

# USENIX Example Paper

Pekka Nikander, Jane-Ellen Long

*Ericsson Research Nomadiclab, Usenix Association*

pekka.nikander@nomadiclab.com, jane-ellen.long@usenix.org

## Abstract

This is an example for a USENIX paper, in the form of an FrameMaker 6 template. Being heavily self-referential, this template illustrates the features included in this template. It is expected that the prospective authors using FrameMaker would create a new document based on this template, remove the content, and start writing their paper.

Note that in this template, you may have a multi-paragraph abstract. However, that it is not necessarily a good practice. Try to keep your abstract in one paragraph, and remember that the optimal length for an abstract is 200–300 words.

## 1 Introduction

For the purposes of USENIX conference publications, the authors, not the USENIX staff, are solely responsible for the content and formatting of their paper. The purpose of this template is to help those authors that want to use Adobe Framemaker to write their papers. This template has been prepared by Pekka Nikander, with the help of Jane-Ellen Long, based on an old USENIX FrameMaker template, dating a few years back. The original author of the old template is currently a mystery — if you know, please enlighten us.

Note that this is *not* a FrameMaker+SGML template, and therefore you *cannot* directly produce sane SGML or XML from this document. Additionally, the quality of HTML produced by standard FrameMaker 6.0 is is not that great. If you are interested in using FrameMaker+SGML to produce your USENIX papers, please contact the authors of this paper. They may be able to help you.

The rest of this paper is organized as follows. Next, in Section 2, we give briefly discuss related work, such as other templates and style manuals. Section 3 discusses the details of this template. After that, Section 4 gives some hints on how you may achieve better performance in preparing your papers, and finally Section 5 includes our conclusions that we have been able to achieve so far.

## 2 Related Work

Preparing publications that are easy to read and good looking is not exactly easy. It requires understanding of style and typography. The purpose of the templates provided by the USENIX organization is to lift the burden of caring about typography from the authors. However, the authors still remain, and will always remain, responsible for the style.

### 2.1 Word and L<sub>A</sub>T<sub>E</sub>X Templates

There is a Microsoft Word template available at the USENIX web site. Note, however, that it produces documents that are somewhat different from those produced using this template. There is also a LaTeX template available. To the extend possible, we have tried to conform to the LaTeX template. There is one large difference, though. USENIX prefers to have the abstract to extend across both of the columns while the LaTeX template places it in one column only. If you are using FrameMaker 6, this template does that. In older versions of FrameMaker, the feature is not supported.

### 2.2 Style manuals

Besides typography, style is the second element of preparing easy-to-read publications. There are tens of good style manuals available. To mention just a couple, *The Elements of Style* by Strunk and White [1] is a classic,

and has remained a bestseller since its introduction in 1930's. From the more contemporary ones, *Writing for Computer Science* by Justin Zobel [2] seems appropriate.

### 3 Implementation

In this section we cover the features included in this template. Our goal has been that the authors do not need to make modifications to the template; instead, they should be able to concentrate on the content and style. With this in mind, this template includes a number of features. On the other hand, we have also tried to keep this document simple and easy to maintain. Therefore there is usually just one way of doing things. For example, this template includes only MLA style references and numerical citations instead of including other styles. Note that this citation system does not follow any widely referenced system, such as the Harvard system. Instead, it is often used in the USENIX papers, and it is easy to automate in FrameMaker.

#### 3.1 Paragraph styles

When using FrameMaker, document structure is best preserved and shown when using appropriate paragraph and character styles. This template includes a largish number of paragraph styles; some of the styles have been included just to make the writing process smoother. For example, the style called *0Heading for Abstract* is identical to the style *0Heading* other than it is centered across both columns, automatically included after *Email*, and that finishing it automatically continues *Abstract* instead of *BodyAfterHeading*.

In this section, we cover all the paragraph styles included, starting from headings, continuing to styles used for body text, and finally covering the special purpose paragraph styles.

