What RTF->TeX Does.

RTF-> TeX is a utility application that attempts to convert a certain style of Microsoft Word RTF (Rich Text Format) file into a file which is capable of being processed with the TeX typesetting program. Both RTF files and files suitable for processing by TeX are plain TEXT files, that is, files of characters. A standard Microsoft Word document is another type of file which *cannot* be processed by RTF->TeX; it must first be saved as an RTF text file with the "Save As…" command in Microsoft Word.

TeX is a language written by Donald Knuth for the typesetting of technical documents, especially those that contain a lot of mathematics. Textures™ is a commercial implementation of the TeX system (Blue Sky Research) and OzTeX is a public domain version.

Microsoft Word has the capability of producing technical documents with its facility for formulas, but it is fairly obvious that the printed output of Word is inferior to that of TeX. Its advantage is that interpreted formulas are displayed on screen as they are typed in, whereas TeX displays the typeset document page by page after a lot of processing. This probably doesn't make much difference if the complete document has already been written out by hand, but seeing directly what you are writing "on the fly" helps you think about what is being typed in. The latest version of Textures™ goes some way towards dealing with the on-screen thinking problem.

This application is a tool to bridge the gap between the ease of use of

Word and the quality of the typeset output of TeX, by taking an RTF file produced by Word, processing it, and producing a text file which, hopefully, can be interpreted by TeX in the *Plain TeX* format. However, the application RTF->TeX makes the restrictive assumption that the RTF file is that of a *mathematical* document written in a language using the English alphabet (that is, without accents) , so most formatting is ignored. Having said that, the output is always a text file which should save some energy in the conversion between the two types of documents. With some exceptions noted below,

all italic characters, formatted characters and formula codes are assumed to be mathematical symbols.

In TeX, mathematical symbols appear between two dollar signs:\$…\$. For example, the following sentence containing the italic character *a*:

Suppose that *at* is the product of polyforous stigmata.

will appear (in the default font) in the corresponding RTF file as:

Suppose that {\i at} is the product of polyforous stigmata.

There may be some other code corresponding to paragraph formatting. After processing with RTF->TeX, the code will translate into

Suppose that \${ at}\$ is the product of polyforous stigmata.

provided that there are no hiccups. If the file produced by RTF->TeX can be interpreted by the TeX typesetting program (there may be many errors), then in the typeset output the sentence will again look like:

Suppose that *at* is the product of polyforous stigmata.

This may not appear to be particularly useful, but with more complicated examples and formulas, the typeset output of TeX looks much better that what's printed with Word.

One problem associated with this assumption is that genuine italics sometimes appears in technical documents for emphasis, such as in a definition, or in propaganda. *The application* RTF->TeX *is not sufficiently intelligent to recognise italics for the purpose of emphasis*. So, the RTF code {\i the purpose of emphasis} translates into ${ the purpose of emphasis}$ which ends up looking like *thepurposeofemphasis* after typesetting by TeX — each of the characters is interpreted as a mathematical symbol. Presumably, there is not a great deal of such emphasis in a technical document, except in the case mentioned below, so changing the TeX file later should not involve a great deal of work (at least, less than would be needed to write the code to recognise emphasis).

This brings us to another point:

it is unlikely that the output of RTF->TeX can be interpreted by TeX without errors, or it may just be plain bad TeX code.

A working knowledge of TeX will usually be needed to produce an interpretable file with proper formatting. Consequently, the market for this utility may be minuscule; presumably the user will have written the Microsoft Word mathematical document as well.

Before describing other limitations of RTF->TeX, we should discuss how it works.

How RTF->TeX works

The application opens the indicated file, looks for the string "{\rtf" and deletes all the formatting code between this and the string "\ endnhere". If these two strings aren't found, it displays an error message and exits from the conversion of that file. If the strings are found, then it scans each character of the file consecutively. If it is not a standard text character, deemed to be a lower or upper case letter of the English alphabet, a number or punctuation, it deems the character to be a symbol. The default font is the font in which text is written in the "normal" style of the Word document, assumed to be "Times" (but which can be changed — see below).

