

# WordTeX

## A WYSIPCTWOTCG<sup>1</sup> Typesetting Tool

Tom Wildenhain

### Abstract

[WordTeX](#) is a plugin for Microsoft Word that attempts the impossible: creating documents that appear to be written in  $\LaTeX$  while irritating people who like  $\LaTeX$ . It is both stupidly impractical and surprisingly useful, offering an editing experience that is initially more enjoyable than  $\LaTeX$  and Word but is asymptotically more complicated than either. In this paper, I will explain the results of my [WordTeX](#) research. I will occasionally include content that has no relevance to the paper and is simply used to showcase how [WordTeX](#) renders certain elements.  $(x + 1)^2 = x^2 + 2x + 1$ .

## 1 Introduction

$\LaTeX$  is a popular typesetting tool for creating complicated, consistently-formatted documents. Many students and scientists use  $\LaTeX$  because the finished documents have a clean, professional look that is hard to achieve in standard word processors.<sup>[citation needed]</sup> However, these high-quality results come at the cost of a steep learning curve, potentially tedious editing experience, and sudden anxiety when homework assignments won't compile before a deadline.

Microsoft Word is a WYSIWYG editor, enabling users to have confidence in the appearance of their documents throughout the editing process. In fact, (real) research shows that Word novices are more productive and

---

<sup>1</sup> What You See Is Pretty Close To What Other Tools Can Get

make fewer errors than  $\text{\LaTeX}$  experts when creating certain types of documents.<sup>2</sup>

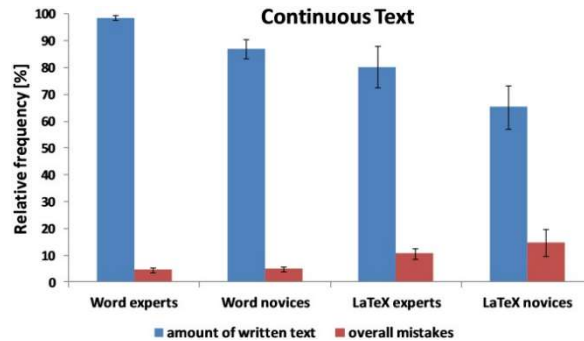


Figure 1: Word users are more productive than  $\text{\LaTeX}$  users for some document types. (Knauff & Nejasmic 2014)

However, when authoring more complicated documents, Word users spend much of their time clicking through menus and fiddling with formatting, leading to inconsistent document structure.

**Word $\text{\TeX}$**  aims to combine the strengths of Word and  $\text{\LaTeX}$ , creating high-quality documents in real time, while annoying both Word and  $\text{\LaTeX}$  fans. It is particularly useful for typesetting homework assignments, although students should be warned that professors may warn students that “using  $\text{\LaTeX}$  is an important skill that will help you in the long term.”

## 2 The **Word $\text{\TeX}$** Template

The main component of **Word $\text{\TeX}$**  is the WordTeX.dot template file. The template includes a set of styles that closely approximate the appearance of  $\text{\LaTeX}$ . The template is designed for Microsoft Office Word 2016 for Windows (or Office 365 ProPlus) and might not work correctly in other versions of Word.<sup>3</sup>

---

<sup>2</sup> Knauff, M., & Nejasmic, J. (2014). An Efficiency Comparison of Document Preparation Systems Used in Academic Research and Development.

<sup>3</sup> I am not yet sponsored by Microsoft.

## 2.1 Fonts

**WordTeX** users should install the LM Roman fonts and Latin Modern Math font (published by GUST) before using the template.

## 2.2 Styles

**WordTeX** uses styles to maintain a consistent look throughout the document. Styles are provided for the title, subtitle, inline code, headings, and other common **LaTeX** elements (Figure 2).

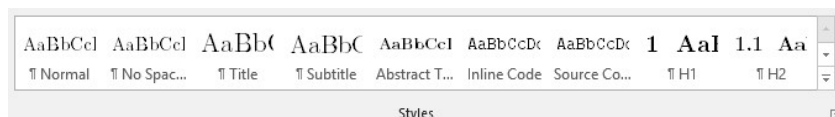


Figure 2: The style gallery

A user can apply a style by selecting it from the style bar or by typing `ctrl+shift+S` and then the name of the style (the latter is much faster). Typing `ctrl+shift+Z` clears character styles (which are applied within a line). Switching to the Normal style clears paragraph styles. Heading styles are automatically numbered, and styles throughout the document are updated if a style is modified.

