command. Use the `-bufsize` switch to increase the buffer size.

Verbatiminput. Verbatiminput files which do not end with a blank line can trip up following commands.

[B]ugs
[t]opic50
[b]rowse00255
[K] Bugs
[E]nableButton("Up");ChangeButtonBinding("Up", "JumpId(`tex2rtf.hlp', `errors')")

$#+K!**Troubleshooting**

Below are some common problems and possible solutions.

Some of the syntax that is OK for true LaTeX but which trips up Tex2RTF, may be detected by the TCHECK program included in the tools directory of the Tex2RTF distribution. Some LaTeX errors may be picked up by the LACHECK program, also found in the tools directory.

Troubleshooting
topic51
browse00256
K Troubleshooting
EnableButton("Up");ChangeButtonBinding("Up", "JumpId(`tex2rtf.hlp', `errors')")

$#+K!**GUI**

Graphical User Interface, such as Windows 3 or X.

GUI
topic61
browse00268
K GUI
EnableButton("Up");ChangeButtonBinding("Up", "JumpId(`tex2rtf.hlp', `topic60')")

$#+K!**HTML**

Hypertext Markup Language; an SGML document type, used for providing hypertext information on the World Wide Web, a distributed hypertext system on the Internet.

HTML
html
browse00269
K HTML
EnableButton("Up");ChangeButtonBinding("Up", "JumpId(`tex2rtf.hlp', `topic60')")

$#+K!**LaTeX**

A typesetting language implemented as a set of TeX macros. It is distinguished for allowing specification of the document structure, while taking care of most layout concerns. It represents the opposite end of the spectrum from WYSIWYG word processors.

LaTeX
latexgloss
browse00270
K LaTeX
EnableButton("Up");ChangeButtonBinding("Up", "JumpId(`tex2rtf.hlp', `topic60')")

$#+K!**RTF**

Rich Text Format: an interchange format for word processor files, used for importing and exporting formatted documents, and as the input to the Windows Help compiler.

RTF
rtf
browse00271
K RTF
EnableButton("Up");ChangeButtonBinding("Up", "JumpId(`tex2rtf.hlp', `topic60')")

$#+K!**wxHelp**

wxHelp is the hypertext help facility used to provide on-line documentation for UNIX-based wxWidgets applications. Under Windows 3.1, Windows Help is used instead.

---

wxHelp
wxhelp
browse00272
K wxHelp
EnableButton("Up");ChangeButtonBinding("Up", "JumpId(`tex2rtf.hlp', `topic60')")

$#+K!**wxWidgets**

wxWidgets is a free C++ toolkit for writing applications that are portable across several platforms. Currently these are Motif, Open Look, Windows 3.1 and Windows NT. Tex2RTF is written using wxWidgets.

---

wxWidgets
wxwidgets
browse00273
K wxWidgets
EnableButton("Up");ChangeButtonBinding("Up", "JumpId(`tex2rtf.hlp', `topic60')")

$#+K!**Menu bar**

Use the menubar for interactive operations.

Menu bar
menubar
browse00009
K Menu bar
EnableButton("Up");ChangeButtonBinding("Up", "JumpId(`tex2rtf.hlp', `topic6')")

$#+K!**Message area**

Tex2RTF writes warning and error messages on this window.

Message area
messagearea
browse00010
K Message area
EnableButton("Up");ChangeButtonBinding("Up", "JumpId(`tex2rtf.hlp', `topic6')")

$#+K!**Status line**

Displays help on menu items as the user drags the cursor over the menus.

Status line
statusline
browse00011
K Status line
EnableButton("Up");ChangeButtonBinding("Up", "JumpId(`tex2rtf.hlp', `topic6')")

$#+K!**Mode indicator**

Displays the output mode Tex2RTF is currently in.

Mode indicator
modeindicator
browse00012
K Mode indicator
EnableButton("Up");ChangeButtonBinding("Up", "JumpId(`tex2rtf.hlp', `topic6')")

$#+KKKKK!**Tex2RTF options**

These are the allowable options in an initialisation file.

Tex2RTF options
topic8
browse00015
K Tex2RTF options
K options in initialisation file
K tex2rtf.ini
K initialisation file
K macros
EnableButton("Up");ChangeButtonBinding("Up", "JumpId(`tex2rtf.hlp', `inifile')")

$#+KKKK!**Space**

Tex2RTF attempts to insert spaces where LaTeX assumes whitespace. However, for the benefit of RTF conversion, you need to use the `\rtfsp` command where a command or brace within a paragraph begins or ends with a macro. For example:

```
Within a paragraph, you need to be careful about commands
\rtfsp{\it that begin at the start of a line.}
```

As normal with LaTeX, two newlines represents a paragraph break, although `\par` can also be used at the end of a paragraph.

You need to have a blank line between section and some environment commands and the first paragraph or your document will look rather weird, e.g. headings running into paragraphs.

wxHelp is more fussy than LaTeX or RTF: you need to use percent characters at line ends liberally to eliminate newlines after commands on single lines.

Space
space
browse00028
K Space
K space
K rtfsp
K par
EnableButton("Up");ChangeButtonBinding("Up", "JumpId(`tex2rtf.hlp', `topic17')")

$#+KK!**Command arguments**

Commands that have one or more arguments can be used in the following three ways:

```
\bf{Some text.}

\begin{bf}
Some text.
\end{bf}

{\bf Some text.}
```

The first method is a normal LaTeX command.

The second method is called an *environment*; LaTeX has specific environments that do not always correspond to normal commands, but Tex2RTF recognizes environments and normal commands interchangeably, so long as the command has no more than two arguments.

With the third method, it is important that the command has its own pair of braces, and that the command immediately follows the first brace. Otherwise, the parser cannot parse the argument(s) properly. With multiple arguments, each should be enclosed in braces.

Optional arguments are specified using square brackets or parentheses.

The braces that start command arguments must not be separated from the other arguments by whitespace. For example, the following produces an error:

```
\image{5cm;0cm}
{picture.eps}
```

and should be replaced by

```
\image{5cm;0cm}{picture.eps}
```

Command arguments
topic18
browse00029
K Command arguments
K LaTeX commands
EnableButton("Up");ChangeButtonBinding("Up", "JumpId(`tex2rtf.hlp', `topic17')")

$#+K!**Avoid the setlength command**

Using the \setlength command doesn't work, since its first argument looks like a command with the wrong number of arguments. Use an alternative form instead, e.g.

```
\parindent 0pt
```

instead of

```
\setlength{\parindent}{0pt}
```

Avoid the setlength command
topic19
browse00030
K Avoid the setlength command
EnableButton("Up");ChangeButtonBinding("Up", "JumpId(`tex2rtf.hlp', `topic17')")

$#+KK!**Units**

Only a subset of LaTeX units may be used for specifying dimensions. Valid units are `pt`, `mm`, `cm` and `in`. Units should usually be specified for dimensions or the results may be unexpected.

Units
topic20
browse00031
K Units
K units
EnableButton("Up");ChangeButtonBinding("Up", "JumpId(`tex2rtf.hlp', `topic17')")

$#+KK!**Labels**

The `\label` command may be used for sections and figure captions, but must come immediately after the section or caption commands with no intervening whitespace.

Labels
topic21
browse00032
K Labels
K labels
EnableButton("Up");ChangeButtonBinding("Up", "JumpId(`tex2rtf.hlp', `topic17')")

$#+KK!**Tables**

For best layout, table rows should be enclosed in a `\row` or `\ruledrow` command, since Tex2RTF can't cope with parsing the LaTeX tabular syntax unaided. However, if you really don't want to go through LaTeX files inserting new syntax, set the *compatibility* flag to TRUE in your `tex2rtf.ini` file. In this mode, Tex2RTF tries to make the best of a bad job, but the results won't be optimal (e.g., no table borders). Without this flag set, normal LaTeX tables can crash RTF readers such as Word for Windows.

Tables
tables
browse00033
K Tables
K tables
EnableButton("Up");ChangeButtonBinding("Up", "JumpId(`tex2rtf.hlp', `topic17')")

$#+K!**What is LaTeX?**

LaTeX is a macro package built on top of the typesetting package,TeX. TeX was written by Donald Knuth in the 1970s, and Leslie Lamport wrote LaTeX as a higher-level, easier way to write TeX.

TeX was quite advanced for its day, and is still used (particularly by academics) because of its free availability and its flexibility in typesetting maths and other symbols. It's more like a programming language than a word processor, with embedded commands prefixed by a backslash and block structure. Like programs, TeX documents are processed by a 'compiler', outputting a .dvi file, which is a device independent file which can be read by many converters for output onto physical devices, such as screens and printers.

A reason for its longevity is the ability to add facilities toTeX, using macro packages that define new commands.

LaTeX is the most popular way to write TeX. Although WYSIWYG word processors and DTP packages are outstripping LaTeX, the increasing interest in hypertext and mark-up languages makes LaTeX relevant as a similar language to SGML documents (such as World Wide Web HTML files).

Also, languages such as LaTeX (and Rich Text Format, which it resembles in many ways) are *complementary* to WYSIWYG packages. These languages allow automatic production and translation of documents, where manual mark-up is impractical or undesirable.

Since the source code of TeX and LaTeX is in the public domain, there are many free and commercial implementations of LaTeX for almost every computer in existance. Of PC implementations, EmTeX is arguably the best and most complete. You can download it from various FTP sites.

If you don't want to use LaTeX itself, you may wish to use a program called lacheck to check your documents before using Tex2RTF, since it catches some mistakes that Tex2RTF doesn't.

[W]hat is LaTeX?
[t]opic23
[b]rowse00035
[K] What is LaTeX?
[E]nableButton("Up");ChangeButtonBinding("Up", "JumpId(`tex2rtf.hlp', `topic22')")

$#+K! Document structure

Here is a sample of a typical LaTeX document:

```
\documentstyle[a4,texhelp]{report}
\title{A title}
\author{Julian Smart}
\date{October 1993}
\begin{document}
\maketitle

\chapter{Introduction}

...

\section{A section}

...

\end{document}
```

The first line is always a `\documentstyle` command. The square brackets enclose optional *style* files (suffix `.sty`) that alter the appearance of the document or provide new commands, and the curly brackets enclose the mandatory style, in this case 'report'.

Before the document begins properly with `\begin{document}`, you can write various commands that have an effect on the appearance of the document or define title page information. The `\maketitle` command writes the title page using information defined previously (title, author, date).

A report has chapters, which are divided into sections, and can be further divided into subsections and subsubsections. To start a new section, you write the appropriate section command with the section heading; there is no specific end section command, since a new section heading or the end of the document will indicate the end of the previous section.

An article is divided into sections, subsections and subsubsections, but has no chapters. This is so an article can be included in a report as a chapter.

Tex2RTF is written to deal with reports best, so stick with the report style if you can.

D ocument structure
t opic24
b rowse00036
K Document structure
E nableButton("Up");ChangeButtonBinding("Up", "JumpId(`tex2rtf.hlp', `topic22')")

$#+K!Command syntax

There are several kinds of commands in LaTeX. Most involve a keyword prefixed with a backslash. Here are some examples:

```
\titlepage

\centerline{This is a centred line}

\begin{center}
This is a centred
paragraph
\end{center}

{\bf This is bold font}
```

The first example has no arguments. The second has one argument. The third example is an *environment* which uses the begin and end keywords instead of a pair of braces to enclose an argument (usually one). The fourth is an example of using a command within a pair of braces: the command applies to the scope within the braces. Tex2RTF treats this form as if it were a command with one argument, with the right brace delimiting the argument. In this case, the command must immediately follow a left brace as shown.

Commands may be nested, but not overlapped.

Command syntax
topic25
browse00037
K Command syntax
EnableButton("Up");ChangeButtonBinding("Up", "JumpId(`tex2rtf.hlp', `topic22')")

$#+KK!**Space**

In LaTeX, white space is mostly ignored, line breaks make no difference. However, LaTeX interprets two successive newlines (a blank line) as denoting a paragraph break. You may also use the `\par` command to end a paragraph.

Space
topic26
browse00038
K Space
K space
EnableButton("Up");ChangeButtonBinding("Up", "JumpId(`tex2rtf.hlp', `topic22')")

$#+KK!**Bibliography**

Tex2RTF recognises standard LaTeX bibliography files (usually with `.bib` extension) and resolves citations. The `\bibliography` command reads the given `.bib` file and includes a list of references at that point in the input. Only numbered, unsorted references are catered for at the moment, with no variation in bibliography style. A **References** heading is placed in the contents section. Note that Tex2RTF must be run twice to ensure the citations are resolved properly.

Tex2RTF can also cope with the `\thebibliography` environment, with `\bibitem` commands, so long as the text following the first `\bibitem` argument is enclosed in braces as if it were a second argument.

Bibliography
bibsection
browse00041
K Bibliography
K bibliography
EnableButton("Up");ChangeButtonBinding("Up", "JumpId(`tex2rtf.hlp', `topic28')")

$#+KKKKK!**Glossary**

Glossaries are formatted according to the following scheme. The `\helpglossary` environment is used together with the `\gloss` command for glossary entries. In LaTeX this is interpreted as a description list, and each glossary entry is an item. In on-line help, each glossary entry is a section.

A labelled glossary entry command may be referenced by `\popref` to provide a quick popup explanation of a term.

Glossary
glossarysection
browse00042
K Glossary
K glossary
K helpglossary
K gloss
K popref
EnableButton("Up");ChangeButtonBinding("Up", "JumpId(`tex2rtf.hlp', `topic28')")

$#+KK!**Index**

The explicit index is assumed to be redundant in on-line help, since search facilities are provided. Therefore the `\printindex` command does nothing in on-line versions. In linear RTF an index field is added, and `\index` marks words for inserting in the index.

In Windows Help, all section headings and C++ function names are treated as keywords. A keyword may be ambiguous, that is, refer to more than one section in the help file. This automatic indexing may not always be adequate, so the LaTeX `\index` command may be used to add keywords.

In wxHelp, all section headings are indexed.

