

# **L<sup>A</sup>T<sub>E</sub>X2RTF**

---

A converter from L<sup>A</sup>T<sub>E</sub>X to RTF  
Edition 0.7.3

**Georg Lehner,**

updated by Wilfried Hennings and Scott Prahl, with contributions by Mikhail Polianski

---

Copyright © 1998-2006 Georg Lehner, Wilfried Hennings, Scott Prah

Permission is granted to make and distribute verbatim copies of this manual provided the copyright notice and this permission notice are preserved on all copies.

Permission is granted to copy and distribute modified versions of this manual under the conditions for verbatim copying, provided also that the sections entitled “Copying” and “GNU General Public License” are included exactly as in the original, and provided that the entire resulting derived work is distributed under the terms of a permission notice identical to this one.

Permission is granted to copy and distribute translations of this manual into another language, under the above conditions for modified versions, except that this permission notice may be stated in a translation approved by the Free Software Foundation.

# 1 Introduction

$\text{\LaTeX}2\text{RTF}$  is a translator program from  $\text{\LaTeX}$  text into “rich text format” files. These files are commonly referred to as RTF files. RTF is a published standard format by Microsoft. This standard can be ambiguous in places and Microsoft ignores parts of the standard it finds inconvenient, but RTF is widely used by many WYSIWIG text editors and is supported by Microsoft Word and most text processors.

$\text{\LaTeX}2\text{RTF}$  translates the text and as much of the formatting information from  $\text{\LaTeX}$  to RTF. Be forewarned that the typeset output is not nearly as good as what you would get from using  $\text{\LaTeX}$  directly. So, why bother translating? Consider,

1. You use  $\text{\LaTeX}$  and hate everything beginning with MS-... Nevertheless, you have to share your documents with people who don't even now that there exist other things than MS-...
2. You know somebody who frequently sends you very fine  $\text{\LaTeX}$  documents. Unfortunately, you are “on the other side” and need to import her files, steal some part, and then desktop publish it in your fine MS-... environment.
3. You like  $\text{\LaTeX}$  and  $\text{\BibTeX}$ . You interact with the rest of the world. You know someone that wants to include your writing in a her Word document.

There are drawbacks to the conversion process. In fact, don't expect any  $\text{\LaTeX}$  file to be converted as you would like, don't expect it to be converted without errors or warnings, and don't be especially surprised when it doesn't convert at all.  $\text{\LaTeX}2\text{RTF}$  is known to have many bugs and many missing features. Paradoxically, this number seems to grow more and more with each day. However, we can categorically state that there are some special cases, in which a  $\text{\LaTeX}$  file will be translated to RTF satisfactorily by  $\text{\LaTeX}2\text{RTF}$  — This was sort of disclaimer, ok? OK!

$\text{\LaTeX}$  is a system for typesetting text and therefore it focuses on the logical structure of a document, whilst RTF is meant to be a transport format for a family of Desktop Publishing Software, dealing mostly with the design of a text.

Although the possible different commands and styles in  $\text{\LaTeX}$  are much more flexible and standardized than in RTF, only a small subset of commands has been implemented to date (see [Section 8.1 \[Unimplemented Features\]](#), page 25).

Some of the capabilities of  $\text{\LaTeX}2\text{RTF}$  are restricted in scope or buggy (see [Section 8.3 \[Known Bugs\]](#), page 25).

RTF is a moving target, because Microsoft does not stop inventing new extensions and features, consequently you cannot view newer RTF files with older word processors. The syntax and semantics of RTF are somewhat artistic, i.e., you can generate a syntactically correct RTF file that cannot be displayed by some/most word processors. For more details on RTF the specification consult the links at <http://latex2rtf.sf.net/>



## 2 Installation

### 2.1 General

The documentation of the program is found in the ‘doc/’ directory in the file ‘`latex2rtf.texi`’ in the GNU TeXInfo format. For your convenience, you can find HTML and PDF versions of the manual there as well.

### 2.2 Obtaining LaTeX2RTF

L<sup>A</sup>T<sub>E</sub>X2RTF is available for many Unix Platforms, for the Macintosh, and for MS-DOS, including all versions of MS Windows.

The latest version of L<sup>A</sup>T<sub>E</sub>X2RTF is available at [SourceForge](http://www.dante.de) and — with some delay — on CTAN sites: e.g., <http://www.dante.de> or <http://www.ctan.org>.

The MS-DOS version (1.8aa and up) will also run under all MS Windows versions (probably except Windows Vista). It requires an i386 processor or better.

The Win32 distribution (starting from L<sup>A</sup>T<sub>E</sub>X2RTF 1.9.15) is the MS-DOS version plus Win32 GUI program (l2rshell).

There are a few people working on L<sup>A</sup>T<sub>E</sub>X2RTF, and some more make contributions, coordinated by Wilfried Hennings (W.Hennings "at" fz-juelich.de). See the [SourceForge](http://www.dante.de) project pages for the latest news.

### 2.3 UNIX

To install,

1. Edit ‘Makefile’ for your local configuration. In particular, pay attention to the `PREFIX` variable. If you do not have root access you might wish to set the makefile variable `PREFIX` to be your home directory.

On some machines the `cc` compiler will issue errors. Therefore the default compiler command in the Makefile is `CC=gcc`.

2. As of version 1.9.13, L<sup>A</sup>T<sub>E</sub>X2RTF supports conversion of L<sup>A</sup>T<sub>E</sub>X equations to bitmaps using the shell script ‘`latex2png`’, found in ‘`scripts/`’. ‘`latex2png`’ requires that both L<sup>A</sup>T<sub>E</sub>X and ‘`ImageMagick`’ are installed. L<sup>A</sup>T<sub>E</sub>X2RTF will translate documents without a working ‘`latex2png`’, but some features will be missing. You can verify that the ‘`latex2png`’ script is working by typing `make` in the ‘`scripts/`’ directory.

3. `make`

If this is not your first time installation, you may want to preserve your old configuration (‘`*.cfg`’) files. Copy them to a safe place before installing.

On IBM AIX, the IBM `make` utility does not support some of the commands used in Makefile. In this case use `gmake` (from GNU) instead.

Sun has decided to support the XPG4 standard on Solaris by an alternative set of binaries. To allow bitmap conversion of equations, two things are needed. First, change the first line of `latex2png` to `#!/usr/xpg4/bin/sh`. Second, define the `XPG4_GREP` environment variable accordingly, for bash-like shells `XPG4_GREP=/usr/xpg4/bin/grep`; `export XPG4_GREP` or for tsch-like shells `setenv XPG4_GREP /usr/xpg4/bin/grep`.

