LaTeX Advanced Introduction

http://www.win.tue.nl/~marko/cursusLaTeX/introAdvanced

Marko Boon
marko@win.tue.nl

23 May, 2007

Introduction to LaTeX and MiKTeX

Topics:

- Introduction
- Text, Symbols and Commands
- Document Layout
- Displaying Text
- Mathematics
- Tables
- Graphics Inclusion

Slides available at:
http://www.win.tue.nl/~marko/cursusLaTeX/intro

Introduction to LaTeX Advanced

- WinEdt - working with a main file
- Installing additional packages
- LaTeX search path
- Scalable and non-scalable graphics
- Table of contents
- Page numbering
- Multicolumn text
- Footnotes
- Fonts
- Headers and footers
- Mathematics
- Floats
- Interactive PDF file
- Index
- Including programming statements
- Slide shows and posters

WinEdt - Working with a main file

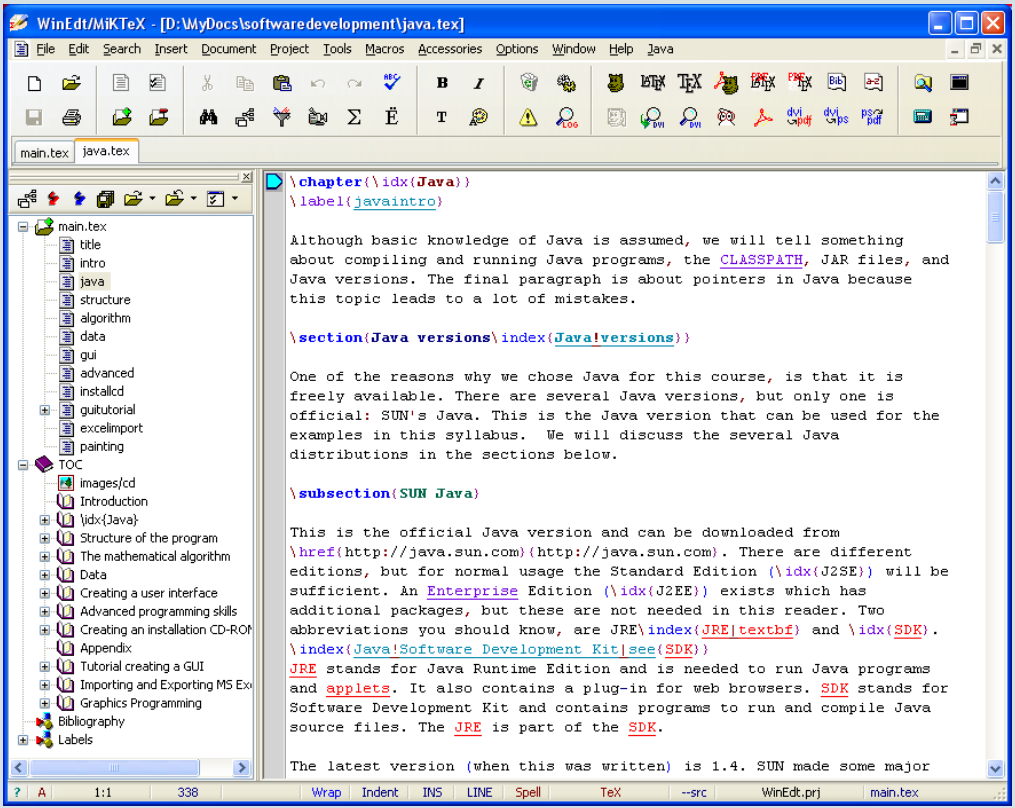
In bigger documents one often has one main file which includes several sub-documents: `\input chapter1` etc.. WinEdt has special functionality to deal with this document structure:


declares the current file as main file. This means that the programs `latex, yap, dvips, gsview, pdflatex` operate on this main file, even if another document is opened in WinEdt.

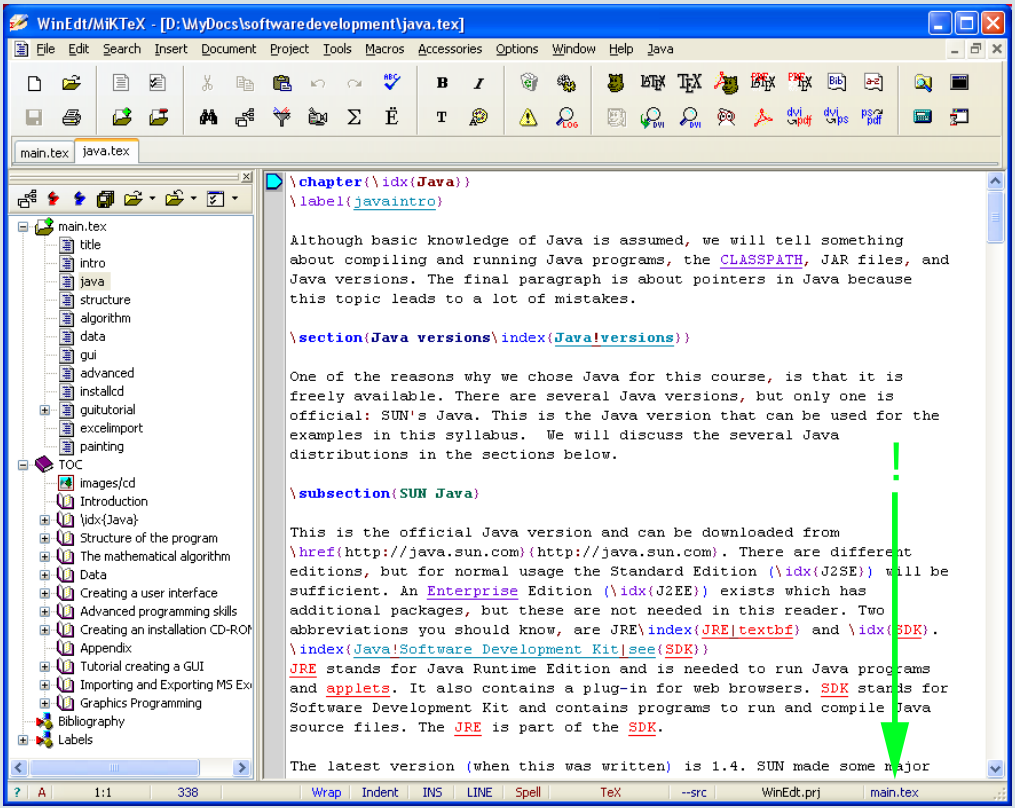

does not consider the current file as main file anymore. The file that is currently open will be LaTeXed.


displays the project tree (main file, input files, table of contents, bibliography and labels. You can click on files in this tree to open them.

WinEdt - Working with a main file



WinEdt - Working with a main file



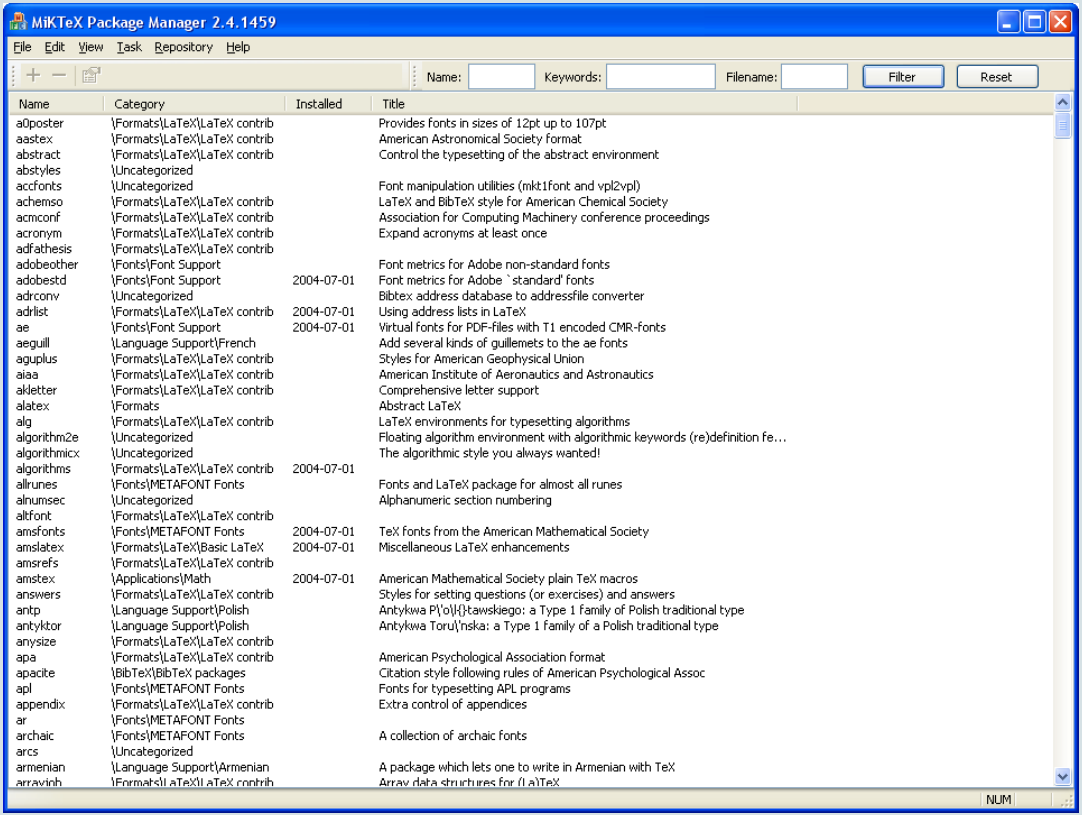
Installing additional packages

Two ways to install additional packages:

