

The LaTeX environment for the EVU conference proceedings

□

Inhaltsverzeichnis

- [1 Why LaTeX?](#)
- [2 What programs do you need?](#)
- [3 Basic Concept](#)
- [4 Some of the packages used](#)
- [5 Basic formatting in LaTeX](#)
- [6 Proofreading and translations](#)
- [7 Keeping track of the changes applied to the LaTeX documents](#)
- [8 Approval by the authors](#)
- [9 Pitfalls and Caveats](#)
 - [9.1 The dblfloatfix package](#)
- [10 Material on the conference CD-ROM](#)
- [11 Uploading the single papers to the EVU website](#)
- [12 Aftermarket for the conference proceedings](#)
- [13 Additional program tools](#)
 - [13.1 Excel2Latex](#)
 - [13.2 SED](#)
- [14 Additional information](#)

Why LaTeX?

LaTeX offers the possibility to create a single PDF file for the conference proceedings in a very stable environment. It offers direct support for PDF files which can be binded in as figures. LaTeX files are pure ASCII files, which means that they can be edited by any program, which simplifies the translation and review process: proofreaders and translators can directly edit the TeX files.

Furthermore, LaTeX is free software, which means that the conference proceedings can be layouted and compiled by any country group without having to buy expensive layout software. One shortcoming of LaTeX is however that it doesn't offer WYSIWYG; if you want to see what LaTeX's PDF output looks like, you have to initiate a compilation, very much like compiling source code into a computer program, with all that comes along with that, like error and warning messages. For the conference proceedings, we use the pdfTeX compiler which directly produces PDF output.

To answer the most often objection directly: *Microsoft Word* is **not suited** for this purpose. It is no layout program and not stable enough to support large documents with lots of figures.

What programs do you need?

In order to layout papers and compile them to a PDF, you need a TeX distribution and an editor. The best choices under Windows are probably MikTeX (<http://www.miktex.org>) for TeX and

TeXnicCenter (<http://www.texniccenter.org>) for the TeX editor. You should first download and install MikTeX and then TeXnicCenter, this will guarantee that all path variables are set correctly.

When installing MikTeX, you have to decide whether you would like to install a complete system or a basic system – both of them will do. TeX has a modular concept based on so-called packages. Each of them offering a certain specific functionality. If you choose a full installation, this will take rather long and will install a lot of packages that you will never need.

If you opt for a basic installation, additional packages will be downloaded from the Internet on the fly when used by your document. This will cause the first compilation of a EVU conference paper to take rather long, as all the additional packages will be downloaded and installed from the Internet. All subsequent compilation processes will then be performed in normal time.

In order to view the PDF output, you need a PDF viewer such as the *Adobe Reader*. For layout purposes you should however possess *Adobe Acrobat Professional* such that you can really edit PDF files, e.g. cut graphics from a larger page. *Adobe Acrobat* will also allow to produce higher compressed versions of PDFs which speed up the transfer process when sent by e-mail: The PDFs compiled from LaTeX code can become quite large as they take any graphics as-is, e.g. leave high-res JPEGs images as they are. In contrast to this *Adobe Acrobat* will recalculate any raster image to a pre-set resolution.

All others who just want to edit the TeX files (without compiling them to PDFs), i.e. proofreaders, translators and the like, just need a text editor. We would recommend Notepad++ (<http://notepad-plus.sourceforge.net>) for this purpose as it will highlight the syntax of the TeX files, i.e. prevent users from altering control codes. TeX files can, however, also be edited with *Microsoft Word*, such that proofreaders and translators do not really have to install specific software.

Basic Concept

All papers of the proceedings reside in a directory tree with one directory for each paper. This directory is named after the first author of the paper (without special characters and umlauts) and contains all files which are related to this paper, especially the TeX files and all figures. Figures must have one of the formats JPEG, PNG or PDF. The paper itself has the same name as the directory with the language indicator attached, e.g. *hugemann_eng.tex* or *hugemann_ger.tex*.

