

L^AT_EX2RTF

A converter from L^AT_EX to RTF
Edition 0.3

Georg Lehner,
updated by Wilfried Hennings and Scott Prahl

Copyright © 1998-2001 Georg Lehner, Wilfried Hennings, Scott Prahl

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 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 \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](#) maintained and extended $\text{\LaTeX}2\text{RTF}$ until 1998.

In 1998 [Georg Lehner](#) 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. 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 latex2rtf-users@lists.sourceforge.net and developers latex2rtf-developers@lists.sourceforge.net. For subscription to these lists: mail_to_latex2rtf-users-request@lists.sourceforge.net?subject=subscribe or mail_to_latex2rtf-developers-request@lists.sourceforge.net?subject=subscribe

As of December 2001, version 1.9.12 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.

2 Introduction

`LATEX2RTF` is a translator program from `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 is somewhat ambiguous in places and Microsoft ignores various parts of the standard that are inconvenient, but RTF is widely used by many WYSIWIG text editors and is supported by Microsoft Word and Word for Windows. RTF can be exported and/or imported by several textprocessors.

`LATEX2RTF` translates the text and much of the formatting information from `LATEX` to RTF, but the typeset output is not nearly as good as what you would get from using `LATEX` directly. So, why bother translating? There are three purposes:

1. You use `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 `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 maybe have both things, `LATEX` and MS-..., or you don't. But you like the way how `LATEX` and friends work, and you don't want to type in a letter to your friends with about 345 characters and end up with a ‘.doc’ file of 32,845 byte. So you edit your documents either with ‘edit.com’ or ‘edlin’ in the structured form that `LATEX` encourages, and then you use MS-..., or MS-...-Viewer or any other RTF-rendering software to print out your file. Yes, this somewhat bizarre, but we are all somewhat bizarre, are we not?

There are drawbacks to the conversion process. In fact, don't expect any `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. `LATEX2RTF` 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 `LATEX` file will be translated to RTF satisfactorily by `LATEX2RTF` — This was sort of disclaimer, ok? OK!

`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 `LATEX` are much more flexible and standardized than in RTF, only a small subset of commands has been implemented to date. See [Section 9.1 \[Unimplemented Features\]](#), page 23.

Some of the capabilities of `LATEX2RTF` are restricted in scope or buggy. See [Section 9.3 \[Known Bugs\]](#), page 23.

RTF is a moving target, because Microsoft does not stop inventing new extensions and features for it. So you cannot view newer RTF files with older word Processors. `LATEX2RTF` eventually will generate RTF output that is unreadable with your Program. 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 look at the RTF-Tools by Paul DuBois, and the corresponding newsgroups, etc. <http://www.primate.wisc.edu/>

3 Availability

$\text{\LaTeX}2\text{RTF}$ is available for many Unix Platforms, for the Macintosh, and for MS-DOS, including all versions of MS Windows 95.

The MS-DOS version (1.8aa and up) will also run under all MS Windows versions. It requires an i386 processor or better. The Win32 Version requires [Cygnum-Cygwin32](#). As the MS-DOS version (1.8aa and up) will also run under all MS windows versions, the Win32 version has been discontinued.

Subsequent to 1.8aa, [Scott Prahl](#) has fixed many bugs, improved handling of tabular environments, significantly improve handling of equations, added macro expansion of `\newcommand` definitions, improved handling of cross references, and added some support for `\includegraphics` files.

You find the latest version on [SourceForge](#) and – with some delay – on CTAN sites: e.g., <http://www.dante.de> or <http://www.ctan.org>.

There are a couple of persons working on the $\text{\LaTeX}2\text{RTF}$ revival, coordinated by [Wilfried Hennings](#). Contact the `mailing_list` for the latest news.

4 Installing LaTeX2RTF

4.1 General

4.2 UNIX

The documentation of the program is found in the ‘doc/’ directory in the file ‘latex2rtf.info’ in the GNU info format. For your convenience, you can find ‘.html’ and ‘.pdf’ versions of the manual there as well.