---

lndex
topic29
browse00043
K Index
K index
EnableButton("Up");ChangeButtonBinding("Up", "JumpId(`tex2rtf.hlp', `topic28')")

$#+K!**abstract:1**

This standard LaTeX environment prepares an abstract page, and is treated as an ordinary chapter or section in on-line help.

---

[a]bstract:1
[a]bstract
[b]rowse00050
[K] abstract 1
[E]nableButton("Up");ChangeButtonBinding("Up", "JumpId(`tex2rtf.hlp', `topic35')")

$#+K!**addcontentsline:3**

Adds a chapter title to the contents page. Linear RTF. Rarely required.

---

[a]ddcontentsline:3
[a]ddcontentsline
[b]rowse00051
[K] addcontentsline 3
[E]nableButton("Up");ChangeButtonBinding("Up", "JumpId(`tex2rtf.hlp', `topic35')")

$#+K!**author:1**

Defines the author, for output when `\maketitle` is used.

---

[a]uthor:1
[a]uthor
[b]rowse00052
[K] author 1
[E]nableButton("Up");ChangeButtonBinding("Up", "JumpId(`tex2rtf.hlp', `topic35')")

$#+K!**backslash:0**

Outputs a backslash in math mode (should be enclosed by two dollar symbols).

---

backslash:0
backslash
browse00053
K backslash 0
EnableButton("Up");ChangeButtonBinding("Up", "JumpId(`tex2rtf.hlp', `topic35')")

$#+K!**bf:1**

Specifies bold font.

---

bf:1
bf
browse00054
K bf 1
EnableButton("Up");ChangeButtonBinding("Up", "JumpId(`tex2rtf.hlp', `topic35')")

$#+K!**bffamily:1**

Specifies bold font.

---

bffamily:1
bffamily
browse00055
K bffamily 1
EnableButton("Up");ChangeButtonBinding("Up", "JumpId(`tex2rtf.hlp', `topic35')")

$#+KKK!**bibitem:2**

For parsing convenience, `\bibitem` requires two arguments: a cite key and item. LaTeX syntax permits writing this as if it were two arguments, even though it is in fact only one. This command is used within a `\thebibliography` environment. The preferred method is to store references in `.bib` files and use the `\bibliography` command to generate a bibliography section automatically.

---

[b]ibitem:2
[b]ibitem
[b]rowse00056
[K] bibitem 2
[K] thebibliography
[K] bibliography
[E]nableButton("Up");ChangeButtonBinding("Up", "JumpId(`tex2rtf.hlp', `topic35')")

$#+K!**bibliographystyle:1**

Currently doesn't affect the style of bibliography, but probably will in the future.

---

bibliographystyle:1
bibliographystyle
browse00057
K bibliographystyle 1
EnableButton("Up");ChangeButtonBinding("Up", "JumpId(`tex2rtf.hlp', `topic35')")

$#+K!**bibliography:0**

Includes the bibliography at this point in the document. See the section on bibliographies.

---

bibliography:0
bibliographycmd
browse00058
K bibliography 0
EnableButton("Up");ChangeButtonBinding("Up", "JumpId(`tex2rtf.hlp', `topic35')")

$#+KKKK!**caption:1**

Specifies a caption (within a `\figure` or `\table` environment). This may be followed immediately by a `\label` command.

---

[c]aption:1
[c]aption
[b]rowse00059
[K] caption 1
[K] figure
[K] table
[K] label
[E]nableButton("Up");ChangeButtonBinding("Up", "JumpId(`tex2rtf.hlp', `topic35')")

$#+K!**cdots:0**

Outputs three dots.

---

cdots:0
cdots
browse00060
K cdots 0
EnableButton("Up");ChangeButtonBinding("Up", "JumpId(`tex2rtf.hlp', `topic35')")

$#+K!**centerline:1**

Centres (or centers!) a line of text.

---

centerline:1
centerline
browse00061
K centerline 1
EnableButton("Up");ChangeButtonBinding("Up", "JumpId(`tex2rtf.hlp', `topic35')")

$#+K!**center:1**

Centres a block of text.

---

center:1
center
browse00062
K center 1
EnableButton("Up");ChangeButtonBinding("Up", "JumpId(`tex2rtf.hlp', `topic35')")

$#+KK!**chapter:1**

Outputs a chapter heading. If the chapter's name is Popups, the chapter title will not be put in the contents, to allow popups to be placed in a document without the popup sections being directly accessible.

---

cchapter:1
cchapter
browse00063
K chapter 1
K popups
EnableButton("Up");ChangeButtonBinding("Up", "JumpId(`tex2rtf.hlp', `topic35')")

$#+K!**chapter*:1**

Outputs a chapter heading with no contents entry.

---

chapter*:1
chaptersX
browse00064
K chapter* 1
EnableButton("Up");ChangeButtonBinding("Up", "JumpId(`tex2rtf.hlp', `topic35')")

$#+K!**cite:1**

Cite a reference. The argument is a reference key as defined in a LaTeX `.bib` file.

---

cite:1
cite
browse00065
K cite 1
EnableButton("Up");ChangeButtonBinding("Up", "JumpId(`tex2rtf.hlp', `topic35')")

$#+KK!**comment:1**

An environment that allows large comments in LaTeX files: the argument is ignored in all formats. Useful for commenting out parts of files that cannot be handled by LaTeX, such as the picture environment. See also `\toocomplex`.

---

[c]omment:1
[c]omment
[b]rowse00066
[K] comment 1
[K] toocomplex
[E]nableButton("Up");ChangeButtonBinding("Up", "JumpId(`tex2rtf.hlp', `topic35')")

$#+KK!**date:1**

Specifies the date of a document; only output by `\maketitle`.

---

[d]ate:1
[d]ate
[b]rowse00067
[K] date 1
[K] maketitle
[E]nableButton("Up");ChangeButtonBinding("Up", "JumpId(`tex2rtf.hlp', `topic35')")

$#+KK!**description:1**

A list environment, where each `\item` command must be followed by optional square-bracketed text which will be highlighted.

---

[d]escription:1
[d]escription
[b]rowse00068
[K] description 1
[K] item
[E]nableButton("Up");ChangeButtonBinding("Up", "JumpId(`tex2rtf.hlp', `topic35')")

$#+K!**document:1**

This environment should enclose the body of a document.

---

document:1
document
browse00069
K document 1
EnableButton("Up");ChangeButtonBinding("Up", "JumpId(`tex2rtf.hlp', `topic35')")

$#+KKK!**documentstyle:1**

Specifies the main style (report, article etc.) and, optionally, style files such as `texhelp.sty`. A report has `\chapters`, while an article's top-level sections are specified using `\section`.

---

[d]ocumentstyle:1
[d]ocumentstyle
[b]rowse00070
[K] documentstyle 1
[K] chapters
[K] section
[E]nableButton("Up");ChangeButtonBinding("Up", "JumpId(`tex2rtf.hlp', `topic35')")

$#+K!**em:1**

Emphasizes text (italic in RTF).

---

em:1
em
browse00071
K em 1
EnableButton("Up");ChangeButtonBinding("Up", "JumpId(`tex2rtf.hlp', `topic35')")

$#+KK!**emph:1**

Same as `\em`.

---

emph:1
emph
browse00072
K emph 1
K em
EnableButton("Up");ChangeButtonBinding("Up", "JumpId(`tex2rtf.hlp', `topic35')")

$#+KK!**enumerate:1**

Enumerate list environment: numbers the `\items`.

---

eenumerate:1
eenumerate
browse00073
K enumerate 1
K items
EnableButton("Up");ChangeButtonBinding("Up", "JumpId(`tex2rtf.hlp', `topic35')")

$#+K!**figure:1**

A figure environment: does nothing special, except allows interpretation of embedded caption commands as figures rather than (say) tables.

---

figure:1
figure
browse00074
K figure 1
EnableButton("Up");ChangeButtonBinding("Up", "JumpId(`tex2rtf.hlp', `topic35')")

$#+K!**flushleft:1**

Flushes the given text to the left margin.

---

flushleft:1
flushleft
browse00075
K flushleft 1
EnableButton("Up");ChangeButtonBinding("Up", "JumpId(`tex2rtf.hlp', `topic35')")

$#+K!**flushright:1**

Flushes the given text to the right margin.

---

flushright:1
flushright
browse00076
K flushright 1
EnableButton("Up");ChangeButtonBinding("Up", "JumpId(`tex2rtf.hlp', `topic35')")

$#+K!**footnote:1**

In linear RTF, a footnote is created. Whether this appears at the end of the section or the bottom of the page appears to depend on the current document style, at least for MS Word 6.0 for Windows. The default seems to be to put the footnotes at the end of the section, which is probably not the best assumption.

In WinHelp RTF, a bracketed number is generated for the footnote and the footnote becomes a popup topic. It is probably preferable to change footnote commands to `\footnotepopup`, or `\popref` references to glossary entries.

This command is not supported for formats other than LaTeX, linear RTF and WinHelp RTF.

---

footnote:1
footnote
browse00077
K footnote 1
EnableButton("Up");ChangeButtonBinding("Up", "JumpId(`tex2rtf.hlp', `topic35')")

$#+KKK!**hline:0**

Within a `\tabular` environment, draws a horizontal rule below the current row. Note that this does not work in RTF for the last row of a table, in which case the command `\ruledrow` should be used instead.

---

hline:0
hline
browse00078
K hline 0
K tabular
K ruledrow
EnableButton("Up");ChangeButtonBinding("Up", "JumpId(`tex2rtf.hlp', `topic35')")

$#+K!**hrule:0**

Draws a horizontal line below the current paragraph. For example:

```
This paragraph should have a horizontal rule following it.\hrule
```

gives:

This paragraph should have a horizontal rule following it.

---

hrule:0
hrule
browse00079
K hrule 0
EnableButton("Up");ChangeButtonBinding("Up", "JumpId(`tex2rtf.hlp', `topic35')")

$#+K!**huge:1**

Outputs the argument in huge text.

---

huge:1
huge1
browse00080
K huge 1
EnableButton("Up");ChangeButtonBinding("Up", "JumpId(`tex2rtf.hlp', `topic35')")

$#+KK!**Huge:1**

Outputs the argument in huger text than \huge.

Huge:1
Huge2
browse00081
K Huge 1
K huge
EnableButton("Up");ChangeButtonBinding("Up", "JumpId(`tex2rtf.hlp', `topic35')")

$#+KK!**HUGE:1**

Outputs the argument in huger text than \Huge.

HUGE:1
HUGE3
browse00082
K HUGE 1
K Huge
EnableButton("Up");ChangeButtonBinding("Up", "JumpId(`tex2rtf.hlp', `topic35')")

$#+K!**include:1**

Include the given file. The command must not be preceded by any whitespace, and spurious whitespace between elements of the command will also trip up Tex2RTF.

---

inclue:1
inclue
browse00083
K include 1
EnableButton("Up");ChangeButtonBinding("Up", "JumpId(`tex2rtf.hlp', `topic35')")

$#+K!**index:1**

In WinHelp mode, adds a keyword to the keyword list for the current topic. This keyword must currently be straight text, with no embedded commands. The conversion process must be run twice (without quitting Tex2RTF inbetween) to resolve the keyword references.

---

iindex:1
iindex
browse00084
K index 1
EnableButton("Up");ChangeButtonBinding("Up", "JumpId(`tex2rtf.hlp', `topic35')")

$#+K!**input:1**

Include the given file. The command must not be preceded by any whitespace, and spurious whitespace between elements of the command will also trip up Tex2RTF.

---

input:1
input
browse00085
K input 1
EnableButton("Up");ChangeButtonBinding("Up", "JumpId(`tex2rtf.hlp', `topic35')")

$#+K!**insertatlevel:2**

Insert some text at a particular level of the document. For example,

```
\insertatlevel{2}{Some text}
```

inserts "Some text" at level 2 (for a report, the current section). This allows you to insert headings into an automatically-generated section contents, for example.

---

iinsertatlevel:2
iinsertatlevel
browse00086
K insertatlevel 2
EnableButton("Up");ChangeButtonBinding("Up", "JumpId(`tex2rtf.hlp', `topic35')")

$#+K!**it:1**

Marks the argument in italic.

---

i t:1
i t
b rowse00087
K it 1
E nableButton("Up");ChangeButtonBinding("Up", "JumpId(`tex2rtf.hlp', `topic35')")

$#+KKK!**itemize:1**

Indents each `\item` of a list and precedes with a bullet. If the file `bullet.bmp` is found by Tex2RTF, this bitmap will be used as the bullet (WinHelp RTF); otherwise, a symbol or bold 'o' will be used instead, depending on output format.

Use `\itemsep` to specify the separation between list items. Currently this only works for linear or WinHelp RTF output. If the value is more than zero, an extra paragraph is inserted.

---

itemize:1
itemize
browse00088
K itemize 1
K item
K itemsep
EnableButton("Up");ChangeButtonBinding("Up", "JumpId(`tex2rtf.hlp', `topic35')")

$#+KKKK!**item:0**

Marks an item of a \itemize, \description or \enumeratelist. Items within a description environment should have an 'optional' argument in square brackets which will be highlighted.

---

item:0
item
browse00089
K item 0
K itemize
K description
K enumerate
EnableButton("Up");ChangeButtonBinding("Up", "JumpId(`tex2rtf.hlp', `topic35')")

$#+K!**itemsep:0**

Use this command to specify the separation between list items. Currently this only works for linear or WinHelp RTF output. If the value is zero, no extra paragraph is inserted; if the value is more than zero, an extra paragraph is inserted.

---

itemsep:0
itemsep
browse00090
K itemsep 0
EnableButton("Up");ChangeButtonBinding("Up", "JumpId(`tex2rtf.hlp', `topic35')")

$#+K!**itshape:1**

Marks the argument in italic.

---

itshape:1
itshape
browse00091
K itshape 1
EnableButton("Up");ChangeButtonBinding("Up", "JumpId(`tex2rtf.hlp', `topic35')")

$#+K!**label:1**

Labels the chapter, section, subsection, subsubsection or figure caption with the given label. This must be an ASCII string, and duplicate items with different case letters are not allowed.