#### 4. `make install`

If your `mkdir` doesn't support the `-p` option, then create the necessary directories by hand and remove the option from the `$MKDIR` variable. If you have other problems, just copy `'latex2rtf'` and `'latex2png'` to a binary directory, and move the contents of the `'cfg/'` directory to the location specified by `$CFG_INSTALL`.

#### 5. `make check`

[OPTIONAL] This tests `LATEX2RTF` on a variety of `LATEX` files. Expect a whole lot of warnings, but no outright errors. (On IBM AIX, use `gmake check`.) Note that this will check the basic functionality of the `'latex2png'` script, and then that of `'latex2rtf'`.

#### 6. `make install-info`

[OPTIONAL] This installs `'info'` files for use with the `info` program.

You no longer need to define the environment variable `RTFPATH`. This is only necessary if you move the directory containing the `'cfg'` files. Just define `RTFPATH` to be the path for the new location of the `'cfg'` directory.

## 2.4 DOS or Windows

The UNIX and Mac distributions do not contain an executable for DOS or Windows. Instead, get the DOS port as file `'latex2rtf-x.x.x_dos.zip'` or the Windows (Win32 = Win95, Win98, WinME, WinNT, Win2000, WinXP) version as file `'latex2rtf-x.x.x_win.exe'` (where x.x.x is the version number) from [SourceForge](#)

The DOS distribution contains a precompiled executable (djgpp compiler) which should run under plain DOS and also in the command prompt (or "console") of any MS Windows system up to XP and Server 2003.

However in Windows Vista the DPMI memory is limited to 32 MB, and we yet don't know whether this is sufficient to run `LATEX2RTF` or not, so please tell us any experiences regarding Windows Vista.

If the memory limitation under Windows Vista turns out to be a problem, one can use an executable compiled with Cygwin, but for now this isn't provided in the distributions.

Moreover, the command processor "command.com" is not provided by Vista, so conversion of pictures and of equations to bitmaps will definitely not work under Vista.

The Win32 version starting from 1.9.14a has the same executable but additionally an optional Win32 GUI interface (LaTeX2RTF Shell) and is wrapped in an automatic installer, see [Section 2.5 \[Win32 systems\]](#), page 6.

To install `LATEX2RTF`, extract all files from the zip archive to `'C:\l2r'` or to another folder (i.e. directory) of your choice, preserving the folder structure (winzip: check "use folder names").

If you extracted them to another folder (e.g. `'C:\Program Files\l2r'`), edit the files `'L2R.BAT'`, `'L2RSEM.BAT'` and `'L2RPREP.BAT'` and change the folder `'C:\l2r'` to where you put them. If there is a blank in any of the folder names (as in e.g. "Program Files"), then you need to enclose both the file-ID and the `cfg` path in double quotes, e.g.

```
"C:\Program Files\l2r\latex2rt" -P "C:\Program Files\l2r\cfg" %1 %2 ...'
```

or use the DOS filename equivalent instead, e.g.

```
'C:\Progra~1\l2r\latex2rt -P C:\Progra~1\l2r\cfg %1 %2 ...'
```

If you install  $\text{\LaTeX}2_{\text{RTF}}$  under WinNT, Win2000 or WinXP and you want other users to be able to use  $\text{\LaTeX}2_{\text{RTF}}$ , you must assign the files and folders appropriate user permissions (at least read & execute). If you don't know about user permissions, put  $\text{\LaTeX}2_{\text{RTF}}$  into 'C:\Program Files\l2r' or what else the standard folder for installing applications is named (e.g. in the German version it is 'C:\Programme\l2r').

Make sure that the folder containing the file 'L2RPREP.BAT' is in your search path, or put this file into a folder which is in your search path (e.g. 'C:\Windows').

To display the current search path, enter 'path' from the command prompt (with no arguments).

To add a folder – e.g. C:\l2r – to the search path:

- DOS, Win3.1, Win95, Win98, WinME
  - Open the file 'C:\Autoexec.bat' either with Edit (DOS) or with Notepad (Windows).
  - At the end of that file, add the line  
'PATH=C:\l2r;%PATH%'
  - Save the file to its original location. Shutdown and reboot the PC.
- WinNT, Win2000, WinXP
  - Right-click "My Computer" (German: "Arbeitsplatz"), then select "Properties";
  - NT: Click the "Environment" tab;  
XP: Click the "Extended" tab, then the "Environment variables" button;
  - Find the line beginning with 'PATH=' and insert the string  
'C:\l2r;'  
so that the complete line looks like  
'PATH=C:\l2r;C:\WINDOWS;...'
  - Click "OK"

To start the program under Windows,

open a command prompt (or console window)

– under DOS you are already at the command prompt –,

use 'cd <folder>' to make the folder with your TeX files the current folder,

then enter either 'L2R <filename> <options>' or 'L2RSEM <filename> <options>'

where <filename> is the name of your TeX file without the .tex extension.

'L2R.BAT' and 'L2RSEM.BAT' only differ in the character being used as parameter separator in the EQ fields, see chapter [Section 4.7 \[Equations\]](#), page 15.

If your  $\text{\LaTeX}$  document refers to external graphic files, e.g. eps, or you want to use the option to convert equations to bitmaps (-M# where # is greater than 3), you must have TeX, ImageMagick and Ghostscript installed. These programs are freely available for download at <http://www.miktex.org/>, <http://www.imagemagick.org> and <http://www.ghostscript.com>. This works only on win32 systems (Win95, Win98, WinME, WinNT4, Win2000, WinXP) because ImageMagick is available only for win32, not for pure DOS.

Either the folders where TeX, ImageMagick and Ghostscript are installed must be in your search path, or you must edit the file 'L2RPREP.BAT', ensure that the pathes in this

file point to the folders where TeX, ImageMagick and Ghostscript are installed on your machine, and call 'L2RPREP' once after opening your command prompt window and before calling 'l2r', 'l2rsem' or 'latex2rt'. Under Windows you can create another command prompt link in which you can include the call to 'L2RPREP.BAT'.

When equations are to be converted to bitmaps, for each equation  $\text{\LaTeX2RTF}$  first writes a temporary .tex file to disk which consists of only the equation to be converted. It then sends the call for latex2pn.bat to the operating system. There were problems that this didn't work although manually calling latex2pn.bat from the command prompt worked. There were several bugs in interfacing between Windows XP and DJGPP which finally caused this. One fix needed was truncating the filename of 'latex2png.bat' to the DOS convention of 8.3 characters so that it is now named 'latex2pn.bat' and the executable program 'latex2rt.exe'. The other fix is now hardcoded in latex2rt.exe, making sure the DOS environment space in which latex2pn.bat is executed is large enough to store the needed variables.

