Conversion from WORD/WordPerfect to LaTeX*

Marion Neubauer

SFB 245
Psychologisches Institut, Universität Heidelberg
Hauptstr. 47–51
D-69117 Heidelberg
Marion.Neubauer@urz.uni-heidelberg.de

Abstract

Production of a large document with many contributors may require conversion of all submitted manuscripts into the same format, for example LaTeX. A large proportion of manuscripts are submitted in the formats of WORD and WordPerfect, two very popular word processing programs. I will discuss different approaches converting such files to LaTeX format.
First of all the differences between the word processors WORD and WordPerfect versus the document preparation system LaTeX will be explained, and problems encountered during text conversion into LaTeX will be discussed. The conversion can be done either by means of a seperate program (external conversion) or using macros, style sheets and a printer driver from within the word processors (internal conversion). Advantages and disadvantages of both methods for different types of text elements such as plain text, lists, tables and mathematical formulas will be discussed. This is followed by an overview of the conversion programs currently available.

Keywords: conversion, WORD, WinWord, WordPerfect, RTF

1 Introduction

First I would like to explain how I come to be working on the subject of this paper. I am employed at the University of Heidelberg and working in the special collaborative program, SFB for short, no. 245, called 'Language and Situation'. Psychologists and linguists at the Universities of Heidelberg and Mannheim and at the Institut für Deutsche Sprache in Mannheim are involved in this program. SFBs are supported by grants for pure research granted by the Deutsche Forschungsgemeinschaft (German Science Foundation) for an initial period of three years, after which a research report is produced and application is made for a grant for a further period of three years. This combined report and application for grant renewal comprised some 1000 pages and was supposed to be produced using LaTeX. The contributions, fiveteen in the current period, were written using various software programs including plain TeX, LaTeX, WORD, WORD for Windows and WordPerfect.

I will concentrate on conversion to LaTeX. Two of the programs described below also support conversion to plain TeX, although I will not go into greater detail.

Firstly, some terms: WORD refers to the program Microsoft WORD versions 5.0 and 5.5 running on the DOS operating system. Unless otherwise stated WinWord refers to WORD for Windows versions 2.0/6.0, WORD for DOS version 6.0 and also WORD on other operating systems. WP stands for WordPerfect for DOS versions 5.x. Most of the points concerning WinWord also apply to WordPerfect for Windows.

2 Overview

WORD, WinWord, and WP are word processors, in contrast to LaTeX, which is a document preparation system. These two concepts are generally understood to be partially or completely incompatible. This is actually the case with the versions available today.

I am assuming a familiarity with LaTeX and its markup language. 'Generic markup means adding information to the text indicating the logical components of a document such as paragraphs, headings and footnotes. The formatting (visual representation) associated with a component is decoupled from it's function (position) in the (hierarchical) structure of the text. LaTeX is, to a large extent, an example of a 'generic markup language' (Goossens, Mittelbach, Samarin: *The LaTeX Companion*, Addision Wesley, 1994, p. 7). For those who are not familiar with WORD I would like to give a brief description of the functions and the main differences between these two programs.

WORD is a so-called WYSIWYG program, i.e. what you see is what you get. This means that manipulated text ap-

*This paper was first published in the Conference Proceedings of the European TeX Conference, EUROTEX'94 in Gdansk. It was prepared within the context of the Sonderforschungsbereich 245, 'Spache und Situation', Univerität Heidelberg/Mannheim.

pears on the screen in its final form — for example, a word which is emphasized appears in italics on the screen. The manipulation itself, e.g. emphasizing a word, is done by marking the text and entering some key combinations or choosing the function from a menu. Character formatting in WORD involves specifying the name and size of a font, type style (italics, bold, bold italics, underlined, crossed out) and also sub- and superscript. WORD uses the IBM standard 256 character code table.

The record length of WORD text files is variable. A carriage return/line feed character is used only at the end of a paragraph. Standard LaTeX requires a carriage return character after every 500 characters, at the most at the size of `buf_size`.[1]

The file written by WORD — I will call it WORD text file — is very different from a LaTeX file. The file has a binary header, the text in ASCII and a binary trailer containing the formatting information in coded form. Pointers are used to apply the formatting information to the text.

WORD also stores formatting information outside the WORD text file, i.e. within an initialisation file of the WORD program, in style sheets and as part of the printer drivers used.

