LaTeX at DSV2:PVT

Tobias Wrigstad
tobias@dsv.su.se

20 november 2002

Kapitel 1

Introduction

1.1 What is this LaTeX, anyway?

Now, I can begin to write some text. Note that the text automatically becomes size 11 pt because of the declaration in the preamble. Text is automatically justified (raka höger- och vänstermarginaler) and new paragraphs are automatically indented. Note (in the code) how linebreaks and double spaces are ignored by LaTeX.

A new paragraph is started by an empty line in the text. Note the indentation. This is clearly visible from the source code. *N.B. You should read this document in parallel with the source code!*

1.1.1 Text formatting

Text can be *emphasized* (which more or less means *italicized* in the book template). Text can also be **bold faced** or in SCRIPT CAPITALS or without serifs i.e. sans serif.

An easy way to show code snippets is to use the `verbatim` (note how I switched to typewriter text just for one word or phrase) environment:

```
This text is preformatted, meaning
that linebreaks
are
    not
        ignored as well as
double           spaces (%% comments are not ignored)
```

In the preformatted text section, text is written exactly as it is typed in the source code. Line breaks are *not* automatically inserted when the end of the line is reached. Instead text can flow into the right margin or even out of the page! Thus, lines must be broken by hand in the `verbatim` environment.

The next level of sectioning lacks a number

Thus, it will not appear in the table of contents.

1.2 The use of tables

First: note how this section's name is showed in the header. All pages (except the first page of every chapter and the title page) automatically becomes a header with a section name and number.

You may wish to use tables in your project documentations. Tables are quite easily created. Below, a simple sample table is shown. For a more comprehensive documentation of how to create tables, see 'The not so short guide to LaTeX'.

The table below is generated with this code:

```
\begin{tabular}{|l|r|c||p{1.5cm}|}
\hline
left justified & right justified &
centered & width specified \\
\hline
\hline
\end{tabular}
```

left justified	right justified	centered	width specified

Text may be formatted with italics, bold face etc. inside a table. However, as we can clearly see from the appearance of this page, the placing of the table is not correct. It almost overlaps with the text of this paragraph. Tables should always be placed in a special environment called a *float*. A float contains a table or a figure and ensures that it is placed correctly in the document. If there is not enough space on a page to place a float, the float *floats* over to the next page, hence the name. Now, we place the table inside a float:

left justified	right justified	centered	width specified

Tabell 1.1: My table

Much better, don't you think? Note that we have also given the float a caption. Since the float was a table float (examine the code), the caption automatically becomes the text 'Tabell'. This would not be possible if we did not place the table in a float since then, LaTeX cannot determine where the table and the explanatory text begins or ends.

c	p
1	lines inside a column with a specified width will be broken automatically if their length exceeds the width of the column. This is not the case for centered, left justified or right justified columns where lines must be broken by hand
10	Ten
1000	Thousand

Tabell 1.2: This table is autmatically placed at the top of the page where it appears in the code

1.3 Figures

Figures should also be placed in floats and be given captions. Figures should contain diagrammes etc. Figures are included with the use of this command:

```
\includegraphics[options]{name of .EPS file}
```

Where the options are a list of keys with values separated with commas:

scale= A scale factor from in per cent.

angle= An angle

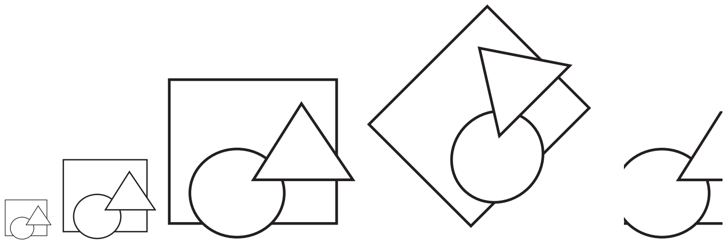
bb= Creates a *bounding box*, i.e. a rectangle within the included image. Everything outside the bounding box is omitted. A bounding box is specified by four values, first the x and y values of the lower left corner and then the x and y values of the upper right corner.

clip= True or false. Specifies if the the contents outside of the bounding box should be omitted or not.