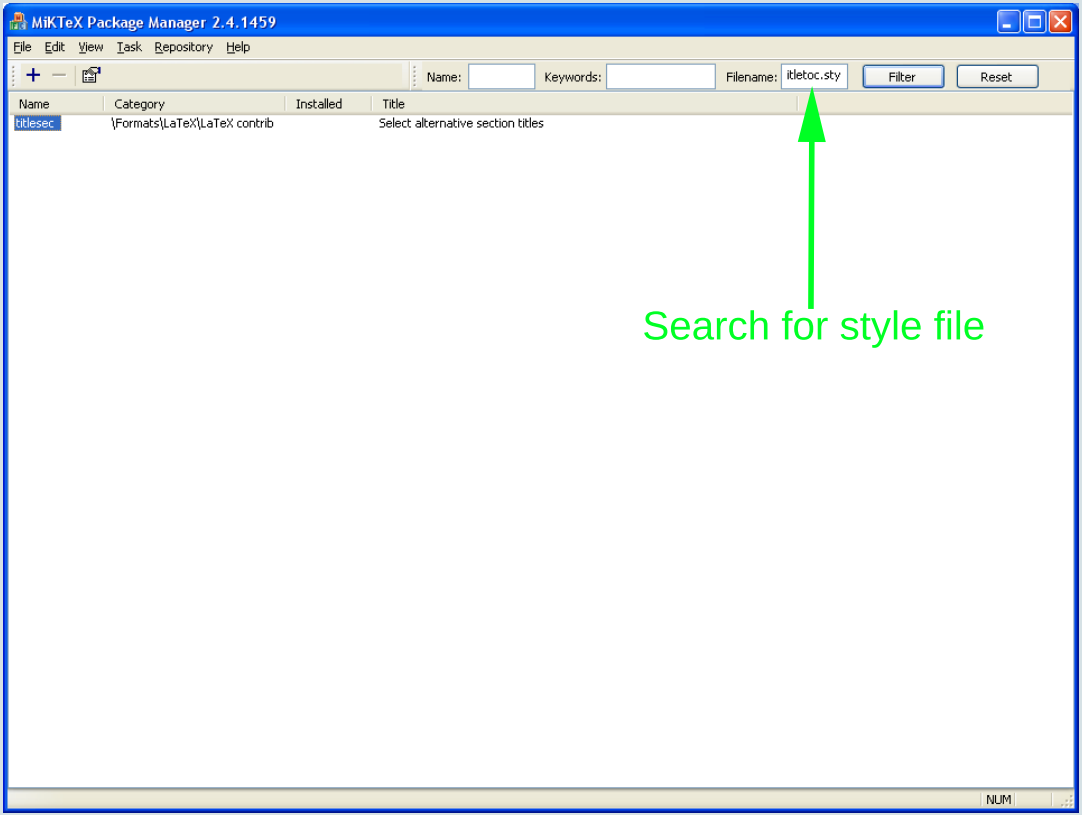
1. MiKTEX Package Manager
   (in the MiKTEX program group in the Start Menu)
2. automatic package installation

Installing additional packages requires Administrator rights.