The root directory of the tree contains the header files with the basic definitions, i.e. the generic *header.tex* and the language-specific header files *header_eng.tex*, *header_ger.tex* and the like. These header files are referred to (via *include*) at the beginning of the LaTeX document of each paper. Every paper has more or less the same intro (header) of TeX commands and the same footer of TeX commands. These basically guarantee that each paper can be compiled on its own (i.e. stand-alone) as well as in the context of the binder document that resides in the root directory of the tree. The compilation of the binder document will create the entire conference proceeding as one single PDF file at the end of the layout process.

During the layout process, each paper is compiled on its own for most of the time, producing a stand-alone PDF for each paper. The compilation of the entire conference proceedings is the very last step, when everything else has been done. There is one master document for every language, which loads all the single LaTeX documents in the correct order. In the final proceedings, each paper will automatically start on a right page (with an odd page number). This is to guarantee that the layouts of the papers will look exactly the same when compiled on their own or when compiled in the context of the entire proceedings.

When compiling the LaTeX-papers first as standalone and last as part of the master document, we have to meet the problem that the start code of each single paper -- *documentclass* and header-section -- has to be ignored in the last step, and the same holds for the end of the standalone documents, i. e. `\end{document}`. This is taken care of by an if-section at the start of each single paper, which checks whether the variable *section* has already been defined (by *document class*), in which case the document class and the header of the single paper is ignored. This trick has been taken from <http://tex.lickert.net/tipps/standaloneinput/standaloneinput.html>, where it is explained in further detail.

The references and the authors' addresses reside in one file *references.tex* for all language versions of a paper. This file is referred to at the footer of each paper.

So in the root directory you have the following files:

- *header.tex* - generic header file with all the packages
- *header_eng.tex* - language-specific header files
- *2009_eng.tex* - binder document for the entire proceedings
- *start_eng* - intro with the welcome message, etc.
- *participants.tex* - participants list

Besides these files there are only a few figures, such as the EVU logo, the photographs of the editors and their signatures. The LaTeX code for the participants list is generated automatically in the Excel file *participants.xls*.

In the paper directories you have:

- *hugemann_eng.tex* - English version of the paper
- *hugemann_ger.tex* - German version of the paper
- *references.tex* - references and addresses for all language versions
- all figures as JPEGs, PNGs or PDFs

Some of the packages used

You'll find all the packages used in the generic header file, accompanied by some comments. We will refrain from naming them all over here. The proceedings use the following larger packages:

- *Koma-Script* - defines "European" document classes
- *Babel* - language support
- *AMS math* - math extensions defined by the American Mathematical Society
- *Siunitx* - SI units for quantities, reformats numbers automatically
- *chapterbib* - allows the use of a bibliography after each paper in the proceedings
- *quotation* - quotations
- *dblfloatfix* - fixes a some bugs connected with page-wide (two-column-wide) floats (i.e. figures)
- *abstract* - defines the environment for the single column abstracts
- *url* - hyphenation of URLs

Basic formatting in LaTeX

The papers will mostly come along as *Word* documents. You can transfer these into LaTeX code in *TeXnicCenter* via the Windows Clipboard. Before doing so, you have to switch off the automatic hyphenation in Word, otherwise the code would be peppered with a lot of superfluous hyphens.

It is however recommended to store the word files in pure text format and treat these with *SED* first, in order to perform a lot of automatic replacements.

Figures should be grouped such that several graphics that make a common statement are grouped to one figure with subfigures. This is disregarded by most authors, but the proceedings of Nice (2008) and Hinckley (2009) give a lot of examples of logical grouping. The authors should provide each figure as a single file, i.e.

