

# Manual for Using the `cnbwp` Class to Write CNB Working Papers

Zdeněk Wagner

version 2013.12

## Abstract

This manual explains how to write Czech National Bank Working Papers using the  $\LaTeX$  typesetting system. The opening sections describe the `cnbwp` class created for this purpose. A separate section tells you how to prepare a list of references. The subsequent sections give more generally applicable advice for creating and inserting figures and tables. Section 11 tells you the form in which the document has to be submitted and how the document is edited. The names of the macros, packages, files and programs mentioned in this manual are listed in the index. The manual also includes sample files, which are listed in Section 12. The class, the manual and the sample files were prepared for the Czech National Bank. Martin Cincibuch from the CNB contributed significantly to the structure and content of this document.

## Contents

1	Typographical Conventions . . . . .	2
2	Installation . . . . .	3
2.1	MikTeX . . . . .	3
2.2	teTeX . . . . .	3
2.3	TeX Live . . . . .	4
2.4	emTeX . . . . .	4
2.5	Scientific Word . . . . .	4
2.6	Other Distributions . . . . .	5
2.7	Preparing Documents from Scientific Word in Other Distributions . . . .	5
3	Testing the Installation . . . . .	5
4	Document Structure . . . . .	5
4.1	Preamble . . . . .	6
4.2	Title Page . . . . .	8
4.2.1	Title . . . . .	8
4.2.2	Acknowledgements . . . . .	9
4.2.3	Abstract . . . . .	9
4.2.4	JEL Codes . . . . .	9
4.2.5	Keywords . . . . .	9


4.3	Nontechnical Summary . . . . .	10
4.4	Macros for the Document Body . . . . .	10
5	References . . . . .	10
5.1	Writing References Using BibTeX . . . . .	11
5.2	Writing References Using Special Macros . . . . .	11
5.2.1	Types of Cited Works . . . . .	12
5.2.2	Specifying Fields . . . . .	13
5.3	How to Choose the Entry Type . . . . .	16
6	Figures and Charts . . . . .	16
7	Preparing Figures . . . . .	17
7.1	Graphics Formats . . . . .	17
7.2	Preparing Vector Figures . . . . .	20
7.3	Virtual Printers . . . . .	20
7.3.1	Printers with PostScript Output . . . . .	20
7.3.2	Printers with PDF Output . . . . .	21
7.4	Converting and Editing Some Graphics Formats . . . . .	21
7.4.1	Converting Figures from WMF to EPS Format . . . . .	21
7.4.2	Converting Between EPS and PDF Formats . . . . .	22
7.4.3	Gnuplot . . . . .	22
7.4.4	Corel Draw . . . . .	22
7.4.5	Adobe Illustrator . . . . .	22
7.4.6	Repairing Corrupted EPS and PDF Files . . . . .	23
8	Preparing Tables . . . . .	24
8.1	Aligning Columns on a Decimal Point . . . . .	24
9	Figures and Tables as Floating Bodies . . . . .	25
9.1	Placing Floating Bodies . . . . .	25
9.2	Documents with a Large Number of Figures and Tables . . . . .	26
9.3	Wide Tables and Charts . . . . .	27
10	Generating the Final Document in PDF Format . . . . .	30
11	Submitting Your Document for Editing and Publication . . . . .	30
12	Sample Files . . . . .	32
13	Changes, Version 2013.12 . . . . .	32
A	Appendix . . . . .	33
B	Index . . . . .	39

## 1 Typographical Conventions

Several font types with special meanings will be used in this document. Text printed in non-proportional font will be used for printing out pieces of code as they should be written in the source file in L<sup>A</sup>T<sub>E</sub>X. This font may also appear directly in the text in descriptions of macros. This means that `\usepackage` and `suchlike` may occur in the text. Names of parameters and other variable objects will be written in *italics*. In the section describing how to create a list of references we will use `\typItem`, where *typ* will substitute for part of the name of a macro. This will generally mean the macros `\bookItem`, `\miscItem` etc.

For the names of files, packages and classes we will use sans-serif font. Examples can be seen in the document title and in the abstract. The exception to this will be URLs, where we need some special characters. We will, therefore, print URLs in non-proportional font as well, e.g. `ftp://ftp.cstug.cz/pub/tex/CTAN/`, but this should not cause any confusion. We will also write full file names with paths in the same way, e.g. `/usr/share/texmf-dist/tex/latex/cnb/cnbwp.cls`. These long names can be split across more than one line without hyphenation.

Emphasised text will be printed in **bold font**, as italics are already being used for a different purpose. Bold font is also used in the headings. In such cases, names of files, packages and classes will be printed in **bold sans-serif font**.

 A large bold exclamation mark at the start of a paragraph indicates very important information. Ignoring the instructions given in such a paragraph will cause a major error or an error with a strange message (such that the cause of the error will be difficult to track down) to occur during document preparation.

## 2 Installation


All macros used for writing CNB Working Papers are implemented in the `cnbwp.cls` class. The class is distributed in the archive file `cnbwp.zip`. The file is saved in the archive with a path conforming to the TDS (T<sub>E</sub>X Directory Structure) standard. The installation method varies slightly depending on the specific T<sub>E</sub>X distribution.

### 2.1 MikT<sub>E</sub>X

Extract the `cnbwp.zip` file to the `X:\localtexmf` directory, where `X`: denotes the disk on which MikT<sub>E</sub>X is installed (usually `C:`). Then open the menu `Start/Programy/MikTeX/MikTeXOptions`. On the `Roots` tab, check that `X:\localtexmf` is on the list of directories searched. If not, add it. Then click `Refresh FNDB`. To determine whether L<sup>A</sup>T<sub>E</sub>X will locate the class, use the command:

```
findtexmf cnbwp.cls
```

If everything is OK, the command will print the full path for the `cnbwp.cls` file.

 The `X:\localtexmf` directory is intended for local files that are not a standard component of MikT<sub>E</sub>X. If you update MikT<sub>E</sub>X this directory will not be overwritten, so you will not lose the files.

### 2.2 teT<sub>E</sub>X

This distribution is the standard in Unix systems and is also available for OS/2 and eComStation. We will describe the installation in Linux only. The installation in OS/2 is similar, differing only in that a backslash is used to separate the directories and the name of the disk on which teT<sub>E</sub>X has been installed must be entered explicitly.

Extract the `cnbwp.zip` file to the `/usr/share/texmf-local` directory. When installing in Unix systems, you must make sure that the `/usr/share/texmf-local/tex/latex/`

cnb/cnbwp.cls file does not have DOS line ends. Once the file has been extracted, the file database usually needs to be refreshed using the `mktexlsr` command.

T<sub>E</sub>X Live is based on this distribution, so everything described in the following section applies to t<sub>E</sub>X as well.

## 2.3 T<sub>E</sub>X Live

T<sub>E</sub>X Live is a popular multiplatform T<sub>E</sub>X distribution. It comes from the same sources as t<sub>E</sub>X, so the installation is similar. The directory to which `cnbwp.zip` should be extracted is usually `/usr/local/texlive/texmf-local` in Unix systems and `X:\TeXLive\texmf-local` in Windows. However, T<sub>E</sub>X Live may be installed in any directory and you can even have several versions of T<sub>E</sub>X Live installed simultaneously. The directory to which `cnbwp.zip` should be extracted can be determined using the command:

```
kpsewhich --expand-var=$TEXMFLOCAL
```

The `texmf-local` directory tree is shared by all T<sub>E</sub>X Live versions installed on a computer. If a newer version is installed, the files will automatically be found.

! The T<sub>E</sub>X Live distribution usually requires `mktexlsr` to be run after the files have been added. In distributions for Windows there is even an item in the T<sub>E</sub>X Live menu for this purpose. If the database is not refreshed using `mktexlsr` and two exclamation marks are given in front of the name of the directory containing the `cnbwp` class, L<sup>A</sup>T<sub>E</sub>X will not be able to locate the class. To determine whether L<sup>A</sup>T<sub>E</sub>X will locate the class, use the command:

```
kpsewhich cnbwp.cls -programe latex
```

If everything is OK, the command will print the full path for the `cnbwp.cls` file.

## 2.4 emT<sub>E</sub>X

This distribution is obsolete and Eberhard Mattes no longer maintains it. It is better to switch to another distribution. However, if you do want to use emT<sub>E</sub>X, extract `cnbwp.zip` to an auxiliary directory, create a `cnb` subdirectory in `X:\emtex\texinput\latex2e` and copy `cnbwp.cls` and `cnbwpsizes.clo` into it.

## 2.5 Scientific Word

This manual is written for version 5.5. In this version, extract `cnbwp.zip` to `X:\sw55\TCITeX`. In other versions the name of the root directory will probably differ. Scientific Word does not adhere strictly to TDS, as the directory names contain both lower-case and upper-case letters. However, the FAT and NTFS file systems are not case sensitive, so this should not cause any problems.

## 2.6 Other Distributions

To install the `cnbwp` class in other distributions you will need to follow the instructions given in the manual supplied with the relevant distribution. The current distributions are usually based on `web2c` and conform to the TDS standard, so the installation procedure is analogous to that for  $\text{\TeX}$  Live, see section 2.3.

## 2.7 Preparing Documents from Scientific Word in Other Distributions

Documents created using Scientific Word require specific macros defined in files in the `SWmacros` directory. These files are freely distributable. So if you want to process a document in another distribution, copy the entire directory to `texmf-local/tex/latex` or `localtexmf/tex/latex`, depending on which distribution you are using.

! Remember that in some distributions after adding files you have to refresh the database either using the `mktexlsr` command or from the menu.

## 3 Testing the Installation

In many distributions the installation can be tested using the `kpsewhich` command, as mentioned in section 2.3. This enables you to check whether  $\text{\LaTeX}$  is able to locate the `cnbwp` class and the macros from the `SWmacros` directory. If you do not have `kpsewhich` in your distribution, all you need to do is write a simple document:

```
\documentclass{cnbwp}
\begin{document}
Hello world.
\end{document}
```

If, when processing this document,  $\text{\LaTeX}$  returns with:

```
! LaTeX error: file `cnbwp.cls' not found
```

the class is not installed correctly. In this event, check whether you followed the instructions given in the relevant section correctly.

## 4 Document Structure

A working paper is a standard document written in  $\text{\LaTeX}$  and thus consists of the standard parts: a preamble and a document body. Everything preceding the command `\begin{document}` is part of the preamble. The list of references is a special part of the document body and is dealt with separately in section 5.

## 4.1 Preamble

All the main macros are implemented in the `cnbwp` class. Each document must therefore start with the command:

```
\documentclass{cnbwp}
```

The `\documentclass` command allows parameters to be entered in square brackets before the class name. This case is no exception. `cnbwp` is derived from the `article` class and assumes all its parameters. The Working Papers format is a fixed one, however, so parameters that alter the paper size, the font size, and number of columns are ignored.