**Headings.** This templates supports three levels of headings: *1Heading*, *2Heading* and *3Heading*. Additionally, *0Heading* is identical to *1Heading* other than they are unnumbered. *3Headings* run in to the paragraphs; for an example, see the beginning of this paragraph.

**Body text.** Most of the body text will use the *Body* paragraph style. However, if you type your text linearly, every paragraph that follows a heading will use the style *BodyAfterHeading* instead. In this template, the two paragraph styles are almost identical, and you need not pay any attention to this distinction. There is three small difference between a *Body* and *BodyAfterHeading*.

Firstly, the first line of *Body* paragraphs is intended by 0.138" while the first line of *BodyAfterHeading* is not. Secondly, the *Body* style allows page breaks to appear after or before two lines of text while *BodyAfterHeading* requires three lines. This distinction is made so that headings are unlikely to appear at very bottom of a page; that doesn't look too good. Thirdly, *BodyAfterHeading* always includes white space above it while *Body* doesn't. Since headings include white space both above and below, usually this distinction doesn't make any difference. Furthermore, if you ever need to reformat your text with a style where all paragraphs following headings are typeset really differently from others, this distinction becomes handy.

If you need to include bullet lists or enumerated lists in your document, the following styles become handy.

- A *Bulleted* paragraph begins with a bullet sign at the left margin, the paragraph style itself doesn't include white space below or above the paragraph, and the rest of the paragraph is intended. Other than that, they are identical to the *Body* style.

A bullet item can be continued with a paragraph named *Bulleted Continud*. This style includes white space above and below, producing more spacely bullet lists. You can use it for intended text, too.

1. Each numbered list should start with the style *Numbered1*. That style resets the numbering to start from one.
2. A plain *Numbered* just continues numbering regardless of its position. That is, even though there were several paragraphs between this and the previous one, this would be numbered as 2.

There is also a paragraph style named *BodyAfterBullets*. It is meant to be used after a bullet or enumerated list whenever you want to have a little bit of space between the list and the following paragraph.

Sometimes you may want to include snippets of code or other such text. *Code* is the provided paragraph style.

*Abstract* is usually used only in the beginning of the paper. It is automatically inserted after a heading style called *0Heading for Abstracts*, and in FrameMaker 6 it crosses both columns. In older versions of FrameMaker this feature is not supported.

If you really need, we have also included a style called *Abstract Continued*. However, its use is not recommended. Instead, try to keep your abstract as one paragraph.

**Special paragraph styles.** The styles called *Author*, *Affiliation* and *Email* are to be used on the first page only, after the title. See the first page on examples how to use them. *CellBody* and *CellHeading* are used in tables, see *Section 3.6*. *Comments* are discussed below, *Equation* in *Section 3.3*, and *Figure* in *Section 3.4*. USENIX prefers not to have bottom-of-page footnotes, and therefore the paragraph style *Footnote* is made to remind you about this. *Reference* paragraph style is used in the reference list; for an example, see the end of this paper. *Table*, *TableAnchor*, *TableTitle*, and *TableFootnote* are discussed in *Section 3.6*. Finally, *Title* was used to typeset the title of this paper.

**Comments.** *This template includes a paragraph style called Comment. It is not meant to be used in any final version of any paper, but just as a tool for the authors. That is, if you want to leave notes to yourself in the paper you are writing, you can use the Comment paragraph style. Comments are shown in red italics to differentiate them from the rest of the text.*

### 3.2 Character styles

There are just a few character styles provided in this template. **Bold** is for the case that you need to be bold, for some reason. Usually *emphasis* is much more appropriate. Sometimes you need to include code snippets inside plain text, or maybe you just want to *comment* something for yourself. *EquationVariables* style is used for equations, see *Section 3.3*. *Publication* style is meant to be used in the References section. *Symbol* is meant to help including characters from the Symbol font, such as  $\alpha\beta\gamma\delta$ . Finally, *TypeWriter* is another monospaced style, meant for other monospaced information but code, and *URL*'s purpose is to provide a style for URLs.