- JPEG - for photographs
- PNG, GIF or TIFF - for sketches, i.e. screenshots, scans of police sketches, etc.
- PDF or original format (e.g. XLS) - for vector drawings

It turns out that a lot of authors don't seem to understand or disregard these rules, such that the layouter has to extract the figures from a *Word* document or a PDF:

- You can extract raster images from a *Word* document by storing it in HTML format. This works best with old versions of Word (i.e. Word 97).
- In order to extract raster images from PDFs, use the *PDF Image Extraction Wizard*.
- Vector images can be cut from PDFs by extracting the according page, deleting everything but the vector graphic and cropping the page to the size of the graphic. This can be performed with Adobe Acrobat.

When sending the first layout back to the author, you should ask for the original files of the graphics you extracted in the first run. This is very important if the resolution of the graphics does not suffice the needs of high resolution printing.

PDFs supplied by the authors should immediately be tested in regard to conformity. A lot of PDF conversion is performed by crude printer drivers which may produce non-conform PDF output. This testing in regard to PDF conformity can also be performed by use of *Adobe Acrobat Professional*. Please keep in mind that the *pdftex* compiler can only deal with older versions of PDF up to 1.3, corresponding to Acrobat Version 4. Newer PDF can however be binded in as long as they do not make use of features that only exist in higher PDF versions.

Proofreading and translations

Before initiating any translation, you should have achieved a stable version of the LaTeX document in one language. As far as possible, proofreaders and translators should apply their changes directly to the LaTeX file. This will spare the layouter from a lot of work and exclude a lot of possible error sources. Any translation should be proofread by a native speaker at last. To enable those who don't possess a LaTeX compiler to perform their revisional work, the layouter should supply them with the LaTeX file as well as with the compiled PDF of the paper. This PDF may have a reduced resolution to speed up the transfer by e-mail.

In the past, we shipped a free copy of the proceedings to those translators who did not attend the conference.

Keeping track of the changes applied to the LaTeX documents

As LaTeX is pure ASCII, there is no built-in version control. With several people working with the files - the author, the layouter, the translator -, all of them possibly several times, you should

however keep track of the changes applied.

Each LaTeX document should therefore start with a header of comment lines explaining the changes applied to the document, with the most recent change at the top. This is what the header of the 2009 papers looked like:

```
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% T. Hugh Woo
% Application of Video-Aided Mapping
% in Verifying Signal Violation - A Case in Taiwan
%
% Version 1.3, 2009-09-03 16:00
%   capitalisation of headings corrected
%
% Version 1.21, 2009-08-28 15:40
%   "Proofread by Richard Lambourn"
%   [normalheadings]
%
% Version 1.2, 2009-08-24
%   Figure 2 (signalling table) exchanged
%
% Version 1.1, 2009-06-30
%   Proofread by Richard Lambourn
%
% Version 1.0
% 2009-06-22
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
```

In order to avoid misunderstandings caused by different date formats, the dates are given as YYYY-MM-DD.

Approval by the authors

It should be obvious that the final version of a paper (in all languages) should be approved by the author. As there might be deviations between a paper compiled on its own and a paper compiled in the context of the binder document (i.e. the entire proceedings), the authors should finally be given the opportunity to have a look at the entire proceedings.

It is recommended to upload the final version (in all languages) to an Internet server and providing the authors with the download links to the paper via e-mail. This version of the proceedings should have a reduced resolution, i.e. about 5 MB in total, intended for screen display.

Pitfalls and Caveats

The LaTeX compiler tries to correct any error in the LaTeX code, e.g. a missing "}", automatically. So if there is a severe error in your code, the compiler will usually produce a whole lot of error messages, originating from LaTeX's attempts to correct follow-up errors resulting from its attempts of correction. Usually, the line number of the code bug is given, which is mostly correct. There are however error reports that do not refer to the correct line. In such case you should truncate the

document by inserting an "`\end{document}`" statement somewhere in the LaTeX code, as this will stop the compilation. Placing this statement in different location will then help to bracket the error source.

