#[1]$[2]+[3]K[4]![5]

Manual for Tex2RTF 1.40

by Julian Smart, Artificial Intelligence Applications Institute,University of Edinburgh

{bmc tex2rtf.wmf}

Contents

1Contents
2Contents
3browse00001
4Contents
5DisableButton("Up")

$6#7+8K9!10Copyright notice

6Copyright notice
7topic0
8browse00002
9Copyright notice
10EnableButton("Up");ChangeButtonBinding("Up", "JumpId(`TEX2RTF.hlp', `Contents')")

$11#12+13K14!15Introduction

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.

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

The latest version of Tex2RTF can be accessed by anonymous ftp from:

```
skye.aiai.ed.ac.uk (192.41.104.6)
```

in the directory /pub/tex2rtf. It is available in SPARC Open Look and Windows 3.1 versions.

Tex2RTF was developed using the free Open Look, Motif and Windows 3.1 C++ class library wxWindows, also available from the above FTP site in the /pub/wxwin/beta directory.

11Introduction
12topic1
13browse00003
14Introduction
15EnableButton("Up");ChangeButtonBinding("Up", "JumpId(`TEX2RTF.hlp', `Contents')")

$16#17+18K19!20Hypertext 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. This 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 macro 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`.

16Hypertext features
17topic13
18browse00017
19Hypertext features
20EnableButton("Up");ChangeButtonBinding("Up", "JumpId(`TEX2RTF.hlp', `Contents')")

$21#22+23K24!25Special sections

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

Bibliography
Glossary
Index

21Special sections
22topic14
23browse00018
24Special sections
25EnableButton("Up");ChangeButtonBinding("Up", "JumpId(`TEX2RTF.hlp', `Contents')")

$[26]#[27]+[28]K[29]![30]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 macro, 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 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.

26Running Tex2RTF
27topic16
28browse00022
29Running Tex2RTF
30EnableButton("Up");ChangeButtonBinding("Up", "JumpId(`TEX2RTF.hlp', `Contents')")

$31#32+33K34!35Tex2RTF 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.

31Tex2RTF for non-LaTeX users
32topic24
33browse00031
34Tex2RTF for non-LaTeX users
35EnableButton("Up");ChangeButtonBinding("Up", "JumpId(`TEX2RTF.hlp', `Contents')")