The command must follow immediately after the section or caption command, with no intervening whitespace.

---

labe l:1
labe l
browse00092
K label 1
EnableButton("Up");ChangeButtonBinding("Up", "JumpId(`tex2rtf.hlp', `topic35')")

$#+K!**large:1**

Marks the argument in large text.

---

large:1
large1
browse00093
K large 1
EnableButton("Up");ChangeButtonBinding("Up", "JumpId(`tex2rtf.hlp', `topic35')")

$#+KK!**Large:1**

Makes the argument display in larger text than `\large`.

Large:1
Large2
browse00094
K Large 1
K large
EnableButton("Up");ChangeButtonBinding("Up", "JumpId(`tex2rtf.hlp', `topic35')")

$#+KK!**LARGE:1**

Makes the argument display in larger text than `\Large`.

LARGE:1
LARGE3
browse00095
LARGE 1
Large
EnableButton("Up");ChangeButtonBinding("Up", "JumpId(`tex2rtf.hlp', `topic35')")

$#+K!**LaTeX:0**

Outputs the annoying LaTeX upper and lower case name.

LaTeX:0
LaTeX
browse00096
K LaTeX 0
EnableButton("Up");ChangeButtonBinding("Up", "JumpId(`tex2rtf.hlp', `topic35')")

$#+K!**ldots:0**

Outputs three dots.

---

ldots:0
ldots
browse00097
K ldots 0
EnableButton("Up");ChangeButtonBinding("Up", "JumpId(`tex2rtf.hlp', `topic35')")

$#+KKKK!**maketitle:0**

Makes the article or report title by outputting the `\title`, `\author` and optionally `\date`.

---

maketitle:0
maketitle
browse00098
K maketitle 0
K title
K author
K date
EnableButton("Up");ChangeButtonBinding("Up", "JumpId(`tex2rtf.hlp', `topic35')")

$#+K!**marginparwidth:1**

Specifies the width of a margin paragraph.

---

marginparwidth:1
marginparwidth
browse00099
K marginparwidth 1
EnableButton("Up");ChangeButtonBinding("Up", "JumpId(`tex2rtf.hlp', `topic35')")

$#+KKK!**marginpar:1**

Inserts a marginal note. It is best to use the Tex2RTF extensions `\marginparodd` and `\marginpareven` for best results.

---

marginpar:1
marginpar
browse00100
K marginpar 1
K marginparodd
K marginpareven
EnableButton("Up");ChangeButtonBinding("Up", "JumpId(`tex2rtf.hlp', `topic35')")

$#+KK!**marginpareven:1**

Inserts a marginal note on even pages. This is required for RTF generation since it is impossible for Tex2RTF to know in advance which side of paper the marginal note will fall upon, and the text has to be positioned using absolute dimensions. If only one sided output is required, use `\marginparodd` instead.

---

marginpareven:1
marginpareven
browse00101
K marginpareven 1
K marginparodd
EnableButton("Up");ChangeButtonBinding("Up", "JumpId(`tex2rtf.hlp', `topic35')")

$#+KKK!**marginparodd:1**

Inserts a marginal note on odd pages. This is required for RTF generation since it is impossible for Tex2RTF to know in advance which side of paper the marginal note will fall upon, and the text has to be positioned using absolute dimensions.

Also, even if one-sided output is required, this command should be used instead of `\marginpar` because the LaTeX command allows it to be used just before a paragraph. Normally, if this were done, the marginal note would not be aligned with the paragraph succeeding it. For example:

```
\marginparodd{{\it Note:} if nothing happens, perhaps you
have not plugged your computer in at the mains.}%
To start using your computer, push the Power button
and wait for text to appear on the screen.
```

Note the percent sign after the `\marginparodd` command: without it,LaTeX refuses to believe that the following text is part of the same paragraph, and will print the note at the wrong place.

You should use `\textwidth` to allow space for marginal notes, and also `\marginparwidth` to specify the size of the marginal note.

In WinHelp, HTML and wxHelp, marginal notes are treated as normal text delineated with horizontal rules above and below.

---

marginparodd:1
marginparodd
browse00102
K marginparodd 1
K textwidth
K marginparwidth
EnableButton("Up");ChangeButtonBinding("Up", "JumpId(`tex2rtf.hlp', `topic35')")

$#+K!**mdseries:1**

Changes to a medium-weight font. Un-emboldens in RTF mode, no effect in other modes.

---

mdseries:1
mdseries
browse00103
K mdseries 1
EnableButton("Up");ChangeButtonBinding("Up", "JumpId(`tex2rtf.hlp', `topic35')")

$#+KK!**multicolumn:3**

Used in \tabular environment to denote a cell that spans more than one column. Only supplied for compatibility with existing LaTeX files, since all it does in RTF is output the correct number of cell commands, with the multicolumn text squashed into one cell.

---

multicolumn:3
multicolumn
browse00104
K multicolumn 3
K tabular
EnableButton("Up");ChangeButtonBinding("Up", "JumpId(`tex2rtf.hlp', `topic35')")

$#+K!**newcommand:3**

Define a new command; arguments are the command, the number of arguments, and the command body. For example:

```
\newcommand{\crazy}[2]{{\bf #1} is crazy but {\bf #2} is not.}
```

The command must have no whitespace at the start of the line or between the three arguments.

New commands may also be defined in the `tex2rtf.ini` file using slightly different syntax (see Macro not found error).

---

newcommand:3
newcommand
browse00105
K newcommand 3
EnableButton("Up");ChangeButtonBinding("Up", "JumpId(`tex2rtf.hlp', `topic35')")

$#+K!**newpage:0**

Inserts a page break.

---

newpage:0
newpage
browse00106
K newpage 0
EnableButton("Up");ChangeButtonBinding("Up", "JumpId(`tex2rtf.hlp', `topic35')")

$#+KK!**nocite:1**

Specifies that this reference should appear in the bibliography, but the citation should not appear in the text.

See also `\cite`.

---

nocite:1
nocite
browse00107
K nocite 1
K cite
EnableButton("Up");ChangeButtonBinding("Up", "JumpId(`tex2rtf.hlp', `topic35')")

$#+KK!**noindent:0**

Sets paragraph indentation to zero. See also `\parindent`.

---

[n]oindent:0
[n]oindent
[b]rowse00108
[K] noindent 0
[K] parindent
[E]nableButton("Up");ChangeButtonBinding("Up", "JumpId(`tex2rtf.hlp', `topic35')")

$#+K!**normalsize:1**

Sets the font size back to normal.

---

normalsize:1
normalsize
browse00109
K normalsize 1
EnableButton("Up");ChangeButtonBinding("Up", "JumpId(`tex2rtf.hlp', `topic35')")

$#+K!**onecolumn:0**

Sets the number of columns to one. LaTeX and linear RTF only.

---

onecolumn:0
onecolumn
browse00110
K onecolumn 0
EnableButton("Up");ChangeButtonBinding("Up", "JumpId(`tex2rtf.hlp', `topic35')")

$#+K!**pageref:1**

In linear RTF, generates a page reference to the given label.

---

pageref:1
pageref
browse00111
K pageref 1
EnableButton("Up");ChangeButtonBinding("Up", "JumpId(`tex2rtf.hlp', `topic35')")

$#+KKK!**pagestyle:1**

If argument is `fancyplain` or `fancy`, Tex2RTF separates the header from the rest of the page with a rule. This command must be defined for headers and footers to work properly. See also `\setheader`,`\setfooter`.

LaTeX and linear RTF only.

---

pagestyle:1
pagestyle
browse00112
K pagestyle 1
K setheader
K setfooter
EnableButton("Up");ChangeButtonBinding("Up", "JumpId(`tex2rtf.hlp', `topic35')")

$#+K!**pagenumbering:1**

The argument may be one of:

alph a, b, ...

Alph A, B, ...

arabic 1, 2, ...

roman i, ii, ...

Roman I, II, ...

LaTeX and linear RTF only.

---

pagenumbering:1
pagenumbering
browse00113
K pagenumbering 1
EnableButton("Up");ChangeButtonBinding("Up", "JumpId(`tex2rtf.hlp', `topic35')")

$#+K!**paragraph:0**

Behaves as for a subsubsection.

---

paragraph:0
paragraph
browse00114
K paragraph 0
EnableButton("Up");ChangeButtonBinding("Up", "JumpId(`tex2rtf.hlp', `topic35')")

$#+K!**paragraph*:0**

Behaves as for a subsubsection.

---

pparagraph*:0
pparagraphX
bprowse00115
K paragraph* 0
EnableButton("Up");ChangeButtonBinding("Up", "JumpId(`tex2rtf.hlp', `topic35')")

$#+K!**parindent:1**

Indents the first line of succeeding paragraphs by the given amount.

---

parindent:1
parindent
browse00116
K parindent 1
EnableButton("Up");ChangeButtonBinding("Up", "JumpId(`tex2rtf.hlp', `topic35')")

$#+KK!**parskip:1**

Changes the spacing between paragraphs. In fact, in RTF this will cause two \par commands to be output if parskip is greater than zero.

---

pparskip:1
pparskip
browse00117
K parskip 1
K par
EnableButton("Up");ChangeButtonBinding("Up", "JumpId(`tex2rtf.hlp', `topic35')")

$#+K!**par:0**

Causes the paragraph to end at this point. LaTeX and Tex2RTF also treat two consecutive newlines as a paragraph break.

---

par:0
par
browse00118
K par 0
EnableButton("Up");ChangeButtonBinding("Up", "JumpId(`tex2rtf.hlp', `topic35')")

$#+K!**printindex:0**

In linear RTF, inserts an index.

---

printindex:0
pprintindex
browse00119
K printindex 0
EnableButton("Up");ChangeButtonBinding("Up", "JumpId(`tex2rtf.hlp', `topic35')")

$#+K!**quote:1**

Indents a short quotation.

---

qquote:1
qquote
browse00120
K quote 1
EnableButton("Up");ChangeButtonBinding("Up", "JumpId(`tex2rtf.hlp', `topic35')")

$#+K!**quotation:1**

Indents a long quotation.

---

qquotation:1
qquotation
browse00121
K quotation 1
EnableButton("Up");ChangeButtonBinding("Up", "JumpId(`tex2rtf.hlp', `topic35')")

$#+KK!**ref:1**

In LaTeX and linear RTF, refers to a `\label` and causes the number of that section or figure to be printed.

---

rref:1
rref
browse00122
K ref 1
K label
EnableButton("Up");ChangeButtonBinding("Up", "JumpId(`tex2rtf.hlp', `topic35')")

$#+K!**rm:1**

Causes the argument to be formatted in a plain, roman font. In fact, does nothing in RTF, HTML and XLP modes.

---

rm:1
rm
browse00123
K rm 1
EnableButton("Up");ChangeButtonBinding("Up", "JumpId(`tex2rtf.hlp', `topic35')")

$#+K!**rmfamily:1**

Causes the argument to be formatted in a plain, roman font. In fact, does nothing in RTF, HTML and XLP modes.

---

rmfamily:1
rmfamily
browse00124
K rmfamily 1
EnableButton("Up");ChangeButtonBinding("Up", "JumpId(`tex2rtf.hlp', `topic35')")

$#+K!**sc:1**

Prints the output in small capitals.

---

sc:1
sc
browse00125
K sc 1
EnableButton("Up");ChangeButtonBinding("Up", "JumpId(`tex2rtf.hlp', `topic35')")

$#+K!**scshape:1**

Prints the output in small capitals.

---

scshape:1
scshape
browse00126
K scshape 1
EnableButton("Up");ChangeButtonBinding("Up", "JumpId(`tex2rtf.hlp', `topic35')")

$#+K!**section:1**

Section header, with an entry in the contents page.

---

section:1
section
browse00127
K section 1
EnableButton("Up");ChangeButtonBinding("Up", "JumpId(`tex2rtf.hlp', `topic35')")

$#+K!**section*:1**

Section header, with no entry in the contents page.

---

section*:1
sectionX
browse00128
K section* 1
EnableButton("Up");ChangeButtonBinding("Up", "JumpId(`tex2rtf.hlp', `topic35')")

$#+K!**sf:1**

Should format in a sans-serif font. Does nothing in Tex2RTF.

---

sf:1
sf
browse00129
K sf 1
EnableButton("Up");ChangeButtonBinding("Up", "JumpId(`tex2rtf.hlp', `topic35')")

$#+K!**sffamily:1**

Should format in a sans-serif font. Does nothing in Tex2RTF.

---

sffamily:1
sffamily
browse00130
K sffamily 1
EnableButton("Up");ChangeButtonBinding("Up", "JumpId(`tex2rtf.hlp', `topic35')")

$#+KK!**shortcite:1**

The same as `\cite`.

---

[s]hortcite:1
[s]hortcite
[b]rowse00131
[K] shortcite 1
[K] cite
[E]nableButton("Up");ChangeButtonBinding("Up", "JumpId(`tex2rtf.hlp', `topic35')")

$#+KK!**sl:1**

In Tex2RTF, the same as `\it`. The LaTeX interpretation is 'slanted text'.

---

^{s}l:1
^{s}l
^{b}rowse00132
^{K} sl 1
^{K} it
^{E}nableButton("Up");ChangeButtonBinding("Up", "JumpId(`tex2rtf.hlp', `topic35')")

$#+KK!**slshape:1**

In Tex2RTF, the same as `\itshape`. The LaTeX interpretation is 'slanted text'.

---

slshape:1
slshape
browse00133
K slshape 1
K itshape
EnableButton("Up");ChangeButtonBinding("Up", "JumpId(`tex2rtf.hlp', `topic35')")

$#+K!**small:1**

Prints the argument in a small font.

---

small:1
small
browse00134
K small 1
EnableButton("Up");ChangeButtonBinding("Up", "JumpId(`tex2rtf.hlp', `topic35')")

$#+K!**special:1**

Simply copies the argument to the output file without processing (except `\}` is translated to `}`, and `\{` is translated to `{`, to allow for insertion of braces).