If you get the message "Out of environment space" (should occur only under plain DOS, Windows 3.1, Windows 95, Windows 98 or Windows ME) then add to your config.sys a line like the following  
`SHELL=C:\COMMAND.COM C:\ /E:4096 /P`  
 or, if there already is such a line, modify it so that it supplies a sufficiently large environment space /E:. Shutdown and reboot the PC.

## 2.5 Win32 systems

To install  $\text{\LaTeX2RTF}$  on a Win32 system (Win9x, WinME, WinNT, Win2000, WinXP), download and execute the 'latex2rtf-x.x.x\_win.exe' (where x.x.x is the version number) and follow the instructions.

If there is a newer version of  $\text{\LaTeX2RTF}$  for DOS but not for Windows, you can download the newer DOS package and unzip it to the same folder where the windows version is installed, thus overwriting the files with their newer versions. The GUI shell is not updated in this case.

To start the program double-click the  $\text{\LaTeX2RTF}$  icon, or drag and drop a '.tex' file onto the icon.

If your  $\text{\LaTeX}$  document refers to external graphic files, e.g. eps, or you want to use the option to convert equations to bitmaps, you must have LaTeX, ImageMagick and Ghostscript installed. These programs are freely available for download at <http://www.miktex.org/>, <http://www.imagemagick.org> and <http://www.ghostscript.com>

If MikTeX, ImageMagick and Ghostscript are installed before installing  $\text{\LaTeX2RTF}$ , the pathes to the executables are automatically found in the windows registry by the installer. Otherwise, these pathes may have to be specified manually in the "Environment" tab of the  $\text{\LaTeX2RTF}$  window.

If you get the message "Out of environment space" (should occur only under Windows 95, Windows 98 or Windows ME) then add to your config.sys a line like the following  
`SHELL=C:\COMMAND.COM C:\ /E:4096 /P`



or, if there already is such a line, modify it so that it supplies a sufficiently large environment space `/E:`. Shutdown and reboot the PC.

## 2.6 Macintosh

If you want a MacOS X version, make sure that you have installed the developer tools CD that is appropriate for your OS version, and then follow the directions above for a UNIX installation. Alternatively you can install using fink <http://fink.sourceforge.net/> or as an i-installer package <http://www.rna.nl/ii.html>.

As of 2006-01-30 there also is a GUI shell for Macintosh.

See <http://www.inf.ethz.ch/personal/fischerk/LaTeX2rtf/index.html>

There is a PPC port of an old version 1.9k for Classic MacOS  $\text{\LaTeX}2\text{RTF}$ . To convert a  $\text{\LaTeX}$  file using this version, drag the file onto the  $\text{\LaTeX}2\text{RTF}$  icon. The translation is best if there are `.aux` and `.bbl` files in the same folder as the `.tex` file to be converted. These should be generated using  $\text{\LaTeX}$  and `bibtex`.

## 2.7 Problems Compiling

The code for  $\text{\LaTeX}2\text{RTF}$  is standard ANSI C. Some possible pitfalls are

- Not correctly defining your compiler in the Makefile. The default is to use `gcc`.
- Encountering errors because the compiler options. During development all compiler warnings are turned on. However, different compilers have different interpretations of `-Wall` and may generate errors that were not found in a different development system. Please report these, but a quick fix is to remove all compiler options.

## 2.8 Problems with make check

All the files in the `test` directory are converted (with varying degrees of success) using  $\text{\LaTeX}2\text{RTF}$  and are tested before most CVS check-ins and with all released tarballs. There will be many warning messages, but there should be no actual error messages. If you do not have a working `latex2png` script, then some of the files will fail to be translated.



## 3 Using LaTeX2RTF

### 3.1 General Assumptions

LaTeX2RTF assumes that the `.tex` file you want to convert is a valid LaTeX document. The chances of a successful LaTeX2RTF conversion are slightly better than the proverbial snowball's if the `.tex` file doesn't `latex` properly. Use LaTeX to find and correct errors before using LaTeX2RTF.

To correctly convert font names you must edit the `fonts.cfg` configuration file. This file is used to specify the needed font names and how the LaTeX default font names should be converted to RTF (see [Section 5.5 \[Font Configuration\]](#), page 19). LaTeX variables and user defined commands are not evaluated. They will be simply ignored. To let LaTeX2RTF know the names of variables you can add them in the `ignore.cfg` file (see [Section 5.4 \[Ignore Command\]](#), page 18).

The environment variable `RTFPATH` may contain a search path for the support files (all files ending in `.cfg`). If no file is found during the search in the search-path or if the environment variable is not set, the compiled-in default for the configuration-file directory is used. If the files are not found at all the program aborts.

In the MS-DOS version the search path is separated by `;` in the Unix version by `:`. For the paths themselves apply `\` and `/`. A separator may appear at the beginning or ending of `RTFPATH`.

Make sure that the configuration files are in the correct directory. LaTeX2RTF will need at least `fonts.cfg`, `direct.cfg`, `ignore.cfg`, `english.cfg`. You may have to change one or more of them to suit your needs (see [Chapter 5 \[Configuration\]](#), page 17).

See [Section 8.2 \[Missing options\]](#), page 25, for actual implementations irregularities.

See [Section 8.4 \[Reporting Bugs\]](#), page 25, for information on how to reach the maintainer.

### 3.2 LaTeX2RTF Options

The LaTeX2RTF command converts a LaTeX file into RTF text format. The text and much of the formatting information is translated to RTF making the new file look similar to the original. The command line syntax is:

```
latex2rtf [-options] inputfile[.tex]
```

The `-options` may consist of one or more of the following

**-a auxfile**

specify an `.aux` file (for table and figure references) that differs from `inputfile.aux`. If this is omitted, the name of the inputfile with the suffix replaced `.aux` will be taken. You must provide both files (`.tex` and the `.aux`) to be able to convert cross-references in a LaTeX file. The `.aux` is created by running the `inputfile.tex` through `latex`.

**-b bblfile**

Unless an `bblfile` is specified with the `-b` option, LaTeX2RTF uses a `inputfile.bbl`. The `bblfile` file is used for citations and is typically created by running `inputfile.aux` through `bibtex`.