$[36]#[37]+[38]K[39]![40]Macro reference

The following lists macros which are recognised by the converters. Other macros not not mentioned can be assumed to be 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}
```


36Macro reference
37topic29
38browse00036
39Macro reference
40EnableButton("Up");ChangeButtonBinding("Up", "JumpId(`TEX2RTF.hlp', `Contents')")

$41#42+43K44!45Bugs and troubleshooting

Bugs
Troubleshooting

41Bugs and troubleshooting
42errors
43browse00190
44Bugs and troubleshooting
45EnableButton("Up");ChangeButtonBinding("Up", "JumpId(`TEX2RTF.hlp', `Contents')")

$[46]#[47]+[48]K[49]![50]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] **Robins, Gabriel**. 1987 (September). *The ISI grapher: a portable tool for displaying graphs pictorially (ISI/RS-87-196).* Technical report. University of South California.

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

46References
47topic38
48browse00199
49References
50EnableButton("Up");ChangeButtonBinding("Up", "JumpId(`TEX2RTF.hlp', `Contents')")

$51#52+53K54!55Glossary

GUI
HTML
LaTeX
Metafile
Open Look
RTF
wxHelp
wxWindows
XView

51Glossary
52topic39
53browse00200
54Glossary
55EnableButton("Up");ChangeButtonBinding("Up", "JumpId(`TEX2RTF.hlp', `Contents')")

$[56]#[57]+[58]K[59]![60]Status of Tex2RTF

Tex2RTF is under continual development, often following users' suggestions. From version 1.33, Tex2RTF is effectively in a second phase of development. In addition to the bare minimum of syntax and facilities for producing useable help systems or linear RTF, commands are being added to allow visually effective, even aesthetically pleasing, documentation to be produced.

Examples are the *indented*, *twocollist* and *marginpar* commands; over time I hope to be able to reproduce most of the popular styles of formatting and presentation in Windows Help files, whilst allowing a reasonable equivalent to be generated in the other formats.

56Status of Tex2RTF
57topic2
58browse00004
59Status of Tex2RTF
60EnableButton("Up");ChangeButtonBinding("Up", "JumpId(`TEX2RTF.hlp', `topic1')")

$[61]#[62]+[63]K[64]![65]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.

61Why use LaTeX?
62topic3
63browse00005
64Why use LaTeX?
65EnableButton("Up");ChangeButtonBinding("Up", "JumpId(`TEX2RTF.hlp', `topic1')")

$66#67+68K69!70Help 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 docmentation 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 [1] is highly recommended.

66Help versus the printed page
67topic4
68browse00006
69Help versus the printed page
70EnableButton("Up");ChangeButtonBinding("Up", "JumpId(`TEX2RTF.hlp', `topic1')")

$[71]#[72]+[73]K[74]![75]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} wxHelp. This is the platform-independent help system for the class library wxWindows [2]. It can display ASCII files with embedded codes for changing font styles, but no formatting is done by wxHelp.

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

71Output Formats
72topic5
73browse00007
74Output Formats
75EnableButton("Up");ChangeButtonBinding("Up", "JumpId(`TEX2RTF.hlp', `topic1')")

$[76]#[77]+[78]K[79]![80]What compromises must I make?

As a LaTeX user, you need to be aware that some macros 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 not at all to the other formats) but an attempt is made to approximate tables so long as special row macros 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* macro 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 (input, include, verbatiminput), is handled, 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. Macros 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 macro (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, and an additonal 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. The class library wxWindows may be extended in future to allow using a better help viewer, such as *xmosaic*. Of course there is nothing to stop xmosaic being used as a help system, but it won't be integrated with wxWindows programs as wxHelp is.

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.

76What compromises must I make?
77topic6
78browse00008
79What compromises must I make?
80EnableButton("Up");ChangeButtonBinding("Up", "JumpId(`TEX2RTF.hlp', `topic1')")

$[81]#[82]+[83]K[84]![85]Changes to LaTeX syntax

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

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.

81Changes to LaTeX syntax
82topic7
83browse00009
84Changes to LaTeX syntax
85EnableButton("Up");ChangeButtonBinding("Up", "JumpId(`TEX2RTF.hlp', `topic1')")

$[86]#[87]+[88]K[89]![90]Tex2RTF change log

Changes in 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

Changes in 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.

Changes in 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.

Changes in 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.

Changes in 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.

Changes in 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.

86Tex2RTF change log
87topic12
88browse00016
89Tex2RTF change log
90EnableButton("Up");ChangeButtonBinding("Up", "JumpId(`TEX2RTF.hlp', `topic1')")

Changes in 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.

Changes in version 1.32:

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

Changes in version 1.31:

{bmc bullet.bmp} *footnote* command now supported, in linear RTF only.
{bmc bullet.bmp} Added -bufsize option, for converting large documents.

Changes in 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.

Changes in 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.

$[91]#[92]+[93]K[94]![95]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.

91Bibliography
92bibsection
93browse00019
94Bibliography
95EnableButton("Up");ChangeButtonBinding("Up", "JumpId(`TEX2RTF.hlp', `topic14')")

$[96]#[97]+[98]K[99]![100]# 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.

96Glossary
97glossarysection
98browse00020
99Glossary
100EnableButton("Up");ChangeButtonBinding("Up", "JumpId(`TEX2RTF.hlp', `topic14')")

$[101]#[102]+[103]K[104]![105]**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.

101Index
102topic15
103browse00021
104Index
105EnableButton("Up");ChangeButtonBinding("Up", "JumpId(`TEX2RTF.hlp', `topic14')")

$[106]#[107]+[108]K[109]![110]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.

106Command line arguments
107topic17
108browse00023
109Command line arguments
110EnableButton("Up");ChangeButtonBinding("Up", "JumpId(`TEX2RTF.hlp', `topic16')")

$[111]#[112]+[113]K[114]![115]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 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

111Initialisation file syntax
112inifile
113browse00024
114Initialisation file syntax
115EnableButton("Up");ChangeButtonBinding("Up", "JumpId(`TEX2RTF.hlp', `topic16')")

$[116]#[117]+[118]K[119]![120]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.

116DDE commands
117topic23
118browse00030
119DDE commands
120EnableButton("Up");ChangeButtonBinding("Up", "JumpId(`TEX2RTF.hlp', `topic16')")

$[121]#[122]+[123]K[124]![125]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.

121What is LaTeX?
122topic25
123browse00032
124What is LaTeX?
125EnableButton("Up");ChangeButtonBinding("Up", "JumpId(`TEX2RTF.hlp', `topic24')")

$126#127+128K129!130Document 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.

126Document structure
127topic26
128browse00033
129Document structure
130EnableButton("Up");ChangeButtonBinding("Up", "JumpId(`TEX2RTF.hlp', `topic24')")

$[131]#[132]+[133]K[134]![135]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.

131Command syntax
132topic27
133browse00034
134Command syntax
135EnableButton("Up");ChangeButtonBinding("Up", "JumpId(`TEX2RTF.hlp', `topic24')")

$[136]#[137]+[138]K[139]![140]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.

To be continued!

136Space
137topic28
138browse00035
139Space
140EnableButton("Up");ChangeButtonBinding("Up", "JumpId(`TEX2RTF.hlp', `topic24')")

$[141]#[142]+[143]K[144]![145]Commands

141Commands
142topic30
143browse00037
144Commands
145EnableButton("Up");ChangeButtonBinding("Up", "JumpId(`TEX2RTF.hlp', `topic29')")

helprefn:2
hline:0
hrule:0
htmlignore:1
htmlonly:1
huge:1
Huge:1
HUGE:1
include:1
indented:2
input:1
itemize:1
item:0
itemsep:0
image:2
index:1
it:1
label:1
large:1
Large:1
LARGE:1
LaTeX:0
latexignore:1
latexonly:1
ldots
maketitle
marginparwidth:1
marginpar:1
marginpareven:1
marginparodd:1
membersection:1
member:1
multicolumn:3
newcommand:3
newpage:0
nocite:1
noindent:0
normalbox:1
normalboxd:1
normalsize:1
onecolumn:0
pageref:1
pagestyle:1
pagenumbering:1
paragraph*
paragraph
param:1
parindent:1
parskip:1
par:0
printindex
popref:2
psboxto:2
quote:1
quotation:1
ref:1

rm:1
row:1
ruledrow:1
rtfignore:1
rtfonly:1
rtfsp:0
sc:1
section*:1
section:1
sectionheading:1
setfooter:6
setheader:6
shortcite:1
small:1
special:1
ss:0
subparagraph*:1
subparagraph:1
subsection*:1
subsection:1
subsubsection*:1
subsubsection:1
tabbing:1
tableofcontents:0
tabular:2
TeX:0
textwidth:1
thebibliography:1
title:1
tiny:1
today:0
toocomplex:1
tt:1
typeout:1
twocolitem:2
twocolitemruled:2
twocollist:1
twocolwidtha:1
twocolwidthb:1
twocolumn:0
underline:1
verbatiminput:1
verbatim:1
verb
winhelpignore:1
winhelponly:1
xlpignore:1
xlponly:1

$146#147+148K149!150Accents

The following LaTeX accents work for RTF 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

146Accents
147comments
148browse00189
149Accents
150EnableButton("Up");ChangeButtonBinding("Up", "JumpId(`TEX2RTF.hlp', `topic29')")

$[151]#[152]+[153]K[154]![155]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 macro 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.

151Bugs
152topic31
153browse00191
154Bugs
155EnableButton("Up");ChangeButtonBinding("Up", "JumpId(`TEX2RTF.hlp', `errors')")

$[156]#[157]+[158]K[159]![160]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.

156Troubleshooting
157topic32
158browse00192
159Troubleshooting
160EnableButton("Up");ChangeButtonBinding("Up", "JumpId(`TEX2RTF.hlp', `errors')")

$[161]#[162]+[163]K[164]![165]GUI

Graphical User Interface, such as Windows 3 or X.

161GUI
162topic40
163browse00201
164GUI
165EnableButton("Up");ChangeButtonBinding("Up", "JumpId(`TEX2RTF.hlp', `topic39')")

$[166]#[167]+[168]K[169]![170]**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.

166HTML
167html
168browse00202
169HTML
170EnableButton("Up");ChangeButtonBinding("Up", "JumpId(`TEX2RTF.hlp', `topic39')")

$[171]#[172]+[173]K[174]![175]LaTeX

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

171LaTeX
172latexgloss
173browse00203
174LaTeX
175EnableButton("Up");ChangeButtonBinding("Up", "JumpId(`TEX2RTF.hlp', `topic39')")

$[176]#[177]+[178]K[179]![180]**Metafile**

Microsoft Windows-specific object which may contain a restricted set of GDI primitives. It is device independent, since it may be scaled without losing precision, unlike a bitmap. A metafile may exist in a file or in memory. wxWindows implements enough metafile functionality to use it to pass graphics to other applications via the clipboard. A placeable metafile is a metafile with a 22-byte header which can be imported into several Windows applications, and is a format used by the Microsoft help compiler.

176Metafile
177topic41
178browse00204
179Metafile
180EnableButton("Up");ChangeButtonBinding("Up", "JumpId(`TEX2RTF.hlp', `topic39')")

$[181]#[182]+[183]K[184]![185]Open Look

A specification for a GUI 'look and feel', initiated by Sun Microsystems. XView is one toolkit for writing Open Look applications under X, and wxWindows sits on top of XView.

181Open Look
182topic42
183browse00205
184Open Look
185EnableButton("Up");ChangeButtonBinding("Up", "JumpId(`TEX2RTF.hlp', `topic39')")

$[186]#[187]+[188]K[189]![190]**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.

186RTF
187rtf
188browse00206
189RTF
190EnableButton("Up");ChangeButtonBinding("Up", "JumpId(`TEX2RTF.hlp', `topic39')")

$[191]#[192]+[193]K[194]![195]**wxHelp**

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

191wxHelp
192wxhelp
193browse00207
194wxHelp
195EnableButton("Up");ChangeButtonBinding("Up", "JumpId(`TEX2RTF.hlp', `topic39')")

$[196]#[197]+[198]K[199]![200]wxWindows

wxWindows 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.

196wxWindows
197wxwindows
198browse00208
199wxWindows
200EnableButton("Up");ChangeButtonBinding("Up", "JumpId(`TEX2RTF.hlp', `topic39')")

$[201]#[202]+[203]K[204]![205]XView

An X toolkit supplied by Sun Microsystems, initially just for porting SunView applications to X, but which has become a popular toolkit in its own right due to its simplicity of use. XView implements Sun's Open Look 'look and feel' for X, but is not the only toolkit to do so.

201XView
202topic43
203browse00209
204XView
205EnableButton("Up");ChangeButtonBinding("Up", "JumpId(`TEX2RTF.hlp', `topic39')")

$206#207+208K209!210Space

Tex2RTF attempts to insert spaces where LaTeX assumes whitespace. However, for the benefit of RTF conversion, you need to use the *\rtfsp* macro 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 that begin
\rtfsp {\it 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.

206Space
207space
208browse00010
209Space
210EnableButton("Up");ChangeButtonBinding("Up", "JumpId(`TEX2RTF.hlp', `topic7')")

$[211]#[212]+[213]K[214]![215]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 seperated 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}
```

211Command arguments
212topic8
213browse00011
214Command arguments
215EnableButton("Up");ChangeButtonBinding("Up", "JumpId(`TEX2RTF.hlp', `topic7')")

$216#217+218K219!220Avoid the setlength macro

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

```
\parindent 0pt
```

instead of

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

216Avoid the setlength macro
217topic9
218browse00012
219Avoid the setlength macro
220EnableButton("Up");ChangeButtonBinding("Up", "JumpId(`TEX2RTF.hlp', `topic7')")

$221#222+223K224!225Units

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.