A number of new parameters may be used in the `cnbwp` class. The first of these concerns the numbering of figures and tables. By default, tables and figures are numbered nonhierarchically. The numbering mode can be changed. It is set using the following parameters:

**hierarchicalnumbering** hierarchical numbering  
**simplenumbering** simple numbering

If neither of these parameters is specified, or if both are specified, simple numbering will be used. Hierarchical numbering can be activated with the command:

```
\documentclass[hierarchicalnumbering]{cnbwp}
```

Figure and table captions are left-aligned by default. In some documents it may look better aesthetically if the captions are centred. The alignment for the document as a whole is set using the following parameters:

**standardcaptions** captions left-aligned  
**centeredcaptions** captions centred

If neither of these parameters is specified, or if both are specified, the figure and table captions will be left-aligned.

In some circumstances it is necessary to align some captions differently than other captions in the document. This can be done using macros that will be explained later on in section 4.4.

The `cnbwp` class implicitly loads the packages needed for formatting Working Papers. These packages are: `ifpdf`, `mathptmx`, `fontenc` with `T1` parameter, `babel` with `czech` and `english` parameters, `natbib`, `url` and `keyval`. The parameters specified in square brackets in the `\documentclass` command are automatically sent to all these packages as well as to any other packages that you might load later on. You will probably most often specify parameters for the `natbib` package. If you wish to change the citation style from author-year to numbered format, use:

```
\documentclass[numbered]{cnbwp}
```

The parameters can be used in any combination you like. It does not matter which order you list them in. If, then, you want a base font size of 10.95 pt, centred captions and numbered literature references, you can write:

```
\documentclass[centeredcaptions,numbered,11pt]{cnbwp}
```

Using the `\usepackage` command you can now load the other packages you need to format your document. In a single `\usepackage` command you can specify two or more packages separated by commas, e.g.:

```
\usepackage{graphicx,amsmath,dcolumn}
```

The `\usepackage` macro also lets you specify parameters in square brackets. These parameters are only sent to the packages specified in the given command. However, you cannot then load more than one package using the same command, as  $\text{\LaTeX}$  would display a message stating that the parameters are not implemented in the package. You need to use, for instance:

```
\usepackage{amsmath,dcolumn}
\usepackage[dvips]{graphicx}
\usepackage[figuresright]{rotating}
```


The parameters specified in the `\documentclass` command will be sent to all packages. Thus the parameter `draft` can be specified globally:

```
\documentclass[draft]{cnbwp}
\usepackage{graphicx}
```


However, you can also use it with the `graphicx` package only:

```
\documentclass{cnbwp}
\usepackage[draft]{graphicx}
```

In both cases the figures will be replaced by a rectangular box of the same size as the figure and the file name will be printed inside this box. In the first case, moreover, overfull boxes will be indicated by black rectangles in the margin.

 The `hyperref` package uses a different definition of the `\url` macro than the `url` package. In addition, this definition causes conflicts in the bibliography. The `cnbwp` class therefore copies the `\url` definition from the `url` package to an auxiliary macro `\CNBurl`, and after `\begin{document}` is executed the `\url` macro definition is refreshed so that it corresponds to the `url` package. If you wish to use the macro from the `hyperref` package, save it after loading the package in a macro with a suitable name, e.g. using:

```
\let\MYurl\url
```

 Never change the `\url` macro definition, as the macros for formatting the list of references depend on this definition. The `\url` macro may be used in the citation database drawn up by `BibTeX`, so you will not necessarily see it directly in the document.

If the definition is changed, L<sup>A</sup>T<sub>E</sub>X will display strange error messages and the formatting of part of the document will probably crash completely.

! L<sup>A</sup>T<sub>E</sub>X does not require the use of separate name files for each package. Packages may therefore contain conflicting definitions that can cause unpredictable errors. Besides the conflicts mentioned above, the `path` package cannot be used at the same time as the `url` package. It is therefore a good idea to begin with a minimal document and only use those packages which are absolutely necessary for processing. Conflicts have to be located and eliminated by trial and error. Sometimes it also helps to change the order in which you load the packages.

At the end of the preamble you should define your own macros needed throughout the document. The `\DeclareRobustCommand` command is especially useful for defining macros. It has the same syntax as `\newcommand`. The only difference is that the defined macro is robust. When using macros defined in this way in the `\section` and `\subsection` commands, you do not need to use `\protect`.

In some cases you will need to adjust the margin settings, as some printer drivers ignore the margins set in the file and move the text. Before you start making changes, check your printer's properties. Some drivers always default to letter format, which means that you have to change the format to A4 before each print job. If the margins still disagree, make adjustments in the registers `\hoffset` (to move the text horizontally) and `\voffset` (to move the text vertically). For instance, if you need to move the text 5 mm to the right and 3 mm upwards, write:

```
\hoffset=5mm
\voffset=-3mm
```

You can ascertain the values by printing out a test figure in `a4-portrait.pdf`.

## 4.2 Title Page

A Working Paper must always start with a title page, which is made up of several components. The following subsections will deal with each of these in turn.

### 4.2.1 Title

The title lies at the boundary between the preamble and the document body. In fact, some macros can be put in the preamble. More precisely, the macros defining the title, the names of the authors and the acknowledgements can be written anywhere between `\documentclass` and `\maketitle`. It can thus be useful to place them right at the beginning of the file.

The title of the work is written as the argument of the `\title` macro. The title will be printed later, after you write the `\maketitle` command, but it will also be put in the header:

```
\title{Twelfth Night, or What You Will}
```

The names of the authors are entered using the `\author` macro that requires two parameters. The first parameter contains the full name. The authors' institution is entered in the second argument, e.g. as follows:



```
\author{Captain Nemo}{Nautilus}
\author{Robinson Crusoe}{Desert Island}
```

## 4.2.2 Acknowledgements

Acknowledgements are an optional part of a Working Paper. They are written in the argument of the `\acknowledge` macro.

The title will be printed by specifying the `\maketitle` macro. This macro must come after `\begin{document}`.

## 4.2.3 Abstract

The English abstract should be placed in the `abstract` environment, the Czech abstract should appear in the `abstrakt` environment, i.e. as follows:

```
\begin{abstract}
Place the abstract here.
```

The abstract may contain more than one paragraph. A blank line should be left between the paragraphs. No blank line is needed at the end of the abstract.

```
\end{abstract}
```

```
\begin{abstrakt}
Zde uvedeme abstrakt práce.
```

Abstrakt může obsahovat více odstavců. Mezi odstavci ponecháme prázdný řádek. Na konci abstraktu prázdný řádek být nemusí.

```
\end{abstrakt}
```

Czech hyphenation rules are activated automatically in the `abstrakt` environment.

## 4.2.4 JEL Codes

JEL codes should be put in the `\JEL` macro:

```
\JEL{E22, E23, E32, E52}
```

## 4.2.5 Keywords

Keywords are the final element of the title page. They are entered using the aforementioned `\Keywords` macro:

```
\Keywords{20,000 leagues, maelstrom, Nautilus, sea, whale}
```

The CNB requires keywords to be given in alphabetical order. However, the above macro does not perform this task, so authors must do it themselves.

### 4.3 Nontechnical Summary

The Nontechnical Summary is about one page long and is printed on a separate page in the document. Like the abstract, it is written in the `nontechsummary` environment:

```
\begin{nontechsummary}
```

Write the Nontechnical Summary in this environment. The title will be created automatically and will not be numbered.

The text usually consists of several paragraphs separated by blank lines.

```
\end{nontechsummary}
```

! The `\Keywords` macro and the `nontechsummary` environment force a page break. If both these elements are missing and the page break is forced manually using the `\newpage` or `\clearpage` command, part of the document will be incorrectly formatted.

### 4.4 Macros for the Document Body

The body of the document is written in the same way as with the article class. Sections are therefore specified using the `\section` macro and subsections using the `\subsection` macro. The `\subsubsection` command is also implemented. Subdivision to `\paragraph` and `\subparagraph` was not envisaged in the design, but no syntax error will occur if these macros are used.

Table and figure captions are written in the `\caption` macro. The captions will be aligned according to the parameter specified in the preamble (see page 6). If you wish to align a caption differently, you must explicitly specify `\standardcaption` to left-align it and `\centeredcaption` to centre it. The numbering format always corresponds to the parameters specified in the preamble, which are explained on page 6. It cannot be changed in the document. This would be nonsensical from the typographical perspective anyway.

The description of a table or a figure may contain a note entered by using the `\Note` macro and information on a source entered by using a `\Source` macro. They are usually placed below the `\caption` macro as follows:

```
\Note{Value Added Tax (VAT), Excises (E), Personal Income Tax (PIT),  
      Social Security Contributions (SSC), Inheritance Tax (IT),  
      Corporate Income Tax (CIT), Other Age-Specific Revenues (OR).}  
\Source{CNB}
```

## 5 References

The list of references (bibliography) can be written in two ways: either using the Bib $\TeX$  system, or not using it. In either case the `natbib` package, which is automatically loaded by `cnbwp`, is used for the actual formatting. Consequently, the appearance of the document does not depend on the method chosen for writing the list of references.

Parameters can be transferred to the `natbib` package as follows. Optional parameters can be written in square brackets in the `\documentclass` command. These parameters

are sent both to the article class, from which the `cnbwp` is derived, and to all the loaded packages. Thus it is possible to specify parameters for the `natbib` package. For example, the citation style can be changed from author-year to numbered format using the `numbered` parameter.

The method for creating a list of references using Bib<sub>TEX</sub> will be described briefly in section 5.1. The macros created for Czech National Bank Working Papers will be explained in detail in section 5.2 and the subsections thereof.

## 5.1 Writing References Using Bib<sub>TEX</sub>

The use of Bib<sub>TEX</sub> is described in Oren Patashnik's document „Bib<sub>TEX</sub>ing“, which comes as part of the standard <sub>TEX</sub> distributions. We will make use of the `abbrvcnb` style derived from the `natbib` bibliographic styles. Let's assume that you have a bibliography database stored in a file called `references.bib`. You should then place the following in the <sub>TEX</sub>document:

```
\bibliographystyle{abbrvcnb}
\bibliography{references}
```

Once the document has been processed by <sub>TEX</sub>, run Bib<sub>TEX</sub>:

```
bibtex document
```

(In place of the word *document* write the file name; here we have assumed that the document is stored in a file called `document.tex`.) The document then needs to be processed twice by <sub>TEX</sub>. Occasionally a third pass is required if a page shift occurs after the references are added to the text.

Document URLs are usually long strings in which automatic hyphenation does not work or the words are hyphenated in inappropriate places, hence confusion can arise. This problem is solved in conformity with the standards for writing URLs in the `url` package, which is also implicitly loaded by the `cnbwp` class. The URL usually appears in a *note* field. This field should therefore be written as follows:

```
note = "\url{ftp://ftp.cstug.cz/pub/tex/CTAN/}",
```

## 5.2 Writing References Using Special Macros

The list of references can be written using special macros directly in the <sub>TEX</sub>file in the `thebibliography` environment. This environment has one required parameter, which sets the width needed for the ordinal. Bib<sub>TEX</sub> enters the number of the citations in this parameter. You may, however, specify any text of suitable width, for instance `XX`, as used in the example (see the Appendix). Each citation is written in the following format:

```
\typItem[citation text]{label}{citation content}
```

In place of the word *type* you need to specify the relevant type of work cited. We will use an identifier derived from the standard types defined in Bib<sub>TEX</sub>. A list and description of these can be found in section 5.2.1.