---

special:1
special
browse00135
special 1
EnableButton("Up");ChangeButtonBinding("Up", "JumpId(`tex2rtf.hlp', `topic35')")

$#+K!**ss:0**

Outputs the German sharp S character ß.

ˢs:0
ˢs
ᵇrowse00136
ᴷ ss 0
ᴱnableButton("Up");ChangeButtonBinding("Up", "JumpId(`tex2rtf.hlp', `topic35')")

$#+K!**subparagraph:1**

Behaves as for a subsubsection.

---

subparagraph:1
subparagraph
browse00137
K subparagraph 1
EnableButton("Up");ChangeButtonBinding("Up", "JumpId(`tex2rtf.hlp', `topic35')")

$#+K!**subparagraph*:1**

Behaves as for a subsubsection.

---

subparagraph*:1
subparagraphX
browse00138
K subparagraph* 1
EnableButton("Up");ChangeButtonBinding("Up", "JumpId(`tex2rtf.hlp', `topic35')")

$#+K!**subsection:1**

Subsection header, with an entry in the contents page.

---

subsection:1
subsection
browse00139
K subsection 1
EnableButton("Up");ChangeButtonBinding("Up", "JumpId(`tex2rtf.hlp', `topic35')")

$#+K!**subsection*:1**

Subsection header, with no entry in the contents page.

---

subsection*:1
subsectionX
browse00140
K subsection* 1
EnableButton("Up");ChangeButtonBinding("Up", "JumpId(`tex2rtf.hlp', `topic35')")

$#+K!**subsubsection:1**

Subsubsection header, with an entry in the contents page.

---

subsubsection:1
subsubsection
browse00141
K subsubsection 1
EnableButton("Up");ChangeButtonBinding("Up", "JumpId(`tex2rtf.hlp', `topic35')")

$#+K!**subsubsection*:1**

Subsubsection header, with no entry in the contents page.

---

subsubsection*:1
subsubsectionX
browse00142
K subsubsection* 1
EnableButton("Up");ChangeButtonBinding("Up", "JumpId(`tex2rtf.hlp', `topic35')")

$#+K!**tabbing:1**

Tabbing environment: doesn't work properly in RTF.

---

tabbing:1
tabbing
browse00143
K tabbing 1
EnableButton("Up");ChangeButtonBinding("Up", "JumpId(`tex2rtf.hlp', `topic35')")

$#+K!**table:1**

An environment for tables. The only thing that Tex2RTF does with this is to interpret an embedded caption command differently from figures.

---

table:1
table
browse00144
K table 1
EnableButton("Up");ChangeButtonBinding("Up", "JumpId(`tex2rtf.hlp', `topic35')")

$#+K!**tableofcontents:0**

Inserts the table of contents at this point. In linear RTF mode, a proper Word for Windows table of contents will be inserted unless either of the variables *insertTOC* or *useWord* is set to *false*.

---

tableofcontents:0
tableofcontents
browse00145
K tableofcontents 0
EnableButton("Up");ChangeButtonBinding("Up", "JumpId(`tex2rtf.hlp', `topic35')")

$#+KKKKK!**tabular:2**

Tabular environment: an attempt is made to output something reasonable in RTF and HTML formats, although currently only simple tables will work. The first argument specifies the column formatting. a pipe symbol (`|`) denotes a vertical border, one of `l, r, c` signifies a normal column of default width, and p followed by a dimension specifies a column of given width. It is recommended that the p is used since Tex2RTF cannot deduce a column width in the same way that LaTeX can.

Horizontal rules are achieved with `\hline`; two together signify a double rule. Note that in HTML, all rows and the table itself are bordered automatically.

Use the Tex2RTF `\row` and `\ruledrow` commands for best effect.

For two-column tables that work in WinHelp files, use `\twocollist` instead.

Example:

```
\begin{tabular}{|l|p{8.5cm}|}\hline
\row{{\bf A.I.}&{\bf Simulation}}\hline\hline
\row{rules&constraints/methods}
\row{planning&design of experiments}
\row{diagnosis&analysis of results}
\ruledrow{learning&detection of connections}
\end{tabular}
```

This produces:

A.I.	**Simulation**
rules	constraints/methods
planning	design of experiments
diagnosis	analysis of results
learning	detection of connections

---

[t]abular:2
[t]abular
[b]rowse00146
[K] tabular 2
[K] hline
[K] row
[K] ruledrow
[K] twocollist
[E]nableButton("Up");ChangeButtonBinding("Up", "JumpId(`tex2rtf.hlp', `topic35')")

$#+K!**TeX:0**

Outputs the annoying TeX upper and lower case name.

TeX:0
TeX
browse00147
K TeX 0
EnableButton("Up");ChangeButtonBinding("Up", "JumpId(`tex2rtf.hlp', `topic35')")

$#+KK!**textbf:1**

Same as `\bf`.

---

textbf:1
textbf
browse00148
K textbf 1
K bf
EnableButton("Up");ChangeButtonBinding("Up", "JumpId(`tex2rtf.hlp', `topic35')")

$#+KK!**textit:1**

Same as `\it`.

---

textit:1
textit
browse00149
K textit 1
K it
EnableButton("Up");ChangeButtonBinding("Up", "JumpId(`tex2rtf.hlp', `topic35')")

$#+KK!**textrm:1**

Same as `\rm`.

---

textrm:1
textrm
browse00150
K textrm 1
K rm
EnableButton("Up");ChangeButtonBinding("Up", "JumpId(`tex2rtf.hlp', `topic35')")

$#+KK!**textsf:1**

Same as `\sf`.

---

textsf:1
textsf
browse00151
K textsf 1
K sf
EnableButton("Up");ChangeButtonBinding("Up", "JumpId(`tex2rtf.hlp', `topic35')")

$#+KK!**textsc:1**

Same as \sc.

---

textsc:1
textsc
browse00152
K textsc 1
K sc
EnableButton("Up");ChangeButtonBinding("Up", "JumpId(`tex2rtf.hlp', `topic35')")

$#+KK!**textsl:1**

Same as `\sl`.

---

textsl:1
textsl
browse00153
K textsl 1
K sl
EnableButton("Up");ChangeButtonBinding("Up", "JumpId(`tex2rtf.hlp', `topic35')")

$#+KK!**texttt:1**

Same as `\tt`.

---

texttt:1
texttt
browse00154
K texttt 1
K tt
EnableButton("Up");ChangeButtonBinding("Up", "JumpId(`tex2rtf.hlp', `topic35')")

$#+KK!**textwidth:1**

Sets the text width (valid for RTF only). This might be used in conjunction with `\marginpar`, for example, to leave space for marginal notes.

---

textwidth:1
textwidth
browse00155
K textwidth 1
K marginpar
EnableButton("Up");ChangeButtonBinding("Up", "JumpId(`tex2rtf.hlp', `topic35')")

$#+KKK!**thebibliography:1**

An environment for specifying the bibliography as a series of `\bibitem` commands; the preferred method is to use `.bib` files and `\bibliography` instead.

---

thebibliography:1
thebibliography
browse00156
K thebibliography 1
K bibitem
K bibliography
EnableButton("Up");ChangeButtonBinding("Up", "JumpId(`tex2rtf.hlp', `topic35')")

$#+KK!**title:1**

Sets the title, to be output when the command `\maketitle` is used.

---

title:1
title
browse00157
K title 1
K maketitle
EnableButton("Up");ChangeButtonBinding("Up", "JumpId(`tex2rtf.hlp', `topic35')")

$#+K!**tiny:1**

Prints the argument in a very small font.

---

tiny:1
tiny
browse00158
K tiny 1
EnableButton("Up");ChangeButtonBinding("Up", "JumpId(`tex2rtf.hlp', `topic35')")

$#+K!**today:0**

Outputs today's date.

---

toda y:0
today
browse00159
K today 0
EnableButton("Up");ChangeButtonBinding("Up", "JumpId(`tex2rtf.hlp', `topic35')")

$#+K!**tt:1**

Outputs the argument in teletype font.

---

tt:1
tt
browse00160
K tt 1
EnableButton("Up");ChangeButtonBinding("Up", "JumpId(`tex2rtf.hlp', `topic35')")

$#+K!**ttfamily:1**

Outputs the argument in teletype font.

---

ttfamily:1
ttfamily
browse00161
K ttfamily 1
EnableButton("Up");ChangeButtonBinding("Up", "JumpId(`tex2rtf.hlp', `topic35')")

$#+K!**typeout:1**

Outputs the text on the Tex2RTF text window.

---

typeout:1
typeout
browse00162
K typeout 1
EnableButton("Up");ChangeButtonBinding("Up", "JumpId(`tex2rtf.hlp', `topic35')")

$#+K!**twocolumn:0**

Sets the number of columns to two. LaTeX and linear RTF only.

---

twocolumn:0
twocolumn
browse00163
K twocolumn 0
EnableButton("Up");ChangeButtonBinding("Up", "JumpId(`tex2rtf.hlp', `topic35')")

$#+K!**underline:1**

Underlines the argument.

---

uderline:1
uderline
browse00164
K underline 1
EnableButton("Up");ChangeButtonBinding("Up", "JumpId(`tex2rtf.hlp', `topic35')")

$#+K!**upshape:1**

Changes to an upright font. Un-italicizes in RTF mode, no effect in other modes.

---

upshape:1
upshape
browse00165
K upshape 1
EnableButton("Up");ChangeButtonBinding("Up", "JumpId(`tex2rtf.hlp', `topic35')")

$#+KK!**verbatiminput:1**

Include the given file as if it were within a `\verbatim` environment. The command must not be preceded by any whitespace, and spurious whitespace between elements of the command will also trip up Tex2RTF.

---

verbatiminput:1
verbatiminput
browse00166
K verbatiminput 1
K verbatim
EnableButton("Up");ChangeButtonBinding("Up", "JumpId(`tex2rtf.hlp', `topic35')")

$#+K!**verbatim:1**

Uses a fixed-width font to format the argument without interpreting any LaTeX commands.

---

verbatim:1
verbatim
browse00167
K verbatim 1
EnableButton("Up");ChangeButtonBinding("Up", "JumpId(`tex2rtf.hlp', `topic35')")

$#+K!**verb**

The `\verb` command is like the `\verbatim` environment, but for small amounts of text. The syntax is:

```
\verb<char><text><char>
```

The character *char* is used as a delimiter; it may be any character not ocurring in the following text, except asterisk.

For example, `\verb$\thing%^&$` produces `\thing%^&`.

---

verb
verb
browse00168
K verb
EnableButton("Up");ChangeButtonBinding("Up", "JumpId(`tex2rtf.hlp', `topic35')")

$#+K!**backgroundcolour:1**

Specifies the page background colour, in HTML only. The argument consists of three numbers from 0 to 255 separated by semicolons, for red, green and blue values respectively.

```
\backgroundcolour{255;255;255}
\backgroundcolour{0;0;255}
```

The first example sets the background to white, the second sets the background to blue.

Instead of using a LaTeX command, you may find it more convenient to use the equivalent `.ini` file setting, *backgroundColour*.

---

backgroundcolour:1
backgroundcolour
browse00170
K backgroundcolour 1
EnableButton("Up");ChangeButtonBinding("Up", "JumpId(`tex2rtf.hlp', `topic36')")

$#+K!**backgroundimage:1**

Specifies the page background image, in HTML only. The argument is a URL for the GIF file to be used as the background.

For example:

```
\backgroundimage{tile.gif}
```

This sets the background to a tile file.

Instead of using a LaTeX command, you may find it more convenient to use the equivalent `.ini` file setting, *backgroundImage*.

---

backgroundimage:1
backgroundimage
browse00171
K backgroundimage 1
EnableButton("Up");ChangeButtonBinding("Up", "JumpId(`tex2rtf.hlp', `topic36')")

$#+K!**backslashraw:0**

Outputs a raw backslash into the output (not LaTeX). Useful when inserting RTF (for example) that cannot be dealt with by Tex2RTF. E.g.

```
\backslashraw{'e3}
```

inserts the text `\'e3` into the RTF file.

---

backslashraw:0
backslashraw
browse00172
K backslashraw 0
EnableButton("Up");ChangeButtonBinding("Up", "JumpId(`tex2rtf.hlp', `topic36')")

$#+KKK!**bcol:2**

Sets the background colour for a block of text (RTF only). Has no known effect in the RTF readers currently tried (Word for Window and Windows Help).

See also `\definecolour`, `\fcol`.

---

bcol:2
bcol
browse00173
K bcol 2
K definecolour
K fcol
EnableButton("Up");ChangeButtonBinding("Up", "JumpId(`tex2rtf.hlp', `topic36')")

$#+KKK!**brclear:0**

Stops aligning content following a left or right-aligned image in HTML only.

See also `\imagel`, `\imager`.

---

bbrclear:0
bbrclear
bbrowse00174
K brclear 0
K imagel
K imager
EnableButton("Up");ChangeButtonBinding("Up", "JumpId(`tex2rtf.hlp', `topic36')")

$#+K!**cextract:0**

Prints a C++ extraction operator (>>).

---

cextract:0
cextract
browse00175
K cextract 0
EnableButton("Up");ChangeButtonBinding("Up", "JumpId(`tex2rtf.hlp', `topic36')")

$#+KK!**chapterheading:1**

Like `\chapter`, but does not increment the chapter number and does not print a chapter number in the printed documentation contents page, or in the chapter heading. Used to implement glossaries and other sections that are not real chapters.

---

[c]hapterheading:1
[c]hapterheading
[b]rowse00176
[K] chapterheading 1
[K] chapter
[E]nableButton("Up");ChangeButtonBinding("Up", "JumpId(`tex2rtf.hlp', `topic36')")

$#+K!**cinsert:0**

Prints a C++ insertion operator (<<).