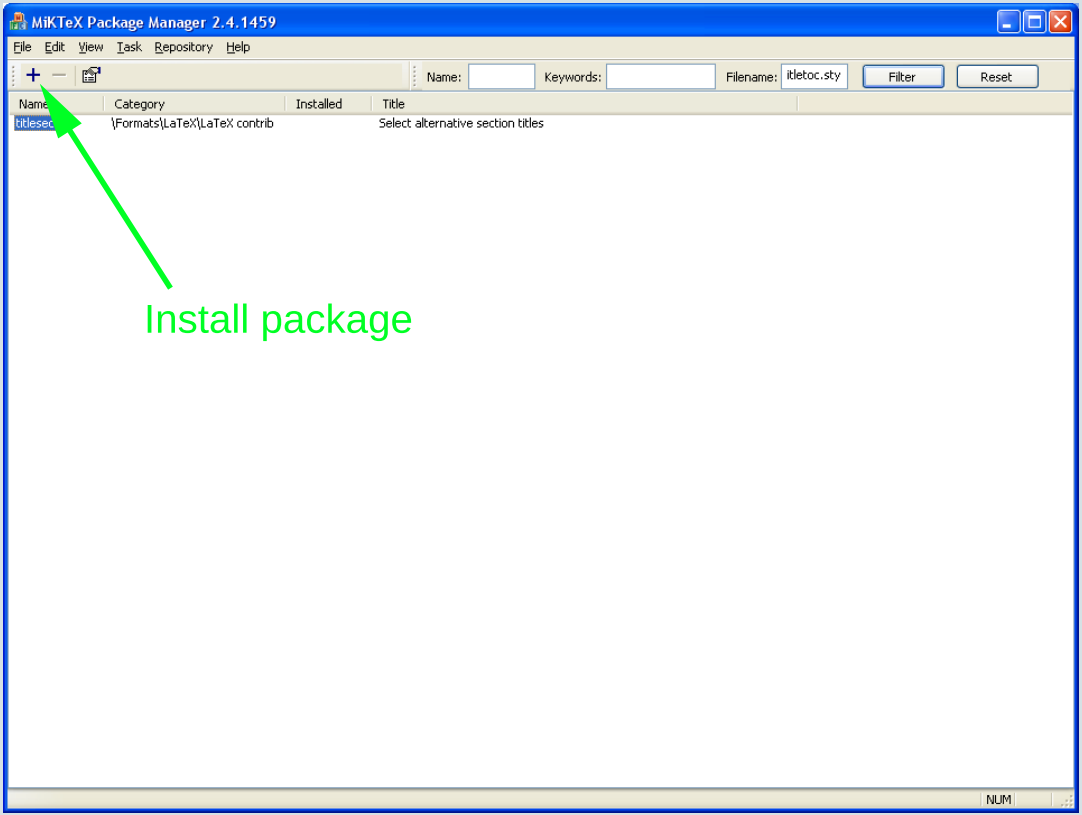
MiKTEX Package Manager



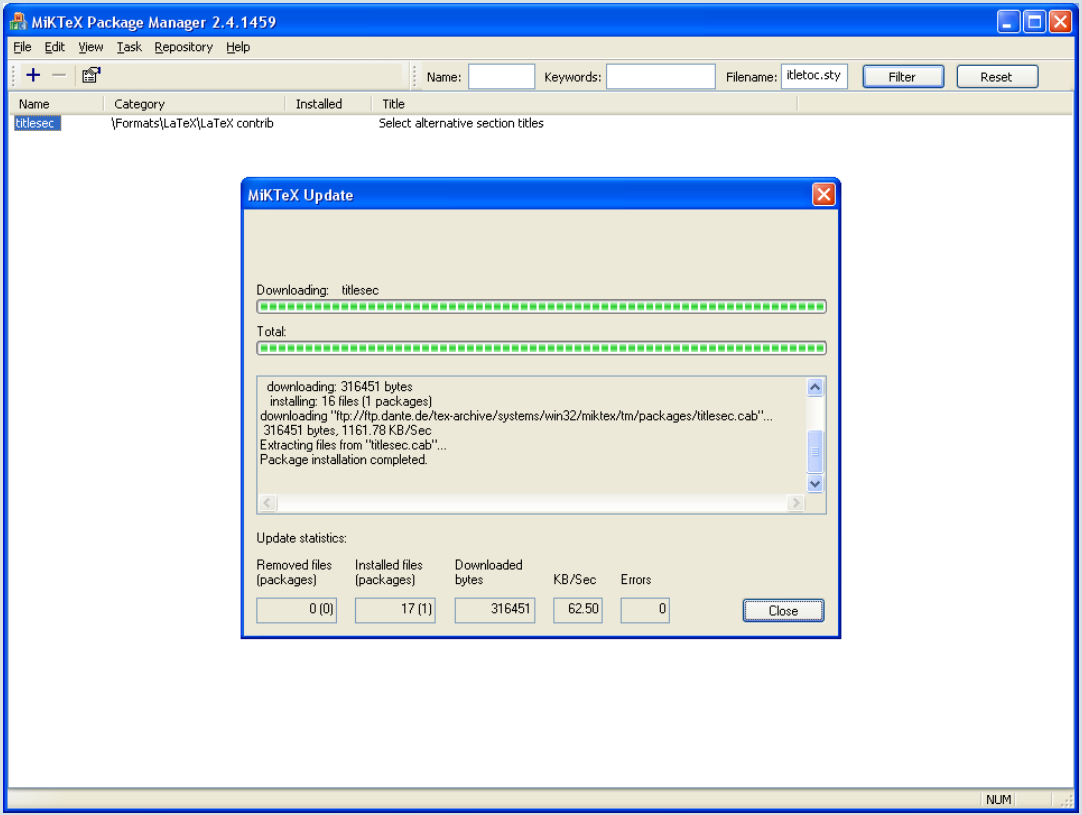
MiKTEX Package Manager



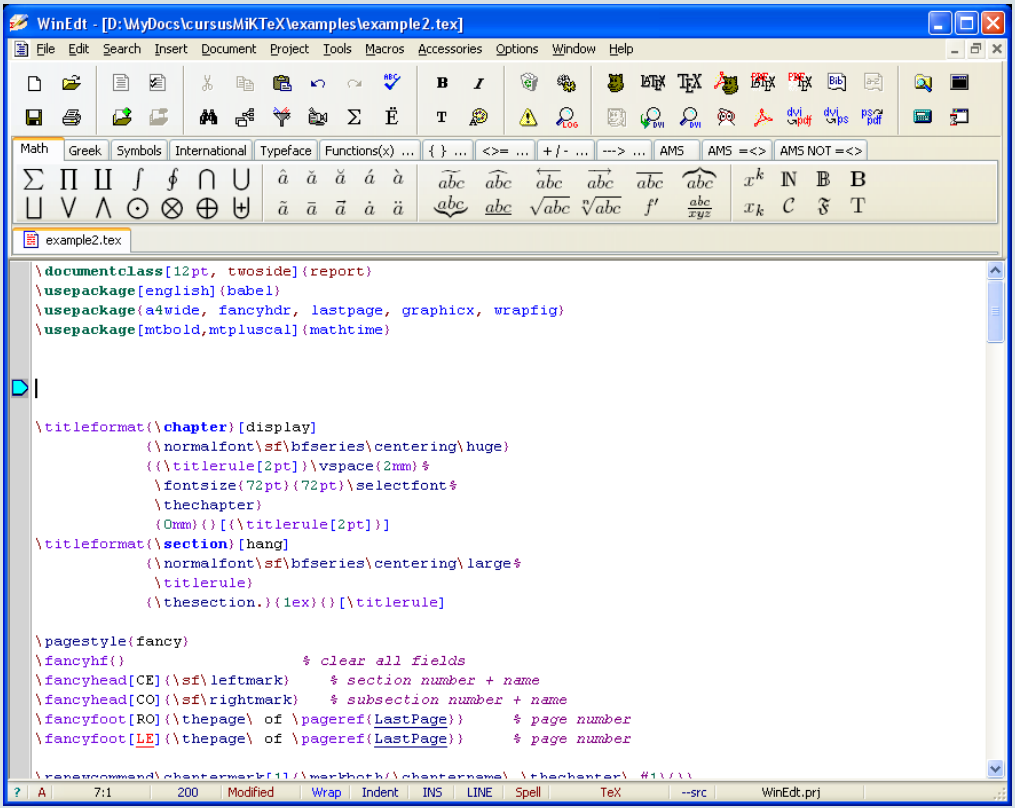
MiKTEX Package Manager



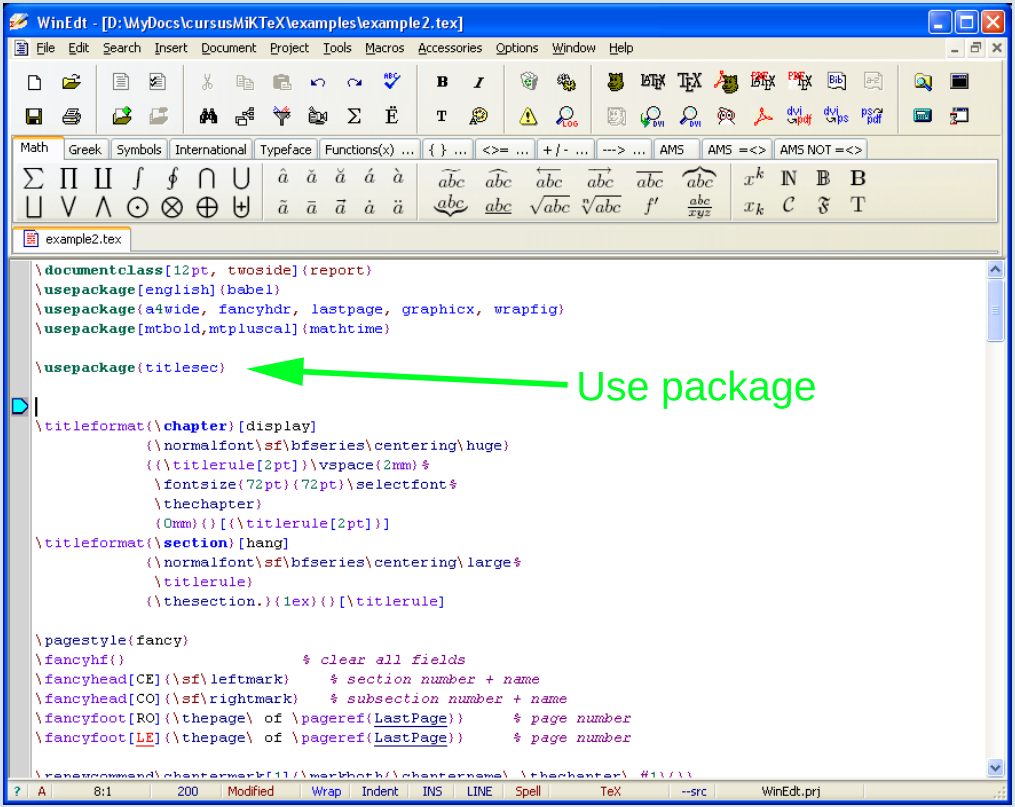
MiKTeX Package Manager



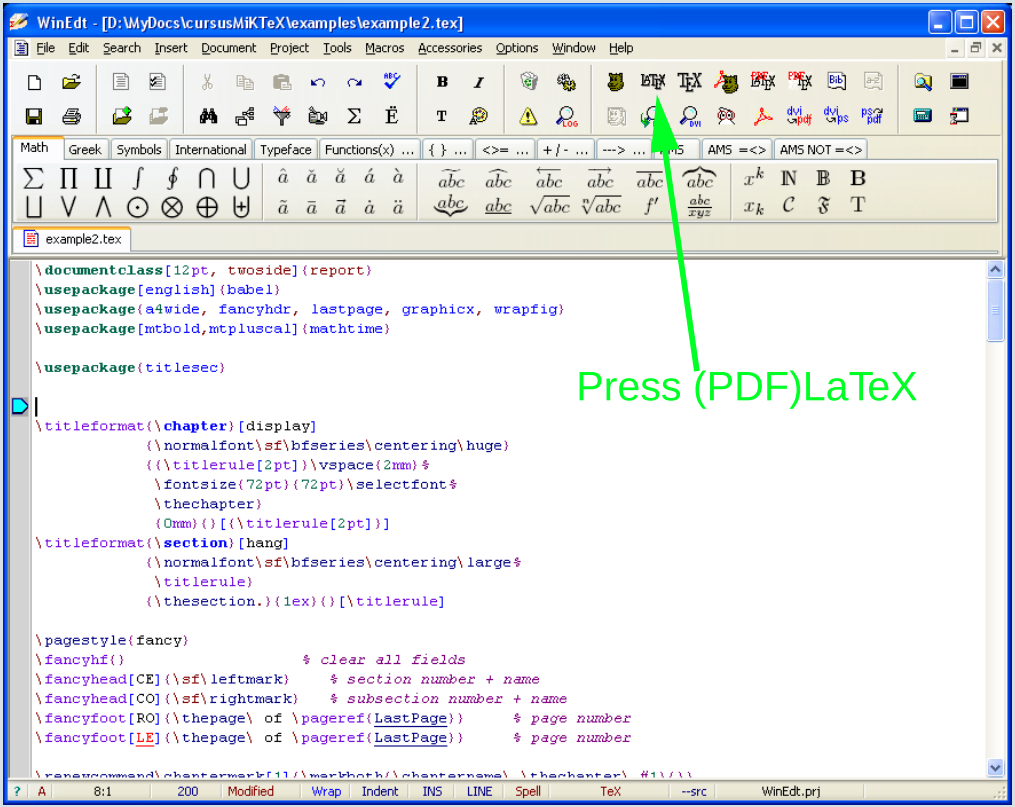
Automatic Package Installation



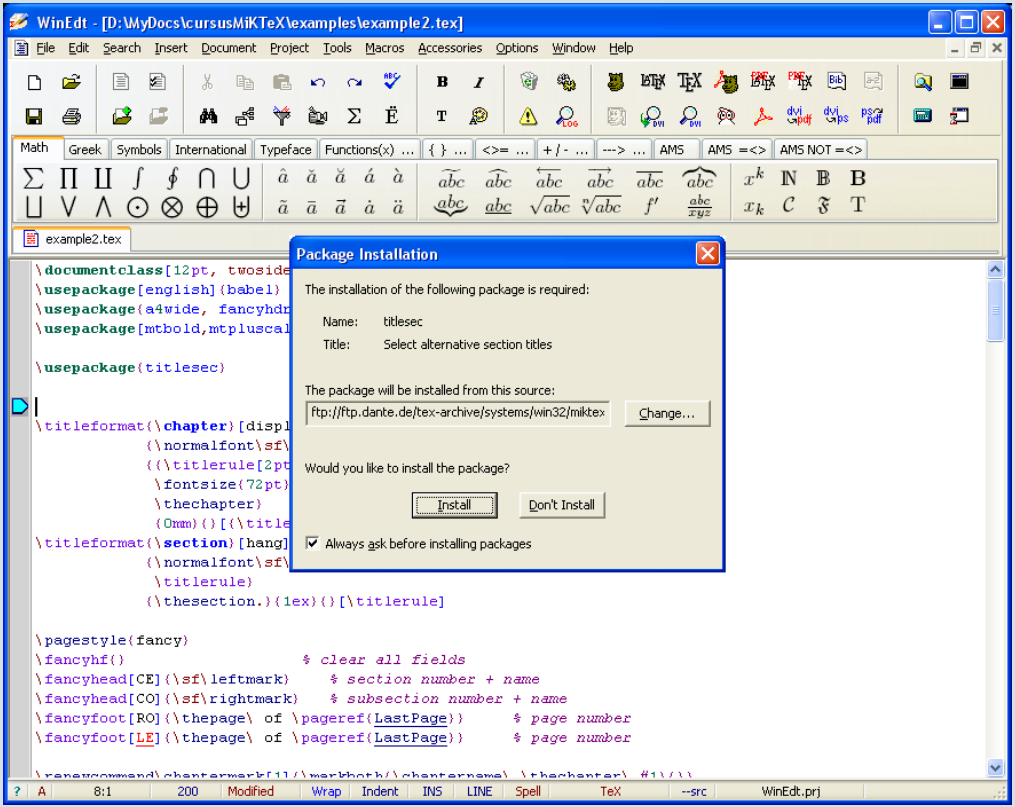
Automatic Package Installation



Automatic Package Installation



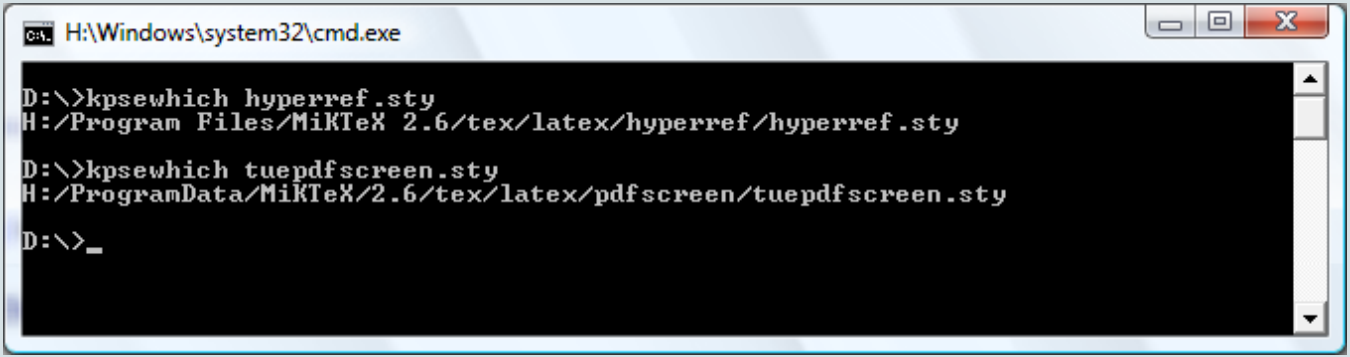
Automatic Package Installation



LaTeX search path

1. local TeXmf tree:
   Windows XP:
   `C:\Documents and Settings\All Users\`
   `Application Data\MiKTeX\2.6\`

   Windows Vista: `C:\ProgramData\MiKTeX\2.6\`
2. standard TeXmf tree: `C:\Program Files\MiKTeX 2.6\`