The figures on the next page were imported with the following commands:

```
\includegraphics[scale=.25]{sample.eps}
\includegraphics[scale=.50]{sample.eps}
\includegraphics{sample.eps}
\includegraphics[angle=45]{sample.eps}
\includegraphics[bb=4mm 0mm 17mm 17mm, clip=true]{sample.eps}
```

They all reside in the same float. The last image is clipped by the use of a bounding box. Normally, using a bounding box is not necessary.



Figur 1.1: The same figure included five times

1.4 Lists

LaTeX supports the generation of lists in many ways; there are bullet lists, enumerated lists, descriptive lists etc. Below, the most common lists are shown. Look at the source code to see how they are generated.

Bullet list

- In a bullet list, all list items are preceeded by a bullet, •

    – Lists can be nested, i.e. a list item can contain a new list (even of a different kind). Note how the appearence of the nested list changes.

- New items are automagically indented

Enumerated list

1. In a enumerated list all list items are preceeded with a number.

    (a) In nested enumerations, different enumeration styles are used.

        i. A maximum of four levels of nesting is supported.

2. New items are automagically indented

Descriptive list

Descriptive list A descriptive list is a list where each item has a name written in bold face. Note how a descriptive list was used to show the options of includegraphics in Section 1.3, page 4.

1.5 Floats

Just a short note on floats. Always place tables and figures in floats. Use table floats for tables and figure floats for figures. Look at the source code on lines 162 and 265. Read the comments and experiment!

1.6 Structuring your documents

This purpose of this document is to be a guide to formatting documents in LaTeX. To be readable codewise – it is very badly structured (even technically). When you write your own documents in LaTeX you must not be as sloppy as this.

1.6.1 File inclusion

A document can be divided into several different files to facilitate different people working at the same document at the same time. Files are included with the command:

```
\input{file name}
```

File name must be without spaces and file extension (extension *must* be TEX). If a file is placed in another directory, the entire path from the main file to the included file must be specified, *e.g.* "`../../../myfile`".

1.6.2 Cross referencing

In large documents (such as those that you are about to write) it is often necessary to refer to a different place in the text. This is easy in LaTeX with the commands:

```
\label{name}
\ref{name}
\pageref{name}
```

The `\label` command associates a name with a place in the document text. This can be issued almost everywhere. By using `\ref`, the number of the section (or float or list item number etc.) where the label appeared is printed; `\pageref` prints the number of page where the label appeared is printed. Depending on where the `\label` was issued, different numbers appear. Experiment! Look at lines 220 and 329 in the source code for an example.

1.6.3 Citation with the bibliography

Hopefully, you will divide your project documentation into several different documents. Needless to say, you will sometimes (or quite often) need to refer between these documents, e.g. to say that X is explained in Y etc.

To refer to external documents, you must first create a bibliography (litteraturförteckning) of documents that you wish to refer. A bibliography is included at the end of this document (look at the source code to see how it is constructed).

To refer to an external document, the command `\cite` is used. It works analogously with `\ref` except that there is no corresponding label. Instead, the names that may be cited are specified by the bibliography at the end of the document. The bibliography at the end of this document contains three documents with the names 'gof-patterns', 'captain69' and 'template'. Either of these names can be used by cite. For example, `\cite{captain69}` produces [2].

In text, you usually write, '…for a more detailed description of the proxy-pattern, see [1]…'

When referring to your own documents, you generally need to specify more exactly the location of the text to which you are referring, e.g. '…for a more detailed description of the proxy-pattern, see [1, pp. 257-263]…'

This is achieved by `\cite[pp. 257-263]{gof-patterns}`. The text in the brackets can be any text, not necessarily page numbers.

1.6.4 Table of contents

In LaTeX, a table of contents is easily generated by issuing the command

```
\tableofcontents
```

The table of contents in the course documentation is automatically generated using this command. (Exercise: insert it in this document and see the results – note that the LaTeX compiler will tell you to rerun the compilation at least one time!)

