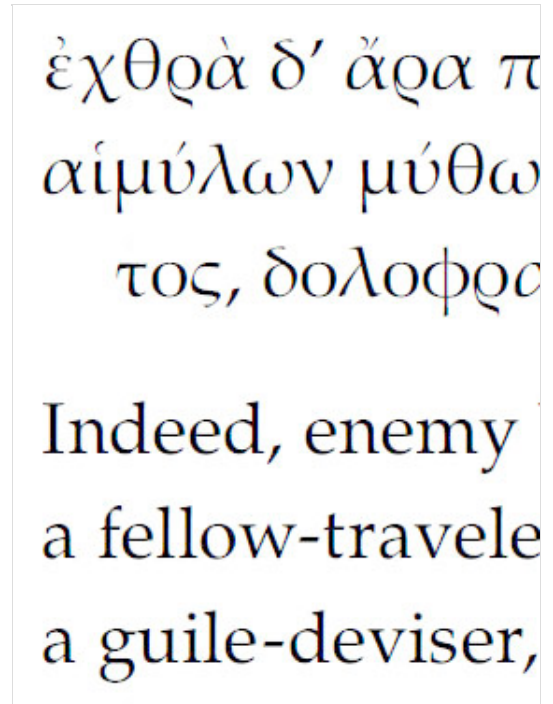


# The Beauty of L<sup>A</sup>T<sub>E</sub>X

There are several reasons why one should prefer L<sup>A</sup>T<sub>E</sub>X to a WYSIWYG word processor like Microsoft Word: *portability*, *lightness*, *security* are just a few of them (not to mention that L<sup>A</sup>T<sub>E</sub>X is *free*). There is still a further reason that definitely convinced me to **abandon MS Word** when I wrote my dissertation: you will never be able to produce *professionally typeset* and *well-structured* documents using most WYSIWYG word processors. L<sup>A</sup>T<sub>E</sub>X is a free typesetting system that allows you to focus on *content* without bothering about the layout: the software takes care of the actual typesetting, structuring and page formatting, producing documents of astonishing elegance. The **software** I use to write in L<sup>A</sup>T<sub>E</sub>X on a Mac compiles documents in **PDF** format (but exporting to other formats such as **RTF** or **HTML** is also possible). It supports **unicode** and all the **advanced typographic features** of **OpenType** and **AAT** fonts, like *Adobe Garamond Pro* and *Hoefler Text*. It allows fine-tuned control on a number of typesetting options, although just using the default configuration results in documents with high typographic quality. In what follows I review some examples, comparing how fonts are rendered in MS Word and in L<sup>A</sup>T<sub>E</sub>X.



## 1. Kerning

Kerning is the process of selectively adjusting the spacing between letters pairs to improve the overall appearance of text. Examples of letter pairs that need kerning treatment are AV, AY, PA, and AT. These letter pairs often look awkward together, and need to either be moved closer together, or further apart manually. Professional typesetting systems and fonts allow fine-grained adjustments for such letter pairs. Popular word processors either lack support for kerning tables or *disable* kerning by default (this is the case with both Microsoft Word for Mac OS v.X and 2008) [Thanks to Mark Dancer and Nicholas Shera for pointing this out].

MS Word (wrong default kerning for the "Ta" letter pair):

Table

[Adobe Garamond Pro, 48pt] [pdf](#) [doc](#)

L<sup>A</sup>T<sub>E</sub>X (correct kerning for the "Ta" letter pair):

Table

[Adobe Garamond Pro, 48pt] [pdf](#) [tex](#)

---

## 2. Real small caps and titling caps

Most word processors create *fake* small capitals by adjusting the size of capitals. Professional fonts contain different sets of glyphs for small capitals and full-size capitals that any serious typesetting system should be able to use in the appropriate context. In particular, *real small capitals* are more than resized versions of uppercase capitals: they have a relatively heavier stroke and are designed to be visually compatible with lowercase characters of the same typeface. Some OpenType fonts have special “titling” alternates that are designed for all-uppercase type set at large sizes and have a lighter stroke.