To install (on a UNIX system)

1. Edit ‘Makefile’ for your local configuration. The install part supports multiple directories to install to, you will normally only need one. Be sure to correctly configure the ‘LIBDIR’ variable to the directory where support files (ending in ‘.cfg’) will be found by the program. This is normally the same as the ‘LIBINSTALL’ variable.
2. `make`
3. 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.
4. `make install`
5. If you have problems with `make install` (for example if your `mkdir` doesn’t support the ‘-p’ option) and you do not need multiple install targets, try: `make simple_install`
6. Define the environment variable `RTFPATH`. This is typically `/usr/local/lib/latex2rtf`
7. Run `make test` to test your installation

Open the ‘latex2rtf’ directory and type

```
make
make test
sudo make install
```

4.3 Windows

The file ‘L2R.BAT’ is for use with the DJGPP generated version ‘LATEX2RT.EXE’.

The UNIX and Mac distributions do not contain an executable for DOS / Windows. Instead, get the DOS / Windows port as file ‘latex2rtf-x.xx_dos.zip’ (where x.xx is the version number) from <http://sourceforge.net/projects/latex2rtf/>

4.4 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 UNIX installation.

There is a classic MacOS PPC port of the 1.9k of $\text{\LaTeX}2\text{RTF}$ and I have made a binary distribution of this application. Unfortunately, because I (Scott Prah) do all development

under MacOS X, the binaries for the Classic version often lag (far) behind the current UNIX version.

To convert a \LaTeX file, just drag the file onto the \LaTeX2RTF application 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 \LaTeX and `bibtex`.

4.5 Problems Compiling

The code for \LaTeX2RTF 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 `-pedantic` and so 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.
- Not defining `RTFPATH`.
- Not defining `HAS_NO_STRDUP` in the Makefile when your system lacks this facility.

4.6 Problems with make test

All the files in the `test` directory are converted (with more or less success) using \LaTeX2RTF and are tested before each CVS check-in and with all released tarballs. If you have successfully compiled \LaTeX2RTF then problems are probably caused by

- Not defining `RTFPATH`.
- Not defining `HAS_NO_STRDUP` in the Makefile when your system lacks this facility.

5 Using LaTeX2RTF

5.1 General assumptions made by LaTeX2RTF

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 7.5 \[Font Configuration\], page 18](#). 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 7.4 \[Ignore Command Configuration\], 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 7 \[Configuring LaTeX2RTF\], page 17](#).

See [Section 9.2 \[Missing and faulty command line options\], page 23](#), for actual implementations irregularities.

See [Section 9.4 \[Reporting Bugs\], page 24](#), for information on how to reach the maintainer.