- Certain operating parameters of the WORD program can be changed at any time via the menu 'options' and apply to any text file loaded from then on. These parameters includes items such as the width of the tabulator.
- Style sheets are comparable to LaTeX style files. Within style sheets, layout functions are assigned to particular key stroke sequences such as page dimensions, layout of lists, etc. Key stroke combinations in style sheets can override standard definitions of WORD. For example, the key stroke combination Alt+I is normally used for italics, but in a custom style sheet it can be re-defined to format an item in a list.
- Printer drivers in WORD are not comparable with DVI drivers since they contain among other things the actual information about available fonts and their possible variations. WORD has a printer driver for every supported output device because the standard version can only use resident printer fonts. This is why changing printers can cause headaches.

WinWord stores almost all formatting information in the text file. Style sheets have a rather different function and WinWord uses the printer drivers of Microsoft Windows which use True Type fonts. Hence, the fonts are independent of the printer.

WP text files include all formatting information. WP use it's own printer drivers, it did not use the drivers from Windows.

3 Problems of text conversion

The following is a list of items which have to be considerd when converting files from a word processor to LaTeX.

- Record length of file,
- spacing:
  - horizontal spacing like blanks, tabulator stops, indentation,
  - vertical spacing, which is often realized with one or more blank lines,
- paragraph formatting: justified, flushleft, flushright, and centered text,
- type size and style (fonts),
- special characters:
  - diacritical symbols,
  - hyphens between compound words,
  - discretionary hyphens,
  - characters with a special meaning in LaTeX, mainly $, %, &, and #,
- headings, to be translated into LaTeX sectioning commands,
- lists and enumerations, which should be converted to the LaTeX list environments 'itemize', 'enumerate', and 'description',
- footnotes,
- mathematical formulas except sub- and superscripts, which are handled with a special font,
- tables, and last but not least
- indices.

In my opinion it is much more important that all characters of the original text are converted than to conserve all formatting information. A very good test of the text integrity is converting a percent sign because it indicates a comment in LaTeX, and % has to be converted to \% so that LaTeX will not get confused.

Personally, I like the conversion programs to handle more than just plain text, but due to the different typographic rules and the large differences of files created by word processors and LaTeX input files, it would be foolish to expect too much from a converter program.

4 Alternatives

All word processors discussed here can store a text in pure ASCII format. Once saved in this format, all formatting information is lost and has to be replaced in the form of LaTeX commands.

The alternative is to convert the formatting information into LaTeX commands by means of a program. I will distinguish two principally different ways, that are (a) 'internal' conversion, and (b) 'external' conversion.

4.1 Internal conversion

Internal conversion is carried out within the word processing program using macros, style sheets and (in the case of WORD) a special printer driver. The big advantage of the

[1] Common LaTeX versions today use a `buf_size` of 2000 characters.

internal conversion is the fact that all information about the text and formatting is accessible and usable. Remember, for example, the parameter of the options menu not stored in the text files.

Internal conversion is very simple and easy to use by people with experience using a word processor but with little or no knowledge of LaTeX. WORD, WP, and WinWord can then serve as a WYSIWYG editor to LaTeX.

The text is typed into WORD using macros and style sheets and then 'printed' to the hard disk using a 'LaTeX' printer driver. In WP and WinWord the text is stored using the normal storing procedure but with the format option 'DOS text format'. The resulting file is a complete LaTeX input file. Special text formatting needs can be met by modifying macros and style sheets. A small problem remains: line numbers, reported by LaTeX in case of an error, cannot be used because the lines are numbered differently within the WYSIWYG editors.

4.2 External conversion

External conversion is performed without the word processor by an external program. The text file is stored either in the text format of the word processor or in Microsoft's Rich Text Format (RTF). Many word processors including WORD, WinWord, and WP are able to write RTF files. RTF itself is a de facto standard. After the conversion, the file has to be edited using a text-oriented editor.

One problem remains: a conversion program can guess that a piece of text in 'Times Roman bold face 30 pt' is a heading, but without further information it cannot determine whether the heading introduces a section, a subsection or an appendix. Within WORD and WinWord the user has the opportunity of using a limited amount of pre-defined headings which can be converted to LaTeX sectioning commands.

5 Conversion Programs

5.1 Overview

Conversion of WORD text files:

- WORD2TeX (DOS), external
- WD2LaTeX (DOS), external
- WORD_TeX (DOS), internal

Conversion of WinWord 2.0 text files:

- WINW2LTX (DOS, Windows), internal

Conversion of files in RTF:

- RTF2TeX (UNIX), external
- RTF2LaTeX (UNIX, Mac), external
- RTFLaTeX (DOS), external

Conversion of WP text files:

- WP2LaTeX (DOS, UNIX), external
- WP2x (UNIX, DOS), external

5.2 Details

WORD2TeX, WD2LaTeX, and WP2x are external conversion programs for WORD or WordPerfect. WORD2TeX additionally includes a primitive version to perform internal conversions. The programs were written between 1989 and 1991 (see references for details), and as far as I know are no longer supported by the authors. They are useful mainly for plain text. Some of the special characters, like the German double s (ß), are converted incorrectly by some of the programs, especially those written by authors from English speaking countries (see Table 1).