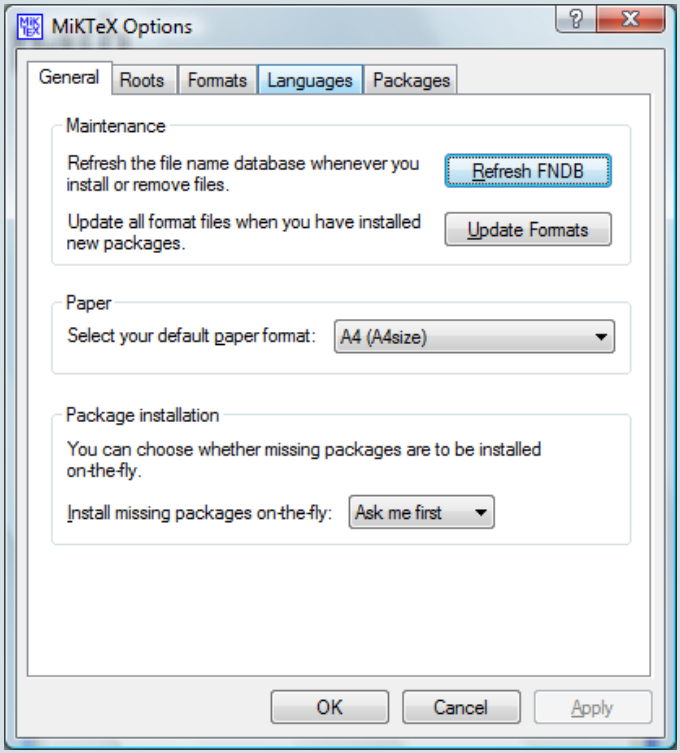


LaTeX search path

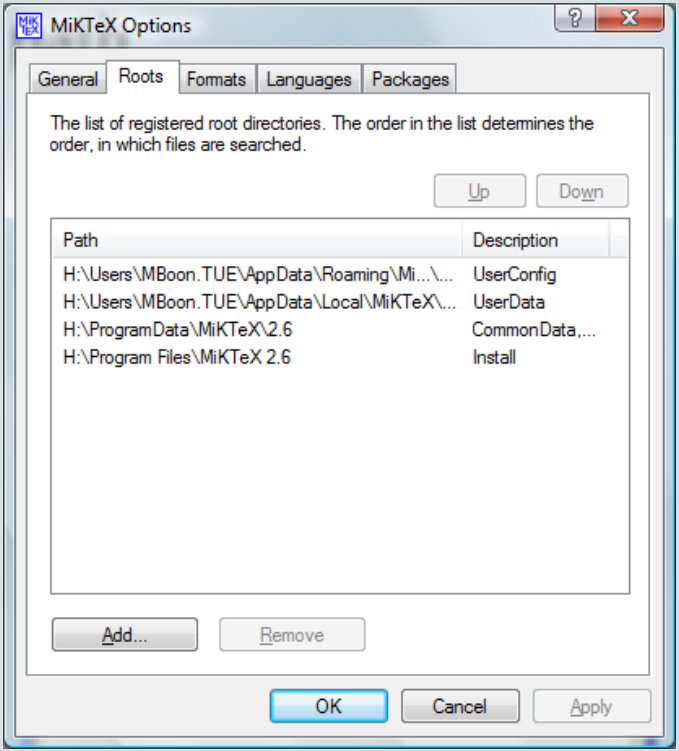
Adding your own TeXmf tree:

1. create a directory that contains the root of your TeXmf tree (e.g. `C:\My Documents\LaTeX`)
2. create a subdirectory **tex** and copy all your style files into this directory
3. optional: create a subdirectory **fonts** and copy all your font files into this directory
4. start MiKTeX Options ( button in WinEdt)
5. add your directory to the TeXmf Root Directories and move it to the top of the list