1.7 Line breaking

In LaTeX, line breaking is a real issue. This is due to that LaTeX must obey a number of typographic rules that is embedded in the system. In some cases, e.g. LaTeX don't know how to break a very long word, it is necessary to manually hyphenate or break the line by hand. There are a couple of different approaches to this. The line below illustrates one problem:

1) ThislinedoesnotcontainanyspacesandwhenLaTeXtriestobreakititwillhaveproblems.

hyhenation When LaTeX doesn't know how to hyphenate a word, you can do it manually with a special (soft) hyphen, `\-`. This will insert a hyphen *if necessary*. If the text is altered later and suddenly, the long word fits into one line without hyphenation, the hyphenation will not be inserted. A long word could have many soft hyphens to enable hyphenation at several different places.

linebreak The command `\linebreak` or `\linebreak[num]` breaks a line and justifies the margins. If you use `\linebreak[num]`, you must specify a number between 0 and 4 that is an indication of how much you want the line broken (higher number indicates higher need for line breaking). This enables the removal of deprecated line breaks that are no longer necessary due to subsequent changes in the text.

newline The command `\newline` breaks a line. There is a shorthand for `\newline` in the command `\\` or `\\[length]`. The latter breaks a line but also specifies how much whitespace should be inserted before next paragraph.

Sometimes, LaTeX is unable to justify the right margin of a section, simply because there is not enough words on the for LaTeX to insert enough space between so to give the line the desired length. This is signalled by the LaTeX-compiler as an "underful hbox".

There is one such example in this document in the last column in the Table 3. Look in the code on the first table on the same page to see the use of `\newline` (line 175 in the code). You may also want to look at the output of a compiler, fix the faulty table and recompile. Note that, depending on the situation, `\linebreak` might actually be the solution (if you want the justified margins). This command does not, however, stretch the length of a word.

1.8 Running LaTeX

This is a brief explanation of how to transform a TEX source file into a viewable/printable document. LaTeX is invoked from the command line with the main file as single parameter:

```
$ernst-hugo>latex my-file
```

This will cause the file to be 'compiled' (in some cases it must be compiled several times due to forward references etc., in this case LaTeX will print a message indicating that the file should be recompiled) and produce a viewable file called a DVI file (DeVice Independent). If there are any compiler errors, these will be reported and the compilation aborted (without producing a DVI file). The DVI file is viewed with the command:

```
$ernst-hugo>xdvi my-file.dvi
```

Some machines may have additional (and better) programs for viewing DVI files, like `kdvi` (available in the k desktop system, `www.kde.org`).

After a DVI file is produced, another program called `dvips` is used to produce a postscript file or feed it to the printer. To create a postscript file from a DVI file (that can be emailed to your contractor):

```
$ernst-hugo>dvips -ta4 -Ppdf -ooutfile.ps my-file.dvi
```

This will cause the contents of my-file.dvi to be printed to the postscript file `outfile.ps`.

Please be careful when using `dvips`! If you forget the `-o` flag, your prints will be spooled to the printer instead of to a PostScript file!

To print the contents of a DVI file to a printer:

```
$ernst-hugo>dvips -ta4 my-file.dvi
```

To convert a PS file to a PDF file:

```
$ernst-hugo>ps2pdf13 mypsfile.ps
```

This creates a new PDF file names `mypsfile.pdf`.

If `ps2pdf13` is not available, but `ps2pdf` is, then the same result can be achieved using `ps2pdf -dCompatibility=1.3 myfile.ps`.

1.9 How do I start?

Copy the first 61 lines of this file and add a `\end{document}`. Strip the comments if you think that you don't need them. This should compile just fine. Alter the title, author and date. Now you have a one-page LaTeX document with only the titlepage. Now, start writing between `\begin{document}` and `\end{document}`. The bibliography is easily copied from this document into your new document and the contents of the bibitems altered[1]. Good luck!

Comment On the course DSV2:PVT, all documentation managers will be supplied with a copy of 'The not so short guide to LaTeX' and also be given an introduction on how to run LaTeX and write a small TEX file. It might be a good idea to assign a project member to work out the details of LaTeX that you think you need. That way, you will produce a local expert that can be called in to solve LaTeX problems as they arise.

[1] Tip: you can use the same bibliography file for all documents and just include it with the input command. Or better still, learn BibTeX, which is a very potent powerful bibliography system for LaTeX.

Litteraturförteckning

[1] *Design Patterns, Elements of Reusable Object-Oriented Software*. Erich Gamma, Richard Helm, Ralph Johnson, John Vlissides. Addison-Wesley 1995.

[2] *Safe as Milk*. Captain Beefheart, Zoot Horn Rollo, Winged Eel Fingerling, Alex Snouffler, John French. Budda Records 1969.

[3] *Title of document*. List of authors in the order they appear in the document Name of publisher, or other remark and year of publication. Document number or version number is good to include.