**-C codepage**

used to specify the character set (code page) used in the  $\text{\LaTeX}$  document. This is only important when non-ansi characters are included in the  $\text{\LaTeX}$  document. Typically this is done in a  $\text{\LaTeX}2_{\epsilon}$  file by using `\usepackage[codepage]{inputenc}` and in this case you need not specify the -C codepage option. If NO `\usepackage[codepage]{inputenc}` is in the  $\text{\LaTeX}2_{\epsilon}$  file, you must inform the converter about the codepage by the -C codepage option. You may select any of the following code pages: ansinew, applemac, cp437, cp437de, cp850, cp852, cp865, decmulti, cp1250, cp1252, latin1, latin2, latin3, latin4, latin5, latin9, next. The default behavior is to use ansinew (code page 1252). Cyrillic support includes conversion of koi8-r, koi8-u, cp1251, cp855, cp866, maccyr, and macukr encodings.

**-d debug\_level**

The ‘-d’ option determines the amount of debugging information to send to stderr while translating. `debug_level=0` means only Errors, ‘1’ Warning Messages (default) also. The `debug_level` can go as high as ‘7’ for insane amounts of debugging fun.

**-D dots\_per\_inch**

used to specify the number of dots per inch for equations converted to bitmaps. This value is also used when picture environments are converted to bitmaps as well as when EPS graphics are converted to png files. The default value is 300 dots per inch.

**-f#**

where # selects which fields to use during conversion:

- f0** do not use fields in RTF. This is handy when primitive RTF editors are being used to view the RTF output.
- f1** use fields for equations but not `\ref` and `\cite`.
- f2** use fields for `\ref` and `\cite` but not equations. This will be useful for versions of OpenOffice that import cross-references properly (as of Sept 2003 in a soon-to-be released version) but do not properly handle fields in equations.
- f3** use fields when possible. This is the default and is most useful when the RTF file is being exported to be used in Word. This retains the most information from the original  $\text{\LaTeX}$  file.

**-F**

use LaTeX to create bitmaps for all figures. This may help when figures are not translated properly with the default settings. This typically requires a functional version of ImageMagick on your machine to work properly.

**-h**

a short usage description

**-i language**

used to set the idiom or language used by the  $\text{\LaTeX}$  document. Typically, this is specified in a  $\text{\LaTeX}2_{\epsilon}$  document by including `\usepackage[language]{babel}` where `language` is one of the languages supported by the `babel` package. All languages listed in the `babel` system

are supported so far as translations for “Chapter,” “References,” and the like. Furthermore, some commands found in the style files for german, french, russian, and czech style are supported (see [Section 5.6 \[Language Configuration\]](#), page 19).

**-l** same as ‘`-i latin1`’ (Note that the default behavior is to use ‘`ansinew`’ which is a superset of ‘`latin1`’). Included for backwards compatibility.

**-M#** where # selects the type of equation conversion. Use

**-M1** convert displayed equations to RTF

**-M2** convert inline equations to RTF

**-M4** convert displayed equations to bitmap

**-M8** convert inline equations to bitmap

**-M16** insert Word comment field that contains the original equation text

**-M32** insert the raw latex equation delimited by `<<:` and `:>>`. This is potential useful when using version 6 of the MathType equation editor.

These switches can be combined to get different effects. Handy examples are

**-M3** convert both inline and displayed equations to RTF (default)

**-M6** convert inline equations to RTF and displayed equations to bitmaps

**-M12** convert both inline and displayed equations to bitmaps

Bitmap conversion requires that you have installed a working latex2png script. Producing bitmaps is slow.

**-o outputfile**

Unless an ‘`outputfile`’ is specified with the `-o` option, the resulting RTF filename is formed by removing ‘`.tex`’ from the ‘`inputfile`’ and appending ‘`.rtf`’.

**-p** Escape parentheses in mathematical formulas. This has no effect unless EQ fields are being generated. When this option is used, then in a ‘(’ or ‘)’ that appears in an EQ field will be preceded by a backslash. Despite documentation to the contrary (which says that all parentheses should be escaped), adding escapes usually produces a worse result than doing nothing. If Word displays some formulas with parentheses as “Error!”, you might try this option as a last resort. See also the `-S` option.

This is an option because it will break typesetting equations with non-matching parentheses (because an unmatched unquoted parenthesis would terminate the field).

**-P /path/to/cfg**

used to specify the directory that contains the `.cfg` files

**-S** used to specify that semicolons should be used to separate arguments in RTF fields (instead of commas). Typically this is needed when the machine that opens the RTF file has a version of Windows that uses ‘,’ for decimal points.

- V            prints version information on standard output and exits.
- W            includes warnings directly in the RTF file
- Z#          add the specified number of extra } to the end of the RTF file. This is useful for files that are not cleanly converted by `LATEX2RTF` .

With no arguments other than switches starting with a “-”, `LATEX2RTF` acts as a filter, i.e., it reads from `stdin` and writes to `stdout`. In addition, diagnostic messages are sent to `stderr`. If these standard channels are not redirected using `<` and `>`, then the input is read from the command line, and both output and error messages are printed on the screen.

If a non-switch argument is present, `LATEX2RTF` assumes it is the name of the input file. The file must have extension “.tex” but the extension is optional. The output file is constructed from the input file name by removing the extension “.tex” and adding “.rtf”.

### 3.3 Debugging

With the ‘-d’ option you can specify how much processing information `LATEX2RTF` reports. If there is a logfile specified the output goes to this file. Nonetheless Warnings and Errors are logged to `stderr` always.

Possible values of ‘-d’ are

0. only errors.
1. Translation Warnings (default).
2. shows preparsing of sections
3. Reasonably high level debugging messages
4. Show all function calls
5. Show all each character as it is processed
6. Show processing of characters as they are output as well

## 4 Features

In this chapter you find what styles  $\text{\LaTeX2RTF}$  is supposed to translate correctly to RTF.

### 4.1 $\text{\LaTeX2e}$

$\text{\LaTeX2RTF}$  understands most of the commands introduced with  $\text{\LaTeX2}_\epsilon$ . It translates both the old 2.09 version of `\documentstyle[options]{format#}` and the newer `\documentclass[options]{format}`.

### 4.2 Unicode Support

As of version 1.9.17,  $\text{\LaTeX2RTF}$  has limited unicode support. LaTeX files that use unicode sequences are just emitted directly into the RTF file. Symbols and odd characters in math sequences may also be converted to a unicode sequence if there is no corresponding symbol in the “Symbol” or “MT Extra” fonts.