The special character "\" precedes every RTF code, which is read in. A formatting code (such as the string to flag the default font) is deleted, otherwise it jumps into *math mode*, inserting the "$" character into the output at the preceding space and reading and processing the ensuing RTF code until a complete word of standard text is encountered, that is, a string of standard text with a space before and after it. Then a "$" sign is inserted at the space preceding the word, and it jumps into *text mode*, merely re-writing the text encountered to output until another non-text character is encountered.

Dead easy — but there's many a slip o' the wrist 'twixt the cup and the lip.

Basic to the conversion algorithm is the notion of a *token*, which is just a portion of processible code. It may be a sequence of text characters, a sequence of mathematical symbols or a number of complete formula codes. Each token is processed and written to disk. For mathematical symbols, a token is the RTF code that ends up between the dollar signs in TeX. It is limited in size (somewhat arbitrarily) to 32K. Anything that breaks this limit is either a portion of a novel or an unintelligible formula, so RTF->TeX ignores it. In principle, RTF->TeX can treat files as large as the available disk space allows.

Further warnings follow.

• Any paragraph , column or tab formatting is lost.

• The code "\i" denoting italics is simply deleted and the rest is treated as a symbol.

• If mathematical symbols in the default font have not been put into italics , then they are treated as plain text.

• An attempt is made to convert Word formula codes to TeX codes.

• Strange things happen with multiple subscripts and superscripts. At this stage, editing the output is more efficient than writing the code to treat every conceivable

possibility.

• No attempt is made to convert PICT and embedded object code into something intelligible to TeX.

• No attempt is made to interpret the overstrike formula code: \|O.

• Formula codes are limited to the 255 characters of a Pascal string, otherwise I'd have to roll my own text procedures (and I still have to do the dishes).

• I have no technical documentation on Microsoft Word RTF format,
so I've guessed what to do by looking at my own RTF files. There are probably lots of things I haven't accounted for, and I may have guessed wrong.

The expression β_F\B(B\B(\I\PR\IN($_{j=1}$,^{n},)Γ[t\S\DO3(j–1)\,t\S\DO3(j)])), for example, generates an RTF formula code of length greater than 255. If you habitually write things like this, then chances are nobody reads your work anyway (only joking).

One RTF code for

the function f(ω) is good

is

the function f({\f23 w}) is good

which should end up as

the function $f(\omega {})$ is good

(as a point of logic: *f* is a function, *f*(ω) is its value at ω, so such expressions, typical in many mathematics texts, are an abomination). However,

the function f(x) is good

will end up as

the function f(x) is good

because parentheses are not viewed as symbols while reading text — they are, however, while reading math code because of their critical presence in formulas. So, if letters denoting symbols have not been put into italics, as mathematical typesetting convention dictates, the results can be variable. The solution is to put the dollar signs in by hand.

Of particular importance is character replacement for special fonts. For example, the capital Greek letter "omega" is represented in the standard Symbol font as Ω. The RTF code for this is {\f23 W}, which should translate into {\Omega} for TeX. The string "\Omega" is read in from a conversion table for the Symbol font whose FOND resource ID is 23. Having the strings in a dialog means that conversion data for fonts other than the Symbol font can be used. The data for a few fonts that contain mathematical symbols are given. Mostly, only the conversion strings not already given in the Symbol and Time fonts are in the dialog resource — why use a bitmapped font when a postscript font has a similar symbol?

A caveat for the rule that the code "\i" is simply deleted is encountered with *proclamations*. These are the statements of theorems, propositions, lemmas and so on, which are conventionally formatted in italics for emphasis. RTF->TeX tries to detect such statements by looking for a special format: two paragraphs, followed by bold-face (code: \b) and one of a number of special proclaim words, such as "Theorem", "Proposition", "Lemma". Subsequent code "\i" is merely replaced with "\sl" until two paragraphs are encountered, which signifies the end of *proclaim mode*. This too causes difficulties, because emphasis italics and symbol italics are not distinguished; subsequent editing is required again. Another problem is paragraph formatting (and crazily formatted paragraph markers) encountered while checking for consecutive paragraphs — the application may simply not recognise a proclamation.

System requirements

RTF->TeX runs under Systems 6.0.x, $x \geq 3$ and 7.0 on HFS volumes. It has been tested on the following machines:
Macintosh IIcx, IIsi, Powerbook 140, Quadra 700, ClassicII, SE without ill effects.