### 3.3 Equation styles

This version of this template does not include polished equation styles. Anyway, if your paper requires heavy equations, you should most probably use LaTeX instead of FrameMaker. On the other hand, it can be argued that

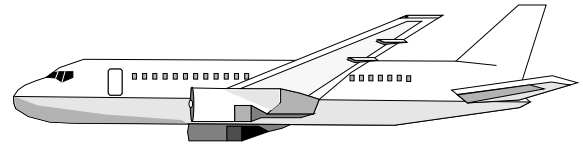


Figure 1: An example figure.

a FrameMaker document should be able to use simple equations. We agree, and if you need those, please contact us so that we can fix this template on demand.

### 3.4 Figures

The best way to include figures in FrameMaker documents is to use anchored frames. To create an anchored frame, select *Special*⇒*Anchored Frame*. For USENIX papers, you usually want to position the figures either *At Top of Column* or *At Bottom of Column*, or sometimes *Below Current Line*. Whatever you do, it is always best to select the *Floating* option so that FrameMaker can place your figure more freely. You should select the anchor point so that the figure stays at its logical location. As an example, Figure 1 is anchored right here: in the middle of this paragraph. If you have turned on viewing text symbols, you'll see the anchor point when editing this text. You can also include figures that go across two columns, like Figure 2. There you just set the width of the figure to be 6.5" instead of 3.125", which is the default in this template.

If you use *At Top of Column*, be sure to set up the bounding box of the anchored frame so that there is enough of space between the figure caption and the following paragraph. Similarly, when using *At Bottom of Column*, be sure to leave enough of space above the figure so that it looks good. However, you may want to leave these kinds of adjustments in the very end of your work, since the location of the figures may change while you revise your paper. Thus, initially it is just best to leave enough of space, within the anchored frame, both above and below of the figure proper.

The figure caption is best included *within* the anchored frame, by inserting a text frame for it, chang-

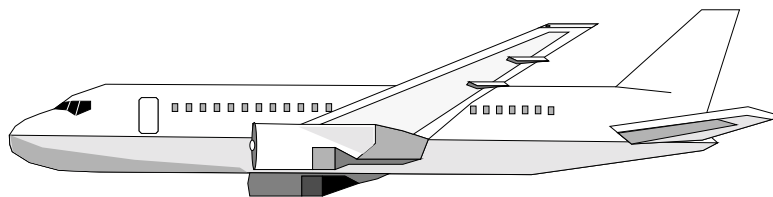


Figure 2: Another example figure.

CellHeading	CellHeading	CellHeading
CellBody	CellBody	CellBody
CellBody	CellBody	CellBody
CellBody	CellBody	CellBody

Table 2: An example of a table that crosses both columns.

ing the paragraph style to *Figure*, and centering the text frame. You can center the text frame by selecting the frame alone, and selecting *Graphics⇒Align...*

### 3.5 Cross references

When referring to appendices, figures included in the text, sections elsewhere in the text, to tables, or citing literature references given at the end of the paper, it is best to use cross references. This template includes specific cross reference styles for the above mentioned things, plus a plain cross reference style called *Numberonly*. We next give examples of these each. A reference to a non-existing appendix would look like this: Appendix . We have already shown Figure 1, and cited the work of Strunk and White [1], and Table 1 will follow. Section 1 is the first section in this paper.

### 3.6 Tables

Typesetting tables in FrameMaker is somewhat problematic. If you can fit in the table within a single column, you can just include a table at the end of a paragraph, like Table 1, next. FrameMaker will almost always place it appropriately.

CellHeading	CellHeading	CellHeading
CellBody	CellBody	CellBody
CellBody	CellBody	CellBody
CellBody	CellBody	CellBody <sup>a</sup>

Table 1: An example table.