221Units
222topic10
223browse00013
224Units
225EnableButton("Up");ChangeButtonBinding("Up", "JumpId(`TEX2RTF.hlp', `topic7')")

$[226]#[227]+[228]K[229]![230]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.

226Labels
227topic11
228browse00014
229Labels
230EnableButton("Up");ChangeButtonBinding("Up", "JumpId(`TEX2RTF.hlp', `topic7')")

$231#232+233K234!235Tables

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.

231Tables
232tables
233browse00015
234Tables
235EnableButton("Up");ChangeButtonBinding("Up", "JumpId(`TEX2RTF.hlp', `topic7')")

$[236]#[237]+[238]K[239]![240]Tex2RTF options

These are the allowable options in an initialisation file.

236Tex2RTF options
237topic18
238browse00025
239Tex2RTF options
240EnableButton("Up");ChangeButtonBinding("Up", "JumpId(`TEX2RTF.hlp', `inifile')")

$[241]#[242]+[243]K[244]![245]**abstract:1**

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

241abstract:1
242abstract
243browse00038
244abstract:1
245EnableButton("Up");ChangeButtonBinding("Up", "JumpId(`TEX2RTF.hlp', `topic30')")

$[246]#[247]+[248]K[249]![250]addcontentsline:3

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

246addcontentsline:3
247addcontentsline
248browse00039
249addcontentsline:3
250EnableButton("Up");ChangeButtonBinding("Up", "JumpId(`TEX2RTF.hlp', `topic30')")

$[251]#[252]+[253]K[254]![255]**author:1**

Defines the author, for output when *maketitle* is used.

251author:1
252author
253browse00040
254author:1
255EnableButton("Up");ChangeButtonBinding("Up", "JumpId(`TEX2RTF.hlp', `topic30')")

$[256]#[257]+[258]K[259]![260]backslash:0

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

256backslash:0
257backslash
258browse00041
259backslash:0
260EnableButton("Up");ChangeButtonBinding("Up", "JumpId(`TEX2RTF.hlp', `topic30')")

$[261]#[262]+[263]K[264]![265]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.

261bcol:2
262bcol
263browse00042
264bcol:2
265EnableButton("Up");ChangeButtonBinding("Up", "JumpId(`TEX2RTF.hlp', `topic30')")

$[266]#[267]+[268]K[269]![270]**bf:1**

Specifies bold font.

266bf:1
267bf
268browse00043
269bf:1
270EnableButton("Up");ChangeButtonBinding("Up", "JumpId(`TEX2RTF.hlp', `topic30')")

$[271]#[272]+[273]K[274]![275]**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.

271bibitem:2
272bibitem
273browse00044
274bibitem:2
275EnableButton("Up");ChangeButtonBinding("Up", "JumpId(`TEX2RTF.hlp', `topic30')")

$[276]#[277]+[278]K[279]![280]bibliographystyle:1

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

276bibliographystyle:1
277bibliographystyle
278browse00045
279bibliographystyle:1
280EnableButton("Up");ChangeButtonBinding("Up", "JumpId(`TEX2RTF.hlp', `topic30')")

$[281]#[282]+[283]K[284]![285]**bibliography:0**

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

281bibliography:0
282bibliography
283browse00046
284bibliography:0
285EnableButton("Up");ChangeButtonBinding("Up", "JumpId(`TEX2RTF.hlp', `topic30')")

$[286]#[287]+[288]K[289]![290]**caption:1**

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

286caption:1
287caption
288browse00047
289caption:1
290EnableButton("Up");ChangeButtonBinding("Up", "JumpId(`TEX2RTF.hlp', `topic30')")

$[291]#[292]+[293]K[294]![295]**cdots:0**

Outputs three dots.

291cdots:0
292cdots
293browse00048
294cdots:0
295EnableButton("Up");ChangeButtonBinding("Up", "JumpId(`TEX2RTF.hlp', `topic30')")

$[296]#[297]+[298]K[299]![300]**centerline:1**

Centres (or centers!) a line of text.

296centerline:1
297centerline
298browse00049
299centerline:1
300EnableButton("Up");ChangeButtonBinding("Up", "JumpId(`TEX2RTF.hlp', `topic30')")

$[301]#[302]+[303]K[304]![305]**center:1**

Centres a block of text.

301center:1
302center
303browse00050
304center:1
305EnableButton("Up");ChangeButtonBinding("Up", "JumpId(`TEX2RTF.hlp', `topic30')")

$[306]#[307]+[308]K[309]![310]**cextract:0**

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

306cextract:0
307cextract
308browse00051
309cextract:0
310EnableButton("Up");ChangeButtonBinding("Up", "JumpId(`TEX2RTF.hlp', `topic30')")

$[311]#[312]+[313]K[314]![315]**chapter*:1**

Outputs a chapter heading with no contents entry.

311chapter*:1
312chaptersX
313browse00052
314chapter*:1
315EnableButton("Up");ChangeButtonBinding("Up", "JumpId(`TEX2RTF.hlp', `topic30')")

$[316]#[317]+[318]K[319]![320]**chapter:1**

Outputs a chapter heading.

316chapter:1
317chapter
318browse00053
319chapter:1
320EnableButton("Up");ChangeButtonBinding("Up", "JumpId(`TEX2RTF.hlp', `topic30')")

$[321]#[322]+[323]K[324]![325]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.

321chapterheading:1
322chapterheading
323browse00054
324chapterheading:1
325EnableButton("Up");ChangeButtonBinding("Up", "JumpId(`TEX2RTF.hlp', `topic30')")

$[326]#[327]+[328]K[329]![330]**cinsert:0**

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

326cinsert:0
327cinsert
328browse00055
329cinsert:0
330EnableButton("Up");ChangeButtonBinding("Up", "JumpId(`TEX2RTF.hlp', `topic30')")

$[331]#[332]+[333]K[334]![335]**cite:1**

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

331cite:1
332cite
333browse00056
334cite:1
335EnableButton("Up");ChangeButtonBinding("Up", "JumpId(`TEX2RTF.hlp', `topic30')")

$[336]#[337]+[338]K[339]![340]**class:1**

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

336class:1
337class
338browse00057
339class:1
340EnableButton("Up");ChangeButtonBinding("Up", "JumpId(`TEX2RTF.hlp', `topic30')")

$[341]#[342]+[343]K[344]![345]clipsfunc:3

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

341clipsfunc:3
342clipsfunc
343browse00058
344clipsfunc:3
345EnableButton("Up");ChangeButtonBinding("Up", "JumpId(`TEX2RTF.hlp', `topic30')")

$[346]#[347]+[348]K[349]![350]**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.

346comment:1
347comment
348browse00059
349comment:1
350EnableButton("Up");ChangeButtonBinding("Up", "JumpId(`TEX2RTF.hlp', `topic30')")

$[351]#[352]+[353]K[354]![355]**copyright:0**

Outputs the copyright symbol.