The first two parameters (optional and required) have the same meaning as for the `\bibitem` macro. The required parameter *label* is used when citing text in the `\cite` macro. The optional parameter *citation text* contains the information that is written in the text at the citation location. This parameter has the general format:

[*short text*(*year*)*long text*]

! The final parameter of `\typItem` macros is processed using the `keyval` package. This means that only declared fields may be used. It is not possible to create an unknown field to add a comment that is not meant to be printed. Such an attempt will lead to an error during the processing of the document. Comments are written after a *procento* sign, as in standard  $\text{\LaTeX}$  texts.

The optional parameter and the first required parameter of `\typItem` macros are transferred unchanged to the `\bibitem` macro.

The `natbib` package can choose between short and long text. If the cited work has only one author, *dlouhý text* is not specified. The entire parameter can then be written in the following format:

[Mouse(1984)]

Both options are specified for works with more than one author. If, for example, you want to cite a book issued in 1984 written by Mickey Mouse, Donald Duck and Bugs Bunny, the parameter should be written in the following format:

[Mouse et.~al.(1984)Mouse, Duck and Bunny]

!  $\text{BibT}_{\text{E}}\text{X}$  generates all pieces of information for the `\typItem` macro automatically. If the information is entered manually, the parameter in the square brackets must be properly formatted. Especially, the value in the round parentheses must be identical to the year value as shown in appendix A. The items must also be sorted by the author.

Selection of the citation text format is based on the required appearance of the resulting text. If you do not require citations in long format, you need not specify that option in the `\typItem` macro.

The `natbib` package requires round brackets in the optional parameter. If you do not know the year of publication of a cited work, you must write at least empty brackets (`()`).

The reference to the cited work is written in *citation content*. The format of this parameter will be described in section 5.2.2.

### 5.2.1 Types of Cited Works

The macros used for citing works are based on the standard types defined in  $\text{BibT}_{\text{E}}\text{X}$ . Here we will list and briefly describe these types and provide a list of the required and optional fields. The meaning of the fields and the way they are written will be explained in the following section.

`\articleItem` An article from a journal or magazine. Required fields: `author`, `title`, `journal`, `year`. Optional fields: `volume`, `number`, `pages`, `month`, `note`.

`\bookItem` A book with an explicit publisher. Required fields: author or editor, title, publisher, year. Optional fields: volume or number, series, address, edition, month, note.

`\bookletItem` A work that is printed and bound, but without a named publisher or institution. Required field: title. Optional fields: author, howpublished, address, edition, month, year, note.

`\conferenceItem` The same as `\inproceedingsItem`.

`\inbookItem` A part of a book, usually a chapter or a range of pages. Required fields: author or editor, title, chapter and/or pages, publisher, year. Optional fields: volume or number, series, type, address, edition, month, note. The meaning of the type field will be explained on page 15.

`\incollectionItem` A part of a book having its own title. Required fields: author, title, booktitle, publisher, year. Optional fields: editor, volume or number, series, type, chapter, pages, address, edition, month, note. The type field has the same meaning as in the previous citation type and is explained on page 15.

`\inproceedingsItem` An article in a conference proceedings. Required fields: author, title, booktitle, year. Optional fields: editor, volume or number, series, pages, address, edition, month, organization, publisher, note.

`\manualItem` Technical documentation. Required field: title. Optional fields: author, organization, address, edition, month, year, note.

`\mastersthesisItem` A Master's thesis. Required fields: author, title, school, year. Optional fields: type, address, month, note.

`\miscItem` Another type used when nothing else fits. It has optional fields only author, title, howpublished, month, year, note.

`\phdthesisItem` A PhD thesis. Required fields: author, title, school, year. Optional fields: type, address, month, note.

`\proceedingsItem` The proceedings of a conference. Required fields: title, year. Optional fields: editor, volume or number, series, address, month, organization, publisher, note.

`\techreportItem` A report published by a school or other institution, usually numbered within a series. Required fields: author, title, institution, year. Optional fields: type, number, address, month, note.

`\unpublishedItem` A document having an author and title, but not formally published. Required fields: author, title, note. Optional fields: month, year.

If none of these types meets your requirements, you can use `\bibitem` directly and format everything manually.

## 5.2.2 Specifying Fields

The content of each field of a cited work is entered in the last parameter of `\typItem` macros. The fields are specified in the following format:

```
{name1 = value1, name2 = value2, name3 = value3}
```

Spaces around equals signs and commas separating fields are ignored. Blank fields are also ignored. It is therefore possible to write two or more commas in succession or a comma before the first field and after the last one. Wherever a space is permitted, a line can also be terminated. In addition,  $\TeX$  ignores all spaces at the start of a line. The list of references can thus be written in a clear, easy-to-read style like that used in the Appendix.


Fields are separated from one another with commas. If you want to write a comma in the field value, you have to enclose the entire value in curly brackets. This will occur most frequently in citations of works with more than one author. Where a publication has only two authors, their names should be joined with the conjunction *and* as follows:

```
author = Donald Duck and Mickey Mouse, ...
```

Here a comma terminates the field. If there are more authors, their names should be enclosed in brackets as follows:

```
author = {Donald Duck, Mickey Mouse, and Bugs Bunny}, ...
```

The first comma (inside the brackets) is part of the list of authors, and the last comma terminates the field. In case of three or more authors the conjunction *and* should be preceded by a comma.

 Omitting a comma or leaving out the brackets around a value containing a comma results in an error message about an unknown field with a strange name. Sometimes errors of this type cause the loss of an entire chunk of the list of references and subsequent parts of the document.

Curly brackets are often used in database files destined for processing by Bib $\TeX$  in order to prevent conversion to upper-case or lower-case letters.  $\LaTeX$  does not perform any conversions, so brackets are not needed. However, various macros can be used, e.g.  $\mbox$  to prevent incorrect hyphenation and  $\-$  to specify where to hyphenate words correctly, and inside the `thebibliography` environment you can even define your own macros, which will be unknown (or defined differently) in other parts of the document.

Fields can be entered in any order, but only the following may be used:

**address** The address of the publisher or institution (e.g. the school in the case of a thesis)

**author** The name(s) of the author(s), in the format as they should appear. Unlike Bib $\TeX$  there is no reformatting (forenames will not be shortened to initials etc.).

**booktitle** Title of a book, part of which is being cited.

**chapter** A chapter number.

**edition** The edition. This should be an ordinal and can also be entered as a word, e.g. *Second*.

**editor** Name(s) of editor(s), entered in the same format as the author names.

**howpublished** How something strange has been published.

**institution** The sponsoring institution of a technical report (`\techreportItem`).

**journal** A journal name or abbreviation.

**month** The month in which the work was published. The day can also be given in this field.

**note** An explanatory note. This field is given in the examples in the Appendix, but it is rarely used in practice. It is used to state the URLs of works published on the Internet. For automatic and correct splitting of long URLs on multiple lines, the `url` package loaded by the `cnbwp` class is used. URLs should therefore be written in the following format:

```
note = \url{ftp://ftp.cstug.cz/pub/tex/CTAN/}, ...
```

**number** The number of a journal, magazine, technical report, or of a work in a series.

**organization** The organisation that sponsors a conference or that publishes a manual.

**pages** A range of pages.

**publisher** The publisher's name.

**school** The name of the school where a thesis was written.

**series** The name of a series (or set) of books.

**title** The work's title.

**type** The type of a work, e.g. *Research Note*. In citations `\inbookItem` and `\incollectionItem` specify the text corresponding to the `chapter` field. The default is `chapter`, but you can specify `section`, for example.

**volume** The volume of a journal or multi-volume book.

**year** The year of publication or, for an unpublished work, the year it was written.

The breakdown into fields need not be strictly observed. The information can be split into separate fields:

```
publisher = John Wiley \& Sons, address = New York, ...
```

The list of references written directly in a  $\LaTeX$  document will probably not be used for further machine processing. It may therefore be useful to combine the two pieces of information into a single field:

```
publisher = {John Wiley \& Sons, New York}, ...
```

Here we had to use brackets, as the combined information contains a comma.

Macros automatically add full stops or commas as individual fields, although only where the text does not end in a punctuation mark. Consider a title typed as:

```
title = Can we spend less money for more music?, ...
```

In this case no punctuation mark will be added, even if the entire title is enclosed in curly brackets.

### 5.3 How to Choose the Entry Type

An entry type must be chosen for making BibTeX database entries and when using `\typItem` macros. The type names are intuitive and it should be straightforward to make the right choice. In cases of doubt, it helps to know which fields are required and which are optional. Choose the type whose structure best fits the purpose.

Some fields seem superfluous. For example, the bibliographic types for Master's (`\mastersthesisItem`) and Ph. D. (`\phdthesisItem`)<sup>1</sup> thesis have exactly the same format. The only difference is the default value of the optional field `type`. If you want to cite a similar publication, for instance a dissertation, you can use any type and specify the optional field:

```
type = Dissertation, ...
```

Works only available from the Internet have not been formally published. For such publications, use `\unpublishedItem` (in BibTeX `@unpublished`). URLs of works should be placed in the `note` field, as mentioned earlier.

## 6 Figures and Charts

TeX is a text-processing program. Apart from primitive graphics created with horizontal and vertical strokes and characters made from specially prepared fonts, no figures can be created directly. You can, however, import<sup>2</sup> figures created in other applications and saved in separate files.

Figure and tables can be (but do not have to be) inserted in the text using the floating environments `figure` and `table`, which among other things allow such objects to be easily numbered, described, placed and, as the case may be, further manipulated. For a description of some of the more advanced techniques for working with floating environments, see also section 9.

The easiest way to load a figure into a floating environment is to use the `graphicx` package. The `\includegraphics` macro has one required parameter – the name of the file containing the figure. A series of different instructions can be specified in the optional parameter in square brackets in the format `key=value`. The syntax is the same as for the `\typItem` macro, which was described in section 5.2.2, except that square brackets are used instead of curly brackets, as the first parameter is optional. Here, too, the parameters are processed using the `keyval` package.

A detailed description of the `\includegraphics` command can be found in the file `grfguide.ps`, which is part of the distribution. Here we will describe only a few of the main parameters:

**scale** is the scale factor. Enter `scale=.5` to reduce by half.

---

<sup>1</sup>In BibTeX `@mastersthesis` and `@phdthesis`.

<sup>2</sup>Technical speaking, there isn't even a command for inserting figures. However, TeX does have the `\special` command, which is not interpreted at all and whose argument is merely sent to the output device. How this command is handled depends on the relevant driver. When we talk about how figures are inserted in TeX or LaTeX documents, we are always referring to some specific device driver.



**width** specifies the preferred width. The dimension can be entered in the units of your choice or as a percentage of other dimensions. If you want a width of 75 % of the width of the text, use `width=.75\textwidth`.