MS Word (fake small caps):

A sample of text in MS Word using Adobe Garamond Pro, 48pt. The text is "Aa AB BC CD". The small caps "AB BC CD" are visually inconsistent with the lowercase letters, appearing as simply smaller versions of the uppercase letters.

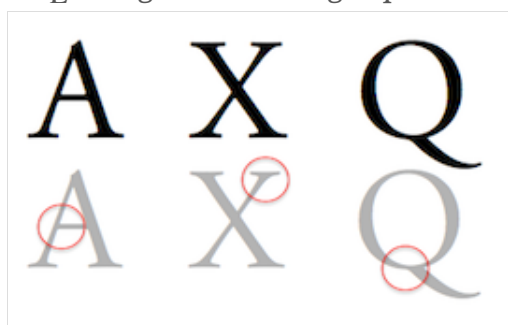
[Adobe Garamond Pro, 48pt] [pdf](#) [doc](#)

L<sup>A</sup>T<sub>E</sub>X (real small caps):

A sample of text in L<sup>A</sup>T<sub>E</sub>X using Adobe Garamond Pro, 48pt. The text is "Aa AB BC CD". The real small caps "AB BC CD" are visually consistent with the lowercase letters, having a heavier stroke and better proportion.

[Adobe Garamond Pro, 48pt] [pdf](#) [tex](#)

L<sup>A</sup>T<sub>E</sub>X (regular vs titling caps):

A comparison of regular and titling caps in Adobe Garamond Pro, 120pt. The top row shows regular caps "A X Q" and the bottom row shows titling caps "A X Q". Red circles highlight the differences in stroke weight and proportions between the two styles.

[Adobe Garamond Pro, 120pt] [pdf](#) [tex](#)

---

## 3. Common ligatures

A good typesetting programme should always use contextual intelligence and substitution tables to determine whether **ligatures** are needed. Common ligatures are essential to professionally typeset text.

MS Word (common ligature errors):

A sample of text in MS Word using Hoefler Text, 48pt. The text is "fire flower fjörd". The words "fire" and "flower" are in red, and "fjörd" is in black. The word "flower" is misspelled as "flower" instead of "flour", and "fjörd" is misspelled as "fjörd" instead of "fjord".

[Hoefler Text, 48pt] [pdf](#) [doc](#)

L<sup>A</sup>T<sub>E</sub>X (correct use of ligatures):

fire flower fjörd

[Hoefer Text, 48pt] pdf tex

---

#### 4. Rare and ancient ligatures

X<sub>Y</sub>L<sup>A</sup>T<sub>E</sub>X in conjunction with professional fonts gives the possibility to exploit the whole set of rare ligatures and decorations, that are automatically added to the text.

MS Word (text with no ligature):

*Aspice, astice, lactosio, Islam, asfissia*  
Aspice, astice, lactosio, Islam, asfissia

[Adobe Minion Pro, 24pt] pdf doc

L<sup>A</sup>T<sub>E</sub>X (text with rare and old-style ligatures):

*Aspice, astice, lactosio, Islam, asfissia*  
Aspice, astice, lactosio, Islam, asfissia

[Adobe Minion Pro, 24pt] pdf tex

Que di<sup>ct</sup>es vous de mon appel,  
Garnier ? Fis je sens ou folie ?  
Toute be<sup>st</sup>e garde sa pel  
Qui la contraint, e<sup>ff</sup>orce ou lie  
S'elle peut, elle se deslie