---

cinsert:0
cinsert
browse00177
K cinsert 0
EnableButton("Up");ChangeButtonBinding("Up", "JumpId(`tex2rtf.hlp', `topic36')")

$#+K!**class:1**

Outputs the argument, an index entry (LaTeX only) and a keyword entry (WinHelp only). Used in class reference documentation.

---

class:1
class
browse00178
K class 1
EnableButton("Up");ChangeButtonBinding("Up", "JumpId(`tex2rtf.hlp', `topic36')")

$#+K!**clipsfunc:3**

Formats a CLIPS function, given the return value, function name, and arguments.

---

clipsfunc:3
clipsfunc
browse00179
K clipsfunc 3
EnableButton("Up");ChangeButtonBinding("Up", "JumpId(`tex2rtf.hlp', `topic36')")

$#+K!**copyright:0**

Outputs the copyright symbol.

---

copyright:0
copyright
browse00180
K copyright 0
EnableButton("Up");ChangeButtonBinding("Up", "JumpId(`tex2rtf.hlp', `topic36')")

$#+KK!**cparam:2**

Formats a CLIPS type and argument. Used within the third argument of a `\clipsfunc` command.

---

cparam:2
cparam
browse00181
K cparam 2
K clipsfunc
EnableButton("Up");ChangeButtonBinding("Up", "JumpId(`tex2rtf.hlp', `topic36')")

$#+KKK!**definecolour:4**

Defines a new colour that can be used in the document (RTF only). This command can also be spelt `\definecolor`.

The first argument is the lower-case name of the colour, and the following three arguments specify the red, green and blue intensities, in the range 0 to 255.

The default colours are equivalent to the following definitions:

```
\definecolour{black}{0}{0}{0}
\definecolour{cyan}{0}{255}{255}
\definecolour{green}{0}{255}{0}
\definecolour{magenta}{255}{0}{255}
\definecolour{red}{255}{0}{0}
\definecolour{yellow}{255}{255}{0}
\definecolour{white}{255}{255}{255}
```

To use colours in a document, use the `\fcol` and `\bcol` commands.

Note that a document that defines its own colours should be converted twice within the same Tex2RTF session.

---

[d]efinecolour:4
[d]efinecolour
[b]rowse00182
[K] definecolour 4
[K] fcol
[K] bcol
[E]nableButton("Up");ChangeButtonBinding("Up", "JumpId(`tex2rtf.hlp', `topic36')")

$#+KKK!**fcol:2**

Sets the foreground colour for a block of text (RTF and HTML).

For example:

```
This sentence is brightened up by some \fcol{red}{red text}.
```

gives:

This sentence is brightened up by some red text.

See also `\definecolour`, `\bcol`.

---

fcol:2
fcol
browse00183
K fcol 2
K definecolour
K bcol
EnableButton("Up");ChangeButtonBinding("Up", "JumpId(`tex2rtf.hlp', `topic36')")

$#+KKKK!**followedlinkcolour:1**

Specifies the followed link colour for the whole page, HTML only. The argument consists of three numbers from 0 to 255 separated by semicolons, for red, green and blue values respectively.

For example:

```
\followedlinkcolour{255;255;255}
\followedlinkcolour{0;0;255}
```

The first example sets the followed link text to white, and the second sets the followed link text to blue.

See also `\backgroundcolour`, `\textcolour`, `\linkcolour`.

Instead of using a LaTeX command, you may find it more convenient to use the equivalent `.ini` file setting, *followedLinkColour*.

---

followedlinkcolour:1
followedlinkcolour
browse00184
K followedlinkcolour 1
K backgroundcolour
K textcolour
K linkcolour
EnableButton("Up");ChangeButtonBinding("Up", "JumpId(`tex2rtf.hlp', `topic36')")

$#+K!**footnotepopup:2**

In linear RTF, a footnote is created following the first argument, as with`\footnote`.

In WinHelp RTF, a the first argument is highlighted and becomes a popup reference to the second argument. See also `\footnote` and `\popref`.

This command is not supported for formats other than LaTeX, linear RTF and WinHelp RTF.

---

footnotepopup:2
footnotepopup
browse00185
K footnotepopup 2
EnableButton("Up");ChangeButtonBinding("Up", "JumpId(`tex2rtf.hlp', `topic36')")

$#+KK!**functionsection:1**

Defines a subsection, adding the C++ function name to the LaTeX index or the WinHelp keyword list.

Should be followed by a `\func` command to specify function details.

---

functionsection:1
functionsection
browse00186
K functionsection 1
K func
EnableButton("Up");ChangeButtonBinding("Up", "JumpId(`tex2rtf.hlp', `topic36')")

$#+KK!**func:3**

Defines a C++ function, given the return type, function name, and parameter list.

Should occur after a `\functionsection` command.

---

func:3
func
browse00187
K func 3
K functionsection
EnableButton("Up");ChangeButtonBinding("Up", "JumpId(`tex2rtf.hlp', `topic36')")

$#+KKKK!**gloss:1**

Marks a glossary entry. In LaTeX, this is a synonym for an `\item`with an optional argument, within a `\description` environment, and the argument is added to the index.

In Windows Help, this is identical to a `\section*` in a report.

If labels are associated with the glossary entries, they can be referenced by`\helpref` or `\popref` jumps. A glossary entry is currently the only type of destination that popref may refer to.

This is an example of making a glossary in a report:

```
\begin{helpglossary}

\gloss{API}\label{api}

Application Programmer's Interface - a set of calls and
classes defining how a library (in this case, wxWidgets)
can be used.

\gloss{Canvas}\label{canvas}

A canvas in XView and wxWidgets is a subwindow...

\gloss{DDE}\label{dde}

Dynamic Data Exchange - Microsoft's interprocess
communication protocol. wxWidgets provides an abstraction
of DDE under both Windows and UNIX.

\end{helpglossary}
```

---

[g]loss:1
[g]loss
[b]rowse00188
[K] gloss 1
[K] item
[K] description
[K] section*
[E]nableButton("Up");ChangeButtonBinding("Up", "JumpId(`tex2rtf.hlp', `topic36')")

$#+KK!**helpglossary:1**

An environment for making a glossary (not standard LaTeX). See `\gloss` for usage.

---

helpglossary:1
helpglossary
browse00189
K helpglossary 1
K gloss
EnableButton("Up");ChangeButtonBinding("Up", "JumpId(`tex2rtf.hlp', `topic36')")

$#+K!**helpignore:1**

Ignores the argument in Tex2RTF generated files, but not LaTeX.

---

helpignore:1
helpignore
browse00190
K helpignore 1
EnableButton("Up");ChangeButtonBinding("Up", "JumpId(`tex2rtf.hlp', `topic36')")

$#+K!**helponly:1**

Only outputs the argument in Tex2RTF generated files.

---

helponly:1
helponly
browse00191
K helponly 1
EnableButton("Up");ChangeButtonBinding("Up", "JumpId(`tex2rtf.hlp', `topic36')")

$#+K!**helpinput:1**

Only includes the given file in Tex2RTF generated files.

---

helpinput:1
helpinput
browse00192
K helpinput 1
EnableButton("Up");ChangeButtonBinding("Up", "JumpId(`tex2rtf.hlp', `topic36')")

$#+K!**helpfontfamily:1**

Specifies the font family for Tex2RTF generated files. The argument may be Swiss or Times.

---

helpfontfamily:1
helpfontfamily
browse00193
K helpfontfamily 1
EnableButton("Up");ChangeButtonBinding("Up", "JumpId(`tex2rtf.hlp', `topic36')")

$#+K!**helpfontsize:1**

Specifies the font size for Tex2RTF generated files.

---

helpfontsize:1
helpfontsize
browse00194
K helpfontsize 1
EnableButton("Up");ChangeButtonBinding("Up", "JumpId(`tex2rtf.hlp', `topic36')")

$#+KK!**helpref:2**

Specifies a jump to a labelled chapter, section, subsection subsubsection or figure.

The first argument is text to be highlighted (mouseable in help systems) and the second is the reference label. In linear documents, the section number is given following the text, unless the `\helprefn` command is used instead, where the section number is suppressed.

Note that when generating HTML, the label *contents* is automatically defined, and may be referenced using `\helpref`.

---

helpref:2
helpref
browse00195
K helpref 2
K helprefn
EnableButton("Up");ChangeButtonBinding("Up", "JumpId(`tex2rtf.hlp', `topic36')")

$#+KK!**helprefn:2**

Specifies a jump to a labelled chapter, section, subsection subsubsection or figure.

The first argument is text to be highlighted (mouseable in help systems) and the second is the reference label. See `\helpref` for the form where the section number is printed in linear documents.

---

helprefn:2
helprefn
browse00196
K helprefn 2
K helpref
EnableButton("Up");ChangeButtonBinding("Up", "JumpId(`tex2rtf.hlp', `topic36')")

$#+K!**htmlignore:1**

Ignores the argument in HTML.

---

htmlignore:1
htmlignore
browse00197
K htmlignore 1
EnableButton("Up");ChangeButtonBinding("Up", "JumpId(`tex2rtf.hlp', `topic36')")

$#+K!**htmlonly:1**

Only outputs the argument in HTML.

---

htmlonly:1
htmlonly
browse00198
K htmlonly 1
EnableButton("Up");ChangeButtonBinding("Up", "JumpId(`tex2rtf.hlp', `topic36')")

$#+KKK!**image:2**

This is translated to a PSBOX macro package `\psboxto` command in LaTeX, the first argument being a sizing command and the second a filename.

In HTML mode, the second argument is used to generate a PostScript file reference.

In RTF mode, the second argument is tried with first a BMP extension and then a WMF extension to find a suitable Windows bitmap file, placeable metafile, or segmented hypergraphics file (.SHG). If a suitable file is found, in Windows Help mode a `bmc` command is inserted into the RTF file with a reference to the file. In linear RTF mode, the bitmap or metafile is converted into hex and inserted into the RTF document.

Note that only RGB-encoded Windows bitmaps, or placeable metafiles, are valid for input to Tex2RTF. You can convert a RLE (run length encoded) bitmap file into a (bigger) RGB file using a program such as Paintshop Pro. A placeable metafile has a special header with dimension information. One may be constructed by a wxWidgets program by calling the function wxMakeMetafilePlaceable. The Microsoft Windows SDK has a sample program that loads and steps through placeable and ordinary metafiles.

Another wrinkle is that programs differ in the methods they use to recognise pictures in RTF files. You may need to use the *bitmapMethod* setting, which can be "hex" (embed the hex data in the file with a `\dibitmap` keyword), "includepicture'' (use the MS Word 6.0 INCLUDEPICTURE field) or "import" (an earlier name for INCLUDEPICTURE).

Here is an example of using the `\image` command.

```
\begin{figure}
$$\image{5cm;0cm}{heart.ps}$$

\caption{My picture}\label{piccy}
\end{figure}
```

The dollars centre the image in the horizontal plane. The syntax of the first argument to `\image` is taken from syntax used by the `\psbox` package: it allows specification of the horizontal and vertical dimensions of the image. Scaling will take place for PostScript and metafile images. A value of zero indicates that the image should be scaled in proportion to the non-zero dimension. Zeros for both dimensions will leave the image unscaled in the case of metafiles, or scaled to fit the page in the case of PostScript.

See also `\imagel`, `\imager` for aligned images in HTML.

---

image:2
image
browse00199
K image 2
K imagel
K imager
EnableButton("Up");ChangeButtonBinding("Up", "JumpId(`tex2rtf.hlp', `topic36')")

$#+KKKK!**imagel:2**

Similar to `\image`, but left-aligns the image with respect to the following content. Use `\brclear` to stop aligning the content to the right of the image.

See also `\imager`.

---

imagel:2
imagel
browse00200
K imagel 2
K image
K brclear
K imager
EnableButton("Up");ChangeButtonBinding("Up", "JumpId(`tex2rtf.hlp', `topic36')")

$#+K!**imagemap:3**

This is translated to an HTML image map reference, or (in LaTeX) a PostScript psbox command. This allows images in HTML to have hotspots, where the user clicks on a part of the image and the browser jumps to a particular file.

The first argument is the same as the first argument to the `\image` command (ignored in HTML). The second argument must be the name of the image map entry, and the second is the filename to be displayed inline.

```
\imagemap{}{tree.gif}{myname}
```

translates to:

```
<a href="/cgi-bin/imagemap/mymap">
  <img src="tree.gif" ismap></a><p>
```

The snag with this is that, apart from the inconvenience of having to register a map file with the server, the map file will also have references to particular HTML files. If they exist in the current document, these names are not known until the file is generated. In which case, the map entries should probably refer to symbolic links that can be easily changed later.

---

imagemap:3
imagemap
browse00201
K imagemap 3
EnableButton("Up");ChangeButtonBinding("Up", "JumpId(`tex2rtf.hlp', `topic36')")

$#+KKKK!**imager:2**

Similar to `\image`, but right-aligns the image with respect to the following content. Use `\brclear` to stop aligning the content to the left of the image.

See also `\imagel`.

---

imager:2
imager
browse00202
K imager 2
K image
K brclear
K imagel
EnableButton("Up");ChangeButtonBinding("Up", "JumpId(`tex2rtf.hlp', `topic36')")

$#+K!indented:2

Environment supplied by Tex2RTF to allow (possibly nested) indentation ofLaTeX and RTF text. The first argument is the amount to be indented.

For example:

```
\begin{indented}{2cm}
This text should be indented by a couple of centimetres.
This can be useful to highlight paragraphs.
\end{indented}
```

produces:

> This text should be indented by a couple of centimetres. This can be useful to highlight paragraphs.

---

iindented:2
iindented
browse00203
K indented 2
EnableButton("Up");ChangeButtonBinding("Up", "JumpId(`tex2rtf.hlp', `topic36')")

$#+K!**latexignore:1**

Ignores the argument in LaTeX.

!atexignore:1
!atexignore
browse00204
K latexignore 1
EnableButton("Up");ChangeButtonBinding("Up", "JumpId(`tex2rtf.hlp', `topic36')")

$#+K!**latexonly:1**

Only prints the argument in LaTeX.