It requires HFS for a PBGetCatInfo call and any System version with WaitNextEvent.

How to use RTF->TeX

Menus

File

Show Text

Displays a scrollable window containing a portion (< 32K) of the TeX code produced by the conversion, if there is any. The text may be selected and copied to another document. The text file written to disk is more complete than the displayed text.

Convert ⌘O

Brings up a custom file dialog to make a list of files to convert. Use the "Add" button to add individual files to the list. Use "Add all" to add all the TEXT files in the folder. The string ".tex" is appended to the name of the RTF text file unless it already has a suffix, in which case the suffix is replaced by "tex". A dialog warns of name conflicts. If a file with the same new name exists already, the second file is not added to the list. Remove a file from the list by selecting the corresponding RTF file in the upper box and clicking the "Remove" button. Double clicking on a file processes that file. The "Convert" button starts processing the files in the list. Up to 100 files may be processed in a batch.

Close ⌘W

Closes te frontmost window.

Quit ⌘Q

Exits from the application.

Edit

Standard editing operations. Only "Copy" is available for the converted text. The other operations apply to the dialogs. The command "Undo" is unimplemented, but the menu looks naked without it.

Font

A collection of font conversion tables are provided with this menu.

Changes to the table only apply while the application is running. It is unlikely that changes will need to be made to the shipped font tables. See below how to make permanent changes. The fonts won't display properly unless they are open in your system.

Options

Default Fonts

Displays a dialog in which the default text and script fonts are set for the RTF code. The code for the default text font is stripped off, and the text is rendered in Times Roman in TeX. The script font code is replaced with the string

"\cal". If the designated fonts are open on the system, then the resource ID is read in from the name, otherwise the FOND ID will need to be entered too.

Proclaim Words
Displays a dialog in which the "proclaim" words are set for the RTF code. These words help signal the application to treat italic code in a special way, as described above.

Converting a Microsoft Word Document

With the Microsoft Word document opened, select the "Save As..." item from the File menu. A dialog appears. In Word 4.0, click on the "File Format" button and click the Interchange Format (RTF) radio button. In Word 5.0, choose "Interchange Format (RTF)" from the pull down menu "Save File as Type". Type in a new name for the RTF document so that the original is preserved (perhaps append ".RTF" to the name) and the save the document in any folder you like.

RTF->TeX will convert the saved RTF document into a file ending in ".tex" in the same folder as the RTF document. Alternatively, put all your RTF files into one folder and convert them all at once.

Using the Convert Dialog

A list of files to be converted is made in the lower box of the dialog which appears after the "Convert" menu item is selected. The buttons are self-explanatory — see the comments on menu item "Convert" above. System 7 users can access a few largely useless balloons.

• **No two files in the list can have the same name, even if they appear in different folders.**

• **To remove a file from the list, select the original file in the upper box and click the "Remove" button—counterintuitive, but cheap.**

Conversion Progress

File conversion can be a lengthy process. A dialog is displayed relating the progress of the conversion process. The lengthening bar indicates how much of the original file has been processed as the converted file is being written to disk.

The characters *H*,*T*,*M*,*P* in the top right hand corner indicate the current state of processing; *H* means the header is being stripped, *T* indicates that text is being read in and *M* means that math code is being processed, *P* means that a proclamation has been encountered. So, a non-RTF file will just display *H* and then the processing will terminate as the end of the file is reached. A file with a large portion of plain text, such as a technical document full of propaganda, will display the symbol *T* for a large proportion of the processing time. Mostly, the symbols *M* and *T* will toggle when technical symbols are frequently interspersed with plain text.

Clicking the "Stop" button terminates the processing of the file. The code processed previously has been written to disk.

• Converted files are saved as Textures™ files of type "TEXT" in the folder in which the original file appears .

• Under System 7, dragging a collection of text files onto the RTF->TeX icon in the Finder will cause it to start up and attempt to process the files. ***Beware—no checking of duplicate file names is done with this method; duplicate files will be overwritten.***

• There is balloon help under System 7 with rather obvious pointers.

• RTF->TeX works in the background.

Changing the font conversion tables

I know the font conversion table dialog is an example of a bad user interface, but it's cheap.

• Any changes to the font conversion tables made while the program is running are *not* saved to disk.