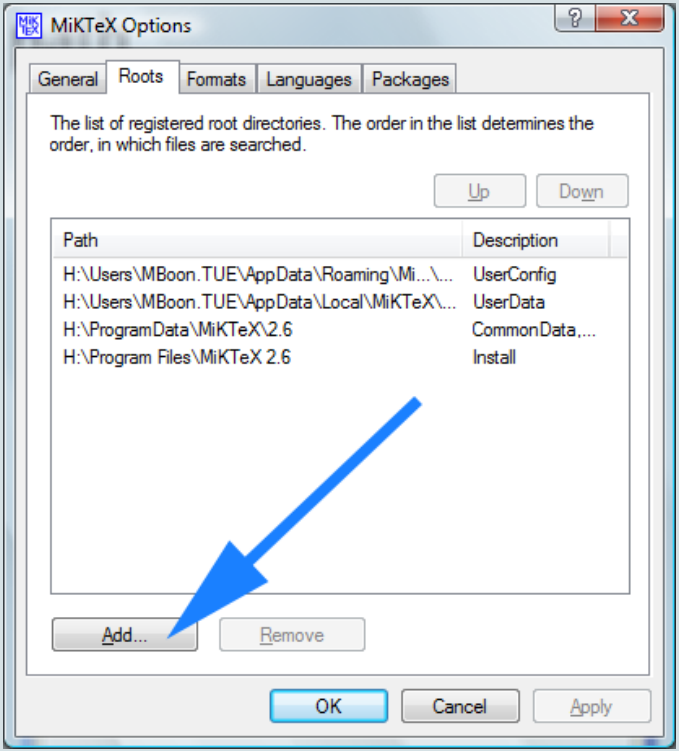
LaTeX search path



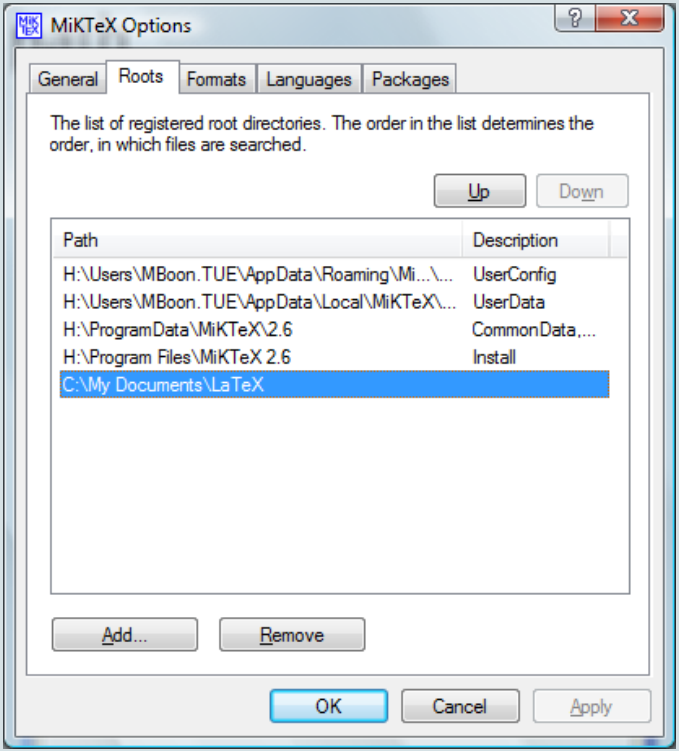
LaTeX search path



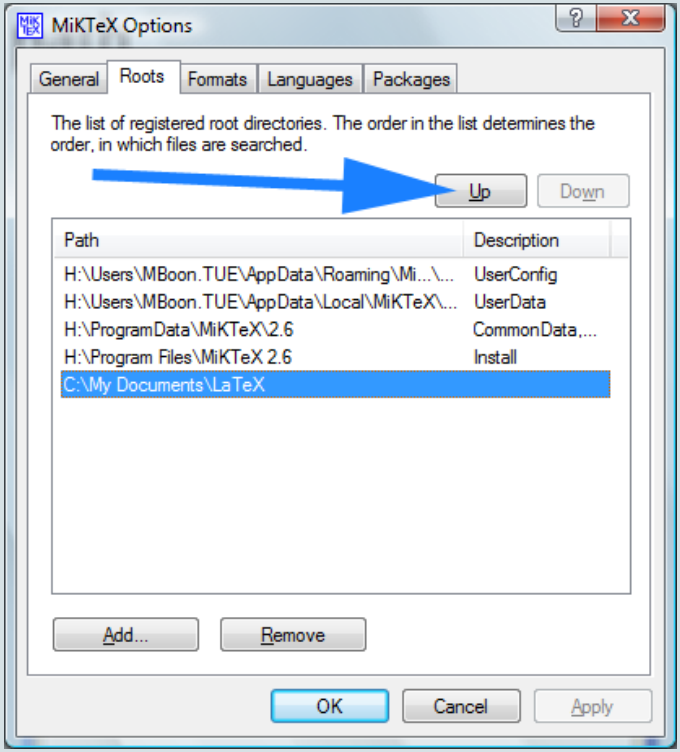
LaTeX search path



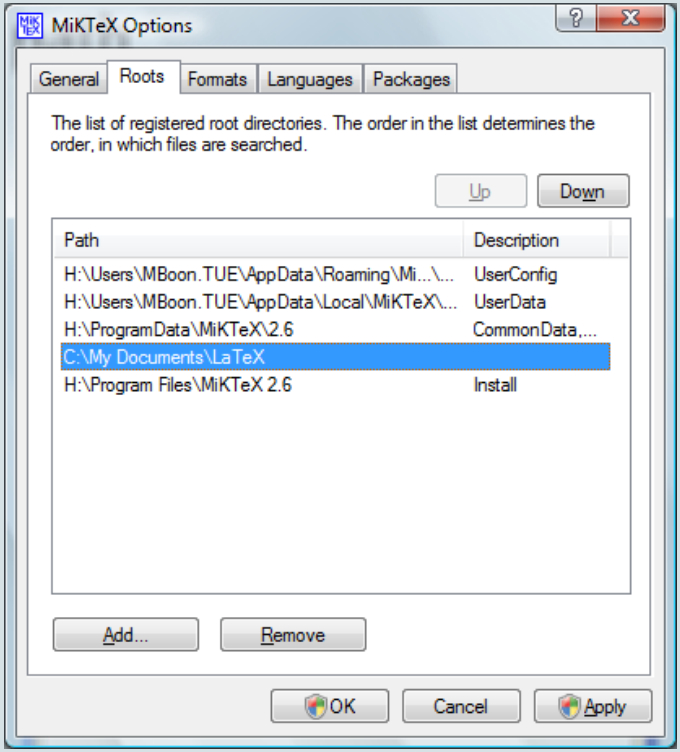
LaTeX search path



LaTeX search path

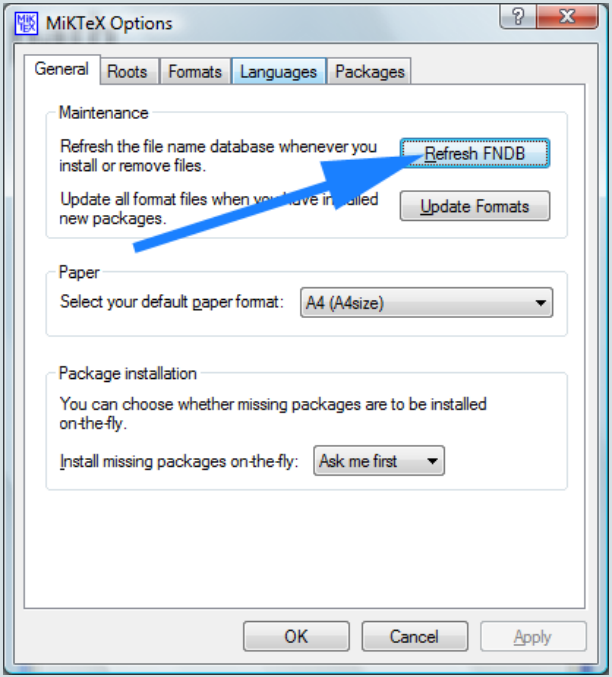


LaTeX search path

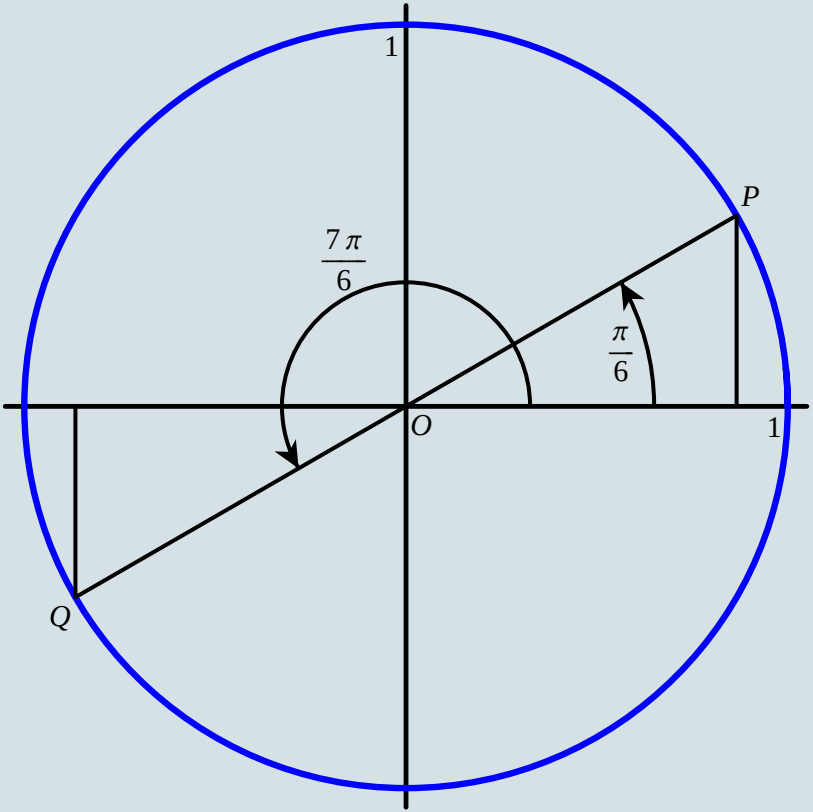


LaTeX search path

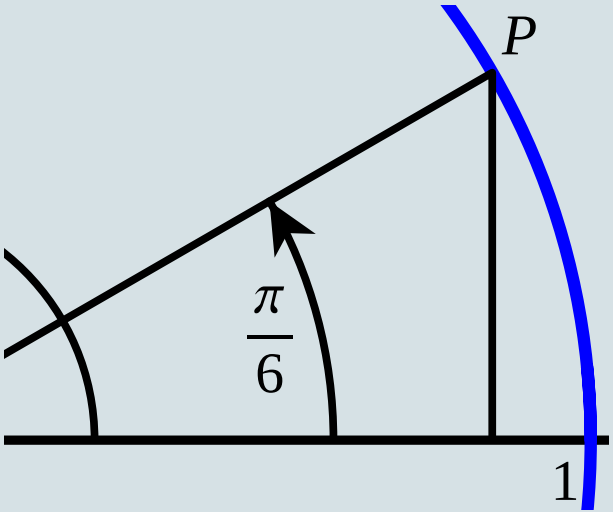
Each time you add a file to your TeXmf tree, you have to refresh the file name database:



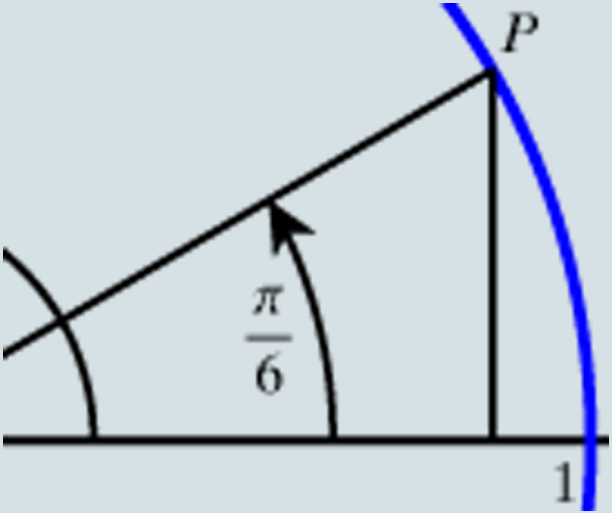
Scalable and non-scalable graphics



Scalable and non-scalable graphics



Scalable graphics formats:
EPS, PDF, WMF, EMF, SVG.



Non-scalable graphics formats:
JPG, GIF, BMP, PNG.
But also: all scalable formats!

Scalable Graphics Software

- Corel Designer,
- CorelDraw,
- Adobe Illustrator,
- Microsoft Visio,
- Microsoft Office Drawing,
- OpenOffice.org Draw,
- all computer algebra software (*Mathematica*, Matlab, Maple)

Non-scalable Graphics Software

- Adobe PhotoShop,
- Paint Shop Pro,
- MS Paint,
- all digital photo editing software!

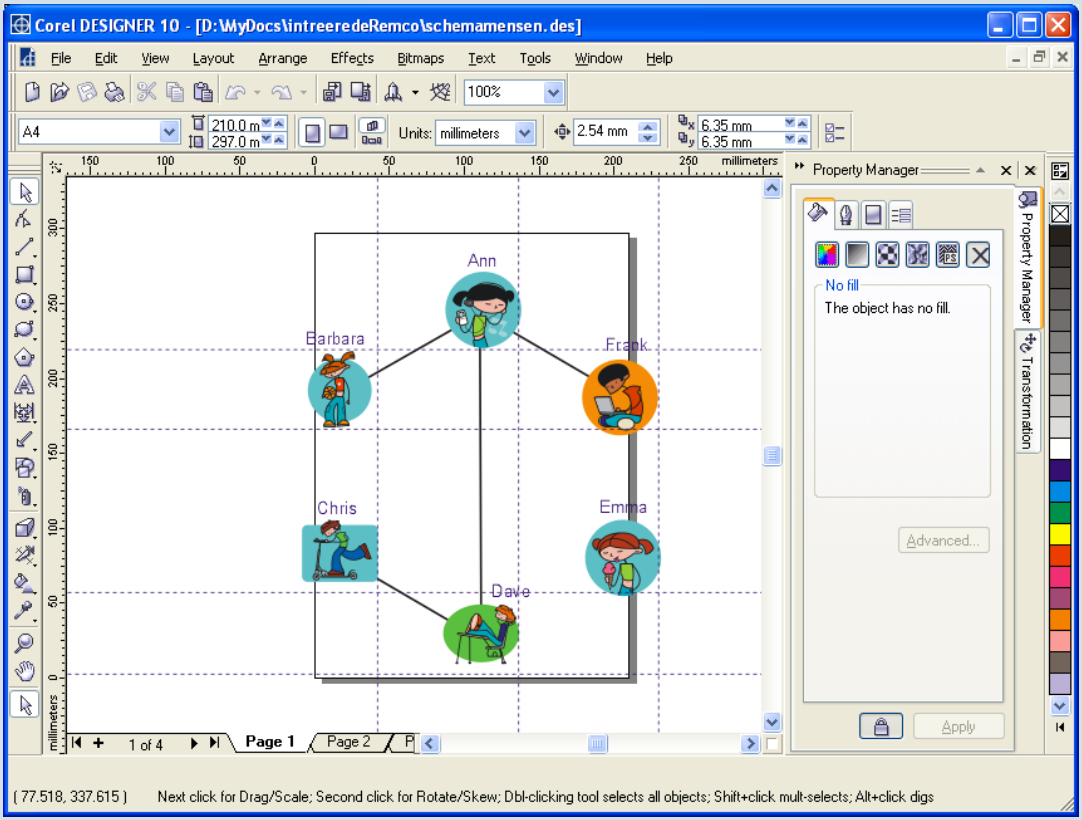
Including JPG/PNG Images

LaTeX (unlike PDFLaTeX) cannot determine the bounding box automatically.

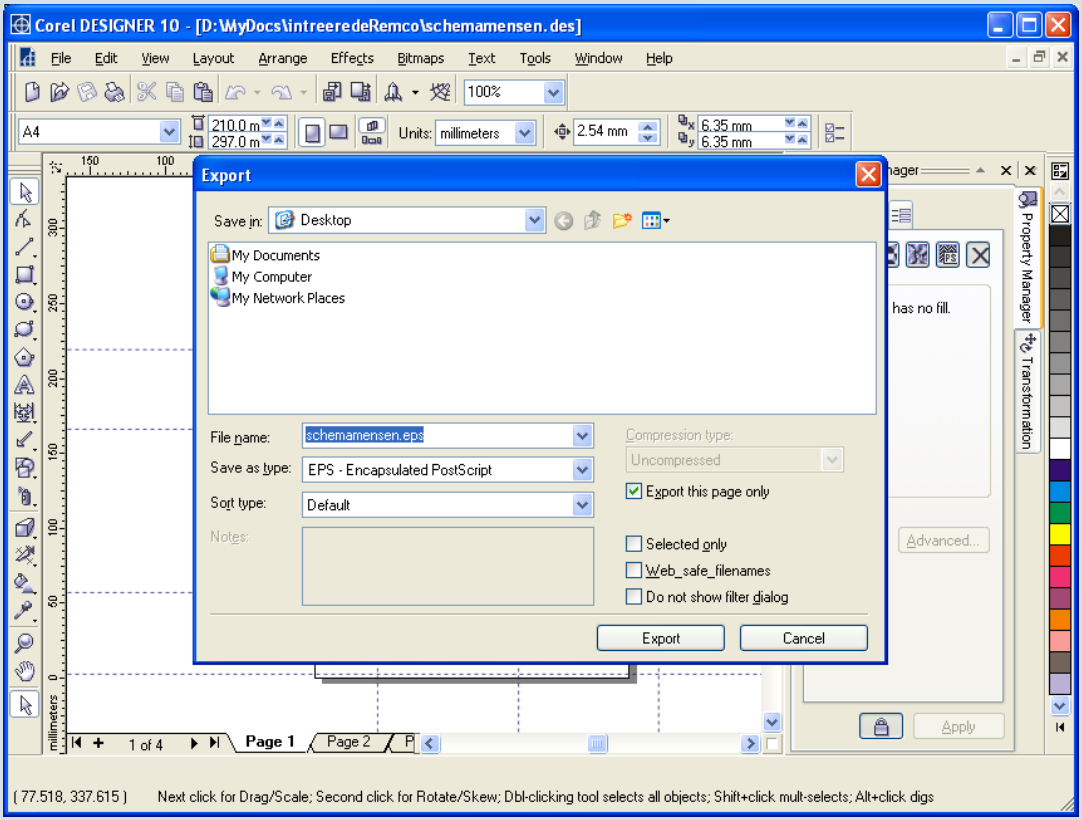
JPEG Image, 2304 × 1728 pixels, taken with 4.0 megapixel digital camera

```
\includegraphics[width=8cm,bb=0 0 2304 1728]
  {images/holiday.jpg}
```