!atexonly:1
!atexonly
browse00205
K latexonly 1
EnableButton("Up");ChangeButtonBinding("Up", "JumpId(`tex2rtf.hlp', `topic36')")

$#+K!**lbraceraw:0**

Outputs a raw left brace into the output (not LaTeX). Useful when inserting RTF (for example) that cannot be dealt with by Tex2RTF.

---

lbraceraw:0
lbraceraw
browse00206
K lbraceraw 0
EnableButton("Up");ChangeButtonBinding("Up", "JumpId(`tex2rtf.hlp', `topic36')")

$#+KKKK!**linkcolour:1**

Specifies the link colour for the whole page, HTML only. The argument consists of three numbers from 0 to 255 separated by semicolons, for red, green and blue values respectively.

For example:

```
\linkcolour{255;255;255}
\linkcolour{0;0;255}
```

The first example sets the link text to white, and the second sets the link text to blue.

See also `\backgroundcolour`, `\textcolour`, `\followedlinkcolour`.

Instead of using a LaTeX command, you may find it more convenient to use the equivalent `.ini` file setting, *linkColour*.

[linkcolour:1
[linkcolour
[browse00207
[K] linkcolour 1
[K] backgroundcolour
[K] textcolour
[K] followedlinkcolour
[EnableButton("Up");ChangeButtonBinding("Up", "JumpId(`tex2rtf.hlp', `topic36')")

$#+K!**membersection:1**

Used when formatting C++ classes to print a subsection for the member name.

---

membersection:1
membersection
browse00208
K membersection 1
EnableButton("Up");ChangeButtonBinding("Up", "JumpId(`tex2rtf.hlp', `topic36')")

$#+K!**member:1**

Used to format a C++ member variable name.

---

member:1
member
browse00209
K member 1
EnableButton("Up");ChangeButtonBinding("Up", "JumpId(`tex2rtf.hlp', `topic36')")

$#+KK!**normalbox:1**

Draws a box around the given paragraph in LaTeX and RTF. In HTML and XLP formats, horizontal rules are drawn before and after the text.

For example:

```
\normalbox{This should be a boxed paragraph for highlighting
important information, such as information for registering
a shareware program.}
```

gives:

This should be a boxed paragraph for highlighting important information, such as information for registering a shareware program.

See also `\normalboxd` for double-bordered text.

---

normalbox:1
normalbox
browse00210
K normalbox 1
K normalboxd
EnableButton("Up");ChangeButtonBinding("Up", "JumpId(`tex2rtf.hlp', `topic36')")

$#+KK!**normalboxd:1**

Draws a double border around the given paragraph in LaTeX and RTF. In HTML and XLP formats, horizontal rules are drawn before and after the text.

For example:

```
\normalboxd{This should be a boxed paragraph for
highlighting important information, such as information
for registering a shareware program.}
```

gives:

This should be a boxed paragraph for highlighting important information,such as information for registering a shareware program.

See also \normalbox for single-bordered text.

---

normalboxd:1
normalboxd
browse00211
K normalboxd 1
K normalbox
EnableButton("Up");ChangeButtonBinding("Up", "JumpId(`tex2rtf.hlp', `topic36')")

$#+KK!**param:1**

Formats a C++ type and argument pair. Should be used within the third argument of a a `\func` command.

---

pparam:1
pparam
browse00212
K param 1
K func
EnableButton("Up");ChangeButtonBinding("Up", "JumpId(`tex2rtf.hlp', `topic36')")

$#+KKK!**popref:2**

Similar to \helprefn, except that in Windows Help, the destination text is popped up in a small window to be dismissed with a mouse click, instead of going to a separate section.

Currently this command can only refer to a labelled glossary entry; see\gloss.

---

popref:2
popref
browse00213
popref 2
helprefn
gloss
EnableButton("Up");ChangeButtonBinding("Up", "JumpId(`tex2rtf.hlp', `topic36')")

$#+KK!**psboxto:2**

Identical to `\image`.

---

[p]sboxto:2
[p]sboxto
[b]rowse00214
[K] psboxto 2
[K] image
[E]nableButton("Up");ChangeButtonBinding("Up", "JumpId(`tex2rtf.hlp', `topic36')")

$#+K!**rbraceraw:0**

Outputs a raw right brace into the output (not LaTeX). Useful when inserting RTF (for example) that cannot be dealt with by Tex2RTF.

---

rbraceraw:0
rbraceraw
browse00215
K rbraceraw 0
EnableButton("Up");ChangeButtonBinding("Up", "JumpId(`tex2rtf.hlp', `topic36')")

$#+K!**registered:0**

Outputs the 'registered' symbol in HTML, and (r) in other formats.

---

registered:0
registered
browse00216
K registered 0
EnableButton("Up");ChangeButtonBinding("Up", "JumpId(`tex2rtf.hlp', `topic36')")

$#+KKK!**row:1**

A Tex2RTF command signifying the row of a table within the `\tabular` environment. See also `\ruledrow`.

---

row:1
row
browse00217
K row 1
K tabular
K ruledrow
EnableButton("Up");ChangeButtonBinding("Up", "JumpId(`tex2rtf.hlp', `topic36')")

$#+KKK!**ruledrow:1**

A Tex2RTF command signifying a ruled row of a table within the `\tabular` environment. See also `\row`.

---

ruledrow:1
ruledrow
browse00218
K ruledrow 1
K tabular
K row
EnableButton("Up");ChangeButtonBinding("Up", "JumpId(`tex2rtf.hlp', `topic36')")

$#+K!**rtfignore:1**

Ignores the argument in linear RTF.

---

rtfignore:1
rtfignore
browse00219
K rtfignore 1
EnableButton("Up");ChangeButtonBinding("Up", "JumpId(`tex2rtf.hlp', `topic36')")

$#+K!**rtfonly:1**

Only outputs the argument in linear RTF.

---

rtfonly:1
rtfonly
browse00220
K rtfonly 1
EnableButton("Up");ChangeButtonBinding("Up", "JumpId(`tex2rtf.hlp', `topic36')")

$#+K!**rtfsp:0**

Outputs a space in RTF. Tex2RTF tries to insert a space where one is implied by a newline, but cannot cope where a line starts or ends with a command, in the middle of a paragraph. Use this command to insert a space explicitly.

---

rtfsp:0
rtfsp
browse00221
K rtfsp 0
EnableButton("Up");ChangeButtonBinding("Up", "JumpId(`tex2rtf.hlp', `topic36')")

$#+KK!**sectionheading:1**

Like `\section`, but does not increment the section number and does not print a section number in the printed documentation contents page, or in the section heading.

---

[s]ectionheading:1
[s]ectionheading
[b]rowse00222
[K] sectionheading 1
[K] section
[E]nableButton("Up");ChangeButtonBinding("Up", "JumpId(`tex2rtf.hlp', `topic36')")

$#+KK!**setfooter:6**

Tex2RTF has a non-standard way of setting headers and footers, but the default macro definitions in `texhelp.sty` may be altered to your current method.

The arguments are as follows:

1. Left footer, even pages
2. Centre footer, even pages
3. Right footer, even pages
4. Left footer, odd pages
5. Centre footer, odd pages
6. Right footer, odd pages

For many documents, the first three arguments will be left empty.

The behaviour for first pages of a chapter, section or document is to have a blank header, but print the footer.

For best results, define headers and footers for *each chapter or section*.

Note that this command works only for LaTeX and linear RTF. See also `\setheader`.

---

setfooter:6
setfooter
browse00223
K setfooter 6
K setheader
EnableButton("Up");ChangeButtonBinding("Up", "JumpId(`tex2rtf.hlp', `topic36')")

$#+KKK!**setheader:6**

Tex2RTF has a non-standard way of setting headers and footers, but the default macro definitions in `texhelp.sty` may be altered to your current method.

The arguments are as follows:

1. Left header, even pages
2. Centre header, even pages
3. Right header, even pages
4. Left header, odd pages
5. Centre header, odd pages
6. Right header, odd pages

For many documents, the first three arguments will be left empty. If `\pagestyle` is not plain or empty, the header will separated from the rest of the page by a rule.

The behaviour for first pages of a chapter, section or document is to have a blank header, but print the footer.

For best results, define headers and footers for *each chapter or section*.

Note that this command works only for LaTeX and linear RTF. See also `\setfooter`.

---

[s]etheader:6
[s]etheader
[b]rowse00224
[K] setheader 6
[K] pagestyle
[K] setfooter
[E]nableButton("Up");ChangeButtonBinding("Up", "JumpId(`tex2rtf.hlp', `topic36')")

$#+K!**sethotspotcolour:1**

If the argument is yes, on or ok, subsequent WinHelp hotspots will be green. If any other value, the hotspots will be the normal text colour. Note that this doesn't apply to section hotspots, only to helpref hotspots.

---

sethotspotcolour:1
sethotspotcolour
browse00225
K sethotspotcolour 1
EnableButton("Up");ChangeButtonBinding("Up", "JumpId(`tex2rtf.hlp', `topic36')")

$#+K!**sethotspotunderline:1**

If the argument is yes, on or ok, subsequent WinHelp hotspots will be underlined (the default). If any other value, the hotspots will not be underlined. Note that this doesn't apply to section hotspots, only to helpref hotspots.

---

sethotspotunderline:1
sethotspotunderline
browse00226
K sethotspotunderline 1
EnableButton("Up");ChangeButtonBinding("Up", "JumpId(`tex2rtf.hlp', `topic36')")

$#+K!**settransparency:1**

WinHelp mode only (version 4 of WinHelp). If the argument is yes, on or ok, subsequent bitmaps will be inserted in transparent mode: areas of white will be made transparent. If the argument is any other value (such as no, ok or false), the bitmaps will not be transparent.

---

settransparency:1
settransparency
browse00227
K settransparency 1
EnableButton("Up");ChangeButtonBinding("Up", "JumpId(`tex2rtf.hlp', `topic36')")

$#+KKKK!**textcolour:1**

Specifies the text foreground colour for the whole page, HTML only. The argument consists of three numbers from 0 to 255 separated by semicolons, for red, green and blue values respectively.

For example:

```
\textcolour{255;255;255}
\textcolour{0;0;255}
```

The first example sets the text to white, and the second sets the text to blue.

See also `\backgroundcolour`, `\linkcolour`, `\followedlinkcolour`.

Instead of using a LaTeX command, you may find it more convenient to use the equivalent `.ini` file setting, *textColour*.

---

textcolour:1
textcolour
browse00228
K textcolour 1
K backgroundcolour
K linkcolour
K followedlinkcolour
EnableButton("Up");ChangeButtonBinding("Up", "JumpId(`tex2rtf.hlp', `topic36')")

$#+KK!**toocomplex:1**

An environment for dealing with complex LaTeX commands that Tex2RTF cannot handle. In normal LaTeX, the argument will be output as normal. In Tex2RTF output, the argument will be output as verbatim text, for the user to hand-translate into the desired output format.

See also `\comment`.

---

toocomplex:1
toocomplex
browse00229
K toocomplex 1
K comment
EnableButton("Up");ChangeButtonBinding("Up", "JumpId(`tex2rtf.hlp', `topic36')")

$#+KKK!**twocolitem:2**

Used to specify a row for a two column list, a Tex2RTF extension to optimize two-column lists for different file formats. See `\twocollist`, `\twocolitemruled`.

---

twocolitem:2
twocolitem
browse00230
K twocolitem 2
K twocollist
K twocolitemruled
EnableButton("Up");ChangeButtonBinding("Up", "JumpId(`tex2rtf.hlp', `topic36')")

$#+KKK!**twocolitemruled:2**

Used to specify a ruled row for a two column list, a Tex2RTF extension to optimize two-column lists for different file formats. See \twocollist, \twocolitem.

---

twocolitemruled:2
twocolitemruled
browse00231
K twocolitemruled 2
K twocollist
K twocolitem
EnableButton("Up");ChangeButtonBinding("Up", "JumpId(`tex2rtf.hlp', `topic36')")

$#+KKKKK!**twocollist:1**

A Tex2RTF environment for specifying a table of two columns, often used in manuals and help files (for example, for listing commands and their meanings). The first column should be one line only, and the second can be an arbitrary number of paragraphs.

The reason that a normal tabular environment cannot be used is that WinHelp does not allow borders in table cells, so a different method must be employed if any of the rows are to be ruled. In LaTeX, a table is used to implement this environment. In RTF, indentation is used instead.

Use this environment in conjunction with `\twocolitem` and `\twocolitemruled`. To set the widths of the first and second column, use `\twocolwidtha` and `\twocolwidthb`.

Example:

```
\htmlignore{\begin{twocollist}}
\twocolitemruled{{\bf Command}}{{\bf Description}}
\twocolitem{File}{The file menu is used to select various
file-related operations, such as saving and loading.}
\twocolitem{Edit}{The Edit menu is used for
selection, copying, pasting, etc.}
\end{twocollist}
```

This produces:

Command	**Description**
File	The file menu is used to select various file-related operations, such as saving and loading.
Edit	The Edit menu is used for selection, copying, pasting, etc.

---

twocollist:1
twocollist
browse00232
K twocollist 1
K twocolitem
K twocolitemruled
K twocolwidtha
K twocolwidthb
EnableButton("Up");ChangeButtonBinding("Up", "JumpId(`tex2rtf.hlp', `topic36')")

$#+KKK!**twocolwidtha:1**

Sets the width of the first column in a two column list to the given dimension. See also `\twocollist` and `\twocolwidthb`.

---

twocolwidtha:1
twocolwidtha
browse00233
K twocolwidtha 1
K twocollist
K twocolwidthb
EnableButton("Up");ChangeButtonBinding("Up", "JumpId(`tex2rtf.hlp', `topic36')")

$#+KKK!**twocolwidthb:1**

Sets the width of the second column in a two column list to the given dimension. See also `\twocollist` and `\twocolwidtha`.