351copyright:0
352copyright
353browse00060
354copyright:0
355EnableButton("Up");ChangeButtonBinding("Up", "JumpId(`TEX2RTF.hlp', `topic30')")

$[356]#[357]+[358]K[359]![360]cparam:2

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

356cparam:2
357cparam
358browse00061
359cparam:2
360EnableButton("Up");ChangeButtonBinding("Up", "JumpId(`TEX2RTF.hlp', `topic30')")

$[361]#[362]+[363]K[364]![365]**date:1**

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

361date:1
362date
363browse00062
364date:1
365EnableButton("Up");ChangeButtonBinding("Up", "JumpId(`TEX2RTF.hlp', `topic30')")

$[366]#[367]+[368]K[369]![370]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.

366definecolour:4
367definecolour
368browse00063
369definecolour:4
370EnableButton("Up");ChangeButtonBinding("Up", "JumpId(`TEX2RTF.hlp', `topic30')")

$[371]#[372]+[373]K[374]![375]**description:1**

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

371description:1
372description
373browse00064
374description:1
375EnableButton("Up");ChangeButtonBinding("Up", "JumpId(`TEX2RTF.hlp', `topic30')")

$[376]#[377]+[378]K[379]![380]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.

376documentstyle:1
377documentstyle
378browse00065
379documentstyle:1
380EnableButton("Up");ChangeButtonBinding("Up", "JumpId(`TEX2RTF.hlp', `topic30')")

$[381]#[382]+[383]K[384]![385]document:1

This environment should enclose the body of a document.

381document:1
382document
383browse00066
384document:1
385EnableButton("Up");ChangeButtonBinding("Up", "JumpId(`TEX2RTF.hlp', `topic30')")

$[386]#[387]+[388]K[389]![390]**em:1**

Emphasizes text (italic in RTF).

386em:1
387em
388browse00067
389em:1
390EnableButton("Up");ChangeButtonBinding("Up", "JumpId(`TEX2RTF.hlp', `topic30')")

$[391]#[392]+[393]K[394]![395]**enumerate:1**

Enumerate list environment: numbers the items.

391enumerate:1
392enumerate
393browse00068
394enumerate:1
395EnableButton("Up");ChangeButtonBinding("Up", "JumpId(`TEX2RTF.hlp', `topic30')")

$396#397+398K399!400fcol:2

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

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.

396fcol:2
397fcol
398browse00069
399fcol:2
400EnableButton("Up");ChangeButtonBinding("Up", "JumpId(`TEX2RTF.hlp', `topic30')")

$[401]#[402]+[403]K[404]![405]**figure:1**

A figure enviroment: does nothin in RTF.

401figure:1
402figure
403browse00070
404figure:1
405EnableButton("Up");ChangeButtonBinding("Up", "JumpId(`TEX2RTF.hlp', `topic30')")

$[406]#[407]+[408]K[409]![410]**flushleft:1**

Flushes the given text to the left margin.

406flushleft:1
407flushleft
408browse00071
409flushleft:1
410EnableButton("Up");ChangeButtonBinding("Up", "JumpId(`TEX2RTF.hlp', `topic30')")

$[411]#[412]+[413]K[414]![415]**flushright:1**

Flushes the given text to the right margin.

411flushright:1
412flushright
413browse00072
414flushright:1
415EnableButton("Up");ChangeButtonBinding("Up", "JumpId(`TEX2RTF.hlp', `topic30')")

$[416]#[417]+[418]K[419]![420]**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.

416footnote:1
417footnote
418browse00073
419footnote:1
420EnableButton("Up");ChangeButtonBinding("Up", "JumpId(`TEX2RTF.hlp', `topic30')")

$[421]#[422]+[423]K[424]![425]footnotepopup:2

In linear RTF, a footnote is created following the first argument, as withfootnote.

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.

421footnotepopup:2
422footnotepopup
423browse00074
424footnotepopup:2
425EnableButton("Up");ChangeButtonBinding("Up", "JumpId(`TEX2RTF.hlp', `topic30')")

$[426]#[427]+[428]K[429]![430]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.

426functionsection:1
427functionsection
428browse00075
429functionsection:1
430EnableButton("Up");ChangeButtonBinding("Up", "JumpId(`TEX2RTF.hlp', `topic30')")

$[431]#[432]+[433]K[434]![435]func:3

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

Should occur after a functionsection command.

431func:3
432func
433browse00076
434func:3
435EnableButton("Up");ChangeButtonBinding("Up", "JumpId(`TEX2RTF.hlp', `topic30')")

$[436]#[437]+[438]K[439]![440]gloss:1

Marks a glossary entry. In LaTeX, this is a synonym for an itemwith 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 byhelpref 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, wxWindows) can be used.