Please keep in mind that any document has to be compiled **twice** in LaTeX in order to resolve the cross-references, citations, establish odd and even page numbers, tables of contents, etc. Usually, you will get some warning messages referring to this in the first compilation run. LaTeX will mark missing links with question marks, e.g. [?] for a missing reference for a citation. The best check for any missing references is to search for question marks in the PDF output, which are usually rare in technical papers. As an alternative, you can check the logfile after the compilation.

LaTeX places floats according to its algorithms and there is no way to "nail" a float at a certain spot. LaTeX tends to place floats "behind" their placing in the LaTeX code, so it's recommended to place the code somewhat before the float (i.e. the figure) is referenced in the text. If the paper uses a lot of figures, the placing of the figures within the LaTeX code has to be adjusted manually applying to the trial and error method (no kidding).

If possible by any means, the reference to a figure should appear on the same double-page as the figure itself. When checking the layout, please ensure that your PDF viewer places the pages correctly, i.e. odd page numbers on the right. Newer versions of the Adobe Reader do not respect this by default.

A lot of hints in regard to LaTeX papers with two-column layout can be found in http://css.paperplaza.net/conferences/support/files/IEEEtran_HOWTO.pdf. In plain LaTeX it is for instance impossible to place a double-column float at the bottom of a page. If you need that functionality, you have to use the package *dblfloatfix.sty*.

You should ensure that you are working with a recent (possibly the newest) version of any package used, to benefit from bugfixes. If you install a fresh version of MiKTeX (currently 2.8), you can be pretty sure that all packages shipped with this version are quite recent. Also any package automatically downloaded by TeXnicCenter will be the current one.

If however TeXnicCenter / MiKTeX refuse to download a certain package, you should update MiKTeX's data repository; please see the manual. In very rare cases, the newest version of TeX packages are still buggy. In this case, the corrected package file (*.sty) has to be copied in the according directory. (This is for instance the case with *dblfloatfix.sty*.)

The *dblfloatfix* package

By default, LaTeX positions page-wide figures (i.e. two-column figures) at the top of a page and there is no way to influence this behaviour by standard means. If you need to place page-wide figures at the page bottom (because there are just too much of them), you have to load the package '*dblfloatfix*', which establishes this functionality.

This package however has a severe bug, causing it to treat a single-column figure occasionally as a two-column figure, leaving blank space in the neighbouring column. This bug can be fixed by inserting one line of code in the style file '*dblfloatfix.sty*'. This is a plain text file in TeX format, which can be edited like any other LaTeX document.

The ready-made solution can be downloaded at <http://newsgroups.derkeiler.com/Archive/Comp/comp.text.tex/2008-08/msg01124.html>. It does however suffice to insert just one line in the style file at line 143:

Original	Altered style file
<pre> \if@tempswa \@bitor \@currtype \@deferlist \if@test \else </pre>	<pre> \if@tempswa \@bitor \@currtype \@deferlist % Test for twocolumn floats (2008/08/21 Mattias Nissler) \@testwrongwidth\@currbox \if@test \else </pre>

Material on the conference CD-ROM

Typically, the printed conference proceedings are accompanied by a CD-ROM handed to the participants together with the printed proceedings. This contains a high-res version (i. e. the one given to the printers) of the conference proceedings (one for each language). The presenters should however be asked for additional material such as

- the slides of their presentation
- original files of their diagrams
- original data of the investigation
- video footage
- colour photographs
- explanatory text, e. g. giving the true frame rate of slomo videos

Again, the directory tree of the CD-ROM should reflect the single papers, the directories being named after the first author of the paper. The root directory of the CD-ROM should contain a HTML mentioning the conference programme and linking to the single papers. Please refer to the 2009's CD to see an example.

Presenters who don't want to supply their original presentation (as this might be re-used by others) should supply a PDF version of it.

Uploading the single papers to the EVU website

After the conference, the single papers should be uploaded to the EVU website, see [example](#). Please mind: Do **not** upload the entire proceedings as a single file. This material should be restricted to those who either attended the conference or bought the conference proceedings.