---

twocolwidthb:1
twocolwidthb
browse00234
K twocolwidthb 1
K twocollist
K twocolwidtha
EnableButton("Up");ChangeButtonBinding("Up", "JumpId(`tex2rtf.hlp', `topic36')")

$#+K!**urlref:2**

Specifies a jump to a URL (univeral resource location).

The first argument is text to be highlighted (mouseable in HTML browsers) and the second is the URL. In linear documents, the URL is given following the text.

Example:

```
See also the \urlref{wxWidgets manual}
  {http://www.aiai.ed.ac.uk/~jacs.html}.
```

(the line is broken only to keep to this manual's page width).

---

uurlref:2
uurlref
browse00235
K urlref 2
EnableButton("Up");ChangeButtonBinding("Up", "JumpId(`tex2rtf.hlp', `topic36')")

$#+K!**winhelpignore:1**

Ignores the argument in WinHelp RTF.

---

winhelpignore:1
winhelpignore
browse00236
K winhelpignore 1
EnableButton("Up");ChangeButtonBinding("Up", "JumpId(`tex2rtf.hlp', `topic36')")

$#+K!**winhelponly:1**

Only outputs the argument in WinHelp RTF.

---

winhelponly:1
winhelponly
browse00237
K winhelponly 1
EnableButton("Up");ChangeButtonBinding("Up", "JumpId(`tex2rtf.hlp', `topic36')")

$#+K!**xlpignore:1**

Ignores the argument in XLP mode (wxHelp files).

---

xlpignore:1
xlpignore
browse00238
K xlpignore 1
EnableButton("Up");ChangeButtonBinding("Up", "JumpId(`tex2rtf.hlp', `topic36')")

$#+K!**xlponly:1**

Only outputs the argument in XLP mode (wxHelp files).

---

xlponly:1
xlponly
browse00239
K xlponly 1
EnableButton("Up");ChangeButtonBinding("Up", "JumpId(`tex2rtf.hlp', `topic36')")

$#+KKKKKKKKKKKKKKKKKKKKKKKKKKKKKKKKKKKK!**Font commands**

{bmc bullet.bmp} \bf

{bmc bullet.bmp} \bffamily

Font commands
topic38
browse00242
K Font commands
K bf
K bffamily
K em
K emph
K huge
K Huge
K HUGE
K it
K itshape
K large
K Large
K LARGE
K mdseries
K normalsize
K rm
K rmfamily
K sc
K scshape
K sf
K sffamily
K sl
K slshape
K small
K textbf
K textit
K textrm
K textsf
K textsc
K textsl
K texttt
K tiny
K tt
K ttfamily
K underline
K upshape
EnableButton("Up");ChangeButtonBinding("Up", "JumpId(`tex2rtf.hlp', `topic37')")

{bmc bullet.bmp} \em

{bmc bullet.bmp} \emph

{bmc bullet.bmp} \huge

{bmc bullet.bmp} \Huge

{bmc bullet.bmp} \HUGE

{bmc bullet.bmp} \it

{bmc bullet.bmp} \itshape

{bmc bullet.bmp} \large

{bmc bullet.bmp} \Large

{bmc bullet.bmp} \LARGE

{bmc bullet.bmp} \mdseries

{bmc bullet.bmp} \normalsize

{bmc bullet.bmp} \rm

{bmc bullet.bmp} \rmfamily

{bmc bullet.bmp} \sc

{bmc bullet.bmp} \scshape

{bmc bullet.bmp} \sf

{bmc bullet.bmp} \sffamily

{bmc bullet.bmp} \sl

{bmc bullet.bmp} \slshape

{bmc bullet.bmp} \small

{bmc bullet.bmp} \textbf

{bmc bullet.bmp} \textit

{bmc bullet.bmp} \textrm

{bmc bullet.bmp} \textsf

{bmc bullet.bmp} \textsc

{bmc bullet.bmp} \textsl

{bmc bullet.bmp} \texttt

{bmc bullet.bmp} \tiny

{bmc bullet.bmp} \tt

{bmc bullet.bmp} \ttfamily

{bmc bullet.bmp} \underline

{bmc bullet.bmp} \upshape

$#+KKKKKKKKKKKKKKKKKKKKK!**Paragraph formatting**

{bmc bullet.bmp} \centerline

{bmc bullet.bmp} \comment

{bmc bullet.bmp} \flushleft

{bmc bullet.bmp} \footnote

{bmc bullet.bmp} \indented

{bmc bullet.bmp} \marginparwidth

{bmc bullet.bmp} \marginpar

{bmc bullet.bmp} \marginpareven

{bmc bullet.bmp} \marginparodd

Paragraph formatting
topic39
browse00243
K Paragraph formatting
K centerline
K comment
K flushleft
K footnote
K indented
K marginparwidth
K marginpar
K marginpareven
K marginparodd
K multicolumn
K newpage
K noindent
K onecolumn
K parindent
K parskip
K par
K quote
K quotation
K textwidth
K twocolumn
K verbatim
K verb
EnableButton("Up");ChangeButtonBinding("Up", "JumpId(`tex2rtf.hlp', `topic37')")

{bmc bullet.bmp} \multicolumn

{bmc bullet.bmp} \newpage

{bmc bullet.bmp} \noindent

{bmc bullet.bmp} \onecolumn

{bmc bullet.bmp} \parindent

{bmc bullet.bmp} \parskip

{bmc bullet.bmp} \par

{bmc bullet.bmp} \quote

{bmc bullet.bmp} \quotation

{bmc bullet.bmp} \textwidth

{bmc bullet.bmp} \twocolumn

{bmc bullet.bmp} \verbatim

{bmc bullet.bmp} \verb

$#+KKKKKKKKKKKKKKKKKK!**Special effects**

{bmc bullet.bmp} \backgroundcolour

{bmc bullet.bmp} \backgroundimage

{bmc bullet.bmp} \backslashraw

{bmc bullet.bmp} \bcol

{bmc bullet.bmp} \definecolour

{bmc bullet.bmp} \fcol

{bmc bullet.bmp} \followedlinkcolour

{bmc bullet.bmp} \helpfontfamily

{bmc bullet.bmp} \helpfontsize

{bmc bullet.bmp} \hrule

{bmc bullet.bmp} \linkcolour

{bmc bullet.bmp} \normalbox

Special effects
topic40
browse00244
K Special effects
K backgroundcolour
K backgroundimage
K backslashraw
K bcol
K definecolour
K fcol
K followedlinkcolour
K helpfontfamily
K helpfontsize
K hrule
K linkcolour
K normalbox
K normalboxd
K sethotspotcolour
K sethotspotunderline
K settransparency
K textcolour
K typeout
EnableButton("Up");ChangeButtonBinding("Up", "JumpId(`tex2rtf.hlp', `topic37')")

{bmc bullet.bmp} \normalboxd

{bmc bullet.bmp} \sethotspotcolour

{bmc bullet.bmp} \sethotspotunderline

{bmc bullet.bmp} \settransparency

{bmc bullet.bmp} \textcolour

{bmc bullet.bmp} \typeout

$#+KKKKKKKKKK!**Lists**

{bmc bullet.bmp} \description

{bmc bullet.bmp} \enumerate

{bmc bullet.bmp} \itemize

{bmc bullet.bmp} \item

{bmc bullet.bmp} \itemsep

{bmc bullet.bmp} \twocolitem

{bmc bullet.bmp} \twocolitemruled

{bmc bullet.bmp} \twocollist

{bmc bullet.bmp} \twocolwidtha

{bmc bullet.bmp} \twocolwidthb

Lists
topic41
browse00245
K Lists
K description
K enumerate
K itemize
K item
K itemsep
K twocolitem
K twocolitemruled
K twocollist
K twocolwidtha
K twocolwidthb
EnableButton("Up");ChangeButtonBinding("Up", "JumpId(`tex2rtf.hlp', `topic37')")

$#+KKKKKKKKKKKKKKK!**Sectioning**

{bmc bullet.bmp} \chapter

{bmc bullet.bmp} \chapter*

{bmc bullet.bmp} \chapterheading

{bmc bullet.bmp} \insertatlevel

{bmc bullet.bmp} \paragraph

{bmc bullet.bmp} \paragraph*

{bmc bullet.bmp} \section

{bmc bullet.bmp} \section*

{bmc bullet.bmp} \sectionheading

{bmc bullet.bmp} \subparagraph

{bmc bullet.bmp} \subparagraph*

{bmc bullet.bmp} \subsection

{bmc bullet.bmp} \subsection*

Sectioning
topic42
browse00246
K Sectioning
K chapter
K chapter*
K chapterheading
K insertatlevel
K paragraph
K paragraph*
K section
K section*
K sectionheading
K subparagraph
K subparagraph*
K subsection
K subsection*
K subsubsection
K subsubsection*
EnableButton("Up");ChangeButtonBinding("Up", "JumpId(`tex2rtf.hlp', `topic37')")

{bmc bullet.bmp} `\subsubsection`

{bmc bullet.bmp} `\subsubsection*`

$#+KKKKKKK!**Pictures**

{bmc bullet.bmp} \brclear

{bmc bullet.bmp} \image

{bmc bullet.bmp} \imagel

{bmc bullet.bmp} \imagemap

{bmc bullet.bmp} \imager

{bmc bullet.bmp} \psboxto

Pictures
topic43
browse00247
K Pictures
K brclear
K image
K imagel
K imagemap
K imager
K psboxto
EnableButton("Up");ChangeButtonBinding("Up", "JumpId(`tex2rtf.hlp', `topic37')")

$#+KKKKKKKKK!**References and jumps**

{bmc bullet.bmp} \footnotepopup

{bmc bullet.bmp} \helpref

{bmc bullet.bmp} \helprefn

{bmc bullet.bmp} \label

{bmc bullet.bmp} \pageref

{bmc bullet.bmp} \popref

{bmc bullet.bmp} \ref

{bmc bullet.bmp} \urlref

R eferences and jumps
t opic44
b rowse00248
K References and jumps
K footnotepopup
K helpref
K helprefn
K label
K pageref
K popref
K ref
K urlref
E nableButton("Up");ChangeButtonBinding("Up", "JumpId(`tex2rtf.hlp', `topic37')")

$#+KKKKKKK!**Tables and figures**

{bmc bullet.bmp} \caption

{bmc bullet.bmp} \figure

{bmc bullet.bmp} \hline

{bmc bullet.bmp} \ruledrow

{bmc bullet.bmp} \tabbing

{bmc bullet.bmp} \tabular

Tables and figures
topic45
browse00249
K Tables and figures
K caption
K figure
K hline
K ruledrow
K tabbing
K tabular
EnableButton("Up");ChangeButtonBinding("Up", "JumpId(`tex2rtf.hlp', `topic37')")

$#+KKKKKKK!**Table of contents**

{bmc bullet.bmp} \addcontentsline

{bmc bullet.bmp} \author

{bmc bullet.bmp} \date

{bmc bullet.bmp} \maketitle

{bmc bullet.bmp} \tableofcontents

{bmc bullet.bmp} \title

Table of contents
topic46
browse00250
K Table of contents
K addcontentsline
K author
K date
K maketitle
K tableofcontents
K title
EnableButton("Up");ChangeButtonBinding("Up", "JumpId(`tex2rtf.hlp', `topic37')")

$#+KKKKKKKKKKK!**Special sections**

{bmc bullet.bmp} \bibitem

{bmc bullet.bmp} \bibliographystyle

{bmc bullet.bmp} \bibliography

{bmc bullet.bmp} \cite

{bmc bullet.bmp} \gloss

{bmc bullet.bmp} \helpglossary

{bmc bullet.bmp} \index

{bmc bullet.bmp} \nocite

{bmc bullet.bmp} \printindex

{bmc bullet.bmp} \shortcite

{bmc bullet.bmp} \thebibliography

Special sections
topic47
browse00251
K Special sections
K bibitem
K bibliographystyle
K bibliography
K cite
K gloss
K helpglossary
K index
K nocite
K printindex
K shortcite
K thebibliography
EnableButton("Up");ChangeButtonBinding("Up", "JumpId(`tex2rtf.hlp', `topic37')")

$#+KKKKKKKKKKKKKK!**Symbols**

{bmc bullet.bmp} \backslash

{bmc bullet.bmp} \cdots

{bmc bullet.bmp} \cextract

{bmc bullet.bmp} \cinsert

{bmc bullet.bmp} \copyright

{bmc bullet.bmp} \LaTeX

{bmc bullet.bmp} \lbraceraw

{bmc bullet.bmp} \ldots

{bmc bullet.bmp} \rbraceraw

{bmc bullet.bmp} \registered

{bmc bullet.bmp} \rtfsp

{bmc bullet.bmp} \ss

{bmc bullet.bmp} \TeX

{bmc bullet.bmp} \today

Symbols
topic48
browse00252
K Symbols
K backslash
K cdots
K cextract
K cinsert
K copyright
K LaTeX
K lbraceraw
K ldots
K rbraceraw
K registered
K rtfsp
K ss
K TeX
K today
EnableButton("Up");ChangeButtonBinding("Up", "JumpId(`tex2rtf.hlp', `topic37')")

$#+KKKKKKKKKKKKKKKKKKKKKKKKK!**Document organisation**

{bmc bullet.bmp} \document

{bmc bullet.bmp} \documentstyle

{bmc bullet.bmp} \helpignore

{bmc bullet.bmp} \helponly

{bmc bullet.bmp} \helpinput

{bmc bullet.bmp} \htmlignore

{bmc bullet.bmp} \htmlonly

{bmc bullet.bmp} \include

Document organisation
topic49
browse00253
K Document organisation
K document
K documentstyle
K helpignore
K helponly
K helpinput
K htmlignore
K htmlonly
K include
K input
K latexignore
K latexonly
K newcommand
K pagestyle
K pagenumbering
K rtfignore
K rtfonly
K setfooter
K setheader
K special
K toocomplex
K verbatiminput
K winhelpignore
K winhelponly
K xlpignore
K xlponly
EnableButton("Up");ChangeButtonBinding("Up", "JumpId(`tex2rtf.hlp', `topic37')")

{bmc bullet.bmp} \input

{bmc bullet.bmp} \latexignore

{bmc bullet.bmp} \latexonly

{bmc bullet.bmp} \newcommand

{bmc bullet.bmp} \pagestyle

{bmc bullet.bmp} \pagenumbering

{bmc bullet.bmp} \rtfignore

{bmc bullet.bmp} \rtfonly

{bmc bullet.bmp} \setfooter

{bmc bullet.bmp} \setheader

{bmc bullet.bmp} \special

{bmc bullet.bmp} \toocomplex

{bmc bullet.bmp} \verbatiminput

{bmc bullet.bmp} \winhelpignore

{bmc bullet.bmp} \winhelponly

{bmc bullet.bmp} \xlpignore

{bmc bullet.bmp} \xlponly

$#+KK!**Macro not found**

This error may indicate that Tex2RTF has not implemented a standard LaTeX command, or that a local macro package is being used that Tex2RTF does not know about. It can cause spurious secondary errors, such as not recognising the end document command.