5.2 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]
```

If `inputfile` is not specified, standard input is read. 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’.
- d**
 The ‘-d’ option determines the amount of debugging information to send to stderr while translating. ‘0’ means only Errors, ‘1’ Warning Messages (default) also. These numbers can go as high as ‘7’.
- i language**
 used to set the idiom or language used by the `LATEX` document. Typically, this is specified in a `LATEX2 ϵ` 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. Specific support for specific options of the german style package are explicitly supported. The french translated names are found in the ‘french.cfg’ file in the ‘cfg/’ directory. See [Section 7.6 \[Language Configuration\]](#), 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.
- 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’.
- C codepage**
 used to specify the character set (code page) used in the `LATEX` document. This is only important when non-ansi characters are included in the `LATEX` document. Typically this is done in a `LATEX2 ϵ` file by using `\usepackage[codepage]{inputenc}` This command is not needed if the above command is already in the `LATEX2 ϵ` file. 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`, and `next`. The default behavior is to use `ansinew` (code page 1252).
- 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` .

5.3 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

6 Features

In this chapter you find what styles `LATEX2RTF` is supposed to translate correctly to RTF.

6.1 LaTeX2e

`LATEX2RTF` understands most of the commands introduced with `LATEX 2ε`. It supports both the old 2.09 version of `\documentstyle[options]{format#}` and the newer `\documentclass[options]{format}`.

6.2 Input Encoding

It is not necessary to specify the ‘-C’ option if you use `\usepackage{isolatin1}` or `\documentstyle[isolatin1]{...}`. `LATEX2RTF` automatically detects these packages/style options and switches to processing of ISO-Latin1 codes.

6.3 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 `LATEX` during processing. For example, this ensures that the `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 `LATEX` file contains `\usepackage[language]{babel}`.

Encountering the ‘german’ package or documentstyle option (by H. Partl of the Viena University) makes `LATEX2RTF` behave like that: German Quotes, German Umlauts by "a, etc. . . . This support is programmed directly into `LATEX2RTF` and supporting similar features for other languages will require patching the source code.

See [Section 7.6 \[Language Configuration\], page 19](#), for details on how to write a ‘`language.cfg`’ file for your language by yourself.

6.4 Cross References

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

Ideally, section references would have been done the same way, but the crux of the problem is that `LATEX` allows you to tag a section with `\label` anywhere in the section. To bookmark the section number, we need to know if any `\label` occurs in that section

before the section is processed. A plausible alternative would be to insert a bookmark when `\label` is encountered with the proper number, and then cross-reference these bookmarks. Unfortunately, if the bookmarks are hidden, then the cross-reference is also hidden. So this did not work.

So the choices were

1. to insert a visible bookmark in the document at the place where the `\label`
2. to use the cross-reference information available in the `.aux` file and lose the link. This is not too bad when the section numbering is not automatic.
3. to process the `LATEX` file twice so that the section labels will be known when the section header is processed. This is hard and breaks using `LATEX2RTF` as pipe.
4. to process each section, subsection, subsubsection as a unit.

I opted for the last because this allows `LATEX2RTF` to bookmark the number in the section header avoids the drawbacks of the other options. This was harder to code, and possibly makes things a bit more fragile.

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.

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

Indexing is reasonable 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.

6.5 Equations

Equations are now converted to RTF fields. This is an interim solution. Ideally the equations would become OLE equation objects in the RTF file, but this needs to be implemented.

It would be nice to have four separate levels of equation translation

1. The first level would just do a simple text translation without any special formatting
2. The second level (which is what is currently implemented) would be to translate equations using fields. These are not readily editable and Word does not do a good job converting these to equation objects

3. The third level would be to fully convert the \LaTeX equations to Word equation objects. The API for this has been published, but is difficult because OLE encoding needs to be written.
4. Finally, the fourth translation would be to convert the equations to images and embed these images in the RTF file. This is what `ltx2rtf` does and the results are not bad.

6.6 Tables

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

6.7 Graphics

One day I will get around to incorporating all graphics formats into the RTF file. This will probably be done using `imagemagick` to do the actual translation to formats that are supported by Word and then embedding the supported format in RTF.

6.8 Pagestyles

If there is no `\pagestyle` command, the RTF output is generated as with plain pagestyle, i.e. each page get's 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.

6.9 Hyperlatex

Hyperlatex support is largely broken at the moment.

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 [Chapter 3 \[Availability\]](#), page 5. There are two handsome 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 aproach since the optional parameter is thrown away right now, See [Chapter 9 \[LaTeX2RTF under Development\]](#), page 23.

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’.

7 Configuring LaTeX2RTF

7.1 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 6.2 \[Input Encoding\]](#), page 13.

7.2 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.

7.3 Direct Conversion Configuration

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.
```

7.4 Ignore Command Configuration

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., <code>\setbox\bak=\hbox</code>
‘COMMAND’	ignores anything to next ‘\’ and from there to the occurrence of anything but a letter e.g., <code>\newbox\bak</code>
‘SINGLE’	ignore single command e.g., <code>\noindent</code>
‘PARAMETER’	ignores a command with one parameter e.g., <code>\foo{bar}</code>
‘PACKAGE’	does not produce a Warning message if PACKAGE is encountered, e.g., ‘PACKAGE,kleenex.’
‘ENVCMD’	processes contents of unknown environment as if it were plain \LaTeX eg. ‘ENV-CMD,envron.’ Therefore <code>\begin{envron} text \end{envron}</code> as ‘text’.
‘ENVIRONMENT’	ignores contents of that environment, e.g., with ‘ENVIRONMENT,ifhtml.’ <code>\begin{ifhtml} text \end{ifhtml}</code> ignores ‘text’.