In addition, contrary to what I have said above about external conversion not needing the word processor itself, WORD2TeX and WD2LaTeX produce better results when the text is stored by WORD with modified page dimensions before being converted.

WP2LaTeX is an external conversion program for WordPerfect written in C. It was updated in February 1994 and the author welcomes any comments about it. The older versions (first written in Pascal and later converted with Pascal-to-C) had a lot of errors, e.g. cutting footnotes after 256 characters. The new version has some improvements concerning formulas, but even they are not converted 100%.

WORD_TeX is an internal conversion program written for WORD version 5.0 only and converts documents to TeX and LaTeX. It has been used in my work to convert reports and applications. I wrote a modified style sheet for the authors which speeds up the conversion process considerably. For authors using WinWord I created a special WinWord style sheet. The text was stored in the 'Word for DOS' format and then converted using the same procedure as for the WORD 5.0 text files.

WORD_TeX is the best solution for internal conversion of text which has still to be typed in. It correctly preserves footnotes and paragraphs (including paragraph formatting) and also converts special characters, different type styles and sizes. Headings are converted into LaTeX sectioning commands and lists into the appropriate list environments. Tables are not converted: only the text of the table is transfered to the LaTeX input file. Sub- and superscripts from WORD text files result in the LaTeX command `\raisebox`. More complex mathematical formulas cannot be converted. WORD_TeX also offers half-automated conversion of quotation marks — the user has to confirm whether they are opening or closing quotation marks — and ligatures. The package is very well documented (18 pages, in German) and with little knowledge of WORD it can be modified to fulfil personal needs.

WINW2LTX is not really a converter but a style sheet for WinWord 2.0 to be used for editing LaTeX code. For example, choosing 'Bullets and Numbering' from the menu for a marked piece of text, the following commands are inserted `\begin{enumerate} \item ... \end{enumerate}` at the appropriate places. After the text has been stored under 'MS-DOS text with layout' it is ready to be processed by LaTeX. The program is still in the development stage. A yet it cannot convert formulas entered with the formula editor.

(As far as I know, there are two internal conversion programs for WP which were described in [1] and [3], but I have not been able to obtain copies of them. See references for details.)

RTF2TeX, RTF2LaTeX, and RTFLaTeX are external conversion programs, and use intermediate files in Microsoft's Rich Text Format (RTF). RTF is a very compact format and has a lot of similarities to TeX commands. It includes many positioning commands which cause problems during conversion. Some programs write redundant RTF code. This redundant code and the large number of positioning commands make LaTeX input files from these converters very difficult to read (and modify). Best results gives RTF-code from Macintoshis or WinWord 2.0b. The standard font for paragraphs should be times roman 12pt.

All three of these programs were written within the last two years and the range of texts that they can preserve is bigger t han with any other conversion program. As stated above, in most cases conversion of an RTF file only preserves the formatting information — information about text types is normally not stored in RTF. In addition, different type sizes are converted to LaTeX size commands, for example 40 pt will get `\LARGE`, 30 pt get `\Large` and so on. Because LaTeX has a limited amount of size commands a 1:1 conversion is not possible, but these programs give a close approximation.

The program RTFLaTeX has an option for producing a LaTeX file for LaTeX with NFSS. A style file belongs to the RTFLaTeX conversion package where serveral RTF commands are defined. After conversion, the style files have to be modified because only a subset of common RTF commands are defined. A bit of trial and error is needed to find out which commands are missing and what they should do.

The main advantage of RTF2LaTeX is the configuration of a character code map for all character codes between ASCII 128 and 255. A user-written configuration file can be used to convert WORD headings and other text styles, and an example is included in the distribution.

There is also a commercial converter from K-Talk Communication Inc. available, named 'Publishing Companion', price $249.00 (US). In an advertisement they promise to convert everything from the list given in the section 'Problems of text conversion', and even newspaper-style columns and mail merge.

6 Conclusions

1. Conversion of moderately complex documents remains a time-consuming job and the converters available today can only take over part of the work.
2. The word processors I have examined
   1. did not support a satisfactory standard markup (e.g. WORD), or
   2. the markup information is not available in the output file (e.g. RTF, except in WORD for Mac).