[Hoefer Text, 24pt] pdf tex

MEet me for a @ffee  
After the L@cture

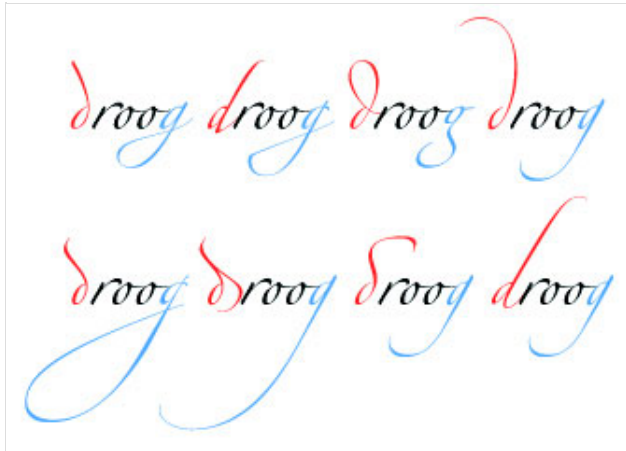
[Skia, 24pt] pdf tex

---

#### 5. Glyph variants

Expert fonts often include variants or alternate shapes for alphabetic characters and numbers. X<sub>Ǝ</sub>T<sub>Ǝ</sub>X with the **fontspec package** offers the possibility to access and select variants on single characters or for a whole text block.

L<sub>A</sub>T<sub>E</sub>X (example of font variants):



[Zapfino, 25pt] [pdf](#) [tex](#)

---

## 6. Transparency

The **fontspec package** allows you to set font transparency in your X<sub>Ǝ</sub>T<sub>Ǝ</sub>X source.

L<sub>A</sub>T<sub>E</sub>X (alpha transparency)



[Hoefler Text Italic, 48+pt] [pdf](#) [tex](#)

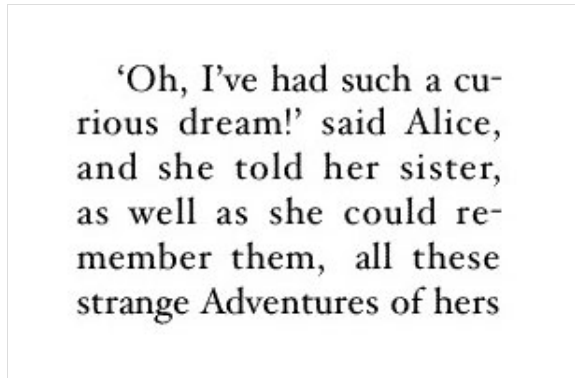
---

## 7. Line breaks, justification and hyphenation

Readability results not only from a good selection of typefaces, but also from a correct distribution of characters and whitespace per line. To attain this goal, most WYSIWYG word processors use relatively dumb justification/hyphenation procedures (i.e. algorithms that establish the position for line breaks by processing text *line by line*). L<sub>A</sub>T<sub>E</sub>X uses an advanced algorithm, based on **seminal work** by Donald Knuth and Michael F.

Plass and enhanced by Frank Liang in 1983 for his PhD dissertation, which considers paragraphs as ‘wholes’ in order to decide where to add line breaks. The algorithm uses language-specific patterns in order to decide the preferred position for hyphenation. The engine then selects line breaks so as to make paragraphs look *as good as possible*. Information that is taken into account for calculating optimal line breaks includes the number of consecutive lines ending with hyphens, word tightness on each line, the change of tightness between consecutive lines. Further development has enabled the L<sup>A</sup>T<sub>E</sub>X engine to allow certain characters to stick into the margin, thus generating an *optically* straight margin - i.e., a margin that *looks* straight without being geometrically so. L<sup>A</sup>T<sub>E</sub>X’s hyphenation settings can be fine-tuned by expert users.

Advanced hyphenation/justification in L<sup>A</sup>T<sub>E</sub>X:



[Hoefler Text, 10pt] [pdf](#) [tex](#)

- For visual examples of the differences between paragraph and line-by-line based justification algorithms, see this [analysis](#) by Maarten Sneeep.
- For a browser-based implementation of the Knuth-Plass hyphenation algorithm check out this [JavaScript library](#) by Bram Stein.

