

## LaTeX Workshop

# Tidying up Documents

Part 2 of our LaTeX series focuses on putting some structure into your LaTeX documents – we will be looking into dividing documents into sections and chapters, creating tables of contents, using various list formats and tables, and working with cross-references and footnotes.

BY HEIKE JURZIK

Intuitive commands are available for dividing LaTeX documents into sections and chapters. Before doing so, you will need to define multi-tiered headings that automatically enumerate your document and use different fonts.

The following structure elements are available for the various LaTeX document classes (*article*, *book*, *report*, [1]):

- `\section{...}`
- `\subsection{...}`
- `\subsubsection{...}`
- `\paragraph{...}`

The text for the heading is enclosed in curly brackets. Note that these commands will need to occur in the correct order: the first `\section` is section number 1, the following `\subsections` are numbered 1.1, 1.2 etc. and the following `\subsubsections` are numbered 1.1.1,



1.1.2 etc. (see Figure 1). The `\paragraph` structure element is used to define the words to be highlighted at the start of a paragraph. The document classes *book* and *report* additionally provide the `\chapter` element.

This kind of structure normally lends itself to a table of contents. You can use the LaTeX `\tableofcontents` to create one and place it at an appropriate position in your document (see Figure 2).

## List Formats

LaTeX provides a collection of list formats, such as numbered and unnumbered formats. The latter type – also referred to as bullet lists – are available in the *itemize* environment; that is, you first type an opening command `\begin{itemize}`, then the individual points, prefacing the `\item` keyword, and close the environment by typing `\end{itemize}`.

Lists of this type can be nested; an *itemize* environment can contain up to four additional levels. Nested levels are indicated by different bullets and indents. Indenting source code is also recommended, to provide for better

readability (see Figure 3). Numbered list formats are available in the *enumerate* environment; they are indicated by consecutive numbers that normally start at 1. Again four levels of nesting are permitted:

```
\begin{enumerate}
\item number 1
\item number 2
  \begin{enumerate}
    \item number 2.1
    ...
  \end{enumerate}
\end{enumerate}
```

The top level of the *enumerate* environment starts with the number 1, small letters are used for the second level, Roman numbers for the third, and capital letters for the fourth. If you need to change the symbols for these levels, you can either apply your changes globally to the document, or locally for a list item.

To change only a single `\item` command, simply append the new symbol in square brackets, e.g. `\item[+]`. To change the document, you will need to modify the preamble for the current

## 1 Section

```
\section{Section}
```

### 1.1 First Subsection

```
\subsection{First Subsection}
```

#### 1.1.1 Subsubsection

```
\subsubsection{Subsubsection}
```

### 1.2 Second Subsection

```
\subsection{Second Subsection}
```

```
\tableofcontents
```

## Contents

1 Section	1
1.1 First Subsection .....	1
1.1.1 Subsubsection .....	1
1.2 Second Subsection .....	1

Figure 1: Structuring LaTeX Documents

Figure 2: Creating a Table of Contents with LaTeX

```

\begin{itemize}
\item Linux Magazine
\begin{itemize}
\item 2002
\item 2003
\begin{itemize}
\item Issue 27: February 2003
\begin{itemize}
\item User tools, P. 68
\item Driving Data, P. 70
\end{itemize}
\end{itemize}
\end{itemize}
\end{itemize}
\end{itemize}

```

Figure 3: Nested Bullet Lists

```

\begin{itemize}
\item Linux Magazine
\begin{enumerate}
\item 2002
\item 2003
\begin{enumerate}
\item Issue 27: February 2003
\begin{description}
\item User tools, P. 68
\item Driving Data, P. 70
\end{description}
\item Issue 28: March 2003
\end{enumerate}
\end{enumerate}
\end{itemize}

```

Figure 4: Mixed Lists

layout in the *.tex* file. To do so, define the required command by typing the following `\renewcommand` new:

```

\renewcommand{\labelitemi}{+}
\renewcommand{\labelenumi}{+}

```

The levels available in the *itemize* environment are called `\labelitemi`, `\labelitemii`, `\labelitemiii`, and `\labelitemiv`. The *enumerate* environment refers to them as `\labelenumi`, `\labelenumii`, `\labelenumiii`, and `\labelenumiv`.

The *description* environment provides so-called definition lists for use with glossaries and the like. The entries each comprise the expression to be defined and the corresponding description:

```

\begin{description}
\item[Linux Magazine] ⚡
The magazine for advanced ⚡
Linux know-how
\item[Linux-Magazin] ⚡
German sister publication ⚡
of Linux Magazine
\end{description}

```

All of the list formats described so far can be nested in a mix and match fashion, but do make sure that you include an opening and closing command for each list format (see Figure 4).

## Tabulators and Tables

LaTeX ignores spaces and tabs in the source code when you launch *latex*. So, you will again need some special LaTeX commands if you want to insert a tab stop.

The so-called *tabbing* environment allows you to:

- set a tab stop by typing `\=`
- jump to a tab stop by typing `\>`

Lines are separated by typing `\\`:

```

\begin{tabbing}
This is \=a tab and ⚡
\= this is another one\\
\end{tabbing}

```

You can then type `\>` in the following line to jump to the tab presets (Figure 5).