The conversion tables reside in dialog resources in the application. To change them, you'll need a copy of Resedit and probably a 13" monitor. Alternatively, send me the resource information, and I may incorporate it in the distributed product.

The tables are standard enough not to require fiddling with too much for the given fonts, so I don't think it's a great blow against user friendliness.

• There are three menu items "Font1", "Font2", "Font3" for other conversion tables—they serve no

purpose at present.

The other fonts included are "Times" (ID: 20), "Symbol" (ID: 23), "Greek12" (ID: 12), "MacMath" (ID: 129), "MT Extra" (ID: 2515). The fonts "Times" and "Symbol" come with the Macintosh system. The font "MT Extra" is distributed with Microsoft Word 5.0 with the equation editor. The fonts "Greek12" and "MacMath" have some special symbols not available elsewhere and are sometimes used in mathematical documents (there ID's probably conflict with other common fonts).

It looks like the RTF code for the characters:

for all $x \in \Omega$

may be one of the following :

```
for all {\i x }{\f23  \'ce W}
for all {\i x} {\f23  \'ce} {\f23 W}
for all {\i x}\~{\f23  \'ce}\~{\f23 W}
```

depending on how it was typed in. I've assumed that \f23 represents the font whose FOND resource ID is 23, namely, the Symbol font. To process the code {\f23 W} , RTF->TeX checks the list of font conversion tables for the one with the "Font ID" 23 in the lower left hand corner. It then looks up the replacement string for the character "W" in the appropriate dialog item, which is "\Omega", so that {\f23 W} becomes \Omega {}. The TeX output of RTF->TeX will be:

for all $x {} \in {} \Omega {}$

which ends up looking like:

for all $x \in \Omega$.

The braces {} around the null code don't cause harm. Because the string "\in" denotes a binary operator in TeX, the spacing of the typeset output of $x {}\in {} \Omega {}$ is better than that of ${x } {\in} {\Omega }$.

Therefore, by changing the Font ID in the static text item on the lower left and corner of the dialog (off-screen on a compact Mac!) with ResEdit and putting in new strings into the conversion table editable text boxes, you can convert characters in your favourite fonts to TeX code. However, you need to know a bit of TeX to guess what the correct replacement strings should be.

The dialogs contain pictures of the font to be converted as a user dialog item. The system will substitute some font in the picture if the appropriate one is unavailable.

Comparison with rtf2tex

I have included an example of a small RTF file ("RTF") produced from a Microsoft Word 5.0 document. By opening this document with Microsoft Word 5.0 (or earlier versions) and agreeing to its conversion if a dialog appears, it can be viewed as a file of mathematical formulas, incidentally related to the author's field of interest.

The file "RTF.tex" is the output from RTF->TeX. The file "RTF.tex1" is the output from the Unix based RTF conversion program "rtf2tex" (available by anonymous ftp from rzsun2.informatik.uni-hamburg.de: 134.100.4.42 in the directory /pub/tex, among other sites). Happily, RTF.tex compiles with plain TeX, without errors, into a reasonably typeset document of three pages. The file "RTF.tex1" generates many errors when interpreted with plain TeX; it doesn't treat Word RTF code as far as I can determine. Presumably, happy users of rtf2tex don't use Word formulas. The production of "RTF.tex1" with rtf2tex is blindingly fast on a Sequent Symmetry S81. The production of "RTF.tex" with RTF->TeX is painfully slow on a Macintosh IIsi or cx (about 9k/min for reasonably dense mathematics).

Status

RTF->TeX is **FREE**! I am actually paid to write and teach mathematics rather than write Macintosh applications to translate file formats, *and* I have to do the dishes in the evening.

If there is any interest shown on the Internet subsequent to this posting I may

• develop the application further to include more features, such as the ability to translate paragraph formatting, extended formula code, multiple subscripts/ superscripts, better System 7 use and much needed code optimisation, or

• distribute the Pascal source code, although that might be too embarrassing.

The application is written in Think Pascal, so portions of it are copyright by Symantec.

Hmm... Must learn C one day.

Internet: brianj@hydra.maths.unsw.edu.au Brian Jefferies
School of Mathematics
University of New South Wales
PO Box 1, Kensington,
N.S.W .2032
AUSTRALIA