**height** specifies the height of the image. Entering `height` and `width` simultaneously will distort the image. If just one of these parameters is entered, the second will be calculated so as to maintain the aspect ratio.

**clip** crops the bounding box. Some programs (e.g. Quattro) create defective EPSs which delete their surroundings. If part of the text above or next to a figure disappears when you insert it, try using the `clip` parameter (without writing an equals sign and value).

**page** is used only when importing an image from a multi-page PDF file. It is used to specify the number of the page you wish to insert.

The graphics format in which the imported images are saved should be chosen according to the method used to create the final PDF document file. PDF creation methods are described in more detail in section 10 and a discussion of some of the most important graphics formats can be found in section 7.

When using `dvips` or `dvipdfm` or when using a virtual postscript printer, EPS is the best graphics format for both bitmap and vector images.

`pdfTeX` can load images in JFIF and PNG formats, but these are bitmap and are used only rarely. Vector images must be prepared in PDF format. Adobe Distiller can be used to convert from EPS to PDF, although `epstopdf` offers the same function. The latter, however, requires GhostScript and perl. These programs are freely distributable and are available for all operating systems.

## 7 Preparing Figures

This section will focus on how to prepare figures. We will start by explaining the differences between the main graphics formats and go on to describe several ways of preparing and editing figures.

### 7.1 Graphics Formats

Images can be divided into two basic formats: bitmap and vector. Bitmap images represent an array of values (a raster) where each element carries information on the colour of a displayed pixel of non-zero size. Vector images differ in that the curves within them are described using mathematical equations. Information on line width, line colour and, as the case may be, fill colour is attached to each curve. Rasterisation occurs at the instant the image is displayed on the monitor or printed.

It may seem that there is no great difference in the result, as even a vector image has to be rasterised, but nothing could be further from the truth. The fundamental difference is that each display or printing device has a different resolution, i.e. a different raster density. A vector image is rasterised at the instant it is displayed, hence it is rasterised at a specific resolution. The size and shape of the dots also matters. This

determines the colour of the dots that are mathematically on the boundary between the coloured and uncoloured parts of the plane. The display device's rasterisation algorithm calculates them in such a way that the curves look smooth.

When a bitmap image is displayed, the prescribed size has to be maintained even if the image was created for a different resolution. This means that there is no other option than to calculate the dots by interpolation. In a bitmap image we no longer have any information on whether the dot lay on the curve. Although the driver of the relevant device knows the size and shape of the dots, it does not have enough information to be able to use them when rasterising.

Example vector and bitmap charts are given in Figure 1. The figure is one of the demonstration examples supplied with gnuplot. The vector and bitmap charts were created from the same source.

The top image is the vector image. It was exported as EPS (Encapsulated PostScript) and converted to PDF so that it could be loaded by pdf $\TeX$  or by X $\LaTeX$ . The curves may look jagged on the screen, because the screen resolution is low, but this effect disappears when the image is printed on a quality printer.

The central image was exported in PNG (Portable Network Graphics) format. The difference in image quality is plain to see. The worse font quality is particularly visible. At some magnifications the axes can disappear altogether. The axes remain jagged even when the image is printed on a high-resolution printer, since, as mentioned earlier, the driver of the relevant device does not know that the pixels should all lie on the same line and merely recalculates the „steps“ at a different resolution.

Raster data take up too much memory. However, images tend to contain repeated colour sequences and monochromatic areas, so data can be stored more efficiently using a compression algorithm. Compression algorithms are divided into two classes: lossless and lossy. Lossless algorithms allow the original data to be restored by decompression. This is impossible in principle with lossy algorithms.

The PNG format, used in the central image, uses a lossless compression algorithm. The bottom image was saved in a format that employs lossy compression – JPEG<sup>3</sup>. This algorithm was developed by the Joint Photographic Expert Group for storage of colour photographs. Research has shown which elements of an image cannot be seen by the human eye. Such elements can thus be removed from a photograph without any problem. The quality is not compromised and the file size is considerably reduced. The quality of the image can be modified, but even with the highest image quality (lowest compression) the compression is always lossy. This format is called JFIF (JPEG File Interchange Format) and the files usually have the extension *jpg*, or less often *jpeg*, *jff* or *jfif*.

The bottom image was created at a low resolution suitable only for the screen and was intentionally saved at a low quality setting to make it obvious at first glance that lossy compression leads to blurring and the formation of colour artefacts, especially in near-monochromatic areas.

**!** Remember that images should always be created in the format best suited to their nature. Charts and diagrams are always mathematical objects, which can be described using equations known from analytical geometry. Complicated curves are generally

---

<sup>3</sup>pronounced *jay peg*.

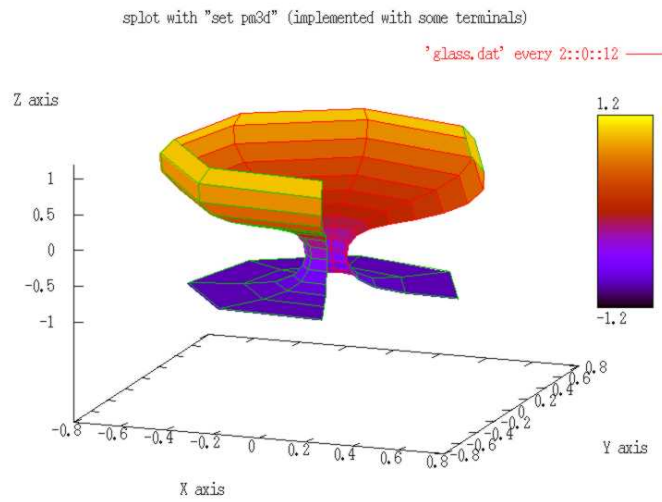
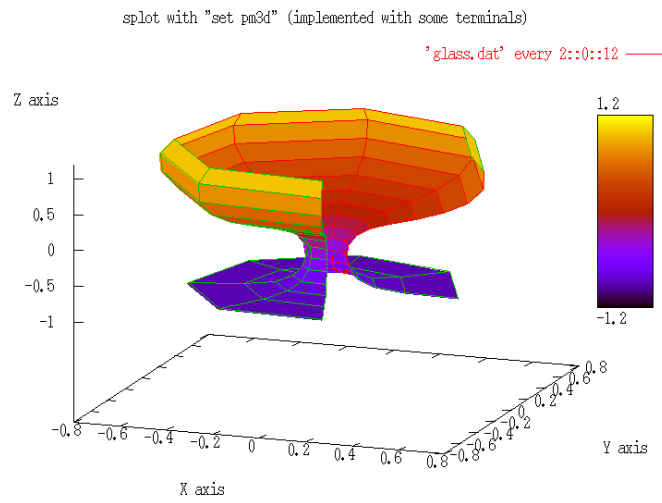
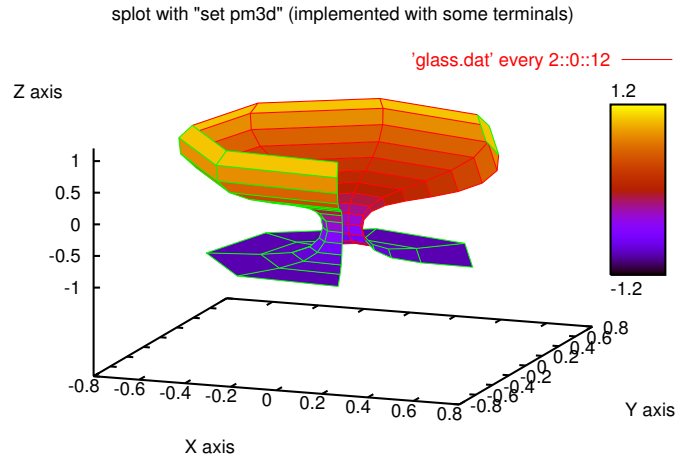


Figure 1: Comparison of vector image (top) with bitmap with lossless compression (centre) and JPEG lossy compression (bottom)

approximated by Bézier curves. Such images should therefore be created in vector format. Bitmap formats should only be used for images that do not have a mathematical representation. Lossless compression formats are preferred. Graphics formats with lossy compression should be used **solely for colour photos and nothing else!**

There are many myths surrounding graphics formats. One such is the belief that EPS is a vector format. This is only partly true. EPS format is basically PostScript with some restrictions. The file may contain vector or bitmap graphics or even a combination of the two. What matters when an image is saved in EPS format is the program that was used to create it. If the image was drawn using a bitmap editor such as Photoshop or Gimp, the resulting EPS will also be bitmap. The same goes for PDF format. Both formats allow lossless and lossy compression of bitmap images.

Many people mistakenly believe that there is no difference between the (PostScript) and EPS formats. Regrettably, this misapprehension is very common even among some writers of commercial software. EPS permits only a subset of the PostScript language and requires extra information before the programs can insert the image in the document. However, some programs create postscript files that only look like EPS. In some cases this doesn't matter, but often such an image cannot be inserted because it causes strange errors (the ensuing file gets corrupted and part of the text, or even the whole of the rest of the document, is lost). Sometimes an expert can repair such a file, but this can take several hours of investigation. Fortunately there are tools available for the job. We will touch on these in section 7.4.6.

Another myth concerns the bitmap format TIFF (Tagged Image File Format; usual file extension: `tif`). It is often claimed that this format uses lossless compression. In reality, though, the compression method can be selected when the file is saved. There are several lossless algorithms to choose from, and the image can even be saved with no compression whatsoever, but JPEG lossy compression may also be used.

## 7.2 Preparing Vector Figures

When preparing a vector figure, it is vital to use a graphics editor or similar program which genuinely works with vector representation. The figure must then be converted into a format that can be imported into the document, i.e. EPS or PDF. It is useful if the program is capable of writing the file in this format directly, as is the case, for example, with the Matlab system. If it is not, an alternative method needs to be used.

## 7.3 Virtual Printers

Virtual printers are an alternative solution where the program does not have EPS or PDF format output. However, this method is only useful if you are using a vector editor. A bitmap graphics editor will create a bitmap EPS or PDF even when a virtual printer is used.

### 7.3.1 Printers with PostScript Output

Printers with PostScript support are available on the market. Drivers for these printers come as a standard component of the operating system installation CD (Windows,

Mac OS, OS/2, eComStation). Some printers support multiple communication languages, such as PCL and HPGL. It is inappropriate to use such drivers in virtual printer mode, as there are usually PJJ or similar commands at the start and the end of the file generated which enable and disable the PostScript interpreter. Such commands do more harm than good as regards subsequent processing. Apple Writer printers are usually the best. To create colour images in EPS format you need to install a colour printer driver, as a monochrome driver may convert the colours to greyscale.

Install the virtual printer as a local printer in Windows and connect it to the file rather than to any physical device. If the driver allows function setting, enable EPS format and select Type 1 font conversion and insertion of fonts in the document. The standard Windows XP installation includes a large number of PostScript printer drivers. We have quite a good experience, for example, with the Apple LaserWriter Pro 600 and Lexmark Color 4079 plus PS2 printer drivers. That said, you will probably have to clip the EPS image (i.e. adjust the bounding box) created by these drivers using the method described in section 7.4.6.