   These are the reasons why an automated external converter is not a viable proposition.
3. Using a word processor as editor in conjuction with an internal conversion program offers an acceptable solution. In the long term a change to a text-oriented editor like Emacs is still recommended.
4. For documents written in WORD I prefer the WORD_TeX converter, for conversion of large documents in WinWord or WP I choose RTF2LaTeX.

References

Most of the programs discussed are available from the CTAN Network (see *TUGboat* **14**(3):342–351, 1993). All programs are accompanied by source code. The location of the files on CTAN servers, e.g.
`ftp.dante.de:/tex-archive`,
or SIMTEL archive mirror, e.g.
`sun0.urz.uni-heidelberg.de:/pub/simtel`,
is given below.

[1] Hoover, A. Z. Using WordPerfect 5.0 to create TeX and LaTeX Documents. *TUGboat* **10**(4):549–558, 1989.

[2] Mintert, St. Recycling. Wie gut sind TeX(t)-Konverter? *ix* **8**:74–80, 1994.

[3] Modest, M. F. Using TeX and LaTeX with WordPerfect 5.0. *TUGboat* **10**(1):67–72, 1989.

[4] Microsoft Corporation. *Arbeiten mit Microsoft WORD für IBM Personal Computer und Kampatible.* Version 5, 1989.

[5] WORD2TeX (1989), Pascal, M. Lenz, Erlangen, Germany.
CTAN: `/support/word2tex`

[6] WD2LaTeX (1991), Pascal, Dr. Connor J. Thomas, Adelaide, Australia.
CTAN: `/systems/msdos/4alltex/disk05/wd2latex.arj`
SIMTEL: `/tex/wd2latex.zip`

[7] WORD_TeX (1992), Günter Partosch, Gießen, Germany.
CTAN: `/support/word_tex`

[8] WINW2LTX (1994), Allin Cottrell, Winston-Salem, USA.
CTAN: `/support/winw2ltx`

[9] RTF2TeX (1991), C, Robert Lupton, Princeton, USA.
CTAN: `/support/rtf2TeX`

[10] RTF2LaTeX Vers. 1.5 (1994), C, Erwin Wechtel, Wien, Austria.
CTAN: `/support/rtf2latex`

[11] RTFLaTeX Vers. 1.63 (1994), Pascal, Daniel Taupin, Orsay, France.
CTAN: `/support/rtflatex`

[12] WP$_2$LaTeX Vers. 3.0b (1994), C, Rob C. Houtepen, Eindhoven, The Netherlands.
CTAN: `/support/wp2latex`

[13] WP2x (1991), C, Raymond Chen, Berkeley, USA.
SIMTEL: `/tex/wp2x110.zip`

Table 1: *Conversion programs currently available*

Name	WORD2TeX		WD2LaTeX	WP2LaTeX	WP2x
Source format	WORD 5.0, 5.5		WORD 5.0, 5.5	WordPerfect	WordPerfect
Traget format	Standard LaTeX		Standard LaTeX	Standard LaTeX	Standard LaTeX
Conversion type	int.	ext.	external	external	external
Line breaking	+	+	+	+	+
Paragraph formatting[a]	−	−	−	−	+
Type style and size[b]	+	−	+	only style	+
Special characters[c]	+ (except &)	+ (execpt &)	+ (except ß)	+	+
Sectioning	+	−	−	+	−
Lists[f]	−	−	−	−	−
Footnotes	−	−	+	+	+
Mathematical formulas	−	−	−	−	−
Tables	−	−	−	−	−
Index	−	−	−	−	−

Name	WORD_TeX	RTFLaTeX	RTF2LaTeX
Source format	WORD 5.0, WinWord stored in WORD 5.0 format	All files stored in RTF	All files stored in RTF
Target format	TeX, Standard LaTeX, GERMAN.STY	Standard LaTeX, NFSS	Standard LaTeX
Conversion type	internal	external	external
Line breaking	+	+	+
Paragraph formatting[a]	+	+	+ (except flushleft)
Type style and size[b]	+	+	+
Special characters[c]	+	+	+[d]
Sectioning	+	−	+[e]
Lists[f]	+	−	−[e]
Footnotes	+	+	+
Mathematical formulas	−	+ (PostScript coded)	+
Tables	−	+	+
Index	−	+	+

+/−: Converted/not converted to the appropriate LaTeX command.

[a] These are justified, flushleft, flushright, and centered text.
[b] All character formatting type styles like bold face, italics, underlined, crossed out, sub- and superscripts.
[c] This includes diacritical characters as well as the different hyphens and also LaTeX specific characters.
[d] ASCII codes above 127 are configurable.
[e] Can be modified with a translation file for special WORD styles.
[f] This should be converted into the LaTeX list environments ‘itemize’, ‘enumerate’, and ‘description’.