rtf2LaTeX 2$_\varepsilon$ Documentation
version 1.0 beta

Ujwal S. Sathyam
setlur@bigfoot.com

April 16, 2009

1 Introduction

rtf2LaTeX 2$_\varepsilon$ is an RTF→LaTeX converter that takes as its input RTF files produced by Microsoft Word and comparable word processors such as Star Office and generates a TeX-able "**.tex**" file. It has the capability to handle fairly complex RTF files containing figures, tables, and starting from version 1.0 equations (courtesy Steve Swanson, Mackichan Software). **rtf2LaTeX 2$_\varepsilon$** is written using standard *C* and should compile on any platform supporting a *C* compiler. I have tested it on the Macintosh, Linux (Intel), Linux (PowerPC), Windows 95/NT, and Solaris.

rtf2LaTeX 2$_\varepsilon$ uses the generic `RTF reader` framework by Paul DuBois. The framework is a general purpose tool for processing RTF files and may be configured in a well-defined manner to allow it to be used with a variety of writers generating different output formats. This provides a method for generating RTF-to-XXX translators. Essentially, **rtf2LaTeX 2$_\varepsilon$** provides the LaTeX 2$_\varepsilon$ writer code to the RTF reader. Paul seems to have stopped developing the reader code, therefore I had to adapt it quite a bit to handle the latest version of RTF.

What you will get

If you expect a WYSIWYG reproduction of your RTF file, you may be disappointed. My main concerns have been translating the essential features of the RTF file such as characters, figures, tables, and equations. I have largely ignored visual formatting such as ruler positions, tabs (until I figure out a good way of doing this), paragraph indentations, and other fluff. I

have always expected the output LaTeX 2$_\varepsilon$ file to require manual editing to put the finishing touches. I just want to make that task a little easier. In my opinion, expecting a WYSIWYG reproduction is not practical and misses the point entirely.

2 Installation

Compiled binaries are available for Macintosh and Windows systems. Users of Unix/Linux systems will have to compile the source code, which is a fairly straightforward process.

2.1 Macintosh

The Macintosh distribution includes a FAT drag-and-drop application on which you can drop multiple RTF files. The output LaTeX files will be created in the same directory as the input RTF files. If Mac users want to build their own binaries, I have included the CodeWarrior 5.3 project file. The project uses the DropUnix application framework (`http:/www.zenspider.com`) that is included in the distribution. If you have CodeWarrior 5.3, you should be able to open the project file and issue a "Make" command. For drag and drop operation, the input RTF files need to be of type 'TEXT' or 'RTF ' (note the space).

2.2 Unix/Linux

There is a *Unix* directory that contains scripts for configuring, building, and installing **rtf2LaTeX 2$_\varepsilon$**. Change to the *Unix* directory and type:

./configure
make

This will compile the sources and create a binary called **rtf2latex2e.bin** in the parent distribution directory. You can optionally issue a "make clean" command to remove the object files.

You can choose to install the binary in a convenient location. If you want to install into the default directory /usr/local/rtf2latex2e, you will need to become super-user at this point. Installation is done by:

make install (as root)

The default installation directory is /usr/local/rtf2latex2e. A symbolic link /usr/bin/rtf2latex2e is created pointing to $(INSTALL_DIR)/rtf2latex2e.bin. If there is already an existing **rtf2LaTeX 2$_\varepsilon$** installation, it will rename that directory to "rtf2latex2e.old". If you do not have super-user privileges,

you can edit the Makefile and change the INSTALL_DIR to somewhere in your home directory, say $(HOME)/rtf2latex2e. **Make sure that the INSTALL_DIR path ends with "rtf2latex2e".**

Finally, the environment variable RTF2LATEX2E_DIR will need to be set from within your shell. The variable has to point to the directory into which rtf2latex2e was installed. You can set the variable using
export RTF2LATEX2E_DIR=directory (bash) or
setenv RTF2LATEX2E_DIR directory (csh)
in your .bashrc or .login file, whichever is read by your shell.

You can also optionally install the ImageMagick image manipulation package available at `http://www.wizards.dupont.com/cristy`. If this is installed, **rtf2LaTeX 2$_\varepsilon$** will use the ImageMagick utility "convert" to attempt to convert embedded PNG, JPEG, and PICT images to EPS. This support is statically built into the Mac and Windows binaries.

2.3 Windows

Windows users get a pre-compiled binary of **rtf2LaTeX 2$_\varepsilon$** to be run from the MS-DOS prompt. Just run it with the RTF file to be converted as the argument.

3 Use

On the Macintosh, drop your RTF file onto the application. The output LaTeX 2$_\varepsilon$ file will be generated in the same folder as the input RTF file. On Linux/Unix and DOS, you have to run the program in a shell:
rtf2latex2e $< rtfFileName >$
If the file name contains spaces, enclose the path in double quotes.