\gloss{Canvas}\label{canvas}

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

\gloss{DDE}\label{dde}

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

\end{helpglossary}
```

436gloss:1
437gloss
438browse00077
439gloss:1
440EnableButton("Up");ChangeButtonBinding("Up", "JumpId(`TEX2RTF.hlp', `topic30')")

$[441]#[442]+[443]K[444]![445]**helpglossary:1**

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

441helpglossary:1
442helpglossary
443browse00078
444helpglossary:1
445EnableButton("Up");ChangeButtonBinding("Up", "JumpId(`TEX2RTF.hlp', `topic30')")

$[446]#[447]+[448]K[449]![450]**helpignore:1**

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

446helpignore:1
447helpignore
448browse00079
449helpignore:1
450EnableButton("Up");ChangeButtonBinding("Up", "JumpId(`TEX2RTF.hlp', `topic30')")

$[451]#[452]+[453]K[454]![455]helponly:1

Only outputs the argument in Tex2RTF generated files.

451helponly:1
452helponly
453browse00080
454helponly:1
455EnableButton("Up");ChangeButtonBinding("Up", "JumpId(`TEX2RTF.hlp', `topic30')")

$[456]#[457]+[458]K[459]![460]helpinput:1

Only includes the given file in Tex2RTF generated files.

456helpinput:1
457helpinput
458browse00081
459helpinput:1
460EnableButton("Up");ChangeButtonBinding("Up", "JumpId(`TEX2RTF.hlp', `topic30')")

$[461]#[462]+[463]K[464]![465]helpfontfamily:1

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

461helpfontfamily:1
462helpfontfamily
463browse00082
464helpfontfamily:1
465EnableButton("Up");ChangeButtonBinding("Up", "JumpId(`TEX2RTF.hlp', `topic30')")

$[466]#[467]+[468]K[469]![470]helpfontsize:1

Specifies the font size for Tex2RTF generated files.

466helpfontsize:1
467helpfontsize
468browse00083
469helpfontsize:1
470EnableButton("Up");ChangeButtonBinding("Up", "JumpId(`TEX2RTF.hlp', `topic30')")

$[471]#[472]+[473]K[474]![475]**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*.

471helpref:2
472helpref
473browse00084
474helpref:2
475EnableButton("Up");ChangeButtonBinding("Up", "JumpId(`TEX2RTF.hlp', `topic30')")

$[476]#[477]+[478]K[479]![480]**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.

476helprefn:2
477helprefn
478browse00085
479helprefn:2
480EnableButton("Up");ChangeButtonBinding("Up", "JumpId(`TEX2RTF.hlp', `topic30')")

$[481]#[482]+[483]K[484]![485]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.

481hline:0
482hline
483browse00086
484hline:0
485EnableButton("Up");ChangeButtonBinding("Up", "JumpId(`TEX2RTF.hlp', `topic30')")

$[486]#[487]+[488]K[489]![490]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.

486hrule:0
487hrule
488browse00087
489hrule:0
490EnableButton("Up");ChangeButtonBinding("Up", "JumpId(`TEX2RTF.hlp', `topic30')")

$[491]#[492]+[493]K[494]![495]**htmlignore:1**

Ignores the argument in HTML.

491htmlignore:1
492htmlignore
493browse00088
494htmlignore:1
495EnableButton("Up");ChangeButtonBinding("Up", "JumpId(`TEX2RTF.hlp', `topic30')")

$[496]#[497]+[498]K[499]![500]**htmlonly:1**

Only outputs the argument in HTML.

496htmlonly:1
497htmlonly
498browse00089
499htmlonly:1
500EnableButton("Up");ChangeButtonBinding("Up", "JumpId(`TEX2RTF.hlp', `topic30')")

$[501]#[502]+[503]K[504]![505]**huge:1**

Outputs the argument in huge text.

501huge:1
502huge1
503browse00090
504huge:1
505EnableButton("Up");ChangeButtonBinding("Up", "JumpId(`TEX2RTF.hlp', `topic30')")

$[506]#[507]+[508]K[509]![510]**Huge:1**

Outputs the argument in huger text that huge.

506Huge:1
507Huge2
508browse00091
509Huge:1
510EnableButton("Up");ChangeButtonBinding("Up", "JumpId(`TEX2RTF.hlp', `topic30')")

$511#512+513K514!515**HUGE:1**

Outputs the argument in huger text that Huge.

511HUGE:1
512HUGE3
513browse00092
514HUGE:1
515EnableButton("Up");ChangeButtonBinding("Up", "JumpId(`TEX2RTF.hlp', `topic30')")

$[516]#[517]+[518]K[519]![520]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.

516include:1
517include
518browse00093
519include:1
520EnableButton("Up");ChangeButtonBinding("Up", "JumpId(`TEX2RTF.hlp', `topic30')")

$[521]#[522]+[523]K[524]![525]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.

521indented:2
522indented
523browse00094
524indented:2
525EnableButton("Up");ChangeButtonBinding("Up", "JumpId(`TEX2RTF.hlp', `topic30')")

$[526]#[527]+[528]K[529]![530]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.

526input:1
527input
528browse00095
529input:1
530EnableButton("Up");ChangeButtonBinding("Up", "JumpId(`TEX2RTF.hlp', `topic30')")

$[531]#[532]+[533]K[534]![535]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.

531itemize:1
532itemize
533browse00096
534itemize:1
535EnableButton("Up");ChangeButtonBinding("Up", "JumpId(`TEX2RTF.hlp', `topic30')")

$[536]#[537]+[538]K[539]![540]**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.

536item:0
537item
538browse00097
539item:0
540EnableButton("Up");ChangeButtonBinding("Up", "JumpId(`TEX2RTF.hlp', `topic30')")

$[541]#[542]+[543]K[544]![545]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.

541itemsep:0
542itemsep
543browse00098
544itemsep:0
545EnableButton("Up");ChangeButtonBinding("Up", "JumpId(`TEX2RTF.hlp', `topic30')")

$[546]#[547]+[548]K[549]![550]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 or placeable metafile. 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 wxWindows 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.

546image:2
547image
548browse00099
549image:2
550EnableButton("Up");ChangeButtonBinding("Up", "JumpId(`TEX2RTF.hlp', `topic30')")

$[551]#[552]+[553]K[554]![555]**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.

551index:1
552index
553browse00100
554index:1
555EnableButton("Up");ChangeButtonBinding("Up", "JumpId(`TEX2RTF.hlp', `topic30')")

$[556]#[557]+[558]K[559]![560]**it:1**

Marks the argument in italic.

556it:1
557it
558browse00101
559it:1
560EnableButton("Up");ChangeButtonBinding("Up", "JumpId(`TEX2RTF.hlp', `topic30')")

$[561]#[562]+[563]K[564]![565]**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.

561label:1
562label
563browse00102
564label:1
565EnableButton("Up");ChangeButtonBinding("Up", "JumpId(`TEX2RTF.hlp', `topic30')")

$[566]#[567]+[568]K[569]![570]**large:1**

Marks the argument in large text.

566large:1
567large1
568browse00103
569large:1
570EnableButton("Up");ChangeButtonBinding("Up", "JumpId(`TEX2RTF.hlp', `topic30')")

$[571]#[572]+[573]K[574]![575]**Large:1**

Makes the argument display in larger text than large.

571Large:1
572Large2
573browse00104
574Large:1
575EnableButton("Up");ChangeButtonBinding("Up", "JumpId(`TEX2RTF.hlp', `topic30')")

$[576]#[577]+[578]K[579]![580]LARGE:1

Makes the argument display in larger text than Large.

576LARGE:1
577LARGE3
578browse00105
579LARGE:1
580EnableButton("Up");ChangeButtonBinding("Up", "JumpId(`TEX2RTF.hlp', `topic30')")

$[581]#[582]+[583]K[584]![585]LaTeX:0

Outputs the annoying LaTeX upper and lower case name.

581LaTeX:0
582LaTeX
583browse00106
584LaTeX:0
585EnableButton("Up");ChangeButtonBinding("Up", "JumpId(`TEX2RTF.hlp', `topic30')")

$[586]#[587]+[588]K[589]![590]**latexignore:1**

Ignores the argument in LaTeX.

586latexignore:1
587latexignore
588browse00107
589latexignore:1
590EnableButton("Up");ChangeButtonBinding("Up", "JumpId(`TEX2RTF.hlp', `topic30')")

$[591]#[592]+[593]K[594]![595]**latexonly:1**

Only prints the argument in LaTeX.

591latexonly:1
592latexonly
593browse00108
594latexonly:1
595EnableButton("Up");ChangeButtonBinding("Up", "JumpId(`TEX2RTF.hlp', `topic30')")

$[596]#[597]+[598]K[599]![600]ldots

Outputs three dots.

596ldots
597ldots
598browse00109
599ldots
600EnableButton("Up");ChangeButtonBinding("Up", "JumpId(`TEX2RTF.hlp', `topic30')")

$[601]#[602]+[603]K[604]![605]**maketitle**

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

601maketitle
602maketitle
603browse00110
604maketitle
605EnableButton("Up");ChangeButtonBinding("Up", "JumpId(`TEX2RTF.hlp', `topic30')")

$606#607+608K609!610**marginparwidth:1**

Specifies the width of a margin paragraph.

606marginparwidth:1
607marginparwidth
608browse00111
609marginparwidth:1
610EnableButton("Up");ChangeButtonBinding("Up", "JumpId(`TEX2RTF.hlp', `topic30')")

$611#612+613K614!615marginpar:1

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

611marginpar:1
612marginpar
613browse00112
614marginpar:1
615EnableButton("Up");ChangeButtonBinding("Up", "JumpId(`TEX2RTF.hlp', `topic30')")

$[616]#[617]+[618]K[619]![620]**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.

616marginpareven:1
617marginpareven
618browse00113
619marginpareven:1
620EnableButton("Up");ChangeButtonBinding("Up", "JumpId(`TEX2RTF.hlp', `topic30')")

$[621]#[622]+[623]K[624]![625]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 macro 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 WhizzyGig Computer 4001, push the Power button and
wait for some kind of response.
```

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.

621marginparodd:1
622marginparodd
623browse00114
624marginparodd:1
625EnableButton("Up");ChangeButtonBinding("Up", "JumpId(`TEX2RTF.hlp', `topic30')")

$[626]#[627]+[628]K[629]![630]membersection:1

Used when formatting C++ classes to print a subsection for the member name.

626membersection:1
627membersection
628browse00115
629membersection:1
630EnableButton("Up");ChangeButtonBinding("Up", "JumpId(`TEX2RTF.hlp', `topic30')")

$[631]#[632]+[633]K[634]![635]**member:1**

Used to format a C++ member variable name.

631member:1
632member
633browse00116
634member:1
635EnableButton("Up");ChangeButtonBinding("Up", "JumpId(`TEX2RTF.hlp', `topic30')")

$[636]#[637]+[638]K[639]![640]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.

636multicolumn:3
637multicolumn
638browse00117
639multicolumn:3
640EnableButton("Up");ChangeButtonBinding("Up", "JumpId(`TEX2RTF.hlp', `topic30')")

$[641]#[642]+[643]K[644]![645]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).