### 4.3 Input Encoding

It is not necessary to specify the ‘-C’ option if you use `\usepackage{isolatin1}` or `\documentstyle[isolatin1]{...}`.  $\text{\LaTeX2RTF}$  automatically detects these packages/style options and switches to processing of ISO-Latin1 codes. The following encodings are supported: ansinew, applemac, cp437, cp437de, cp850, cp852, cp865, decmulti, cp1250, cp1252, latin1, latin2, latin3, latin4, latin5, latin9, next, koi8-r, koi8-u, cp1251, cp855, cp866, maccyr, and macukr. The encoding used in RTF files is cp1252. If cyrillic fonts are present, then these are represented in the RTF file using cp1251 (Windows Cyrillic).

### 4.4 Language Support

The following languages from the Babel package are supported: afrikaans, german, nynorsk, spanish, bahasa, dutch, icelandic, polish, swedish, basque, english, portuges, turkish, brazil, esperanto, irish, romanian, usorbian, breton, estonian, italian, samin, welsh, catalan, finnish, latin, scottish, croatian, lsorbian, serbian, czech, french, magyar, slovak, danish, galician, norsk, slovene,

The only thing that these files do is to translate various words usually emitted by  $\text{\LaTeX}$  during processing. For example, this ensures that the  $\text{\LaTeX2RTF}$  will provide the correct translation of the word “Chapter” in the converted document.

You can select any of the above languages using the ‘-l’ option. This is not needed if your  $\text{\LaTeX}$  file contains `\usepackage[language]{babel}`.

Encountering the ‘german’ package or documentstyle option (by H. Partl of the Viena University) makes  $\text{\LaTeX2RTF}$  behave like that: German Quotes, German Umlauts by “a, etc. . . This support is programmed directly into  $\text{\LaTeX2RTF}$  and supporting similar features for other languages will require patching the source code.

There is similar support for ‘french’ packages.

There is reasonable support for english, latin1, latin2, and cyrillic languages.

See [Section 5.6 \[Language Configuration\], page 19](#), for details on how to write a ‘language.cfg’ file for your language by yourself.

## 4.5 Cross References

Cross references include everything that you might expect and then some: bibliographic citations, equation references, table references, figure references, and section references. Section, equation, table and figure references are implemented by placing RTF bookmarks around the equation number (or table number or figure number).

Page references work but are implemented as “warm” cross-references. This means that Word does not automatically update the page references when the file is opened. To update the page references you must select the entire document (in Word) and press F9.

Bibliographic references currently require that a valid ‘.aux’ file be present. This is where LaTeX2RTF obtains the reference numbers. It would be nice if LaTeX2RTF just automatically numbered the references when there was no ‘.aux’ file, but LaTeX2RTF does not do this yet.

LaTeX2RTF relies on BibTeX to convert and format bibliographic entries. Usually the style file for a particular BibTeX format does not use any special LaTeX commands and therefore the bibliography file ‘file.bbl’ can be processed by LaTeX2RTF without difficulty. As a consequence, LaTeX2RTF can handle most bibliography styles without problem.

There are several latex style packages that add additional latex commands to enhance bibliographic formatting. LaTeX2RTF currently supports the following bibliographic packages:

apacite, apalike, authordate, harvard, natbib (also with apanat1b). These packages have many, many options and you may encounter problems with formatting in special cases.

As of LaTeX2RTF 1.9.17, the natbib command bibpunct is supported.

Footnotes are implemented and appear at the bottom of each page.

Indexing is reasonably-well supported. The simple mark-up of `makeindex`

```
\index{topic!subtopic@textit{subtopic}}
```

is supported. The rest of the fancy indexing stuff is not implemented. The index is created at the location of the `\printindex` command. When a file with an index is first opened in Word, you must select the entire file and update the page references and fields by pressing F9.

Currently, there is no support for `\labels` of `\items` in enumerate environments.

The conversion of cross-references is not perfect because of the different mechanisms in the LaTeX and Word worlds. In particular, if there are multiple `\label` in a figure, table, or section environment then only the first gets processed. It is also possible to confuse the LaTeX2RTF in eqnarray environments.

## 4.6 Page Formatting

LaTeX2RTF will handle some basic page formatting options, including `\doublespacing` (as implemented in the `setspace` package), and the margin setting options provided by the `geometry` package including commands in the `ratio`, `centering` and `margin` families. Not all geometry options are implemented yet, in part because there are no corresponding rtf commands for many of them.



## 4.7 Equations

There are five separate levels of equation translation based on the `-M` switch, see [Section 3.2 \[LaTeX2RTF Options\]](#), page 9. Each equation is now converted either to an EQ field or to a bitmap.

This is an interim solution (for some definition of “interim”). Ideally the equations would become OLE equation objects in the RTF file, but this needs to be implemented.

Some functions in the EQ fields have two or more parameters with a separator between each two. Unfortunately, the interpretation of these separators depends on the country specific settings in the MS Windows system in which the rtf file is opened. E.g. in English versions of MS Windows, the default parameter separator is the comma, in German versions the default is the semicolon. If the parameter in the RTF file does not match the Windows setting, some EQ fields are not interpreted correctly. You can check and set the separator in [Windows control panel - country settings - numbers - list separator]. By default, `latex2rtf` uses the comma as separator. If `latex2rtf` is called with the command line parameter `-S`, the semicolon is inserted as parameter delimiter.

## 4.8 Math and Special Symbols

Many mathematical and special symbols are supported by the conversion table in the file `direct.cfg`, see [Section 5.3 \[Direct Conversion\]](#), page 18. However to display (or print) them correctly from the rtf file, these symbols must be available in the fonts in the system in which the rtf file is opened.

Required fonts are:

- "Times" / "Times New Roman", preferably with Unicode extension (i.e. supporting all European languages including Cyrillic, Greek, and Hebrew). "Times" or "Times New Roman" is standard on all systems, but not its Unicode extension.
- "Symbol", which is standard on all systems.
- "MT Extra". This font is installed with the Microsoft Equation Editor, which comes with Microsoft Word, or its full version MathType. If you don't have Word or MathType, you can get this font from <http://www.mathtype.com/en/dl/fonts/>
- "Lucida Sans Unicode" which contains many symbols that aren't in the first three fonts. This is a new feature in `LATEX2RTF` version 1.9.17 and may or may not work, depending on the system in which the rtf file is opened.

If you need these symbols, don't have "Lucida Sans Unicode" but have another font containing these symbols, you can edit `direct.cfg` and `fonts.cfg` and replace the font name in both files. For a description of `direct.cfg` see [Section 5.3 \[Direct Conversion\]](#), page 18.

## 4.9 Tables

The table code is currently barely working. It needs to be rewritten.