The average CNB computer network user may not have the option of installing non-network printers. In such cases, it may be necessary to log in to the system as a local administrator in order to install a virtual local printer. After installation, this printer will be available to other users of the station. We recommend testing the functionality of the newly installed virtual printer only after logging in to the standard user account (i.e. u0xxxx).

To create a PostScript figure from a chart in MS Excel, for example, print the relevant chart on a virtual PostScript printer. It is a good idea to select 'Custom' under 'Printed chart size' on the 'Chart' tab in the File/Page setup dialogue box. This will ensure that the dimensions of the printed chart can easily be modified in MS Excel by setting the size of the 'Chart Area' object. When printing, **do not tick Print to file**, but wait until the driver itself asks you to enter the file name.

### **7.3.2 Printers with PDF Output**

Adobe Acrobat offers a virtual printer with PDF output called PDFWriter.

There are also numerous PDF-generating shareware programs that can be installed as virtual printers. Each has its own installation program. Again, where possible you should try to enable Type 1 font conversion and insertion of fonts in the document.

## **7.4 Converting and Editing Some Graphics Formats**

When producing documents it is often necessary to convert files between EPS, PDF and WMF formats and to modify such graphics files in other ways.

### **7.4.1 Converting Figures from WMF to EPS Format**

Scientific Word creates figures in WMF (Windows Metafile) format. To process a document in the standard T<sub>E</sub>X installation you will need to convert the figures to EPS format. To do so you can use the shareware program `wmf2eps`, an unregistered version of which is supplied with Scientific Word version 5.5. To work properly the program requires in-

stallation of a virtual printer allowing print to EPS. Suitable drivers are supplied with the program in the PSpint subdirectory. The installation process is described in the README.TXT file in the software distribution. The program itself is very user friendly and intuitive.

#### 7.4.2 Converting Between EPS and PDF Formats

A PS or EPS file can be converted to PDF using either Adobe Distiller or the ps2pdf program contained in the GhostScript distribution (or the epstopdf program).

Conversely, to convert from PDF to EPS or PS use either the full Adobe Acrobat (Save As EPS or Save As PS) or pdftops with the `-eps` parameter. `pdftops` is part of the freely distributable `xpdf`, see <http://www.foolabs.com/xpdf/>.

GhostScript also offers PDF to EPS conversion, but unfortunately it rasterises the fonts at the same time. Such figures are not usable.

#### 7.4.3 Gnuplot

Gnuplot is a flexible, freely distributable program for plotting mathematical graphs. It is available for all operating systems. Figure 1 was created using this utility. It supports outputs in many formats, including EPS and (in the latest version) PDF.

The program offers a wide range of curve colour, dot colour and dot shape options. These options depend on the output format selected, and the same type of curve can even be displayed in a different colour in different output formats. The best way to get information is to use the `test` command, which will give you a test figure showing all the options that the relevant format has to offer. To find out what options an EPS-format output offers, use the following commands:

```
set term postscript eps color solid lw 2 18
set output "test.eps"
test
set output
quit
```

The test figure will be in the `test.eps` file in the current directory.

#### 7.4.4 Corel Draw

Corel Draw is a commercial vector image editor. It offers export to EPS format. Bear in mind, though, that Corel Draw contains many fonts that are not available on other computers. When exporting to EPS, therefore, you have to convert such fonts into curves. In versions 8 onwards the software performs this operation automatically.

#### 7.4.5 Adobe Illustrator

Adobe Illustrator is a commercial vector editor whose native format AI is based on PostScript. When a file is opened in a standard text editor, it is indistinguishable from EPS at first glance. However, PostScript is a programming language that has one shared

feature with  $\TeX$ : it allows definition of user commands (using the `def` operator). The AI format contains postscript commands, but for these to be interpreted the defined macros contained in Adobe Illustrator are needed. If you insert an AI image directly into another document, everything will seem to work, but only until you try to print the document. On the page containing this figure, the PostScript interpreter will report an undefined error. So don't forget to save the figure in EPS format.

#### 7.4.6 Repairing Corrupted EPS and PDF Files

In section 7.1, on page 20, we mentioned that images in EPS and PDF files can get corrupted. We shall now describe how to repair them.

The first common error is an incorrect or unsuitable bounding box. The bounding box indicates the dimensions of the figure in the EPS file. According to the specification, the entire figure must be inside the box, but there is no mention that the box has to be tight-fitting. If the bounding box is too large, it is not an error, but it is not useful. The software uses this information when inserting the figure in the document. If the bounding box is too large it leaves a large empty area around the figure. This is best repaired using the freely distributable software GhostScript and its handy frontend GhostView. Both programs are needed. There are various different frontends for Linux (e.g. `gfv` for Gnome). Switch the orientation of the figure to Portrait and select the rather inappropriately named function `PS to EPS`. This does not, in fact, convert PostScript to EPS, but just generates a tight-fitting bounding box with the right dimensions. In most cases the automatic setting works correctly. Use manual if automatic fails or if you want to remove part of the figure and you plan to use the `clip` parameter in the `\includegraphics` macro.

! GhostView cannot determine the necessary paper size for displaying an image in EPS format. In addition, some programs, in particular virtual printers, do not insert the figure in the bottom left-hand corner. If all you see is an empty page, tick EPS Clip in the Options menu.

The page dimensions in a PDF file can be changed using the full Adobe Acrobat, which offers this function. I am not aware of any freely distributable alternative.

The situation is worse if the file only looks like an EPS. The simplest case is where there are PDL or other language commands at the start and end of the file. GhostView can usually cope with these, but problems will arise when the figure is inserted. The remedy is straightforward. Open the file in a standard text editor and delete everything before the first occurrence of the characters `%!PS` (in EPS these must be at the start of the first line) and everything after the text `%%EOF`. With a rudimentary knowledge of PostScript you can repair other errors. Usually all you have to do is delete the forbidden commands, but this can be a tough task even for an expert. It is easier to convert the image to PDF and then back to EPS (see 7.4.2) if you don't want to, or cannot, process the document using `pdf $\TeX$` .

## 8 Preparing Tables

Tables are an important component of technical papers. However, they are only useful if they are well arranged. Producing tables is usually the most time-consuming part of the document preparation process. Tables are prepared using the `tabular` environment, and in this section we will present a number of techniques that will make it easier to typeset tables and help you achieve the desired look.

Otherwise, as with figures, tables can be inserted in the text using the `table` floating environment. For a description of some of the more advanced techniques for working with floating environments, see also section 9.

### 8.1 Aligning Columns on a Decimal Point

Decimals are often used in tables. The number of digits in each number of a column can differ, but it is vital that the numbers are aligned on the decimal point. The `tabular` environment does not offer this option, but a solution can be found by using the `dcolumn` package. This package adds column type `D` to the `tabular` environment. Specification of column type `D` requires three parameters: the separator used in the source text, the separator that you want to be printed and the maximum number of decimal places. If this last parameter is negative the decimal point will be in the centre of the column, which can cause the column to be too wide. The first two parameters are usually the same, although this need not be the case. Let's assume that we have exported a table from a table software package which uses decimal commas. We, however, want decimal points. To avoid the need to mess with the file, we make use of the option provided by the first two parameters. An example is given in table 1, the source code for which can be seen in figure 2.

Table 1: Demonstration of table with decimal point alignment

<b>Jméno</b>	<b>Výška [m]</b>	<b>Váha [kg]</b>
Zlatovláska	1.68	62.3
Dlouhý	12.6	98.1
Široký	1.83	386
Bystrozraký	1.74	74.2

The heading in column type `D` would not be correctly positioned. We therefore have to use the `\multicolumn` macro, in which we can change the alignment of that particular cell. Note that the **following** vertical line, not the preceding one, is part of the alignment specification. The preceding line must be specified for the first column only. If, for some (in this case unwise) reason, the heading *Jméno* should be right-aligned, we would have to use:

```
\multicolumn{1}{|r|}{\bfseries Jm\`eno}
```

The new column type definition is enabled by the fact that the `array` package is implicitly loaded. This package offers users the `\newcolumntype` macro, which is used to



Figure 2: Source code for table 1

---

```

\setlength\extrarowheight{2pt}
\begin{tabular}{|l|D{,}{.}{2}D{,}{.}{1}|}\hline
\bfseries Jm\`eno & \multicolumn{1}{c}{\bfseries V\`y\v{s}ka [m]} &
\multicolumn{1}{c}{\bfseries V\`aha [kg]}\\\hline
Zlatovl\`aska & 1,68 & 62,3 \\
Dlouh\`y & 12,6 & 98,1 \\
\v{S}irok\`y & 1,83 & 386 \\
Bystrozrak\`y & 1,74 & 74,2 \\
\end{tabular}

```

---

define new types. The syntax is the same as that for `\newcommand`, except that we are defining a column type, not a macro, and we cannot declare an optional parameter with a default value. If, using the example from figure 2, we define a new column type:

```
\newcolumntype{d}[1]{D{,}{.}{#1}}
```

we can write the table preamble in a plainer way:

```
\begin{tabular}{|l|d{2}d{1}|}
```

In the table we have defined an extra dimension `\extrarowheight`, which is also defined in the array package. Letters with Czech diacritics are too high and there would be almost no space between the text and the line. The `\extrarowheight` dimension register increases the size of the space between the text and the line above it. The parameter in square brackets of the `\` macro just increases the size of the space below the text.

## 9 Figures and Tables as Floating Bodies

Figures and tables are inserted in the text using the floating environments `figure` and `table`, which among other things allow such objects to be easily numbered, described, placed and, as the case may be, further manipulated. Users should consult the  $\text{\LaTeX}$  system documentation for a description of basic manipulation of floating bodies. Here we describe some of the more advanced techniques, which will enable you to put floats exactly where you want them, to process documents with a large number of figures and charts and to insert rotated wide tables or charts.

### 9.1 Placing Floating Bodies

The optional parameter defining where the float should be placed is specified in square brackets. However,  $\text{\LaTeX}$  treats such parameters only as recommendations. If it is unable to execute them, it will try to place the float on a separate page. If even this is impossible, it will place this and all subsequent floats at the end of the document or chapter

(in the case of a book). You should therefore specify more than one option so that L<sup>A</sup>T<sub>E</sub>X has a choice. Stating the [h] specifier alone usually results in the float being shifted to a separate page or to the end of the document.

In some cases the float **has to be** in a specific location. You thus have to tell L<sup>A</sup>T<sub>E</sub>X that you do not mean the specifier as a recommendation but as a categorical command. The specifier [H] is used for this purpose. It was originally implemented in the here package, but this is no longer in the standard distributions. This specifier is now implemented in the float package.

! When [H] is used, the environment ceases to be floating. L<sup>A</sup>T<sub>E</sub>X will place it in the desired location regardless of whether it fits in the space remaining on the page. You should therefore use this specifier only in the final version of the document.