641newcommand:3
642newcommand
643browse00118
644newcommand:3
645EnableButton("Up");ChangeButtonBinding("Up", "JumpId(`TEX2RTF.hlp', `topic30')")

$[646]#[647]+[648]K[649]![650]**newpage:0**

Inserts a page break.

646newpage:0
647newpage
648browse00119
649newpage:0
650EnableButton("Up");ChangeButtonBinding("Up", "JumpId(`TEX2RTF.hlp', `topic30')")

$651#652+653K654!655**nocite:1**

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

See also cite.

651nocite:1
652nocite
653browse00120
654nocite:1
655EnableButton("Up");ChangeButtonBinding("Up", "JumpId(`TEX2RTF.hlp', `topic30')")

$[656]#[657]+[658]K[659]![660]**noindent:0**

Sets paragraph indentation to zero. See also parindent.

656noindent:0
657noindent
658browse00121
659noindent:0
660EnableButton("Up");ChangeButtonBinding("Up", "JumpId(`TEX2RTF.hlp', `topic30')")

$661#662+663K664!665normalbox: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.

661normalbox:1
662normalbox
663browse00122
664normalbox:1
665EnableButton("Up");ChangeButtonBinding("Up", "JumpId(`TEX2RTF.hlp', `topic30')")

$666#667+668K669!670normalboxd: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.

666normalboxd:1
667normalboxd
668browse00123
669normalboxd:1
670EnableButton("Up");ChangeButtonBinding("Up", "JumpId(`TEX2RTF.hlp', `topic30')")

$[671]#[672]+[673]K[674]![675]**normalsize:1**

Sets the font size back to normal.

671normalsize:1
672normalsize
673browse00124
674normalsize:1
675EnableButton("Up");ChangeButtonBinding("Up", "JumpId(`TEX2RTF.hlp', `topic30')")

$676#677+678K679!680onecolumn:0

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

676onecolumn:0
677onecolumn
678browse00125
679onecolumn:0
680EnableButton("Up");ChangeButtonBinding("Up", "JumpId(`TEX2RTF.hlp', `topic30')")

$[681]#[682]+[683]K[684]![685]**pageref:1**

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

681pageref:1
682pageref
683browse00126
684pageref:1
685EnableButton("Up");ChangeButtonBinding("Up", "JumpId(`TEX2RTF.hlp', `topic30')")

$[686]#[687]+[688]K[689]![690]**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.

686pagestyle:1
687pagestyle
688browse00127
689pagestyle:1
690EnableButton("Up");ChangeButtonBinding("Up", "JumpId(`TEX2RTF.hlp', `topic30')")

$691#692+693K694!695**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.

691pagenumbering:1
692pagenumbering
693browse00128
694pagenumbering:1
695EnableButton("Up");ChangeButtonBinding("Up", "JumpId(`TEX2RTF.hlp', `topic30')")

$[696]#[697]+[698]K[699]![700]**paragraph***

Behaves as for a subsubsection.

696paragraph*
697paragraphX
698browse00129
699paragraph*
700EnableButton("Up");ChangeButtonBinding("Up", "JumpId(`TEX2RTF.hlp', `topic30')")

$[701]#[702]+[703]K[704]![705]**paragraph**

Behaves as for a subsubsection.

701paragraph
702paragraph
703browse00130
704paragraph
705EnableButton("Up");ChangeButtonBinding("Up", "JumpId(`TEX2RTF.hlp', `topic30')")

$[706]#[707]+[708]K[709]![710]**param:1**

Formats a C++ type and argument pair. Should be used within the third argument of a a func command.

706param:1
707param
708browse00131
709param:1
710EnableButton("Up");ChangeButtonBinding("Up", "JumpId(`TEX2RTF.hlp', `topic30')")

$[711]#[712]+[713]K[714]![715]**parindent:1**

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

711parindent:1
712parindent
713browse00132
714parindent:1
715EnableButton("Up");ChangeButtonBinding("Up", "JumpId(`TEX2RTF.hlp', `topic30')")

$716#717+718K719!720**parskip:1**

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

716parskip:1
717parskip
718browse00133
719parskip:1
720EnableButton("Up");ChangeButtonBinding("Up", "JumpId(`TEX2RTF.hlp', `topic30')")

$[721]#[722]+[723]K[724]![725]**par:0**

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

721par:0
722par
723browse00134
724par:0
725EnableButton("Up");ChangeButtonBinding("Up", "JumpId(`TEX2RTF.hlp', `topic30')")

$[726]#[727]+[728]K[729]![730]**printindex**

In linear RTF, inserts an index.

726printindex
727printindex
728browse00135
729printindex
730EnableButton("Up");ChangeButtonBinding("Up", "JumpId(`TEX2RTF.hlp', `topic30')")

$[731]#[732]+[733]K[734]![735]**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; seegloss.

731popref:2
732popref
733browse00136
734popref:2
735EnableButton("Up");ChangeButtonBinding("Up", "JumpId(`TEX2RTF.hlp', `topic30')")

$[736]#[737]+[738]K[739]![740]**psboxto:2**

Identical to image.

736psboxto:2
737psboxto
738browse00137
739psboxto:2
740EnableButton("Up");ChangeButtonBinding("Up", "JumpId(`TEX2RTF.hlp', `topic30')")

$[741]#[742]+[743]K[744]![745]**quote:1**

Indents a short quotation.

741quote:1
742quote
743browse00138
744quote:1
745EnableButton("Up");ChangeButtonBinding("Up", "JumpId(`TEX2RTF.hlp', `topic30')")

$[746]#[747]+[748]K[749]![750]**quotation:1**

Indents a long quotation.

746quotation:1
747quotation
748browse00139
749quotation:1
750EnableButton("Up");ChangeButtonBinding("Up", "JumpId(`TEX2RTF.hlp', `topic30')")

$[751]#[752]+[753]K[754]![755]**ref:1**

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

751ref:1
752ref
753browse00140
754ref:1
755EnableButton("Up");ChangeButtonBinding("Up", "JumpId(`TEX2RTF.hlp', `topic30')")

$[756]#[757]+[758]K[759]![760]**rm:1**

Causes the argument to be formatted in a plain, roman font.

756rm:1
757rm
758browse00141
759rm:1
760EnableButton("Up");ChangeButtonBinding("Up", "JumpId(`TEX2RTF.hlp', `topic30')")

$[761]#[762]+[763]K[764]![765]row:1

A Tex2RTF command signifying the row of a table within the tabular environment. See also ruledrow.

761row:1
762row
763browse00142
764row:1
765EnableButton("Up");ChangeButtonBinding("Up", "JumpId(`TEX2RTF.hlp', `topic30')")

$[766]#[767]+[768]K[769]![770]ruledrow:1

A Tex2RTF command signifying a ruled row of a table within the tabular environment. See also row.

766ruledrow:1
767ruledrow
768browse00143
769ruledrow:1
770EnableButton("Up");ChangeButtonBinding("Up", "JumpId(`TEX2RTF.hlp', `topic30')")

$[771]#[772]+[773]K[774]![775]**rtfignore:1**

Ignores the argument in linear RTF.

771rtfignore:1
772rtfignore
773browse00144
774rtfignore:1
775EnableButton("Up");ChangeButtonBinding("Up", "JumpId(`TEX2RTF.hlp', `topic30')")

$[776]#[777]+[778]K[779]![780]**rtfonly:1**

Only outputs the argument in linear RTF.

776rtfonly:1
777rtfonly
778browse00145
779rtfonly:1
780EnableButton("Up");ChangeButtonBinding("Up", "JumpId(`TEX2RTF.hlp', `topic30')")

$[781]#[782]+[783]K[784]![785]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.