As the example shows, *tabbing* is not exactly a practical way of creating tables. Fortunately, LaTeX provides an additional environment for tables, known as the *tabular* environment. The format is as follows:

```

\begin{tabular}[position]⚡
{column type}
...
\end{tabular}

```

The parameter indicating the position can either be a *[t]* (justifies the top line of the table with the current environment) or a *[b]* (justifies the bottom line of the table with the current environment). If you leave out the parameter the vertical center of the table is justified. The second option indicates the number of columns and their justification; the *{column type}* parameter can include the following definitions:

- *l* – left-justified column
- *r* – right-justified column
- *c* – centered column
- *|* – vertical line separates columns

At the same time, the number of columns is enclosed in square brackets.

Thus, you can use the following syntax to define three left-justified columns and use a single separating line:

```

\begin{tabular}[t][|l|l|l|]
...

```

The following are available inside a table environment:

- `&` – separates the columns
- `\\` – indicates a line end
- `\hline` – a horizontal line the width as the table

The column width is defined by the length of the content; advanced LaTeX users can, of course, use individual parameters and other formatting syntax to modify any defaults. Let's just look at one last feature: creating multicolumn headers. The `\multicolumn{number-of-cols}{justification}{text}` command allows you to define, the number of columns, the justification, and the text for a heading. Thus, a few simple instructions allow you to create easily readable tables (see Figure 6).

## Cross-References

LaTeX documents can include cross-references to other positions in the

```

Issue Cover Date Cover Story
27 Feb. 03 Security

\begin{tabbing}
Issue \=Cover Date \=Cover Story \\
27 \>Feb. 03 \>Security \\
\end{tabbing}

```

Figure 5: Setting Tab Stops

Index of Linux Magazine		
Issue	Cover Date	Cover Story
27	Feb. 03	Security

```

\begin{tabular}[t]{t}{|l|l|l|}
\hline
\multicolumn{3}{|c|}{Index of Linux Magazine} \\
\hline
\hline
Issue & Cover Date & Cover Story \\
\hline
27 & Feb. 03 & Security \\
\hline
\hline
\end{tabular}

```

Figure 6: Simple Tables in LaTeX

document wherever you need them. Thus, you can refer to other chapters or to pages in the document. Cross-references of this type comprise two elements – a label and the reference itself. You can use the `\label{name}` command to insert the former. Two commands are available for referencing labels:

- `\ref{name}` – uses the chapter number
- `\pageref{name}` – uses the page number

Label names must be unique, and they are also case-sensitive:

```

...
\section{Cross-References}

LaTeX documents can include cross-references to other positions in the document wherever you need them. Thus, you can refer to other chapters

```

## 1 Cross-References

LaTeX documents can include cross-references to other positions in the document wherever you need them. Thus, you can refer to other chapters or to pages in the document. Cross-references of this type comprise two elements – a label and the reference itself. You can use...

See also the section 1.1, which you will find on p. 1.

### 1.1 How to use labels

...

Figure 7: LaTeX setting cross-references

It is just as convenient and easy to use footnotes in LaTeX documents. A single command is all it takes to define and enumerate a footnote and automatically place it in small type at the end of a page<sup>1</sup>. You can change the enumeration<sup>13</sup> or the typeface or font in the footer text itself<sup>2</sup>, or even use mathematical formulae<sup>3</sup>.

It is just as convenient and easy to use footnotes in LaTeX documents. A single command is all it takes to define and enumerate a footnote and automatically place it in small type at the end of a page\footnote{in the {\tt article} document class}. You can change the enumeration\footnote[13]{by using a number in square brackets} or the typeface or font in the footer text itself\footnote{\it italic} or {\bf bold}}, or even use mathematical formulae\footnote{\sum\_{i=0}^n n\$}.

<sup>1</sup>in the `article` document class  
<sup>13</sup>by using a number in square brackets  
<sup>2</sup>*italic* or **bold**  
<sup>3</sup> $\sum_{i=0}^n n$$

Figure 8: Footnotes – convenient and simple

...

See also the section `\ref{example}`, which you will find on p. `\pageref{example}`.

```

\subsection{How to use labels}
\label{example}

```

...

The information pertaining to these cross-references is stored in an `.aux` file. The file is created when you run `latex`, and updated on successive runs. In order to position labels and references correctly, LaTeX needs to re-run and evaluate the commands; this is made evident by the output provided by the first `latex` run:

```

huhn@asteroid:~$ latex test.tex

```

...

LaTeX Warning: Reference `'example'` on page 1 undefined on input line 22.

...

LaTeX Warning: Label(s) may have changed. Rerun to get cross-references right.

Your cross-references are tidied up after re-running (see Figure 7).

It is just as convenient and easy to use footnotes in LaTeX documents. A single command, `\footnote{footertext}`, is all it takes to define and enumerate a footnote and automatically place it in small type at the end of a page in the `article` document class. The first line of each footnote is indented; the first footnote is separated from the body text by a short horizontal line.

If you use the `book` or `report` style, instead of `article`, enumeration will respect chapter numbers, that is the first footnote in a new `\chapter` will be number “1”. You can change the typeface or font in the footer text itself, or even use mathematical formulae (see Figure 8).

## INFO

- [1] Heike Jurzik: “Making up with LaTeX” – LaTeX-Workshop: Part 1, Linux Magazine, Issue 26, p46
- [2] Helmut Kopka, Patrick Daly: “A guide to LaTeX”, Addison-Wesley, ISBN 020142777X.