## 4.10 Graphics

There is now rudimentary support for `\includegraphics`. Three file types will be inserted into the RTF file without needing conversion: `.pict`, `.jpeg`, and `.png` files. EPS files are

converted to PNG using `convert` from the ImageMagick package. Some options are handled properly.

## 4.11 Pagestyles

If there is no `\pagestyle` command, the RTF output is generated as with plain pagestyle, i.e. each page has its page number centered at the bottom.

You must turn this off with the `\pagestyle{empty}` command in the `LaTeX` file if you don't want pagenumbers. The headings and myheadings styles are silently ignored by now. The twosided option to the `\documentstyle` or `\documentclass` produces the corresponding RTF tokens. Note that these features require RTF Version 1.4.

## 4.12 Hyperlatex

Hyperlatex support is largely broken at the moment, but continues to improve.

Otfried Schwarzkopf has created the “Hyperlatex Markup Language” which is a “little package that allows you to use `LaTeX` to prepare documents in HTML.” It brings an Emacs lisp program with it to convert the Hyperlatex file to HTML. Hyperlatex can be obtained from the CTAN-sites, see [Section 2.2 \[Obtaining LaTeX2RTF\], page 3](#). There are two convenient commands that avoid typing: `\link` and `\xlink` that generate an “internal” label which then is used in the following `\Ref` and `\Pageref` commands.

`LaTeX` makes it possible to write `'\link{anchor}[ltx]{label}'`, which typesets: ‘anchor ltx’. `LaTeX2RTF` does NOT support this approach since the optional parameter is thrown away right now, see [Chapter 8 \[LaTeX2RTF under Development\], page 25](#).

Note that you have to update your `‘.cfg’` files if you are upgrading, since there are a lot of HTML oriented commands in Hyperlatex that we simply can ‘ignore’.

## 5 Configuration

### 5.1 Input processing

On processing input `LATEX2RTF` first converts the `LATEX` special characters. If it encounters one of the standard commands it is converted internally. If a command is not known to `LATEX2RTF` it is first looked up in ‘`direct.cfg`’ and the RTF code specified there is output. If not found there it is looked up in the section ‘`ignore.cfg`’. This file includes a lot of `LATEX` commands that do not affect the output (cross reference information and the like), or that we are not able or willing to convert to RTF.

You can use ‘`ignore.cfg`’ if you get tired of seeing

```
WARNING: command: ‘foo’ not found - ignored
```

and you don’t need ‘foo’ in your RTF document. It would be nice to send your additions to the `LATEX2RTF` mailing list for inclusion in later distributions.

`LATEX2RTF` accepts Unix, MS-DOS, and Macintosh line ending codes (`\n`, `\r\n` and `\r`). The files it creates are the line ending for the platform on which `LATEX2RTF` was compiled.

The `LATEX` file may have been created with a wide variety of character sets. If the `LATEX` lacks the `\package[codepage]{inputenc}` definition, then you may need to use the command line switch to manually select the proper code page (see [Section 4.3 \[Input Encoding\]](#), page 13).

As of `LATEX2RTF` 1.9.17, there are two options for including content that will be ignored by `LATEX` but parsed by `LATEX2RTF`. The first is simply to begin a line with `%latex2rtf:`, as in the following:

```
%latex2rtf: These lines will only appear in the latex2rtf output,
%latex2rtf: not in the LaTeX output.
```

More sophisticated processing is available with the `\iflatex2rtf` command:

```
\newif\iflatex2rtf %all \newif constructs ignored by latex2rtf
\latex2rtffalse %ignored by latex2rtf, so file works in latex

\iflatex2rtf
This code is processed only by latex2rtf
\else
This code is processed only by latex
\fi
```

Note that `\iflatex2rtf` will only work within a section; you cannot use this command to conditionally parse code that crosses section boundaries. Also, it will only work on complete table or figure environments. Due to the mechanism used by `LATEX2RTF` in processing these environments, at this time the only way to conditionally parse tables and figures is to include two complete versions of the environment in question, nested within an appropriate `\iflatex2rtf` structure.

## 5.2 Output formatting

On writing output, `LaTeX2RTF` generates `\n` as line ending code. Your RTF Reader should accept this on any platform. If you ftp your RTF file from or to MS-DOS platforms the line ending code can be converted to `\r\n`. As this should also be legal to any RTF Reader the resulting RTF rendering should not be affected.

`LaTeX2RTF` does not offer a whole lot of flexibility in how files are translated, but it does offer some. This flexibility resides in four files `'direct.cfg'`, `'ignore.cfg'`, `'fonts.cfg'`, and `'language.cfg'`. These files are documented in the next four sections.

## 5.3 Direct Conversion

The file `'direct.cfg'` is used for converting `LaTeX` commands by simple text replacement. The format consists of lines with a `LaTeX` command with backslash followed by comma. The rest of the line until a `'.'` character will be written to the RTF file when the command is found in the `LaTeX` file. Lines starting with a `'#'` character are ignored. After the `'.'` everything is ignored to end of line. To select a specific font use `*fontname*`, where `fontname` be defined in `'fonts.cfg'`. To write the `'*'` character use `**`.

```
\alpha,{\f*Symbol* a}. #alpha in the Symbol Font
\copyright,\a9.
```

Changes in version 1.9.17:

More math symbols are supported by using Unicode coding. However many of these symbols are not (yet) contained in the standard MS Windows fonts but only in "Lucida Sans Unicode". So on the system where the rtf file is opened, this font must be available, otherwise the symbols are not displayed correctly.

As it seems that this font is not commonly available or supported, symbols which are available in the standard fonts ("Times" / "Times New Roman", "Symbol", "MT Extra") are taken from these fonts (see [Section 4.8 \[Math and Special Symbols\]](#), page 15).

An alternative `direct.cfg` is available by which MT Extra is not used and the symbols which normally are taken from MT Extra are instead taken from "Lucida Sans Unicode". To use "Lucida Sans Unicode" instead of MT Extra, rename `direct.cfg` (to e.g. `direct_mt.cfg`) and rename `direct_ucs.cfg` to `direct.cfg`.