781rtfsp:0
782rtfsp
783browse00146
784rtfsp:0
785EnableButton("Up");ChangeButtonBinding("Up", "JumpId(`TEX2RTF.hlp', `topic30')")

$[786]#[787]+[788]K[789]![790]**sc:1**

Prints the output in small capitals.

786sc:1
787sc
788browse00147
789sc:1
790EnableButton("Up");ChangeButtonBinding("Up", "JumpId(`TEX2RTF.hlp', `topic30')")

$[791]#[792]+[793]K[794]![795]section*:1

Section header, with no entry in the contents page.

791section*:1
792sectionX
793browse00148
794section*:1
795EnableButton("Up");ChangeButtonBinding("Up", "JumpId(`TEX2RTF.hlp', `topic30')")

$[796]#[797]+[798]K[799]![800]section:1

Section header, with an entry in the contents page.

796section:1
797section
798browse00149
799section:1
800EnableButton("Up");ChangeButtonBinding("Up", "JumpId(`TEX2RTF.hlp', `topic30')")

$[801]#[802]+[803]K[804]![805]**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.

801sectionheading:1
802sectionheading
803browse00150
804sectionheading:1
805EnableButton("Up");ChangeButtonBinding("Up", "JumpId(`TEX2RTF.hlp', `topic30')")

$[806]#[807]+[808]K[809]![810]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.

806setfooter:6
807setfooter
808browse00151
809setfooter:6
810EnableButton("Up");ChangeButtonBinding("Up", "JumpId(`TEX2RTF.hlp', `topic30')")

$[811]#[812]+[813]K[814]![815]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.

811setheader:6
812setheader
813browse00152
814setheader:6
815EnableButton("Up");ChangeButtonBinding("Up", "JumpId(`TEX2RTF.hlp', `topic30')")

$[816]#[817]+[818]K[819]![820]**shortcite:1**

The same as cite.

816shortcite:1
817shortcite
818browse00153
819shortcite:1
820EnableButton("Up");ChangeButtonBinding("Up", "JumpId(`TEX2RTF.hlp', `topic30')")

$[821]#[822]+[823]K[824]![825]**small:1**

Prints the argument in a small font.

821small:1
822small
823browse00154
824small:1
825EnableButton("Up");ChangeButtonBinding("Up", "JumpId(`TEX2RTF.hlp', `topic30')")

$826#827+828K829!830special: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).

826special:1
827special
828browse00155
829special:1
830EnableButton("Up");ChangeButtonBinding("Up", "JumpId(`TEX2RTF.hlp', `topic30')")

$[831]#[832]+[833]K[834]![835]**ss:0**

Outputs the German sharp S character ß.

831ss:0
832ss
833browse00156
834ss:0
835EnableButton("Up");ChangeButtonBinding("Up", "JumpId(`TEX2RTF.hlp', `topic30')")

$[836]#[837]+[838]K[839]![840]**subparagraph*:1**

Behaves as for a subsubsection.

836subparagraph*:1
837subparagraphX
838browse00157
839subparagraph*:1
840EnableButton("Up");ChangeButtonBinding("Up", "JumpId(`TEX2RTF.hlp', `topic30')")

$[841]#[842]+[843]K[844]![845]subparagraph:1

Behaves as for a subsubsection.

841subparagraph:1
842subparagraph
843browse00158
844subparagraph:1
845EnableButton("Up");ChangeButtonBinding("Up", "JumpId(`TEX2RTF.hlp', `topic30')")

$[846]#[847]+[848]K[849]![850]**subsection*:1**

Subsection header, with no entry in the contents page.

846subsection*:1
847subsectionX
848browse00159
849subsection*:1
850EnableButton("Up");ChangeButtonBinding("Up", "JumpId(`TEX2RTF.hlp', `topic30')")

$[851]#[852]+[853]K[854]![855]subsection:1

Subsection header, with an entry in the contents page.

851subsection:1
852subsection
853browse00160
854subsection:1
855EnableButton("Up");ChangeButtonBinding("Up", "JumpId(`TEX2RTF.hlp', `topic30')")

$[856]#[857]+[858]K[859]![860]subsubsection*:1

Subsubsection header, with no entry in the contents page.

856subsubsection*:1
857subsubsectionX
858browse00161
859subsubsection*:1
860EnableButton("Up");ChangeButtonBinding("Up", "JumpId(`TEX2RTF.hlp', `topic30')")

$[861]#[862]+[863]K[864]![865]subsubsection:1

Subsubsection header, with an entry in the contents page.

861subsubsection:1
862subsubsection
863browse00162
864subsubsection:1
865EnableButton("Up");ChangeButtonBinding("Up", "JumpId(`TEX2RTF.hlp', `topic30')")

$[866]#[867]+[868]K[869]![870]tabbing:1

Tabbing environment: doesn't work properly in RTF.

866tabbing:1
867tabbing
868browse00163
869tabbing:1
870EnableButton("Up");ChangeButtonBinding("Up", "JumpId(`TEX2RTF.hlp', `topic30')")

$[871]#[872]+[873]K[874]![875]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 the variable *insertTOC* is set to *false*.

871tableofcontents:0
872tableofcontents
873browse00164
874tableofcontents:0
875EnableButton("Up");ChangeButtonBinding("Up", "JumpId(`TEX2RTF.hlp', `topic30')")

$876#877+878K879!880tabular:2

Tabular environment: an attempt is made to output something reasonable in RTF format, 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.

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

876tabular:2
877tabular
878browse00165
879tabular:2
880EnableButton("Up");ChangeButtonBinding("Up", "JumpId(`TEX2RTF.hlp', `topic30')")

$881#882+883K884!885TeX:0

Outputs the annoying TeX upper and lower case name.

881TeX:0
882TeX
883browse00166
884TeX:0
885EnableButton("Up");ChangeButtonBinding("Up", "JumpId(`TEX2RTF.hlp', `topic30')")

$[886]#[887]+[888]K[889]![890]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.

886textwidth:1
887textwidth
888browse00167
889textwidth:1
890EnableButton("Up");ChangeButtonBinding("Up", "JumpId(`TEX2RTF.hlp', `topic30')")

$[891]#[892]+[893]K[894]![895]**thebibliography:1**

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

891thebibliography:1
892thebibliography
893browse00168
894thebibliography:1
895EnableButton("Up");ChangeButtonBinding("Up", "JumpId(`TEX2RTF.hlp', `topic30')")

$[896]#[897]+[898]K[899]![900]**title:1**

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

896title:1
897title
898browse00169
899title:1
900EnableButton("Up");ChangeButtonBinding("Up", "JumpId(`TEX2RTF.hlp', `topic30')")

$[901]#[902]+[903]K[904]![905]**tiny:1**

Prints the argument in a very small font.

901tiny:1
902tiny
903browse00170
904tiny:1
905EnableButton("Up");ChangeButtonBinding("Up", "JumpId(`TEX2RTF.hlp', `topic30')")

$[906]#[907]+[908]K[909]![910]**today:0**

Outputs today's date.

906today:0
907today
908browse00171
909today:0
910EnableButton("Up");ChangeButtonBinding("Up", "JumpId(`TEX2RTF.hlp', `topic30')")

$[911]#[912]+[913]K[914]![915]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.

911toocomplex:1
912toocomplex
913browse00172
914toocomplex:1
915EnableButton("Up");ChangeButtonBinding("Up", "JumpId(`TEX2RTF.hlp', `topic30')")

$[916]#[917]+[918]K[919]![920]**tt:1**

Outputs the argument in teletype font.

916tt:1
917tt
918browse00173
919tt:1
920EnableButton("Up");ChangeButtonBinding("Up", "JumpId(`TEX2RTF.hlp', `topic30')")

$[921]#[922]+[923]K[924]![925]typeout:1

Outputs the text on the Tex2RTF text window.