The single papers can be extracted from the entire proceedings (e.g. with Adobe Acrobat) and stored with reduced resolution (simple printing). The abstract should become the descriptive text in the paper database.

Aftermarket for the conference proceedings

There is a (limited) aftermarket for the conference proceedings, especially for the German and English version. These versions have been sold in numbers of about 20 copies after each conference. Which means that you should print some more copies than you actually need for the participants. The copies are sold on account of the organising country group, either by the EVU country group itself or by the EVU headquarters.

Additional program tools

For the layouter, we would also recommend some additional tools:

- 'SED (*the Streamline EDitor*) <http://gnuwin32.sourceforge.net/packages/sed.htm>) to perform some automatic search and replace operations when transferring the Word documents into LaTeX code.
- To easify the conversion of Excel tables into TeX format, we recommend the Excel Add-In *Excel2Latex* (see below).
- Use the *PDF image Extraction Wizard* <http://www.rlvision.com/pdfwiz/about.asp> to extract raster images (especially photographs) from PDFs.
- A diff tool such as *Tkdiff* (<http://sourceforge.net/projects/tkdiff>) to check changes made to TeX files.

Excel2Latex

The Excel-Addin *Excel2Latex* is an XLA file (i.e. a compiled macro). The Add-In should be placed in Excel's Add-In directory, which seems to change its location with each new Excel version. The best thing to do is to look for the folder containing several XLA files and placing the file therein. Then open the Add-In Manager and activate the check box of the Add-In.

The Add-In works for Excel versions 2000 and higher. In Excel 97, you run into an error at first and have to replace the VBA function *Replace* by *WorksheetFunction.Substitute*. After that, save the XLA file and the macro will also function in Excel 97.

When using the Add-In, mark the spreadsheet area that you want to convert into a LaTeX table and run the macro from the Tools menu. This will open a text box displaying the LaTeX code, which has to be copied into the LaTeX document.

The Add-In file can be found at [CTAN](#) and [DANTE](#).

SED

The ready-made Windows installation program *sed-4.2-1-setup.exe* (with varying version number) can be downloaded at <http://gnuwin32.sourceforge.net/packages/sed.htm>. By default, it will install SED in %PROGRAMFILES%\GnuWin32\bin. This path can be included in the computer's search path, such that SED can be called directly in any folder. This is at least recommended for the test phase.

In order to perform the most frequent substitutions on a paper, save its Word document as a plain text file (File > Save as ... > plain text). SED will use this text file for input and will perform most of the required substitutions automatically. For test purposes, store the listing below in a text file named *evu.txt* in an empty folder.

For the initial testing, we also set up a test file that SED can work on, which you will also find below. Store this listing in a file named *paper.txt* in the same folder as the definition file *sed.txt*. Open a DOS box, move to the test folder and type

```
sed -f evu.txt paper.txt
```

or, if you did not change the path environment variable

```
%programfiles%\gnuwin32\bin\sed -f evu.txt paper.txt
```

This will display the altered text on the screen. In order to dump the text in a file called *paper.tex* use a redirect:

```
sed -f evu.txt paper.txt >paper.tex
```

For frequent and convenient use, you should use the batch file below (last listing) and save it in the [SendTo](#) folder. After that, you can right-click on any text file and perform the needed conversion on it, creating a file with the same name and the extension *tex*.