! This is a good place to explain the difference between the `\newpage` and `\clearpage` / `\cleardoublepage` macros. The first two instruct L<sup>A</sup>T<sub>E</sub>X to start a new page. `\cleardoublepage` goes to a right-hand page<sup>4</sup>. In addition, the `\clearpage` and `\cleardoublepage` macros will print all the as yet unplaced floats stored in the buffer.

## 9.2 Documents with a Large Number of Figures and Tables

The floating-body placement algorithm is controlled by the values of several registers. The values are chosen for normal documents, but they may not be appropriate for technical reports with a large number of tables and figures. The `\topfraction` register defines the maximum fraction of the page that can be occupied by floats at the top of the page. Analogously, `\bottomfraction` defines the maximum fraction of the page for floats placed at the bottom of the page and `\textfraction` is the minimum fraction of a page that must be devoted to text. The registers are changed using the `\renewcommand` command, as they are macros. In documents with a large number of floats they can be set to extreme values by means of:

```
\renewcommand\topfraction{.99}
\renewcommand\bottomfraction{.99}
\renewcommand\textfraction{.01}
```

A floating page is only created if the floats occupy at least the fraction defined in the `\floatpagefraction` register. This value does not usually need changing.

L<sup>A</sup>T<sub>E</sub>X also has counters, which are used to set the highest number of floats allowed. `topnumber` and `bottomnumber` set the maximum number of floats at, respectively, the top and bottom of a page, while `\totalnumber` limits the total number of floats allowed on a page. These do not usually need changing unless the document contains a large number of very small floats<sup>5</sup>. The values are set using `\setcounter` command, for example:

```
\setcounter{totalnumber}{17}
```

---

<sup>4</sup>This only applies to documents in two-sided printing style which have the `twoside` parameter specified either explicitly or implicitly. The `cnbwp` declares it implicitly.

<sup>5</sup>In such case it would be worth considering grouping them using the `subfigure` package.

! The current register values can be determined in various ways. Here we will demonstrate the use of the  $\TeX$  commands `\show` and `\showthe`. The former shows the macro definition, so we will use it to determine the value of a *fraction* type register. A *number* type register must be printed using `\showthe`. The following two lines will output the current values of the `\topfraction` and `\topnumber` registers:

```
\show\topfraction
\showthe\value{topnumber}
```

On the terminal and in the log file (we have omitted the blank lines here) we will then get an output like this:

```
> \topfraction=\long macro:
->.7.
1.875 \show\topfraction

> 2.
<recently read> \c@topnumber
1.876 \showthe\value{topnumber}
```

We have thus determined that `\topfraction = 0.7` and `topnumber = 2`. The full stops at the end of the definitions are added by the `\show` and `\showthe` commands.

### 9.3 Wide Tables and Charts

Some tables can be very wide. One way of printing them is to rotate them through  $90^\circ$ . This is done using the rotating package and the `sidewaystable` environment. (Figures can be rotated analogously in the `sidewaysfigure` environment.) Unfortunately, the rotating package rotates tables one way on right-hand pages and the other way on left-hand pages, which is inadmissible from a typographical perspective. To eliminate this problem, the package must be loaded in the preamble in the following way:

```
\usepackage[figuresright]{rotating}
```

An example of a wide table, containing an array of random numbers calculated in Octave by the `5 \times \text{randn}(20, 10)` command, is given in Table 2. The source code can be found in figure 3. The table will be rotated so that the left-hand margin is placed at the bottom margin of the typeset image (i.e. flush with the bottom margin of the text on the other pages). This rarely creates an aesthetic effect. A centred table looks better and can be obtained using the `\centering` command. The caption also needs to be centred by explicitly specifying `\centeredcaption` instead of `\caption`.

Table centering need not always be desirable. The page setup can be altered by shifting the table. Prepare the page without centering the table and print it out or display it in GhostView (this program allows you to make measurements). Let's assume that you want to shift the table 27 mm towards the top margin of the page. This means that you have to widen the left-hand margin of the floating environment by the same amount. This is done by specifying the following at the beginning of the environment:

Table 2: Široká tabulka

#	Col. 1	Col. 2	Col. 3	Col. 4	Col. 5	Col. 6	Col. 7	Col. 8	Col. 9	Col. 10
Row 1	-6.412	-2.654	-5.300	4.358	-6.473	-4.573	2.391	-0.497	-4.262	-0.341
Row 2	2.799	-8.109	5.647	-0.214	4.665	2.971	13.699	-5.059	-0.088	4.090
Row 3	8.427	5.467	-7.061	-0.347	-6.955	6.352	-3.955	7.768	-9.852	4.618
Row 4	-1.978	-1.226	1.136	1.733	3.874	15.072	4.112	-1.931	4.127	1.177
Row 5	10.896	6.859	-0.623	6.685	-6.378	2.714	-2.670	7.862	-2.314	-4.094
Row 6	-5.013	-2.391	-1.763	-1.499	-4.053	2.453	1.550	-3.939	3.366	-0.780
Row 7	8.644	-1.787	-5.782	1.244	-0.806	3.506	1.810	-3.908	-0.626	-1.933
Row 8	-4.965	-2.494	1.539	6.265	0.892	-2.730	3.311	-0.006	3.735	0.408
Row 9	1.237	-3.029	-0.773	9.400	-6.009	-0.487	4.281	4.520	-5.744	-3.628
Row 10	0.051	-8.717	-1.366	1.811	-1.599	-10.179	-1.355	6.024	4.912	0.728
Row 11	1.894	-8.089	-6.445	-9.112	-4.753	2.555	2.751	0.952	-0.291	-1.523
Row 12	-3.103	-0.002	2.733	-9.805	-3.154	-1.985	4.259	-2.340	-2.236	-2.372
Row 13	3.729	1.978	6.627	-9.898	3.746	-3.595	-6.425	10.043	4.578	7.770
Row 14	9.511	-8.231	1.815	-5.189	-1.213	0.767	-2.620	6.613	-1.119	-3.838
Row 15	-0.699	-10.599	5.787	-11.333	-4.810	2.769	0.255	-6.831	-1.643	-2.870
Row 16	7.190	-2.291	7.532	2.650	-5.878	-4.859	7.792	-1.337	-5.075	-7.241
Row 17	-5.918	0.987	5.037	-0.556	-2.653	-7.008	3.491	-1.028	0.573	4.620
Row 18	-3.957	-4.265	1.325	3.102	-5.731	-3.944	-6.565	5.178	2.477	-1.948
Row 19	-1.228	-0.170	-3.048	-2.966	9.791	9.006	9.186	-2.971	8.657	-2.838
Row 20	-2.340	-4.932	-3.904	4.164	-5.838	-7.320	1.451	4.955	7.439	-4.407

Figure 3: Source code for table 2

```

\begin{sidewaystable}[p]
\setlength{\extrarowheight}{2pt}
\newcommand\mcol[1]{\multicolumn{1}{r}{Col. #1}}
\centeredcaption{\v{S}irok'a tabulka\label{widetable}\bigskip\centering}
\begin{tabular}{l*{10}{D{.}{.}{3}}}\hline
\bfseries \# & \mcol{1}& \mcol{2}& \mcol{3}& \mcol{4}&
& \mcol{5}& \mcol{6}& \mcol{7}& \mcol{8}& \mcol{9}& \mcol{10}\\\hline
Row 1 & -6.412 & -2.654 & -5.300 & 4.358 & -6.473 & -4.573 & 2.391 & -0.497 & -4.262 & -0.341\\
Row 2 & 2.799 & -8.109 & 5.647 & -0.214 & 4.665 & 2.971 & 13.699 & -5.059 & -0.088 & 4.090\\
Row 3 & 8.427 & 5.467 & -7.061 & -0.347 & -6.955 & 6.352 & -3.955 & 7.768 & -9.852 & 4.618\\
Row 4 & -1.978 & -1.226 & 1.136 & 1.733 & 3.874 & 15.072 & 4.112 & -1.931 & 4.127 & 1.177\\
Row 5 & 10.896 & 6.859 & -0.623 & 6.685 & -6.378 & 2.714 & -2.670 & 7.862 & -2.314 & -4.094\\
Row 6 & -5.013 & -2.391 & -1.763 & -1.499 & -4.053 & 2.453 & 1.550 & -3.939 & 3.366 & -0.780\\
Row 7 & 8.644 & -1.787 & -5.782 & 1.244 & -0.806 & 3.506 & 1.810 & -3.908 & -0.626 & -1.933\\
Row 8 & -4.965 & -2.494 & 1.539 & 6.265 & 0.892 & -2.730 & 3.311 & -0.006 & 3.735 & 0.408\\
Row 9 & 1.237 & -3.029 & -0.773 & 9.400 & -6.009 & -0.487 & 4.281 & 4.520 & -5.744 & -3.628\\
Row 10 & 0.051 & -8.717 & -1.366 & 1.811 & -1.599 & -10.179 & -1.355 & 6.024 & 4.912 & 0.728\\
Row 11 & 1.894 & -8.089 & -6.445 & -9.112 & -4.753 & 2.555 & 2.751 & 0.952 & -0.291 & -1.523\\
Row 12 & -3.103 & -0.002 & 2.733 & -9.805 & -3.154 & -1.985 & 4.259 & -2.340 & -2.236 & -2.372\\
Row 13 & 3.729 & 1.978 & 6.627 & -9.898 & 3.746 & -3.595 & -6.425 & 10.043 & 4.578 & 7.770\\
Row 14 & 9.511 & -8.231 & 1.815 & -5.189 & -1.213 & 0.767 & -2.620 & 6.613 & -1.119 & -3.838\\
Row 15 & -0.699 & -10.599 & 5.787 & -11.333 & -4.810 & 2.769 & 0.255 & -6.831 & -1.643 & -2.870\\
Row 16 & 7.190 & -2.291 & 7.532 & 2.650 & -5.878 & -4.859 & 7.792 & -1.337 & -5.075 & -7.241\\
Row 17 & -5.918 & 0.987 & 5.037 & -0.556 & -2.653 & -7.008 & 3.491 & -1.028 & 0.573 & 4.620\\
Row 18 & -3.957 & -4.265 & 1.325 & 3.102 & -5.731 & -3.944 & -6.565 & 5.178 & 2.477 & -1.948\\
Row 19 & -1.228 & -0.170 & -3.048 & -2.966 & 9.791 & 9.006 & 9.186 & -2.971 & 8.657 & -2.838\\
Row 20 & -2.340 & -4.932 & -3.904 & 4.164 & -5.838 & -7.320 & 1.451 & 4.955 & 7.439 & -4.407\\
\hline
\end{tabular}
\end{sidewaystable}

```

```
\begin{sidewaystable}[p]% beginning of rotated table
\setlength{\leftskip}{27mm}
```

This command will shift the table and the standard caption, although it will not work correctly when `\centeredcaption` is used.

! Note that the `\label` command must be written inside the floating environment to which it relates and after the `\caption` macro or the alternative thereto defined in the `cnbwp` class. If it is written outside this environment or before the `\caption` macro, it will relate to the `\section` or `\subsection` in which it occurs.