## 5.4 Ignore Command

The file `'ignore.cfg'` is used for defining how to ignore specific commands. This file is used for recognition of `LaTeX` variables, user defined variables, and some simple commands. All variables are ignored but the converter must know the names to correctly ignore assignments to variables. Lines in this file consist of a variable name with backslash, followed by comma and the type of the variable followed by `'.'`. Possible types are

- `'NUMBER'`    simple numeric value
- `'MEASURE'`    numeric value with following unit of measure
- `'OTHER'`    ignores anything to the first character after `'='` and from there to next space.  
e.g., `\setbox\bak=\hbox`
- `'COMMAND'`    ignores anything to next `'\'` and from there to the occurrence of anything but a letter e.g., `\newbox\bak`

<code>'SINGLE'</code>	ignore single command e.g., <code>\noindent</code>
<code>'PARAMETER'</code>	ignores a command with one parameter e.g., <code>\foo{bar}</code>
<code>'PACKAGE'</code>	does not produce a Warning message if <code>PACKAGE</code> is encountered, e.g., <code>'PACKAGE,kleenex.'</code>
<code>'ENVCMD'</code>	processes contents of unknown environment as if it were plain $\text{\LaTeX}$ eg. <code>'ENVCMD,environtext\end{environtext'.</code> Therefore <code>\begin{environtext\end{environtext}</code> as <code>'text'.</code>
<code>'ENVIRONMENT'</code>	ignores contents of that environment, e.g., with <code>'ENVIRONMENT,ifhtml.'</code> <code>\begin{ifhtml}text\end{ifhtml}</code> ignores <code>'text'.</code>

The types are in upper case exactly as above. Do not use spaces. Lines starting with a `'#'` character are ignored. After the `'.'` everything is ignored to end of line. Example:

```
\pagelength,MEASURE.
```

## 5.5 Font Configuration

The file `'fonts.cfg'` contains the font name mapping. For example, this file determines what font is used to represent `\rm` characters in the RTF file.

A line consists of a font name in  $\text{\LaTeX}$  followed by comma and a font name in RTF. The end is marked by a `'.'`. No spaces are allowed. The  $\text{\LaTeX}$  font will be converted to the RTF font when it is found in the  $\text{\LaTeX}$  file. If multiple translations for the same  $\text{\LaTeX}$  font are specified, only the first is used. All fonts in a  $\text{\LaTeX}$  file that are not in this file will be mapped to the default font. All RTF fonts listed in this file will be in every RTF file header whether used or not. Lines starting with a `'#'` character are ignored. After the `'.'` everything is ignored to end of line.

To add a RTF font not used as substitute for a  $\text{\LaTeX}$  font — for example a Symbol font used in `'direct.cfg'` — use a dummy  $\text{\LaTeX}$  name like in the following

```
Dummy3,MathematicalSymbols.
```

Make sure you use the correct font name. Take care of spaces in font names. The default fonts are named Roman `\rm`, Slanted `\sl`, Sans Serif `\sf`, Typewriter `\tt`, or Calligraphic `\cal`.

## 5.6 Language Configuration

The file(s) `'language.cfg'` control the translation of  $\text{\LaTeX}$ 's “hardcoded” sectioning names. The standard  $\text{\LaTeX}$  styles have some fixed Title names like `'Part'`, `'Reference'` or `'Bibliography'` that appeared in English or German in the output with the original versions of  $\text{\LaTeX}$ 2RTF.

It is unlikely that you will need to create a new `'language.cfg'` file. However, just look at one of the existing files and follow the pattern. The format is really simple.



## 6 Error Messages and Logging

As stated in the Debugging section, `LATEX2RTF` provides a means to control the amount of debugging information through the ‘`-d#`’ switch. By using a debugging level of 4, you can get a pretty good idea of what `LATEX` command caused the problem and what line that command might be found on.

### ‘Fatal error messages’

indicate a bug in the source code. PLEASE report them, if they do not appear in the documentation (see [Section 8.4 \[Reporting Bugs\]](#), page 25).

### ‘Error messages’

always abort the program and are caused by conditions that prevent further conversion of the input file. Typically this is caused by `LATEX2RTF` getting hopelessly confused by the number of braces in the `LATEX` file.

### ‘Warning messages’

inform you, that there is some conversion loss from `LATEX` to RTF, or that the output file has some restrictions on some RTF Readers. Most of these warnings can be suppressed by adding the offending command to the ‘`ignore.cfg`’ file.

Error and Warning messages should follow the GNU Coding standards, i.e. they have the format

```
inputfile':line: Error|Warning: message
```

You can also control the level of debugging output by inserting `\verbositylevel{#}` in the `LATEX` file. This is very handy if you have a large `LATEX` file that is failing in only a small section. For example,

```
problem free latex file ....
\verbositylevel{5}
problematic code
\verbositylevel{0}
```

will cause a huge amount of debugging information to be emitted for the problematic code.

Error reporting and logging still has many inconsistencies, but it gets better with each release. Don't try to make any sense in debugging levels above 4, these are for my own delight only and can change significantly between versions.

The ‘`inputfile`’ may be incorrectly identified if it is incorporated through `\input` or `\include`. The line may be also be wrong at times. See [Section 8.3 \[Known Bugs\]](#), page 25.





## 7 History & Copyright

In 1994 the first Version of  $\text{\LaTeX}2\text{RTF}$  was written by Fernando Dorner and Andreas Granzer of the Viena University supervised by Ralf Schlatterbeck in a one-semester course. They created a simple  $\text{\LaTeX}$  parser and added most of the infrastructure for the program. This was version 1.0 of  $\text{\LaTeX}2\text{RTF}$ . In 1995, work on  $\text{\LaTeX}2\text{RTF}$  was continued in another one-semester course by Friedrich Polzer and Gerhard Trisko. The result was  $\text{\LaTeX}2\text{RTF}$  version 1.5. Ralf Schlatterbeck (ralf "at" zoo.priv.at) maintained and extended  $\text{\LaTeX}2\text{RTF}$  until 1998.

In 1998 Georg Lehner (jorge.lehner "at" gmx.net) found the reference to  $\text{\LaTeX}2\text{RTF}$  on the [TeX Conversion Webpage](#) of Wilfried Hennings and added some functionality and took over the maintainence of the program. The last version release by Georg is 1.8aa. The bulk of development post 1.8aa was done by Scott Prahl. Wilfried Hennings now coordinates the development of the program and maintains the project on [SourceForge](#) where there are also (low volume) mailing lists for users and developers. Mailing via one of these lists requires subscription to the list (to prevent spam). For subscription to these lists visit the page: [users list](#) or [developers list](#)

As of October 2004, version 1.9.16 of  $\text{\LaTeX}2\text{RTF}$  is available. One day there shall be a jump to Version 2.0, but this is not history but future ...

The contents of this manual were composed by copying shamelessly what was available in the original sources and documentation.