## 2.3 Math

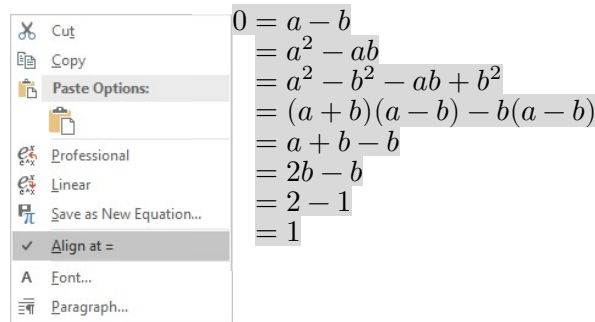
Math in **WordTeX** uses the Latin Modern Math font. To enter or exit math mode a user can type `alt+=`. Latex commands like `\sum` and `\subseq` are supported and render in real time. Word uses parentheses for grouping instead of curly braces. For example, `\sum_{i=1}^{10} i=55` renders as:

$$\sum_{i=1}^{10} i = 55$$

### 2.3.1 Alignment

A set of aligned equations can be created by selecting “align at =” from the equation context menu.

Let  $a = b \neq 0$ .



0 =  $a - b$   
=  $a^2 - ab$   
=  $a^2 - b^2 - ab + b^2$   
=  $(a + b)(a - b) - b(a - b)$   
=  $a + b - b$   
=  $2b - b$   
=  $2 - 1$   
=  $1$

### 2.3.2 Functions

Functions like  $\sin(x)$  and  $\log_2(x)$  are automatically written without italics if they appear in the recognized function list. The equation options dialog (accessible from the conversions section) can be used to add more functions to the list.

### 2.3.3 Blackboard Bold

Blackboard bold letters like  $\mathbb{R}$  and  $\mathbb{N}$  can be inserted by typing `\doubleR` or `\doubleN`. They unfortunately look different from the L<sup>A</sup>T<sub>E</sub>X Blackboard Bold font, since they use glyphs from the Latin Modern Math font. Future researchers might be able to edit the font to include the `\mathbb` glyphs. The commands for these symbols can be shortened (see the section on macros).

## 2.4 Proofs

Proofs are started with the word “proof” set in the proof character style. To end a proof, a user can type `tab` and then `\qed` to insert a  $\square$  symbol.

*Proof.* This is a proof. It is written in [WordT<sub>E</sub>X](#).  $\therefore$  [WordT<sub>E</sub>X](#) can make proofs.  $\square$

## 2.5 Code

```
def print_code_instructions():  
    assert includes_styles(inline and block_code)  
    # Syntax highlighting support is planned and will  
    # hopefully be released soon.
```

## 2.6 Macros

L<sup>A</sup>T<sub>E</sub>X includes a powerful macro system allows users to define custom commands. Word’s AutoCorrect is somewhat similar, but only performs basic text replacement. For example, you can add a math AutoCorrect entry that replaces `\R` with  $\mathbb{R}$ , but AutoCorrect entries can’t take arguments.

Despite these limitations, AutoCorrect is in fact exactly as powerful as L<sup>A</sup>T<sub>E</sub>X macros, as they are both Turing Complete.<sup>4</sup> We can easily simulate a Turing Machine using AutoCorrect entries by representing the state as a string that “reads” characters by adding different entries for every combination of adjacent characters.

<code>\$ \stateA 1 0 1 1 0 0 1 1 0 1 \$</code>	
<code>\$ \stateB 0 1 1 0 0 1 1 0 1 \$</code>	
<code>\$ 0 1 1 0 0 1 1 \stateB 0 1 \$</code>	
<code>\$ 0 1 1 0 0 1 1 0 \stateC \$</code>	
<code>\$ 0 1 1 0 0 \stateC 1 1 0 \$</code>	

<code>\stateA 1</code>	<code>\stateB</code>
<code>\stateB 5</code>	<code>\stateC \$</code>
<code>\stateB 0</code>	<code>0 \stateB</code>
<code>\stateB 1</code>	<code>1 \stateB</code>

Figure 3: An example computation trace. `$` marks the ends of the input.