- a. An example table footnote. Discouraged.

However, if your table must extend both columns, the situation is more tricky. According to my experience, the best method is to include a two-column wide anchored frame, insert a text frame into the anchored frame, and place the table there, see Table 2. There is the problem, though, that FrameMaker doesn't let you easily put the table on the top of the text frame. Here I have used a trick: the first and only paragraph in the text frame uses

a special paragraph style, *TableAnchor*, whose font size is 0 (zero). That allows you to place the table at the top of the text frame.

Right now, you can use footnotes in Tables. That results in a paragraph style called *TableFootnote*. However, their use is discouraged.

## 4 Performance

In this section we give a number of hints and advice on how you can make your life with FrameMaker smoother. We first cover some basic hints wrt. producing documents in general, and then proceed to some hints and tricks in getting decent PDF and HTML.

### 4.1 Editing in general

The first and foremost thing to remember is that FrameMaker is very much a template based system. That is, you should use paragraph and character templates as much as possible. If you can't do something with the provided paragraph or character styles, it is much better to define a new one than formatting a particular word or paragraph "by hand". Thus, using *Format⇒Font*, *Format⇒Size* or *Format⇒Style* is **heavily** discouraged. Additionally, if you need to create a new paragraph style, please let us know and we may include it in a future version of this template.

The fastest way of changing the style of the current paragraph is hitting CTRL+9, typing the first letter of the desired format, and hitting enter. The character format can be changed with CTRL+8. In both cases, looking at the left lower corner helps. (Try it now.)

If you are using FrameMaker in Unix or Macintosh, you can use many of the emacs key bindings. The Windows version of FrameMaker doesn't follow that convention by default, but it can be tweaked to do so.

FrameMaker also includes decent spelling checker and thesaurus. Use them.

## 4.2 Producing PDF

The newer versions of FrameMaker come bundled with Acrobat Distiller. Consequently, they produce good PDF out of the box. To produce PDF, simply select *File⇒Save As...*, set format to PDF..., and save. FrameMaker will prompt you to change the file extension to *.pdf* (if you didn't change it yourself), and open a dialog to define the document structure. To include proper meta information into the generated PDF file, it is a good idea to go to the *Document Info* pane and type in relevant information.

## 4.3 Producing HTML

In general, FrameMaker is able to produce HTML, but the quality of the produced HTML is not that great. However, partly by good luck and partly by finely crafting the HTML knots within FrameMaker, HTML produced from this template looks reasonably good when viewed either with Mozilla or a reasonably new version of Internet Explorer. This is largely due to the fairly sane Cascading Style Sheet produced. Thus, when you submit your HTML version to USENIX, please remember to include the CSS file as well.

If you want to achieve better results than the default conversion gives, we advice you either to use Quadralay WebWorks Publisher (included with FrameMaker), or to edit the produced HTML file by hand.

## 5 Conclusions

Each good paper concludes the most significant findings in the end.

## Acknowledgments

A polite author always includes acknowledgments. Thank everyone, especially those who funded the work.

## Availability

Please include a section at the end of your paper providing availability information. If the system you describe is available to others, and if more information (reports, etc.) may be obtained, indicate terms and contact information.

This document is available at the USENIX Web site, <http://www.usenix.org/XXX>. A newer version may sometimes be found at Pekka Nikander's Home Page, <http://www.iki.fi/~pnr/>.

## References

- [1] William Strunk Jr, and E.B. White, *The Elements of Style, 4th Ed*, Allyn and Bacon, August, 1999, ISBN 020530902X
- [2] Justin Zobel, *Writing for Computer Science*, Springer-Verlag, December 1997, ISBN 9813083220
- [3] Jane Ellen Long, *Detailed Instructions for Accepted Authors*, Usenix Organization WWW page for 2002 Usenix Annual Technical Conference authors, <http://www.usenix.org/events/useenix02/instrux/details.html>