## 10 Generating the Final Document in PDF Format

Czech National Bank Working Papers are published in PDF format. We will therefore focus on the most common methods of generating a PDF file from a text written in  $\LaTeX$ .

The first option is to generate the PDF document directly, using `pdf $\TeX$` . If you cannot, or do not want to, use this, you can convert the DVI output file to PostScript using `dvips`. The postscript file can then be converted to PDF using a software package (commercial and freely distributable products are both available). See also section 7.4.

Scientific Word does not contain `dvips`. The PDF can thus only be generated using a virtual printer. Adobe Distiller tends to produce better results than PDFWriter, although for short documents the difference is not usually significant. As Scientific Word generates a  $\LaTeX$  file, this file can also be transferred onto a computer with the full  $\TeX$  installation and the PDF can be generated in the usual way using `pdf $\TeX$`  or via PostScript and conversion to PDF. Figures in WMF format have to be converted to EPS using `wmf2eps`, as mentioned in section 7.4.1.

Another option is to generate the PDF from the DVI using `dvipdfm`, which is part of the MikTeX distribution, for example.

To print the document from PDF you may need to adjust the margin settings. The drivers of some physical printers ignore the margins set in the file and shift the text on the paper. Adjusting margin settings is described in section `/refpreamble`.

## 11 Submitting Your Document for Editing and Publication

When preparing your document for submission, you should bear in mind that the editor is not necessarily a  $\LaTeX$  expert and might not even have any  $\TeX$  distribution installed. You should therefore submit a file that can be printed out using standard, easy-to-install software, hence preferably in PDF format or, as a last resort, as a PostScript file.

By agreement with our editor, CNB research publications are currently submitted for editing in two formats: PDF and the ( $\LaTeX$ ) source text saved as an MS Word document. The editor opens the source text in MS Word and edits it with Track Changes enabled, using the PDF version for guidance.

The CNB editor has been told to emend linguistic phenomena only. He does not alter formatting commands that start with a backslash, including their arguments in curly

or square brackets. In other words, he only corrects the words he sees in the printed version. For example, he also leaves apostrophes and hyphens unchanged. However, it is the author's responsibility to further process the edited text and either accept or reject each change. We do not recommend accepting all the changes in one step. Finally, the edited file is saved as plain text with the extension `.tex` and can be processed using the  $\text{\LaTeX}$  system.

The main document can load other files using the `\input` or `\include` commands. The name of the file to be loaded is written in a parameter of this macro. The `.tex` extension need not be given. However, keep these commands to a minimum when writing documents for CNB WP purposes. Excessive use reduces clarity and leads to complications during editing.

Do not use letters with diacritics. There are too many codings for Czech and Slovak, and different  $\text{\TeX}$  distributions treat these codings differently. You should assume that your document will be worked on by a somewhat informed layman who may not be able to cope with recoding. So, always write diacritics using  $\text{\TeX}$  sequences.

All the files needed for typesetting the documents should be in the same directory or (in exceptional cases) in subdirectories of the directory containing the main document. The command

```
\includegraphics{d:/projects/figures/image1.eps}
```


is bad for three reasons:

1. The author of the document might forget that a vital file is in a different directory and will forget to bundle it.
2. The person processing the document might have different disk partitioning and the command will return a File Not Found message.
3. For security reasons, some  $\text{\TeX}$  installations do not permit absolute paths or access to the parent directory.

Unix users should resist the temptation to use symbolic links in such cases. For one thing, it can easily occur that only the symbolic links, and not the files, are bundled with the document. In addition, the document has to be processable on a file system which does not support symbolic links. If you use subdirectories, enter the names relatively, for example:

```
\includegraphics{figures/image1.eps}
```

Always consider, however, whether it is really necessary to create subdirectories. For the uninitiated, it is usually clearer if all related files are saved in the same directory.

 Do not use file names that contain spaces and/or diacritics. The  $\text{\TeX}$ book states that a space terminates a file name. Given the popularity of spaces with Microsoft, there are distributions around which can cope with this. It is, however, a **very non-standard distribution** and may not even work on the same operating system with a different  $\text{\TeX}$  distribution. Directory and file names containing diacritics are an even worse problem. Whether they work depends on too many factors – not only the  $\text{\TeX}$  distribution, but

also the locales settings in the operating system. Such names should be avoided, as they are highly likely to cause problems.

All the necessary files must be packed into a single archive with a directory structure. ZIP is the best format. Add a short explanatory text in a file named `readme.txt`, especially if the document comprises multiple files.

$\TeX$  was created at a time when texts were entered into computers on punch cards. Consequently, it does not matter how the text is divided into lines and how many spaces there are between words. The important thing is to leave a blank line between paragraphs.

As an interesting alternative a PDF file may directly be used for carrying out the proof. Commenting must be enabled by full version of Adobe Acrobat and the proof can then be done by Acrobat Reader.

## 12 Sample Files

This manual includes a number of sample files. The main document template is in the file `cnbpaper.tex` and the entire document has been converted into PDF format (`cnbpaper.pdf`) using  $\text{pdf}\mathcal{L}\mathcal{A}\mathcal{T}\mathcal{E}\mathcal{X}$ . However, the file can also be processed using the standard  $\mathcal{L}\mathcal{A}\mathcal{T}\mathcal{E}\mathcal{X}$  and the `dvips` program.

Some examples are saved in separate files, partly to ensure that they are consistent with this manual, and partly so that users do not have to work with a lengthy sample document when writing their texts.

Table 1, demonstrating decimal point alignment, is located in `numtable.tex`. Neither the `\caption` command nor a floating environment is specified in this file. These commands can be found in the main document template.

The wide table 2 can be found in `widematrix.tex`. In this case the file also contains environment definition commands.

The files `graf18.eps` and `graf18.pdf` are the vector charts from figure 1. The template uses the `ifpdf` package to determine whether  $\text{pdf}\mathcal{L}\mathcal{A}\mathcal{T}\mathcal{E}\mathcal{X}$  is being used, and the file will accordingly be loaded in the correct format. If `ifpdf` is not contained in the distribution, it will be assumed that  $\text{pdf}\mathcal{L}\mathcal{A}\mathcal{T}\mathcal{E}\mathcal{X}$  is unavailable.

The file `biblio.tex` contains a list of references written using the macros from section 5.2. The full list is given in the Appendix.

The file `cnbsample.bib` contains the list of references for use with  $\text{Bib}\mathcal{T}\mathcal{E}\mathcal{X}$ . It was also used to generate the above mentioned file `biblio.tex`.

If you intend to use `cnbpaper.tex` as the template for your document, think about which packages you really need. Delete any unnecessary `\usepackage` commands. You may need some additional packages, such as `amsmath`.

## 13 Changes, Version 2013.12

The following changes were made in December, 2013:

1. Switches `11pt` and `12pt` for changing the font size were removed.
2. The syntax of the `\author` macro was changed.



3. Macros `\shortauthor` and `\shorttitle` were removed.
4. Usage of the `\acknowledge` macro was changed.
5. The `abstrakt` environment for entering the Czech abstract was added.
6. The `babel` package is loaded in order to activate the Czech hyphenation rules in the `abstrakt` environment.
7. Captions formatting modified to comply with the new requirements.
8. New macros `\Note` and `\Source` were implemented.
9. Formatting of the author names in the `abbrvcnb` style was modified.
10. The `times` package replaced with the `mathptmx`.
11. The manual was updated.

## A Appendix

The annex gives examples of all types of works written using the macros from section 5.2 and the subsections thereof. You will not normally use the `note` field.

The entire set of examples is also available in `biblio.tex`.

```
% Requires cnbwp version 2005/12/23 or newer
```

```
\begin{thebibliography}{29}
\manualItem[man()]{manual-minimal}{
  title = {The Definitive Computer Manual},
  note = {This is a minimal MANUAL entry},
}

\proceedingsItem[pro(1983)]{proceedings-minimal}{
  title = {Proc. Fifteenth Annual Symposium on the Theory of Computing},
  year = {1983},
}

\articleItem[Aamport(1986b)]{article-full}{
  author = {Aamport, L.~A.},
  title = {The Gnats and Gnus Document Preparation System},
  journal = {\mbox{G-Animal's} Journal},
  year = {1986b},
  volume = {41},
  number = {7},
  pages = {73+},
  month = {Jul},
  note = {This is a full ARTICLE entry},
}

\articleItem[Aamport(1986a)]{article-minimal}{
  author = {Aamport, L.~A.},
```

```

    title = {The Gnats and Gnu Document Preparation System},
    journal = {\mbox{G-Animal's} Journal},
    year = {1986a},
}

```

```

\bookItem[Knuth(1981c)]{book-full}{
    author = {Knuth, D.~E.},
    title = {Seminumerical Algorithms},
    publisher = {Addison-Wesley},
    year = {1981c},
    volume = {2},
    series = {The Art of Computer Programming},
    address = {{Reading, Massachusetts}},
    edition = {second},
    month = {10~Jan},
    note = {This is a full BOOK entry},
}

```

```

\bookItem[Knuth(1981)]{book-minimal}{
    author = {Knuth, D.~E.},
    title = {Seminumerical Algorithms},
    publisher = {Addison-Wesley},
    year = {1981},
}

```

```

\inbookItem[Knuth(1973a)]{inbook-full}{
    author = {Knuth, D.~E.},
    title = {Fundamental Algorithms},
    chapter = {1.2},
    pages = {10--119},
    publisher = {Addison-Wesley},
    year = {1973a},
    volume = {1},
    series = {The Art of Computer Programming},
    type = {section},
    address = {{Reading, Massachusetts}},
    edition = {second},
    month = {10~Jan},
    note = {This is a full INBOOK entry},
}

```

```

\inbookItem[Knuth(1973)]{inbook-minimal}{
    author = {Knuth, D.~E.},
    title = {Fundamental Algorithms},
    chapter = {1.2},
    publisher = {Addison-Wesley},
    year = {1973},
}

```

```

\bookletItem[Knvth(1988)]{booklet-full}{
  title = {The Programming of Computer Art},
  author = {Knvth, J.~C.},
  howpublished = {Vernier Art Center},
  address = {{Stanford, California}},
  month = {Feb},
  year = {1988},
  note = {This is a full BOOKLET entry},
}

\incollectionItem[Lincoll(1977b)]{incollection-full}{
  author = {Lincoll, D.~D.},
  title = {Semigroups of Recurrences},
  booktitle = {High Speed Computer and Algorithm Organization},
  publisher = {Academic Press},
  year = {1977b},
  editor = {Lipcoll, D.~J., D.~H. Lawrie, and A.~H. Sameh},
  number = {23},
  series = {Fast Computers},
  type = {part},
  chapter = {3},
  pages = {179--183},
  address = {New York},
  edition = {third},
  month = {Sep},
  note = {This is a full INCOLLECTION entry},
}

\incollectionItem[Lincoll(1977a)]{incollection-minimal}{
  author = {Lincoll, D.~D.},
  title = {Semigroups of Recurrences},
  booktitle = {High Speed Computer and Algorithm Organization},
  publisher = {Academic Press},
  year = {1977a},
}

\bookItem[Lipcoll et~al.(1977c)Lipcoll, Lawrie, and Sameh]{whole-collection}{
  editor = {Lipcoll, D.~J., D.~H. Lawrie, and A.~H. Sameh},
  title = {High Speed Computer and Algorithm Organization},
  publisher = {Academic Press},
  year = {1977c},
  number = {23},
  series = {Fast Computers},
  address = {New York},
  edition = {third},
  month = {Sep},
  note = {This is a cross-referenced BOOK (collection) entry},
}