SED's definition file:

```
#####  
# SED definition file for EVU papers  
#  
# Version 1.0, 2010-09-06 by Wolfgang Hugemann  
#  
# General comments:  
# All substitution are global (i.e. every occurrence in a line),  
# hence every command ends with the option 'global', i.e. /g  
#  
# Ranges are detected first and substituted, such that  
# any numbers left are not part of a range.  
#####  
  
# change units to their (SI) abbreviations  
# change singular or plural, i.e. [s]*  
s/metre[s]*/m/g  
s/second[s]*/s/g  
s/mile[s]* per hour/mpg/g  
  
# number mph -> number~mph  
s/ \([0-9][0-9.,]*\) [ ]*mph/ \1~mph/g  
  
# change Anglo-American times, e.g. 8.30am -> 8.30~am  
s/\([0-9][0-9.,]*\) [ ]*\([ap]\)m/\1~\2m/g  
# detect number ranges separated by "to" n1 to n2 -> n1 - n2  
s/\([0-9][0-9.,]*\) [ ]*to[ ]*\([0-9][0-9.,]*\) / \1 - \2/g  
  
# detect number ranges separated by "bis" (German) n1 to n2 -> n1 - n2  
s/\([0-9][0-9.,]*\) [ ]*bis[ ]*\([0-9][0-9.,]*\) / \1 - \2/g  
  
# Ranges of physical values with single-letter units, e.g. m, s, g  
s/\([0-9][0-9.,]*\) [ ]*- [ ]*\([0-9][0-9.,]*\) [ ]*\([gms]\{1,\}\)\([ ,. )&$\) / \SIrange{\1}{\2}{\3}\4/g  
  
# Ranges of physical values containing a divisor and possibly ending on "²",  
e.g. km/h, m/s, m/s²  
s/\([0-9][0-9.,]*\) [ ]*- [ ]*\([0-9][0-9.,]*\) [ ]*\([a-zA-Z]\{1,\}\)\/\([a-zA-Z]\{1,\}\)^2*\)\([ ,. )&$\) / \SIrange{\1}{\2}{\3}\4/g
```

```

# Percent ranges -> \SIrange{n1}{n2}{\percent}
s/\([0-9][0-9.,]*\) [ ]*-\[ ]*\([0-9][0-9.,]*\) [ ]*\([
\\]*%\)/\\SIrange{\1}{\2}{\\percent}/g

# number % -> \SI{number}{\percent}
s/\([0-9][0-9.,]*\) [ \\]*%\\SI{\1}{\\percent}/g

# Physical values with single-letter units, e.g. m, s, g
s/\([0-9][0-9.,]*\) [ ]*\([gms]\{1,\}\)\([ ,.)&$)\)/\\SI{\1}{\2}\3/g

# Physical values containing a divisor and possibly ending on "²", e.g. km/h,
m/s, m/s²
s/\([0-9][0-9.,]*\) [ ]*\([a-zA-Z]\{1,\}\[/[a-zA-Z]\{1,\}²*\)\([
,.)&$)\)/\\SI{\1}{\2}\3/g

# Number ranges (without physical units)
# other ranges must have been substituted by the previous statements
s/\([0-9][0-9.,]*\) [ ]*-\[ ]*\([0-9][0-9.,]*\)\\numrange{\1}{\2}/g

# plain numbers (without physical units)
# other numbers must have been substituted by the previous statements
s/\([0-9]\{1,\}\[,]\{0-9}\{1,\}\)\([ ,.)&$)\)/\\num{\1}\2/g

# "text" -> \enquote{text}
s/"\(.*\)"/\\enquote{\1}/g

```

(To be stored in *evu.txt*.)

Test file:

The lines

The test lasted for 2.5 seconds in a speed speed range of 20.5 to 23.5 km/h.
The engine ran on 50% of its "maximum" power.

should be stored in *paper.txt* and should result in an file *paper.tex* containing the lines:

The test lasted for \SI{2.5}{s} in a speed speed range of
\range{20.5}{23.5}{km/h}.
The engine ran on \SI{50}{\percent} of its \enquote{maximum} power.

Batch file for the SendTo folder:

```

"%PROGRAMFILES%\gnuwin32\bin\sed" -f "%PROGRAMFILES%\gnuwin32\bin\evu.txt" %1
> "%~dpn1.tex"

```

Please note that the redirect symbol ">" is **not** escaped by a "^" (caret or circumflex), as the complete line is interpreted by SED and is not a genuine DOS command line.

Additional information

- [Guidelines for presenters at EVU conferences](#)