3.1 Command-line options

No very useful command line options are supported yet. There are the obvious "-h" for help and "-v" for version number. The only other supported option is "-t <output-map-file>" where you can specify an output map file other than the default "TeX-map".

3.2 The r2l-pref preference file

rtf2LaTeX 2$_\varepsilon$ reads a preference file "*r2l-pref*" where you can specify various options such **ignoreRulerSettings**, **ignoreColor**, etc. The options

are self-explanatory. There are also some Macintosh-specific options at the end for converting embedded PICT images to EPS. The PICT→EPS conversion routine requires the Apple Laserwriter driver to be installed. All the options in the "*r2l-pref*" file are available as a "Preferences" menu item in the Macintosh application. In the Unix/Linux and Windows versions, the default "*r2l-pref*" file in the installation directory can be overridden by a file with the same name in the current working directory.

3.3 The output map file

The default output map file is "TeX-map". The output map file determines the LaTeX 2$_\varepsilon$ representation of characters. You could use a different output map file, e.g. "TeX-map.applemac" that generates characters in accordance to the Macintosh character set. Obviously, such a LaTeX file might need to use an appropriate "input" encoding package. Packages required by an output map file can be specified in the file itself, so that the packages are always loaded when the output map file is used. For example, the "TeX-map.applemac" file specifies "%\usepackage[applemac]{inputenc}". The percent sign is not a comment, but tells **rtf2LaTeX 2$_\varepsilon$** to treat the following string as a output map qualifier. You can add other qualifiers such as "%\usepackage[T1]{fontenc}".

3.4 The r2l-head file

rtf2LaTeX 2$_\varepsilon$ also reads a file (if present) called "*r2l-head*". In this file, you can specify any additional packages that you want to use in your LaTeX 2$_\varepsilon$ file, e.g. a babel hyphenation package or a font encoding. The contents of this file are just copied into the preamble of the LaTeX 2$_\varepsilon$ file. In the Unix/Linux and Windows versions, the default "*r2l-head*" file in the installation directory can be overridden by a file with the same name in the current working directory.

3.5 The r2l-map file

rtf2LaTeX 2$_\varepsilon$ also reads a file (if present) called "*r2l-map*". In this file, you can customize options for the document class in the LaTeX 2$_\varepsilon$ file and mappings for section headings. You can also specify how text style is to be handled in the LaTeX 2$_\varepsilon$ file, e.g. "\textbf" vs. {\bf ...}. In the Unix/Linux and Windows versions, the default "*r2l-map*" file in the installation directory can be overridden by a file with the same name in the current working directory.

4 Features

rtf2LaTeX 2$_\varepsilon$ is designed to convert journal articles, reports, and letters written in Microsoft Word. That means I would like it to handle the following:

- **Text Style:** Some amount of stylized text like color, **bold**, *italic*, underlined, and relative size like small, normal, big, very big, and large. This is a little weak in older RTF files as the older RTF spec is a little crappier than the newer one. All other font information is disregarded, as TeX can do better anyway.

- **Figures:** **rtf2LaTeX 2$_\varepsilon$** can read figures of format PICT, WMF, PNG, and JPEG embedded into RTF files. These are the most common formats encountered in RTF files. When **rtf2LaTeX 2$_\varepsilon$** encounters an embedded figure, it reads out the figure into a separate file. The output format of the figure is the same as the format it is embedded in. **rtf2LaTeX 2$_\varepsilon$** will then attempt to convert the image to EPS using both internal and external conversion routines. For PNG and JPEG images, the ImageMagick (`http://www.wizards.dupont.com/cristy`) package is used to generate EPS files. PICT images on non-Mac systems are also converted to EPS using ImageMagick, but the performance can be quite poor. On Macs, you have the option of using either the Laserwriter driver for EPS generation or ImageMagick. Generally, using the Laserwriter yields better results.

  The ImageMagick support is statically built into the binaries for the Mac and Windows platforms. **rtf2LaTeX 2$_\varepsilon$** for Unix and Linux will attempt to externally call the ImageMagick routine "convert" if available. You will need to install ImageMagick separately for EPS conversion on these platforms.

  rtf2LaTeX 2$_\varepsilon$ for the Macintosh also converts embedded WMF files to PICT format. All major implementations of TeX on the Macintosh can handle PICT images. Most of the image conversion code was written by `Scott Prahl`.

- **Equations:** The most common source of the RTF file is Microsoft Word. Equations in Word are created in Equation Editor (MathType), and when saved into an RTF file, the equation is embedded as an OLE object. **rtf2LaTeX 2$_\varepsilon$** uses the free `cole` library to extract the

embedded equations from the OLE structured format. The equation is then converted into LaTeX format. Only equations created by Equation Editor 3.0 (supplied along with MS Word) have been tested. MS Word also embeds the equation as a picture for older RTF readers. If the native equation conversion fails, or if the option to convert equations is disabled, **rtf2LaTeX 2$_\varepsilon$** reads that picture and outputs the equation as a picture file. The equation converter capability was provided by Steve Swanson from Mackichan Software (`http://www.mackichan.com`), makers of Scientific Word and Workplace.