You can get round this by defining a macro file (default name `tex2rtf.ini`) containing command definitions, such as:

```
\crazy      [2]{{\bf #2} is crazy but #1 is not}
\something  [0]{}
\julian     [0]{Julian Smart}
```

New commands may be defined in LaTeX files, but custom macro files will have to be defined when local style files are being used. See Initialisation file syntax for further details.

The 'Macro not found' error can also be caused by a syntax error such as an unbalanced brace or passing the wrong number of arguments to a command, so look in the vicinity of the reported error for the real cause.

Here is one obscure situation that causes this error:

```
\begin{center}
{\large{\underline{A}}}
\end{center}
```

The problem is too many curly brackets. This should be rewritten as:

```
\begin{center}
{\large \underline{A}}
\end{center}
```

Often you get a 'Macro not found' error for `\end{document}`. This is a spurious side-effect of an earlier error, usually an incorrect number of arguments to a command. The location of the true error is then anywhere in the document. To home in on the error, try putting a verbatim environment `\begin{comment}...\end{comment}` around much of the document, and then move the `\begin{comment}` line down until the error manifests itself.

[M]acro not found
[m]acronotfound
[b]rowse00257
[K] Macro not found
[K] macro not found error
[E]nableButton("Up");ChangeButtonBinding("Up", "JumpId(`tex2rtf.hlp', `topic51')")

$#+KK!**Unresolved reference**

References and citations are usually resolved on a second pass of Tex2RTF. If this doesn't work, then a missing label or bibliographical entry is to blame.

Unresolved reference
topic52
browse00258
K Unresolved reference
K references, unresolved
EnableButton("Up");ChangeButtonBinding("Up", "JumpId(`tex2rtf.hlp', `topic51')")

$#+K!Output crashes the RTF reader

This could be due to confusing table syntax. Set *compatibility* to *TRUE* in `.ini` file; also check for end of row characters backslash characters on their own on a line, and insert correct number of ampersands for the number of columns. E.g.

```
hello & world\\
\\
```

becomes

```
hello & world\\
&\\
```

Output crashes the RTF reader
topic53
browse00259
K Output crashes the RTF reader
EnableButton("Up");ChangeButtonBinding("Up", "JumpId(`tex2rtf.hlp', `topic51')")

$#+K!**Erratic list indentation**

Try increasing the value of the variable *listItemIndent* (default 40 points) to give more space between label and following text. A global replacement of `\item [` with `\item[` may also be helpful to remove unnecessary space before the item label.

Erratic list indentation
topic54
browse00260
K Erratic list indentation
EnableButton("Up");ChangeButtonBinding("Up", "JumpId(`tex2rtf.hlp', `topic51')")

$#+K!**Missing figure or section reference**

Ensure all labels *directly* follow captions or sections (no intervening white space).

Missing figure or section reference
topic55
browse00261
K Missing figure or section reference
EnableButton("Up");ChangeButtonBinding("Up", "JumpId(`tex2rtf.hlp', `topic51')")

$#+K!**Linear RTF looks odd**

For viewing by programs other than MS Word, you should set the variable *useWord* to *false*. This will turn off some of the special RTF keywords recognised by Word (and possibly other advanced RTF readers).

Linear RTF looks odd
topic56
browse00262
K Linear RTF looks odd
EnableButton("Up");ChangeButtonBinding("Up", "JumpId(`tex2rtf.hlp', `topic51')")

$#+K!**Paragraphs preceding lists are formatted weirdly.**

If a list has spurious spacing in it, e.g. before a `\item` command, the preceding paragraph can take on some of the list's indentation. This may be a WinHelp bug, or an aspect of RTF I don't fully understand. The solution is to remove unnecessary space.

Paragraphs preceding lists are formatted weirdly.
topic57
browse00263
K Paragraphs preceding lists are formatted weirdly.
EnableButton("Up");ChangeButtonBinding("Up", "JumpId(`tex2rtf.hlp', `topic51')")

$#+KK!**Unresolved references in Word for Windows**

If question marks appear instead of numbers for figures and tables, select all (e.g. CTRL-A), then press F9 *twice* to reformat the document twice. For the second format, respond with *Update Entire Table* to any prompts.

Unresolved references in Word for Windows
topic58
browse00264
K Unresolved references in Word for Windows
K Microsoft Word
EnableButton("Up");ChangeButtonBinding("Up", "JumpId(`tex2rtf.hlp', `topic51')")

$#+KK!**The Windows 95 help file contents hierarchy looks wrong**

WinHelp version 4 (or the WIN32 Help Compiler) does not allow a book in the contents list to be followed by a page at the same level. A book must be followed by a book, for some strange reason, otherwise the page will be tacked on to the pages of the book above it, i.e. placed at the wrong level.

To get around this, Tex2RTF inserts a book in some places, if there was a book preceding it on the same level. This results in more navigation than necessary, but is better than a wrong contents page.

Tthe Windows 95 help file contents hierarchy looks wrong
topic59
browse00265
K The Windows 95 help file contents hierarchy looks wrong
K WinHelp files
EnableButton("Up");ChangeButtonBinding("Up", "JumpId(`tex2rtf.hlp', `topic51')")

$#+KKKKKKK!**General options**

Option	Description
compatibility	Set to true for maximum LaTeX compatibility, e.g. if tables crash RTF readers. Should be false (default) if the Tex2RTF guidelines are followed, e.g. use of `\row` command in tabular environment.
conversionMode	One of RTF, WinHelp, XLP (or wxHelp), and HTML.
ignoreInput	Adds the filename to the list of files ignored by the `\input` command. The only default filename in the list is `psbox.tex`.
isInteractive	If true, runs in interactive mode (the default).
runTwice	If true, runs the converter twice.
ignoreBadRefs	If true (or yes), ignores bad helpref references and simply writes the text in the first argument. Useful when a program such as HelpGen generates references to classes documented in another manual.

[G]eneral options
[g]eneraloptions
[b]rowse00016
[K] General options
[K] compatibility
[K] conversionMode
[K] ignoreInput
[K] isInteractive
[K] runTwice
[K] ignoreBadRefs
[E]nableButton("Up");ChangeButtonBinding("Up", "JumpId(`tex2rtf.hlp', `topic8')")

$#+KKKKKKKKKKKKKKKKKKKK!**Presentation options**

Option	Description
authorFontSize	Specifies the point size for the author and date (RTF only).
chapterFontSize	Specifies the point size for chapter headings (RTF only).
documentFontSize	One of 10, 11 and 12, to specify the main font size independently of the LaTeX document style command.
sectionFontSize	Specifies the point size for section headings (RTF only).
subsectionFontSize	Specifies the point size for subsection headings (RTF only).
titleFontSize	Specifies the point size for the title (RTF only).
chapterName	The string used when referencing chapters. The default is "chapter".
sectionName	The string used when referencing sections. The default is "section".

Presentation options
topic9
browse00017
K Presentation options
K options, presentation
K authorFontSize
K chapterFontSize
K documentFontSize
K sectionFontSize
K subsectionFontSize
K titleFontSize
K chapterName
K sectionName
K subsectionName
K subsubsectionName
K indexName
K contentsName
K abstractName
K tablesName
K tableName
K figuresName
K figureName
K glossaryName
K referencesName
EnableButton("Up");ChangeButtonBinding("Up", "JumpId(`tex2rtf.hlp', `topic8')")

subsectionName	The string used when referencing subsections. The default is "subsection".
subsubsectionName	The string used when referencing subsubsections. The default is "subsubsection".
indexName	The string used for printing the index heading. The default is "Index".
contentsName	The string used for printing the contents heading. The default is "Contents".
abstractName	The string used for printing the abstract heading. The default is "Abstract".
tablesName	The string used for printing the list of tables heading. The default is "List of Tables".
tableName	The string used when referencing a table. The default is "table".
figuresName	The string used for printing the list of figures heading. The default is "List of Figures".
figureName	The string used when referencing a figure. The default is "figure".
glossaryName	The string used for printing the glossary heading. The default is "Glossary".
referencesName	The string used for printing the references heading. The default is "References".

$#+KKKKKKKKKKKKKKKKKK!**RTF and WinHelp options**

Option	Description
bitmapMethod	Can be "hex" (embed the hex data in the file with a \dibitmap keyword), "includepicture" (use the MS Word 6.0 INCLUDEPICTURE field) or "import" (an earlier name for INCLUDEPICTURE). "hex" may be used for importing into MS Works, but this doesn't work for Word 6.0. The default is "includepicture".
contentsDepth	The depth of headings that is displayed in the table of contents. The default is 4 but you may wish to reduce this, for example for manuals that document C++ and have a large number of headings for member functions.
defaultColumnWidth	The width in points for columns in tables where the width of the column is not set by using *p* in the tabular argument. The default is 100.
footerRule	If true, draws a rule above footers (linear RTF only).
generateHPJ	If true, generates a .HPJ project file (WinHelp mode only).

RTF and WinHelp options
rtfwinhelpoptions
browse00018
K RTF and WinHelp options
K options, RTF
K RTF
K bitmapMethod
K contentsDepth
K defaultColumnWidth
K footerRule
K generateHPJ
K headerRule
K listLabelIndent
K listItemIndent
K indexSubsections
K mirrorMargins
K useWord
K useHeadingStyles
K useUpButton
K winHelpContents
K winHelpVersion
K winHelpTitle
EnableButton("Up");ChangeButtonBinding("Up", "JumpId(`tex2rtf.hlp', `topic8')")

headerRule	If true, draws a rule below headers (linear RTF only).
listLabelIndent	Specifies the size of list item label indentation, in points. The default is 18.
listItemIndent	Specifies the size of list item indentation, in points. The default is 40.
indexSubsections	If true (the default), subsection and subsubsection titles are indexed in RTF mode.
mirrorMargins	If true, margins are mirrored in twosided documents (linear RTF only).
useWord	If true (the default), Word for Windows RTF formatting is used where possibly, e.g. for the table of contents, list of tables, and list of figures.
useHeadingStyles	If true (the default), sections are marked with appropriate heading styles for generating the table of contents in RTF.
useUpButton	If true (the default), WinHelp files will be generated with an **Up** button to make browsing easier. Note that you need to put an extra line in the CONFIG section of your .HPJ file: `CreateButton("Up", "&Up", "JumpId('name.hlp', 'Contents')")` where `name.hlp` is the name of your help file.
winHelpContents	If yes, ok or true, a WinHelp `.cnt` file will be generated (used in Windows 95 for either old WinHelp files or new WinHelp 4 files).
winHelpVersion	The version of WinHelp being targetted. This affects the generated `.hpj` file and features such as transparent bitmaps which are new to version 4 or later. The default is 3.
winHelpTitle	Windows Help file title, inserted into the project file if *generateHPJ* is true.

$#+KKKKKKKKKKKKKK!**HTML options**

Option	Description
htmlBrowseButtons	Allows generation of Contents, Up, browse back and browse forward buttons on each HTML page except title page. Specify none, text or bitmap. If you specify bitmap, make sure that the files `contents.gif`, `up.gif`, `back.gif` and `forward.gif` are in the directory where the HTML files will reside: samples are given in the docs directory.
truncateFilenames	If true, uses `.htm` suffix instead of `.html`, and truncates filenames within HTML documents.
htmlIndex	If true, specifies generation of an `.htx` index file for an HTML document. This file can be used in wxHelp version 2 or other programs. The file consists of a number of lines, each line with three fields separated by bar characters: the indexed phrase, the file, and a label in the file.
htmlWorkshopFiles	If true, specifies generation of `.hpp, .hhc` and `.hhk` files which can be used to create both MS HTML Help and wxHTML Help files. wxHTML Help is the HTML help facility that can be used by wxWidgets 2 applications (see the wxWidgets manual and the wxWidgets HTML sample).
upperCaseNames	If true, filenames in links are in upper case. By default

---

HTML options
htmloptions
browse00019
K HTML options
K options, HTML
K HTML
K htmlBrowseButtons
K truncateFilenames
K htmlIndex
K htmlWorkshopFiles
K upperCaseNames
K backgroundColour
K backgroundImage
K textColour
K linkColour
K followedLinkColour
K combineSubSections
K htmlFaceName
EnableButton("Up");ChangeButtonBinding("Up", "JumpId(`tex2rtf.hlp', `topic8')")

	filenames are in lower case.
backgroundColour	Specifies the RGB background colour for the document, e.g. `255;255;255` for white. The default is white.
backgroundImage	Specifies the RGB background image for the document, e.g. `tile.gif`.
textColour	Specifies the RGB text colour for the document, e.g. `0;0;0` for black.
linkColour	Specifies the RGB link colour for the document, e.g. `0;0;255` for blue.
followedLinkColour	Specifies the RGB followed link colour for the document, e.g. `0;0;255` for blue.
combineSubSections	If true (or yes), switches off the generation of separate HTML files below section level. This can reduce the number of HTML files substantially. A subsection contents list is inserted before the first subsection.
htmlFaceName	A string specifying the overall font face, such as ""Arial, Lucida, Helvetica".