Corel Designer



Corel Designer



Corel Designer

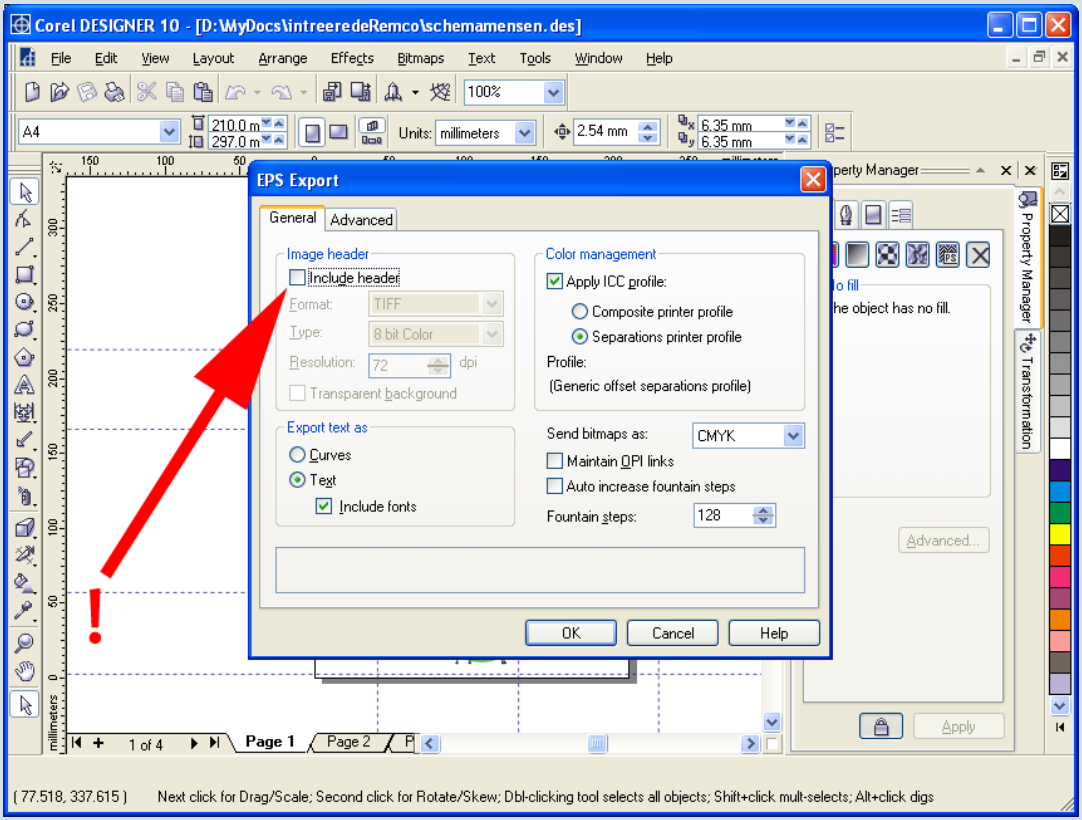


Table of Contents

The table of contents is generated and printed with the command `\tableofcontents` (normally after title page and abstract).
All entries are created automatically, based on the sectioning commands. You have to run `latex` twice to get all references right! It is recommended to use the `texify` command.

To create additional entries manually, use the command:
`\addcontentsline{toc}{section type}{entry text}`

```
\appendix
\addcontentsline{toc}{chapter}{\noindent Appendix}
\chapter{Statistical Tables}
```

Page Numbering

You can set the page number manually:

```
\setcounter{page}{14}
```

You can also control the style of the page numbering:

```
\pagenumbering{style}
```

The allowed styles are:

arabic	normal (Arabic) numerals: 1, 2, 3, 4
roman	lowercase Roman numerals: i, ii, iii, iv
Roman	uppercase Roman numerals: I, II, III, IV
alph	for lowercase letters: a, b, c, d
Alph	for uppercase letters: A, B, C, D

Book Structure

To simplify the structuring of the book, use the commands:

```
\frontmatter
\mainmatter
\backmatter
```

Front matter: preface, table of contents
Main matter: main body of text
Back matter: bibliography, index

Front matter has Roman page numbering and suppresses the numbering of chapters. Back matter also has unnumbered chapters. The page number is reset for the main matter.

Multicolumn Text

- The document class option **twocolumn** sets the entire document in two columns per page
- Individual pages may be output in one or two columns:

```
\onecolumn
\twocolumn[header text]
```

  Please note that these commands start a new page.

- To select a different number of columns within one page, use the **multicols** environment which is defined in the package **multicol**:

```
\usepackage{multicol}
...
\begin{multicols}{3}[header text]
Text set in 3 columns.
\end{multicols}
```

Footnotes

Footnotes are generated with the command `\footnote{text}`.
Example:

```
This section is about footnotes.\footnote{The
standard footnote marker is a small, raised number.}
```

This section is about footnotes.[1]

[1] The standard footnote marker is a small, raised number.

TU/e Fonts

In preamble:

```
\usepackage[T1]{fontenc}
```

To change the font defaults:

```
\renewcommand{\sfdefault}{zmb}
\renewcommand{\rmdefault}{zsc}
\renewcommand{\ttdefault}{pcr}
\fontfamily{\rmdefault}
\selectfont
```

TU/e Scala (zsc):
0123456789
ABCDEFGHIJKLMNOPQRSTUVWXYZ
abcdefghijklmnopqrstuvwxyz

TU/e Meta (zmb):
0123456789
ABCDEFGHIJKLMNOPQRSTUVWXYZ
abcdefghijklmnopqrstuvwxyz

TU/e logos (zlo):

TU/e technische universiteit eindhoven

Fonts

- Mathtime

```
\usepackage[T1]{fontenc}
\usepackage{mathtime}
```

This is an *italic* or **bold** test

- Helvetica (looks like Arial)

```
\renewcommand{\sfdefault}{phv}
\renewcommand{\rmdefault}{phv}
```

This is an *italic* or **bold** test

Non-standard headers and footers

The package **fancyhdr** (previously known as **fancyheadings**) defines commands that let you control headers and footers:

```
\lhead{...}     \chead{...}     \rhead{...}

\lfoot{...}     \cfoot{...}     \rfoot{...}
```

If you want to distinguish between odd and even pages, it becomes slightly more complicated:

```
\fancyhead[RO, LE]{...}

\fancyfoot[C]{...}
```

L = Left, C = left, R = right, O = odd, E = even.

You have to specify that pagestyle should be fancy (instead of plain or empty).

fancyhdr - Example

```
\pagestyle{fancy}
\fancyhf{}
\fancyhead[CE]{\sffamily\leftmark}
\fancyhead[CO]{\sffamily\rightmark}
\fancyfoot[RO]{\thepage\ of \pageref{LastPage}}
\fancyfoot[LE]{\thepage\ of \pageref{LastPage}}

\renewcommand\chaptermark[1]{%
  \markboth{\chaptername\ \thechapter\ #1}{}}
\renewcommand\sectionmark[1]{%
  \markright{\thesection\ #1}}
\renewcommand\headrulewidth{0.4pt}
```

Please note that the LastPage reference is only available if you load the package lastpage. You have to LaTeX your document twice before it works.

Exercise 1

1. Create a directory `LaTeX` in your Documents directory. Add this directory to your LaTeX search path. Make a subdirectory `tex`.
2. Download `snowwhite.tex` and `snowwhite.jpg`. Put the JPG file in the newly created TeXmf tree and Refresh the Filename Database.
3. Include the image `snowwhite.jpg` on the title page. Make sure that your document runs with `latex` and `pdflatex`!
4. Change the page numbering to uppercase Roman numbering.
5. find the second line in chapter 4.2 (They were seven dwarfs ...) and create a footnote at this place. It should say: Walt Disney was the first one to give names to the dwarfs: Dopey, Grumpy, Doc, Happy, Bashful, Sneezy and Sleepy.
6. Change the default fonts of the document to TU/e fonts.
7. Create headers and footers just like in the **fancyhdr** example.

Mathematical formulas

In a text:

> For a rectangular triangle, we know from *Pythagoras' theorem* that $a^2 + b^2 = c^2$ where a and b are the length of two sides adjoining the straight angle while c is the length of the side opposite the straight angle.

Compare this with:

> For a rectangular triangle, we know from *Pythagoras' theorem* that $a^2+b^2=c^2$ where a and b are the length of two sides adjoining the straight angle while c is the length of the side opposite the straight angle.