## 8 LaTeX2RTF under Development

### 8.1 Unimplemented Features

- LaTeX2RTF ignores some optional parameters of `\documentstyle`
- Add the code to produce the corresponding chapter, section, and page numbering with headings and myheadings pagestyles. Implement `\markboth` and `\markright`.
- To support `\tableofcontents` there would be two approaches: Transfer sectioning information, title text and then produce page numbers by the rtf- reader. Scan and label all of the sectioning commands while reading and then construct the sectioning information using these labels. Needs two passes on LaTeX input.

### 8.2 Missing options

Missing or buggy command line options.

- ‘-d’            Information logging and Error reporting is not implemented consistently. Need to test and track problems with the linenumber and with the file name.
- ‘--long\_names’            It would be useful to implement the GNU long option names, e.g.: ‘-debug’, ‘-output\_file’, ‘-quiet’, etc. This could be done by switching to the GNU getopt package.

### 8.3 Known Bugs

1. The first parameter of a `\link{anchor}[ltx]{label}` is converted to the rtf-output. Label is stored to hyperref for later use, the optional parameter is ignored. [ltx] should be processed as Otfried recommends it, to use for exclusive LaTeX output.e.g: `\link{readhere}[\~\Ref]{explaining: chapter}`. Since {explaining:chapter} is yet read by LaTeX and hyperlatex when [...] is evaluated it produces the correct reference. LaTeX2RTF is only strolling from left to right through the text and can't remember what she will see in the future.
2. The diagnostics routine does not output the correct (actual) inputfilename. (‘.aux’, ‘.bbl’, `\input`).

### 8.4 Reporting Bugs

Report bugs to to the bug tracking system at [SourceForge](#). Only report bugs for the latest version of LaTeX2RTF that is available. Please provide the following information and observe the following guidelines when reporting a bug in the program:

1. State the version of LaTeX2RTF that you are using. You can get the version by specifying the ‘-V’ option to LaTeX2RTF .
2. Specify the your operating system and version. Be sure to check the file ‘Makefile’ for settings that may be specific to your machine, especially for some versions of SunOS there may be settings which are needed to compile successfully. Do this before submitting a bug report.

3. If the program produces wrong output or does not work for you, include a short  $\text{\LaTeX}$  file along with a description of the problem. Isolating the bug into a small  $\text{\LaTeX}$  file does two things. First, it provides a file that can be used to test future versions of  $\text{\LaTeX2RTF}$  and second, it certainly improves the chances that the bug will get some attention. Do not send me large  $\text{\LaTeX}$  or RTF files, I simply do not have the time to wade through large files to search for a bug!
4. Be patient. I am maintaining the program in my free time. I did not write most of the code. Often I do not have the time to answer to your question. I will, however, try to fix reported bugs in upcoming releases.

## 8.5 Todo List

Scott's ToDo list

- Use lex/yacc to implement getSection
- Add support for pagestyle
- Improve graphic/graphicx support
- Better support for ignoring commands

Georg's todo list

- Make this Manual more consistent, the ToDo and Known Bug List shorter and the Features List longer.
- Harmonize all of the error and warning messages.
- Put warnings everywhere applicable about producing RTF 1.4 tokens.
- Provide an Error and Warning recovery guide to the user.
- Add a chapter with lists of all  $\text{\LaTeX}$  commands that convert, and that do not convert to RTF, including their status (for future releases, never, partially functional, ...).

## 9 Index

### %

`%latex2rtf` ..... 17

### A

`apacite` ..... 14  
`apalike` ..... 14  
`apanat1b` ..... 14  
`authdate` ..... 14

### B

`babel` ..... 13  
`bibliography` ..... 14  
`bibliography, apacite` ..... 14  
`bibliography, apalike` ..... 14  
`bibliography, apanat1b` ..... 14  
`bibliography, authdate` ..... 14  
`bibliography, BibTeX` ..... 14  
`bibliography, harvard` ..... 14  
`bibliography, natbib` ..... 14  
`BibTeX` ..... 14

### C

`conditional parsing` ..... 17  
`Copyright issues` ..... 23

### D

`direct.cfg` ..... 17  
`double spacing` ..... 14

### E

`equations` ..... 15

### F

`footnotes` ..... 14

### G

`geometry package` ..... 14

`graphics` ..... 15

### H

`harvard` ..... 14  
`History of the program` ..... 23  
`hyperlatex` ..... 16

### I

`iflaxtextortf` ..... 17  
`ignore.cfg` ..... 17  
`index` ..... 14  
`input processing` ..... 17

### L

`language support` ..... 13  
`line spacing` ..... 14

### M

`margins` ..... 14  
`math` ..... 15

### N

`natbib` ..... 14

### P

`page formatting` ..... 14  
`pagetypes` ..... 16

### S

`setspace package` ..... 14  
`special symbols` ..... 15

### T

`tables` ..... 15



# Table of Contents

<b>1</b>	<b>Introduction</b>	<b>1</b>
<b>2</b>	<b>Installation</b>	<b>3</b>
2.1	General	3
2.2	Obtaining LaTeX2RTF	3
2.3	UNIX	3
2.4	DOS or Windows	4
2.5	Win32 systems	6
2.6	Macintosh	7
2.7	Problems Compiling	7
2.8	Problems with <code>make check</code>	7
<b>3</b>	<b>Using LaTeX2RTF</b>	<b>9</b>
3.1	General Assumptions	9
3.2	LaTeX2RTF Options	9
3.3	Debugging	12
<b>4</b>	<b>Features</b>	<b>13</b>
4.1	LaTeX2e	13
4.2	Unicode Support	13
4.3	Input Encoding	13
4.4	Language Support	13
4.5	Cross References	14
4.6	Page Formatting	14
4.7	Equations	15
4.8	Math and Special Symbols	15
4.9	Tables	15
4.10	Graphics	15
4.11	Pagestyles	16
4.12	Hyperlatex	16
<b>5</b>	<b>Configuration</b>	<b>17</b>
5.1	Input processing	17
5.2	Output formatting	18
5.3	Direct Conversion	18
5.4	Ignore Command	18
5.5	Font Configuration	19
5.6	Language Configuration	19
<b>6</b>	<b>Error Messages and Logging</b>	<b>21</b>

<b>7</b>	<b>History &amp; Copyright .....</b>	<b>23</b>
<b>8</b>	<b>LaTeX2RTF under Development .....</b>	<b>25</b>
8.1	Unimplemented Features .....	25
8.2	Missing options .....	25
8.3	Known Bugs .....	25
8.4	Reporting Bugs .....	25
8.5	Todo List .....	26
<b>9</b>	<b>Index .....</b>	<b>27</b>