---

## 8. Getting expert fonts

X<sub>Ǝ</sub>T<sub>E</sub>X gives the **best results** with expert fonts such as those based on **OpenType technology** but works with standard **TrueType fonts** as well. Zillions of expert fonts can be purchased online from digital foundries, but Mac OS comes bundled with a number of excellent fonts with expert features (e.g. Hoefler Text, Optima, Skia, Apple Chancery, Zapfino). More free OpenType fonts are available on the net. Check out for example the **Gentium**, **Charis SIL** and **Doulos SIL** fonts from **SIL**, **Cardo** by David J. Perry, the free fonts designed by **Jos Buivenga** (the creator of Fontin), this **collection** of professional quality fonts selected by Vitaly Friedman or the amazing **Font Squirrel**.

---

## 9. Typesetting with L<sup>A</sup>T<sub>E</sub>X: selected links

- SIL International, the X<sub>Ǝ</sub>T<sub>E</sub>X typesetting system »
- Luc Devroye, **Fonts with TeX** - possibly the most comprehensive selection of resources on fonts in T<sub>E</sub>X/L<sup>A</sup>T<sub>E</sub>X.
- Andrew Roberts, **LaTeX vs. MS Word**.

- Allin Cottrell, [Word Processors: Stupid and Inefficient](#).
  - Kevin O'Malley, [L<sup>A</sup>T<sub>E</sub>X: It's Not Just for Academia - part 1 :: part2](#).
  - [T<sub>E</sub>X showcase](#) - edited by Gerben Wierda, it features examples of what you can do with T<sub>E</sub>X.
  - [Humanities books typeset with T<sub>E</sub>X](#) - examples of professional T<sub>E</sub>X typesetting of books in the humanities, selected by the Association of American University Presses.
  - [T<sub>E</sub>X font sampler](#) - booklet illustrating and comparing how different typefaces look like in L<sup>A</sup>T<sub>E</sub>X.
  - [LaTeX typefaces](#) - comparison of fonts available with the standard L<sup>A</sup>T<sub>E</sub>X distribution.
  - [LaTeX - A document preparation system](#) - official site of the L<sup>A</sup>T<sub>E</sub>X project.
  - [TUG.org](#) - The T<sub>E</sub>X Users Group (TUG) was founded in 1980 for educational and scientific purposes, to provide an organization for those who have an interest in typography and font design, and are users of the T<sub>E</sub>X typesetting system invented by [Donald Knuth](#).
- 

## 10. Acknowledgments

Many of the examples in this article are based on the [documentation](#) of the `fontspec` package by *Will Robertson*, who deserves most of the credits for making expert font features in X<sub>Y</sub>T<sub>E</sub>X so easy to use. Thanks to all those who helped improve this article with valuable feedback: *Bastien Guerry, Nicholas Shera, Mark Dancer, Olaf 'Rhialto' Seibert, David Crossland, Tiago Tresoldi, Ehud Kaplan, Henri Langenhoven*.

---

## 11. Technical notes

These examples were created on a Mac, partly on Mac OS 10.3.9, Microsoft Word v.X and TeXShop 1.35, partly on Mac OS 10.5.3 with Word:Mac 2008 and TeXShop 2.x, the X<sub>Y</sub>T<sub>E</sub>X engine with the `fontspec` package, and using the following fonts: Adobe Garamond Pro, Adobe Minion Pro (commercial fonts), Hoefler Text, Skia, Zapfino (fonts bundled with Mac OS X). This article, the PDF samples and T<sub>E</sub>X sources are licensed under a [Creative Commons Attribution-Share Alike 3.0 Unported License](#). A backlink is sufficient for attribution. All materials used in this article can be obtained via [GitHub](#). The T<sub>E</sub>X, L<sup>A</sup>T<sub>E</sub>X and X<sub>Y</sub>T<sub>E</sub>X logos on this page are rendered via a [nifty XHTML and CSS hack](#) by [Edward O'Connor](#). [ COinS metadata ]