Mathematical formulas are created as follows:

```
We get: $a^2+b^2=c^2$, $a^{13}$, $b_3$ or $b_13$
```

results in

We get: $a^2 + b^2 = c^2$, a^{13}, b_3 or b_13

Mathematical formulas are created as follows:

```
We get
\[
  a^2+b^2=c^2, a^{13}, b_3 \mbox{ or } b_13
\]
```

results in

We get

$$a^2 + b^2 = c^2, a^{13}, b_3 \text{ or } b_13$$

We can also number our equations:

```
We get
\begin{equation} \label{one}
  a^2+b^2=c^2, a^{13}, b_3 \mbox{ or } b_13
\end{equation}
```

results in

We get

$$a^2 + b^2 = c^2, a^{13}, b_3 \text{ or } b_13 \tag{1}$$

We can also have multiple equations:

```
\begin{eqnarray}
  x & = & r\sin \varphi \label{11} \\
  y & = & r\cos \varphi \nonumber \\
  z & = & z \label{33}
\end{eqnarray}
```

$$x = r \sin \varphi \tag{2}$$
$$y = r \cos \varphi$$
$$z = z \tag{3}$$

or without numbers:

```
\begin{eqnarray*}
  x & = & r\sin \varphi   \\[-0.2cm]
  y & = & r\cos \varphi   \\
  z & = & z
\end{eqnarray*}
```

$$
\begin{aligned}
x &= r \sin \varphi \\
y &= r \cos \varphi \\
z &= z
\end{aligned}
$$

Obviously we can do more:

```
$\frac{n}{n+p^2} \int_0^\infty
\sqrt[n]{x^n-\sin y} \textrm{d}x$
```

$$\frac{n}{n+p^2} \int_0^\infty \sqrt[n]{x^n - \sin y}\mathrm{d}x$$

On the other hand:

```
\[
\frac{n}{n+p^2} \int_0^\infty
\sqrt[n]{x^n-\sin y}\, \textrm{d}x
\]
```

$$\frac{n}{n+p^2}\int_0^\infty \sqrt[n]{x^n-\sin y}\,\mathrm{d}x$$

and finally:

```
$\displaystyle \frac{n}{n+p^2} \int_0^\infty
\sqrt[n]{x^n-\sin y}\; \textrm{d}x$
```

$$\displaystyle \frac{n}{n+p^2} \int_0^\infty \sqrt[n]{x^n-\sin y}\; \textrm{d}x$$

Functions

```
$\sin x,\; sin x, \;  \mbox{sin} x$
```

$\sin x,\ sinx,\ \mathrm{sin}x$

Brackets

```
$\displaystyle (\frac{n}{\frac{n}{n+p}+1})
    + \left( \frac{n}{\tfrac{n}{n+p}+1} \right)$
```

$$(\frac{n}{\frac{n}{n+p}+1}) + \left(\frac{n}{\frac{n}{n+p}+1} \right)$$

Inline floats

The package **wrapfig** makes it possible to place text next to floats:

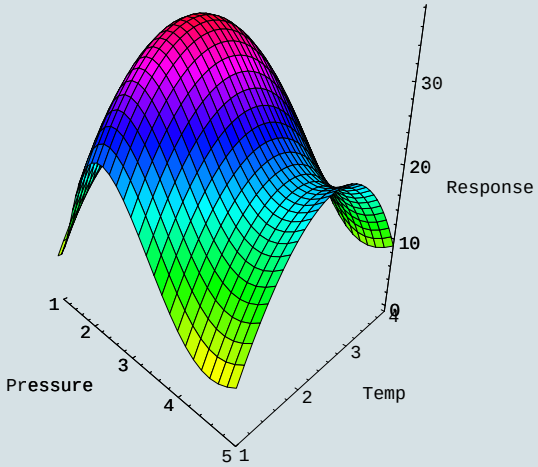
```
\begin{wrapfigure}{placement}[overhang]{width}
\includegraphics[width=\linewidth]{image}
\end{wrapfigure}
```

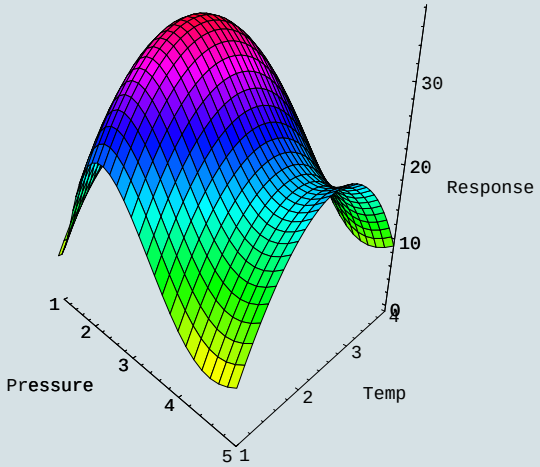
placement	horizontal placement: l (left) or r (right). For two-sided documents: i (inside edge) or o (outside edge).
overhang	overhang of the float into the margin (default: opt).
width	width of the figure or table (use `wraptable` for tables).

Two floats next to each other

1. No caption

```
\includegraphics[width=0.45\linewidth]{images/pic1}
\hfill
\includegraphics[width=0.45\linewidth]{images/pic2}
```





Two floats next to each other

2. One caption

```
\begin{figure}[ht]
\includegraphics[width=0.45\linewidth]{images/pic1}
\hfill
\includegraphics[width=0.45\linewidth]{images/pic2}
\caption{a response surface.}
\label{fig:surface}
\end{figure}
```

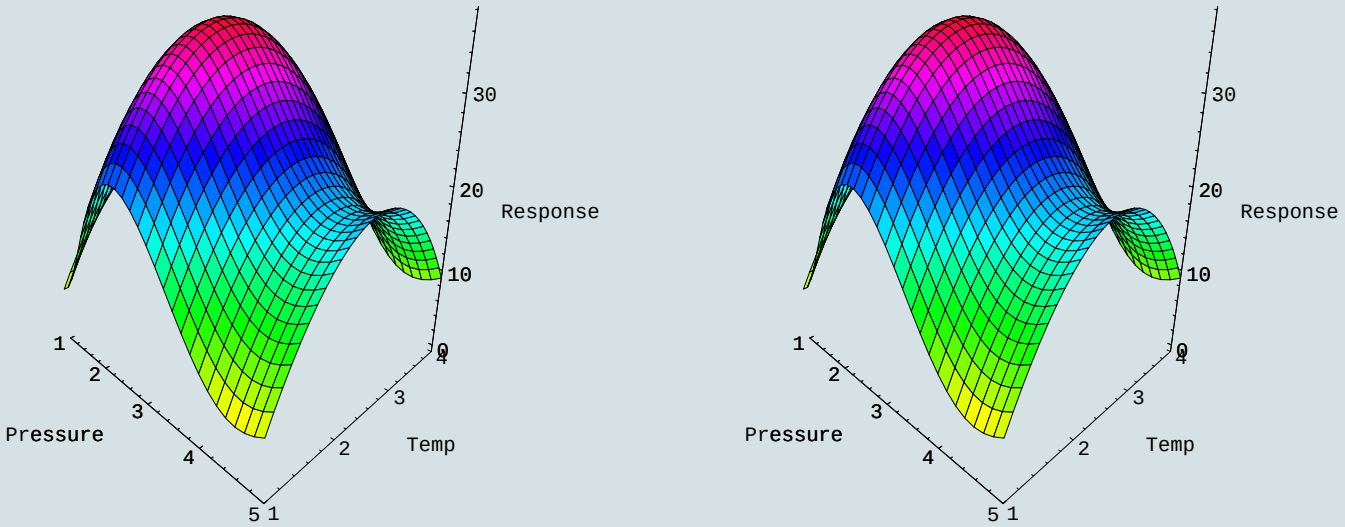


Figure 1: a response surface.

Two floats next to each other

3. Two captions

Now we need the package **caption**. This package has very extensive functionality to change the appearance of captions. In this case we are only going to use the new command `\captionof`.

```
\parbox[t]{0.45\textwidth}{
  \includegraphics[width=\linewidth]{pic1}
  \captionof{figure}{the first figure}
}
\hfill
\parbox[t]{0.45\textwidth}{
  \includegraphics[width=\linewidth]{pic2}
  \captionof{figure}{the second figure}
}
```

For tables, just repace **figure** by **table**.

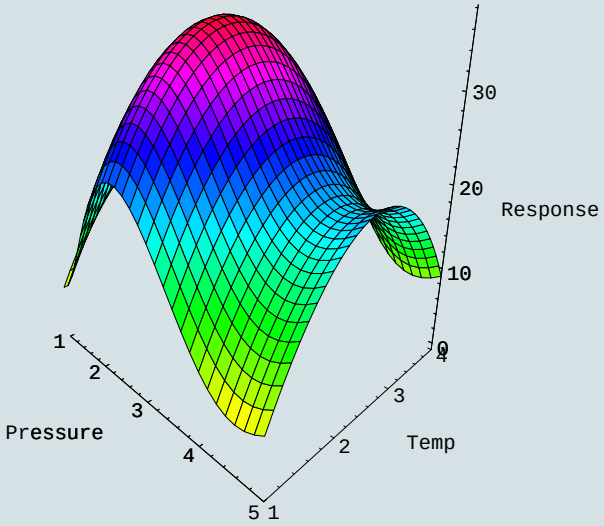


Figure 2: the first figure

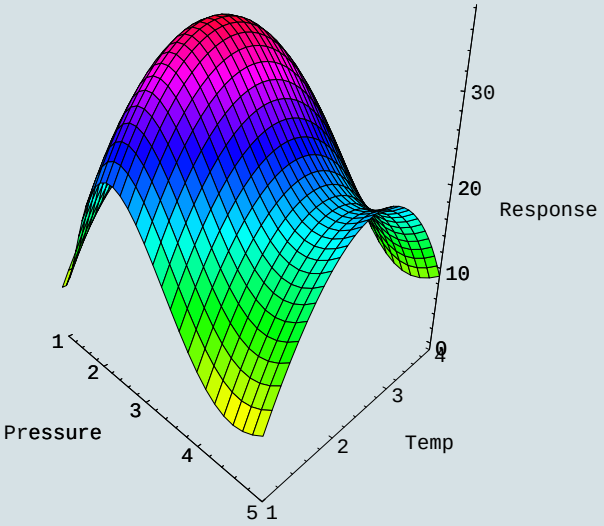


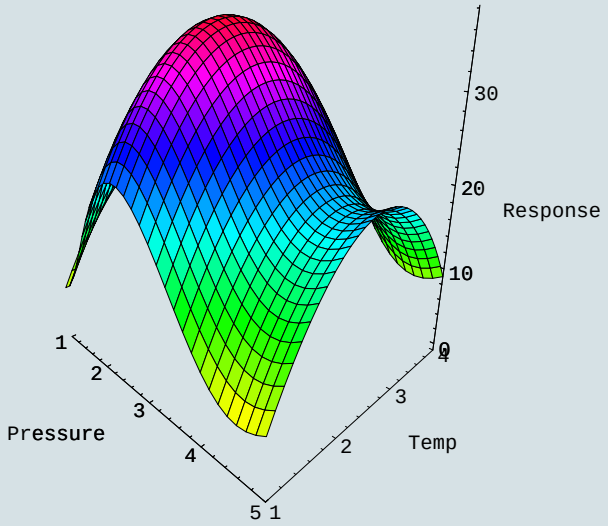
Figure 3: the second figure

Two floats next to each other

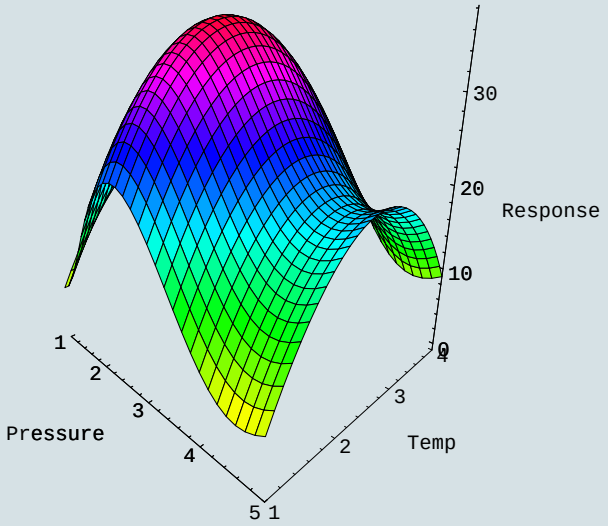
4. Sub-captions

Now we need the package **subfig**:

```
\begin{figure}[ht]
\begin{center}
\subfloat[First figure]{
  \includegraphics[width=0.45\textwidth]{pic1}
}
\subfloat[Second figure]{
  \includegraphics[width=0.45\textwidth]{pic2}
}
\caption{Two figures}
\end{center}
\end{figure}
```