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.
```

7.5 Font Configuration

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

A line consists of a font name in \LaTeX followed by comma and a font name in RTF. The end is marked by a ‘.’. No spaces are allowed. The \LaTeX font will be converted to the RTF font when it is found in the \LaTeX file. If multiple translations for the same \LaTeX font are specified, only the first is used. All fonts in a \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 \LaTeX font — for example a Symbol font used in ‘`direct.cfg`’ — use a dummy \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`.

7.6 Language Configuration

The file(s) `'language.cfg'` control the translation of LaTeX's "hardcoded" sectioning names. The standard 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 LaTeX2RTF.

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.

8 Error Messages and Logging of LaTeX2RTF Activity

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 9.4 \[Reporting Bugs\]](#), page 24.

‘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 9.3 \[Known Bugs\]](#), page 23.

9 LaTeX2RTF under Development

9.1 Unimplemented Features

- `\LaTeX2RTF` ignores some of the optional parameters of `\documentstyle`
- Need to finish code page support. Typically characters that need to be constructed using RTF `\field` commands are not implemented.
- Add the code to produce the corresponding chapter, section informations 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 let produce pagenumber 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.
- Include the GNU gettext package to internationalize `\LaTeX2RTF` 's Messages.
- Switch over to GNU `getopt()` and long options.

9.2 Missing and faulty command line options

In this section you find comments about missing and buggy command line options.

- ‘-V’ The version information output is not compatible with the GNU Coding Standards.
- ‘-d’ Information logging and Error reporting is not implemented consistently. Need to test and track problems with the `linenumber` and with the file name.
- ‘-?’ There should be an option to intersperse RTF-Output with the `\LaTeX` input that produced it to aid debugging.
- ‘-q’ There should be a ‘-q’ (quiet) option, to suppress Warning Messages. By now this can be achieved by the ‘-d0’ option.
- ‘-rmajor.minor’
 There should be an option that restrict the generation of RTF code with version greater than major,minor. Actually this is done at compile time. There are some Warning messages if “newer” RTF Code is generated, but it is not consistent at all.

It would be useful to implement the GNU long option names, e.g.: ‘-debug’, ‘-output_file’, ‘-quiet’, etc.

9.3 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', \backslash input).

9.4 Reporting Bugs

Report bugs to [Georg Lehner](#)). Ralf Schlatterbeck is no longer the maintainer. 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 \LaTeX file along with a description of the problem. Isolating the bug into a small \LaTeX file does two things. First, it provides a file that can be used to test future versions of \LaTeX2RTF and second, it certainly improves the chances that the bug will get some attention. Do not send me large \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.

9.5 Todo List

Scott's ToDo list

- Improve \backslash input
- Add support for pagestyle
- Add better graphic/graphicx support
- Convert equations to MathType OLE objects
- Create bitmaps of equations

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 \LaTeX commands that convert, and that do not convert to RTF, including their status (for future releases, never, partially functional, ...).

The following comes from Ralf:

- Redesign the input routines to not use `lseek` so that `LATEX2RTF` can read from pipes.
- Change the Makefile to use default rules for generating `.o` files.
- Also use generic rules in some other places (clean target)
- For ignored commands the number of arguments to ignore should also be given
- Environment definition produced with `\newtheorem` should be supported
- Change how the current version is computed (currently `version.h` defines the version of the whole package this should change to a CVS (or RCS) version tag from which a `version.h` file should automatically be generated).

10 Index of Commandline and Configuration file Options

(Index is nonexistent)

Table of Contents

1	History & Copyright	1
2	Introduction	3
3	Availability	5
4	Installing LaTeX2RTF	7
	4.1 General	7
	4.2 UNIX	7
	4.3 Windows	7
	4.4 Macintosh	7
	4.5 Problems Compiling	8
	4.6 Problems with <code>make test</code>	8
5	Using LaTeX2RTF	9
	5.1 General assumptions made by LaTeX2RTF	9
	5.2 LaTeX2RTF Options	9
	5.3 Debugging	10
6	Features	13
	6.1 LaTeX2e	13
	6.2 Input Encoding	13
	6.3 Language Support	13
	6.4 Cross References	13
	6.5 Equations	14
	6.6 Tables	15
	6.7 Graphics	15
	6.8 Pagestyles	15
	6.9 Hyperlatex	15
7	Configuring LaTeX2RTF	17
	7.1 Input processing	17
	7.2 Output formatting	17
	7.3 Direct Conversion Configuration	17
	7.4 Ignore Command Configuration	18
	7.5 Font Configuration	18
	7.6 Language Configuration	19
8	Error Messages and Logging of LaTeX2RTF	
	Activity	21

9	LaTeX2RTF under Development	23
9.1	Unimplemented Features	23
9.2	Missing and faulty command line options	23
9.3	Known Bugs	23
9.4	Reporting Bugs	24
9.5	Todo List	24
10	Index of Commandline and Configuration file Options	27