921typeout:1
922typeout
923browse00174
924typeout:1
925EnableButton("Up");ChangeButtonBinding("Up", "JumpId(`TEX2RTF.hlp', `topic30')")

$[926]#[927]+[928]K[929]![930]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.

926twocolitem:2
927twocolitem
928browse00175
929twocolitem:2
930EnableButton("Up");ChangeButtonBinding("Up", "JumpId(`TEX2RTF.hlp', `topic30')")

$[931]#[932]+[933]K[934]![935]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.

931twocolitemruled:2
932twocolitemruled
933browse00176
934twocolitemruled:2
935EnableButton("Up");ChangeButtonBinding("Up", "JumpId(`TEX2RTF.hlp', `topic30')")

$[936]#[937]+[938]K[939]![940]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, loading, exporting, importing, saving as
and various other bits and pieces.}
\twocolitem{Edit}{The Edit menu is used for selection, copying, pasting
and various other bits and pieces.}
\end{twocollist}
```

This produces:

Command	**Description**
File	The file menu is used to select various file-related operations, such as saving, loading, exporting, importing, saving as and various other bits and pieces.
Edit	The Edit menu is used for selection, copying, pasting and various other bits and pieces.

936twocollist:1
937twocollist
938browse00177
939twocollist:1
940EnableButton("Up");ChangeButtonBinding("Up", "JumpId(`TEX2RTF.hlp', `topic30')")

$[941]#[942]+[943]K[944]![945]**twocolwidtha:1**

Sets the width of the first column in a two column list to the given dimension. See also twocollist and twocolwidthb.

941twocolwidtha:1
942twocolwidtha
943browse00178
944twocolwidtha:1
945EnableButton("Up");ChangeButtonBinding("Up", "JumpId(`TEX2RTF.hlp', `topic30')")

$946#947+948K949!950twocolwidthb:1

Sets the width of the second column in a two column list to the given dimension. See also twocollist and twocolwidtha.

946twocolwidthb:1
947twocolwidthb
948browse00179
949twocolwidthb:1
950EnableButton("Up");ChangeButtonBinding("Up", "JumpId(`TEX2RTF.hlp', `topic30')")

$[951]#[952]+[953]K[954]![955]twocolumn:0

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

951twocolumn:0
952twocolumn
953browse00180
954twocolumn:0
955EnableButton("Up");ChangeButtonBinding("Up", "JumpId(`TEX2RTF.hlp', `topic30')")

$[956]#[957]+[958]K[959]![960]**underline:1**

Underlines the argument.

956underline:1
957underline
958browse00181
959underline:1
960EnableButton("Up");ChangeButtonBinding("Up", "JumpId(`TEX2RTF.hlp', `topic30')")

$961#962+963K964!965verbatiminput: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.

961verbatiminput:1
962verbatiminput
963browse00182
964verbatiminput:1
965EnableButton("Up");ChangeButtonBinding("Up", "JumpId(`TEX2RTF.hlp', `topic30')")

$[966]#[967]+[968]K[969]![970]**verbatim:1**

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

966verbatim:1
967verbatim
968browse00183
969verbatim:1
970EnableButton("Up");ChangeButtonBinding("Up", "JumpId(`TEX2RTF.hlp', `topic30')")

$[971]#[972]+[973]K[974]![975]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%^&`.

971verb
972verb
973browse00184
974verb
975EnableButton("Up");ChangeButtonBinding("Up", "JumpId(`TEX2RTF.hlp', `topic30')")

$[976]#[977]+[978]K[979]![980]**winhelpignore:1**

Ignores the argument in WinHelp RTF.

976winhelpignore:1
977winhelpignore
978browse00185
979winhelpignore:1
980EnableButton("Up");ChangeButtonBinding("Up", "JumpId(`TEX2RTF.hlp', `topic30')")

$[981]#[982]+[983]K[984]![985]**winhelponly:1**

Only outputs the argument in WinHelp RTF.

981winhelponly:1
982winhelponly
983browse00186
984winhelponly:1
985EnableButton("Up");ChangeButtonBinding("Up", "JumpId(`TEX2RTF.hlp', `topic30')")

$[986]#[987]+[988]K[989]![990]**xlpignore:1**

Ignores the argument in XLP mode (wxHelp files).

986xlpignore:1
987xlpignore
988browse00187
989xlpignore:1
990EnableButton("Up");ChangeButtonBinding("Up", "JumpId(`TEX2RTF.hlp', `topic30')")

$[991]#[992]+[993]K[994]![995]**xlponly:1**

Only outputs the argument in XLP mode (wxHelp files).

991xlponly:1
992xlponly
993browse00188
994xlponly:1
995EnableButton("Up");ChangeButtonBinding("Up", "JumpId(`TEX2RTF.hlp', `topic30')")

$996#997+998K999!1000Macro not found

This error may indicate that Tex2RTF has not implemented a standard LaTeX macro, 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 macro, 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}
```

996Macro not found
997macronotfound
998browse00193
999Macro not found
1000EnableButton("Up");ChangeButtonBinding("Up", "JumpId(`TEX2RTF.hlp', `topic32')")

$1001#1002+1003K1004!1005Unresolved 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.

1001Unresolved reference
1002topic33
1003browse00194
1004Unresolved reference
1005EnableButton("Up");ChangeButtonBinding("Up", "JumpId(`TEX2RTF.hlp', `topic32')")

$1006#1007+1008K1009!1010Output 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\\
&\\
```

1006Output crashes the RTF reader
1007topic34
1008browse00195
1009Output crashes the RTF reader
1010EnableButton("Up");ChangeButtonBinding("Up", "JumpId(`TEX2RTF.hlp', `topic32')")

$[1011]#[1012]+[1013]K[1014]![1015]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 replace of `\item [` to `\item[` may also be helpful to remove unnecessary space before the item label.

1011Erratic list indentation
1012topic35
1013browse00196
1014Erratic list indentation
1015EnableButton("Up");ChangeButtonBinding("Up", "JumpId(`TEX2RTF.hlp', `topic32')")

$1016#1017+1018K1019!1020Missing figure or section reference

Ensure all labels *directly* follow captions or sections (no intervening white space).

1016Missing figure or section reference
1017topic36
1018browse00197
1019Missing figure or section reference
1020EnableButton("Up");ChangeButtonBinding("Up", "JumpId(`TEX2RTF.hlp', `topic32')")

$1021#1022+1023K1024!1025Unresolved 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.

1021Unresolved references in Word for Windows
1022topic37
1023browse00198
1024Unresolved references in Word for Windows
1025EnableButton("Up");ChangeButtonBinding("Up", "JumpId(`TEX2RTF.hlp', `topic32')")

$1026#1027+1028K1029!1030General 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.

1026General options
1027topic19
1028browse00026
1029General options
1030EnableButton("Up");ChangeButtonBinding("Up", "JumpId(`TEX2RTF.hlp', `topic18')")

$1031#1032+1033K1034!1035Presentation 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".
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".
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".

1031Presentation options
1032topic20
1033browse00027
1034Presentation options
1035EnableButton("Up");ChangeButtonBinding("Up", "JumpId(`TEX2RTF.hlp', `topic18')")

$[1036]#[1037]+[1038]K[1039]![1040]RTF/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".
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).
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.
winHelpTitle	Windows Help file title, inserted into the project file if **generateHPJ** is true.

1036RTF/WinHelp options
1037topic21
1038browse00028
1039RTF/WinHelp options
1040EnableButton("Up");ChangeButtonBinding("Up", "JumpId(`TEX2RTF.hlp', `topic18')")

$[1041]#[1042]+[1043]K[1044]![1045]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.

1041HTML options
1042topic22
1043browse00029
1044HTML options
1045EnableButton("Up");ChangeButtonBinding("Up", "JumpId(`TEX2RTF.hlp', `topic18')")