Unfortunately, AutoCorrect macros only evaluate once from left to right, so the Turing Machine will stop running if it ever moves to the left. To continue evaluation, a user can repeatedly press the Convert button in the Equation Tools tab. Be warned: if the Turing Machine moves to the right without halting, evaluation will not stop, and Word will freeze. I do not know if Microsoft is aware of this issue.

## 2.7 Printing

While Word has a built-in export to PDF option, it unfortunately does not embed otf fonts (like the Latin Modern fonts). The best option is to print to the Microsoft Print to PDF printer, which will embed the fonts.

---

<sup>4</sup> <https://www.sharelatex.com/blog/2012/04/24/latex-is-more-powerful-than-you-think.html>

### 3 Conversions

While **WordTeX** is superior to **LaTeX** in many ways, sometimes **LaTeX** source is required for a conference or assignment.<sup>5</sup> Thanks to Pandoc, **WordTeX** files can be converted to **LaTeX** source code.<sup>6</sup> Mathematical expressions ( $\int_0^4 x dx = 8$ ), `inline code`, and most *formatting* is converted. In fact, here's the source for this paragraph:

```
\hypertarget{conversions}{%
\section{Conversions}\label{conversions}}

While WordTeX is superior to LaTeX in many ways, sometimes LaTeX source
is required for a conference or assignment.\footnote{Fortunately,
  SIGBOVIK does not have such ridiculous restrictions.} Thanks to
Pandoc, WordTeX files can be converted to LaTeX source code.
\footnote{\url{https://pandoc.org/}}
Mathematical expressions ( $\int_0^4 x dx = 8$ ),
\texttt{inline\ code}, and most \emph{formatting} is converted.
In fact, here's the source for this paragraph:
```

The **WordTeX** plugin adds Copy as LaTeX and Paste From LaTeX buttons to Word. They use Pandoc to convert between formats in real time. The conversion isn't perfect, but is fairly close.

## 4 Similarity to LaTeX

### 4.1 Experiment

I conducted a double-blind randomized study to determine whether documents typeset using **LaTeX** and **WordTeX** are distinguishable (Fig. 4). The test subject was blindfolded, and the experimenter (also blindfolded) told her to write L or W on each paper she believed to be a **LaTeX** or **WordTeX** document, respectively. Data was collected until the results supported the hypothesis that **LaTeX** and **WordTeX** are indistinguishable.

---

<sup>5</sup> Fortunately, SIGBOVIK does not have such ridiculous restrictions.

<sup>6</sup> <https://pandoc.org/>



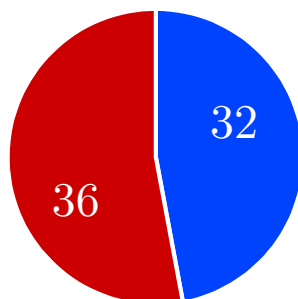
Figure 4: Experimental setup

## 4.2 Results

The papers were misidentified most of the time.

	L <sup>A</sup> T <sub>E</sub> X	WordT <sub>E</sub> X
Identified as L <sup>A</sup> T <sub>E</sub> X	16	18
Identified as WordT <sub>E</sub> X	18	16

It is clear from the following chart that the red side is bigger than the blue side ( $p < 0.5$ ).



■ Correct ■ Incorrect

### 4.3 Conclusions

Running a  $\chi^2$  test, the  $p$ -value is 0.628, which is not significant. Therefore, documents typeset in  $\LaTeX$  do not appear to be significantly different from those typeset in [WordTeX](#).

## 5 Summary

[WordTeX](#) is a typesetting system that supports the basic functionality of  $\LaTeX$  while utilizing the editing convenience of Word. Word's Turing-complete macros ensure that [WordTeX](#) is just as powerful as  $\LaTeX$  (for questionable definitions of "powerful"). The [WordTeX](#) plugin allows for easy conversion between [WordTeX](#) and  $\LaTeX$ . Experimental results suggest that [WordTeX](#) and  $\LaTeX$  documents are indistinguishable. In light of these results, I encourage all scientists, students, and professors to abandon  $\LaTeX$  immediately and use [WordTeX](#) for future work.