(a) First figure



(b) Second figure

Figure 4: Two figures

Create an interactive PDF file

If you load the package **hyperref**, the DVI and PDF file created from your LaTeX document will be interactive. Added interactivity is:

- all internal references will be clickable (including the table of contents).
- you can create external links (e.g. to web pages).
- you can create a list of bookmarks in your PDF document.

Usage is simple: just add the line

```
\usepackage{hyperref}
```

to your document (always load this package after all the other packages!).

A useful new command, similar to the HTML `<a href="..."> </a>` command is:

```
\href{URL}{description}
```

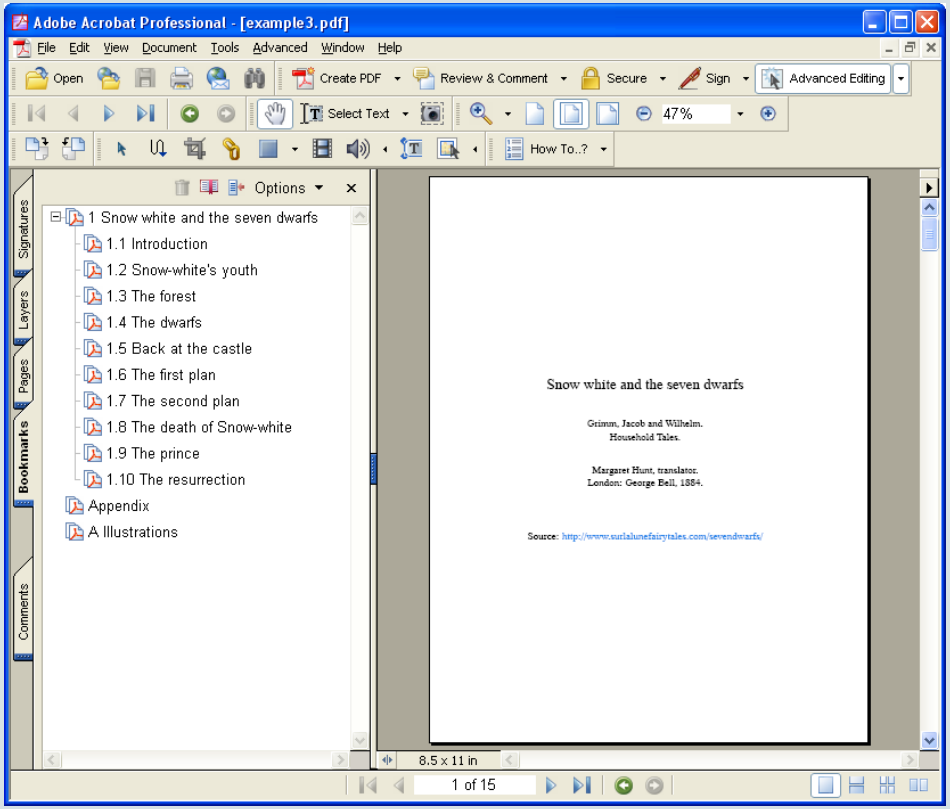
Customizing hyperref

You can customize the appearance of hyperlinks in your document using the command \hypersetup:

```
\definecolor{mycolour}{rgb}{0.19,0.54,0.92}
\hypersetup{
    colorlinks=true,
    linkcolor=mycolour, urlcolor=mycolour,
    pdfpagemode=UseOutlines,
    bookmarksopen=true,
    bookmarksnumbered=true,
    plainpages=false, pdfpagelabels,
    pdftitle={Snow white and the seven dwarfs},
    pdfauthor={Grimm, Jacob and Wilhelm.}
}
```

You have to include the package color to use \definecolor. You can use WinEdt's colour-picker to find RGB values of a selected colour.

hyperref - Example



Distinguishing between LaTeX and PDFLaTeX

The command \hypersetup also affects the DVI and PostScript file. Usually you want the printed version of your document to be black and not interactive, while the PDF version supports interactivity and has coloured links. The package ifpdf defines a command \ifpdf that can be used to distinguish between LaTeX and PDFLaTeX.

```
\ifpdf
\hypersetup{
    colorlinks=true,
    linkcolor=mycolour,
    urlcolor=mycolour
}
\else
\hypersetup{
    colorlinks=false
}
\fi
```

Creating a master index

To create a master index, just follow these steps:

1. include the package **makeidx**.
2. add the command `\makeindex` **before** `\begin{document}`
3. add words to the index with the command `\index{word}`. Please note that this command does not display the word. It might be useful to define a command:

```
\newcommand{\idx}[1]{#1\index{#1}}
```

4. put the following commands at the location where you want the index:

```
\newpage\cleardoublepage
\printindex
```

   This makes sure that the index will start on an odd page.

5. run LaTeX twice, then run `makeindex`  and run LaTeX again.

Special index formats

Use this to point to another word:

```
\index{looking-glass|see{mirror}}
```

Use this to make the page number bold:

```
\index{forest|textbf}
```

Use this to make sub categories:

```
\index{plan!first}
\index{plan!second}
```

Use this to span multiple pages:

```
\index{Snow white|(}
\index{Snow white|)}
```

Including programming statements

The package **listings** formats listings. It defines the following commands:

- `\lstlisting{...}` for inline programming statements.
- `\begin{lstlisting} ... \end{lstlisting}` for multi-line listings.
- `\lstinputlisting{filename}` imports a complete source file

Customizing listings

Using the command \lstset you can customize the language and appearance of the listing:

```
\lstset{
    language=Java,
    basicstyle=\color{black}\ttfamily,
    commentstyle=\color{green}\itshape\ttfamily,
    keywordstyle=\color{blue}\bfseries\ttfamily,
    showstringspaces=false,
    frame=single, % boxed listings
    backgroundcolor=\color{white}
}
```

Supported languages: too many to mention. Included are Basic, C, C++, Delphi, Fortran, HTML, Java, Mathematica, Matlab, Pascal, Perl, PHP, SAS, SQL, TeX, VBScript, XML.

Customizing listings

Alternatively, you can specify options like this:

```
\definecolor{myyellow}{rgb}{1.00,1.00,0.50}
\begin{lstlisting}[language=Pascal,
                   backgroundcolor=\color{myyellow}]
  readln(N);
  for i := 1 to N do
  begin
    writeln(random)
  end
\end{lstlisting}
```

```
  readln(N);
  for i := 1 to N do
  begin
    writeln(random)
  end
```

Create a PDF slide show or poster

The package **pdfscreen** was written for PDF slide show presentations. Unfortunately this package contained some bugs, so another package was written: **tuepdfscreen**. This package can be used to create PDF slide shows. The default appearance is in the TU/e style (colours, fonts) but this can be modified. In fact, any Powerpoint style can be converted to PDF which makes it suitable for TU/ePDFScreen.

Detailed information about TU/ePDFScreen can be found in the file **slides.tex** which is located in your local TEXmf tree, in the sub directory `examples/presentatie`.

This file contains information about all features, like navigation buttons, page numbering, page transitions, step-by-step appearance of lists, including pictures and movies.

This style can also be used for posters. See: **poster.tex**.

Converting your LaTeX document to a slide show

1. make sure that your document runs with (LaTeX and) PDFLaTeX.
2. include the line

```
\usepackage[eleuk]{tuepdfscreen}
```

3. use the **slide** or **slidetop** environments to divide your text into different slides.
4. if want pages to appear in multiple steps, use the **\pause** command to define breaks.
5. run PDFLaTeX on the file.
6. if you used the **\pause** command, run the program **AddPause** that can be found in the MiKTeX Start Menu program group.

Converting your LaTeX document to a slide show

```
\documentclass[a4paper]{article}


\begin{document}

\section*{Mathematics}

\begin{eqnarray*}
\lim_{x \rightarrow 0} \frac{\sin x}{x} &=& 1\\
\pause
\sum_{k=0}^\infty x^k &=& \frac{1}{1-x} (|x| < 1)
\end{eqnarray*}

\end{document}
```

Converting your LaTeX document to a slide show

```
\documentclass[a4paper]{article}
\usepackage[eleuk]{tuepdfscreen}

\begin{document}
\begin{slidetop}
\section*{Mathematics}

\begin{eqnarray*}
\lim_{x \rightarrow 0} \frac{\sin x}{x} &=& 1\\
\pause
\sum_{k=0}^\infty x^k &=& \frac{1}{1-x} (|x| < 1)
\end{eqnarray*}
\end{slidetop}%
\end{document}
```

Mathematics

$$\lim_{x \to 0} \frac{\sin x}{x} = 1$$

$$\sum_{k=0}^{\infty} x^k = \frac{1}{1-x} (|x| < 1)$$

Exercise 2

Load the package hyperref with the following options:
LaTeX only:

```
colorlinks=false
```

PDFLaTeX only:

```
pdfpagemode=UseOutlines,
bookmarksopen=false,
bookmarksnumbered=true,
pdftitle={Snow white and the seven dwarfs},
pdfauthor={Grimm, Jacob and Wilhelm.},
colorlinks=true,
linkcolor=tuedarkblue,
urlcolor=tuedarkblue,
plainpages=false,
pdfpagelabels
```