- **Tables:** Yeah, it does tables!! However, this is the weakest link in the chain and the messiest part of the code. This is largely due to the fact that RTF does not have a separate 'Table' group. It is also due to the fact that TeX likes to know in advance the number of columns in the table, and RTF does not tell us that. I spent a lot of time to support tables to this extent. A lot of the test files have tables in them. To get an idea of the type of tables that **rtf2LaTeX 2$_\varepsilon$** can handle, take a look at **table1.rtf**, **Script.rtf**, and **RTF-Spec.rtf**. I use longtable.sty package for generic table handling to take care of tables that span several pages.

- **Paragraph Style:** I care for alignment issues like centering, left, and right justification. Useful in letters. All other visual formatting like indentation is currently ignored until I figure out how to translate RTF's paragraph syntax into appropriate LateX commands/environments.

- **Character mapping:** Character mapping is largely complete for the most common latin scripts. Characters are translated by referencing character set maps and the output map file "TeX-map". The platform and locale dependent character set, eg. latin-2 (Eastern European), is converted to an internal platform-independent representation by reading the appropriate character map file, in this case cp1250.map. For example, character 192 (hex 0xc1) represents "$\acute{A}$" in the latin-2 character set. **rtf2LaTeX 2$_\varepsilon$** maps this character to "*Aacute*". This mapping is then finally translated into the LaTeX representation "\'{A}" using the output map file *Tex-map*. This two-step character mapping allows for easy addition of support for additional character sets such as latin-5 (Turkish). Also, some users prefer to use the *inputenc* package to represent characters above ASCII value 127, ie. type "$\acute{A}$" instead of "\'{A}". **rtf2LaTeX 2$_\varepsilon$** can do this automatically by using the appro-

priate mapping in the Tex-map file. Sample Tex-map files for cp1252 (ANSI), cp1250 (latin-2), and applemac (Macintosh) are provided to illustrate this approach. To use any of these files, rename the file to "Tex-map". The appropriate "\usepackage[...]{inputenc}" entry in the r2l-head file will cause the inputenc package to be loaded.

- **Footnotes:** It was quite simple to add footnote support. I initially had trouble converting footnotes within tables, but it works now.
- **Hyperlinks:** I have added support for translating hyperlinks using the hyperref package. This is still somewhat experimental. There is an option in the r2l-pref file to turning off this option.

Features I would like to support in future versions are:

- Unicode: This should really get rid of the need for different character set maps. Word 98 on the Mac already puts out Unicode.
- Lists

5 Test files

There are four test files in the *examples* directory of the rtf2latex2e distribution that demonstrate the capabilities of the converter. You can also download a larger set of test files to see how the program behaves. These test files are in a tarred gzipped archive in the same place where you downloaded the rtf2latex2e distribution. "*RTF-test-files*" contains several RTF files that have been successfully tested on **rtf2LaTeX 2$_\varepsilon$**. By success, I mean that **rtf2LaTeX 2$_\varepsilon$** processes the RTF file without any problems (except maybe giving a few warnings) and produces a ".tex" file that is LaTeX 2$_\varepsilon$-able!! It does not mean that the LaTeX 2$_\varepsilon$ output file will look exactly the same as the RTF input file. In fact, most of the time, it will not. Some features like I do not care to convert, others like Unicode support will be implemented in future versions.

6 Acknowledgements

I would not even have attempted this thing had it not been for Paul DuBois' very nicely designed RTF tool. I did not have to bother with parsing the RTF tokens and understanding it. All I had to do was write code to act upon the token. Thanks, Paul, for simplifying it. Another great help has

been the DropUnix application framework by Ryan Davis that makes porting between command-line Unix and drag-and-drop Macintosh a matter of changing one line of code. DropUnix itself is based on the drag-and-drop DropShell framework by Leonard Rosenthol, Marshall Clow, and Stephan Somogyi.

Steve Swanson of Mackichan Software, makers of Scientific Word and Workplace (`http://www.mackichan.com`), contributed the equation converter code. With this ability, **rtf2LaTeX 2$_\varepsilon$** has advanced to version 1.0. Hopefully, this essential feature addition along with **rtf2LaTeX 2$_\varepsilon$**'s other capabilities will make this program the *de facto* tool for converting word processor documents to LaTeX 2$_\varepsilon$.

Finally, I have to thank Scott Prahl for providing constant feedback and encouragement to keep this going. Scott also joined me in the development effort and contributed the image conversion code.

7 Legalese