```

}

```
\manualItem[Manmaker(1986)]{manual-full}{  
  title = {The Definitive Computer Manual},  
  author = {Manmaker, L.},  
  organization = {Chips-R-Us},  
  address = {Silicon Valley},  
  edition = {silver},  
  month = {Apr-May},  
  year = {1986},  
  note = {This is a full MANUAL entry},  
}
```

```
\manualItem[Manmaker(1984)]{manual-question}{  
  title = {Are Computer Manuals Useful?},  
  author = {Manmaker, L.},  
  organization = {Chips-R-Us},  
  year = {1984},  
}
```

```
\mastersthesisItem[Masterly(1988b)]{mastersthesis-full}{  
  author = {Masterly, {'E}.},  
  title = {Mastering Thesis Writing},  
  school = {Stanford University},  
  year = {1988b},  
  type = {Master's project},  
  address = {English Department},  
  month = {Jun-Aug},  
  note = {This is a full MASTERSTHESIS entry},  
}
```

```
\mastersthesisItem[Masterly(1988a)]{mastersthesis-minimal}{  
  author = {Masterly, {'E}.},  
  title = {Mastering Thesis Writing},  
  school = {Stanford University},  
  year = {1988a},  
}
```

```
\miscItem[Missilany(1984)]{misc-full}{  
  author = {Missilany, J.-B.},  
  title = {Handing out random pamphlets in airports},  
  howpublished = {Handed out at O'Hare},  
  month = {Oct},  
  year = {1984},  
  note = {This is a full MISC entry},  
}
```

```
\inproceedingsItem[Oaho et~al.(1983a)Oaho, Ullman, and
```

```

Yannakakis]{inproceedings-minimal}{
author = {Oaho, A.~V., J.~D. Ullman, and M.~Yannakakis},
title = {On Notions of Information Transfer in VLSI Circuits},
booktitle = {Proc. Fifteenth Annual Symposium on the Theory of Computing},
year = {1983a},
}

```

```

\inproceedingsItem[Oaho et~al.(1983b)Oaho, Ullman, and
Yannakakis]{inproceedings-full}{
author = {Oaho, A.~V., J.~D. Ullman, and M.~Yannakakis},
title = {On Notions of Information Transfer in VLSI Circuits},
booktitle = {Proc. Fifteenth Annual Symposium on the Theory of Computing},
year = {1983b},
editor = {Oz, W.~V. and M.~Yannakakis},
number = {17},
series = {All ACM Conferences},
pages = {133--139},
address = {Boston},
month = {Mar},
organization = {The OX Association for Computing Machinery},
publisher = {Academic Press},
note = {This is a full INPROCEEDINGS entry},
}

```

```

\unpublishedItem[Overwood et~al.()Overwood, {\~N}et, and
\={P}ot]{unpublished-minimal2}{
author = {Overwood, U., N.~{\~N}et, and P.~\={P}ot},
title = {Lower Bounds for Wishful Research Results},
}

```

```

\unpublishedItem[Overwood et~al.(1988)Overwood, {\~N}et, and
\={P}ot]{unpublished-full2}{
author = {Overwood, U., N.~{\~N}et, and P.~\={P}ot},
title = {Lower Bounds for Wishful Research Results},
month = {{Nov, Dec}},
year = {1988},
}

```

```

\proceedingsItem[Oz and Yannakakis(1983)]{proceedings-full}{
title = {Proc. Fifteenth Annual Symposium on the Theory of Computing},
year = {1983},
editor = {Oz, W.~V. and M.~Yannakakis},
number = {17},
series = {All ACM Conferences},
address = {Boston},
month = {Mar},
organization = {The OX Association for Computing Machinery},
publisher = {Academic Press},
}

```

```

    note = {This is a full PROCEEDINGS entry},
}

\phdthesisItem[Phony-Baloney(1988b)]{phdthesis-full}{
    author = {Phony-Baloney, F.~P.},
    title = {Fighting Fire with Fire: Festooning French Phrases},
    school = {Fanstord University},
    year = {1988b},
    type = {PhD Dissertation},
    address = {Department of French},
    month = {Jun-Aug},
    note = {This is a full PHDTHESIS entry},
}

\phdthesisItem[Phony-Baloney(1988a)]{phdthesis-minimal}{
    author = {Phony-Baloney, F.~P.},
    title = {Fighting Fire with Fire: Festooning French Phrases},
    school = {Fanstord University},
    year = {1988a},
}

\techreportItem[T'\{e}rrific(1988)]{techreport-full}{
    author = {T'\{e}rrific, T.},
    title = {An  $\Theta(n \log n / \log \log n)$  Sorting Algorithm},
    institution = {Fanstord University},
    year = {1988},
    type = {Wishful Research Result},
    number = {7},
    address = {{Computer Science Department, Fanstord, California}},
    month = {Oct},
    note = {This is a full TECHREPORT entry},
}

\techreportItem[Terrific(1988)]{techreport-minimal}{
    author = {Terrific, T.},
    title = {An  $\Theta(n \log n / \log \log n)$  Sorting Algorithm},
    institution = {Fanstord University},
    year = {1988},
}

\unpublishedItem["{U}nderwood et~al.()\{U}nderwood, {\~N}et, and
    \={P}ot]{unpublished-minimal}{
    author = {"{U}nderwood, U., N.~{\~N}et, and P.~\={P}ot},
    title = {Lower Bounds for Wishful Research Results},
    note = {Talk at Fanstord University (this is a minimal UNPUBLISHED entry)},
}

\unpublishedItem["{U}nderwood et~al.(1988)\{U}nderwood, {\~N}et, and

```

```

\={P}ot]{unpublished-full}{
author = {"{U}nderwood, U., N.~{\~N}et, and P.~\={P}ot},
title = {Lower Bounds for Wishful Research Results},
note = {Talk at Fanstord University (this is a full UNPUBLISHED entry)},
month = {{Nov, Dec}},
year = {1988},
}

\miscItem[Wagner()]{url-demo}{
author = {Wagner, Z.},
title = {Internet Home Page},
note = {\url{http://icebearsoft.euweb.cz}},
}

\end{thebibliography}

```

## B Index

<b>Symbols</b>	Bib <sub>T</sub> E <sub>X</sub> . . . . . 11	\documentclass . . . . . 6, 7
\- . . . . . 14	\bookItem . . . . . 13	dvipdfm . . . . . 30
<b>A</b>	\bookletItem . . . . . 13	dvips . . . . . 30
a4-portrait.pdf . . . . . 8	booktitle . . . . . 14	<b>E</b>
abbrvcnb . . . . . 11	\bottomfraction . . . . . 26	edition . . . . . 14
abbrvnat . . . . . 11	bottomnumber . . . . . 26	editor . . . . . 14
abstract . . . . . 9	BoundingBox . . . . . 23	Encapsulated PostScript 18
abstrakt . . . . . 9	<b>C</b>	EPS . . . . . 17, 18, 20–23, 30
\acknowledge . . . . . 9	\caption . . . . . 10, 27	EPS Clip . . . . . 23
Acrobat . . . . . 21–23, 32	\centeredcaption 10, 27, 30	epstopdf . . . . . 17, 22
Acrobat Reader . . . . . 32	centeredcaptions . . . . . 6	\extrarowheight . . . . . 25
address . . . . . 14	\centering . . . . . 27	<b>F</b>
Adobe Acrobat . . . . . 21–23, 32	chapter . . . . . 14, 15	figure . . . . . 16, 25
Adobe Distiller . . . . . 17, 22	\cite . . . . . 12	float . . . . . 26
Adobe Illustrator . . . . . 22	\cleardoublepage . . . . . 26	\floatpagefraction . . . . . 26
AI . . . . . 22	\clearpage . . . . . 26	<b>G</b>
amsmath . . . . . 32	clip . . . . . 17, 23	ggv . . . . . 23
array . . . . . 24, 25	cnbpaper.pdf . . . . . 32	GhostScript . . . . . 17, 22, 23
\articleItem . . . . . 12	cnbpaper.tex . . . . . 32	GhostView . . . . . 23, 27
author . . . . . 14	cnbsample.bib . . . . . 32	Gimp . . . . . 20
\author . . . . . 8	\conferenceItem . . . . . 13	gnuplot . . . . . 22
<b>B</b>	Corel Draw . . . . . 22	graf18.eps . . . . . 32
\bibitem . . . . . 12, 13	<b>D</b>	graf18.pdf . . . . . 32
biblio.tex . . . . . 32, 33	dcolumn . . . . . 24	graphicx . . . . . 16
\bibliography . . . . . 11	\DeclareRobustCommand . 8	grfguide.ps . . . . . 16
\bibliographystyle . . . . . 11	Distiller . . . . . 17, 22	

<b>H</b>		<code>\miscItem</code> . . . . . 13	<b>S</b>	
<code>[H]</code> . . . . . 26		month . . . . . 15	<code>scale</code> . . . . . 16	
<code>[h]</code> . . . . . 26		<code>\multicolumn</code> . . . . . 24	<code>school</code> . . . . . 15	
height . . . . . 17			Scientific Word . . . . . 21, 30	
here . . . . . 26		<b>N</b>		<code>\section</code> . . . . . 10
hierarchicalnumbering . . . . . 6		<code>natbib</code> . . . . . 6, 10–12	<code>series</code> . . . . . 15	
<code>\hoffset</code> . . . . . 8		<code>\newcolumntype</code> . . . . . 24	<code>\setcounter</code> . . . . . 26	
howpublished . . . . . 14		<code>\newcommand</code> . . . . . 8	<code>\shortauthor</code> . . . . . 33	
HPGL . . . . . 21		<code>\newpage</code> . . . . . 26	<code>\shorttitle</code> . . . . . 33	
hyperref . . . . . 7		nontechsummary . . . . . 10	<code>\show</code> . . . . . 27	
		<code>\Note</code> . . . . . 10	<code>\showthe</code> . . . . . 27	
<b>I</b>		note . . . . . 11, 15, 16	<code>sidewaysfigure</code> . . . . . 27	
<code>ifpdf</code> . . . . . 6, 32		number . . . . . 15	<code>sidewaystable</code> . . . . . 27	
Illustrator . . . . . 22		numbered . . . . . 7	<code>simplenumbering</code> . . . . . 6	
<code>\inbookItem</code> . . . . . 13, 15		numbered . . . . . 11	<code>\Source</code> . . . . . 10	
<code>\includegraphics</code> . . . . . 16, 23		<code>numtable.tex</code> . . . . . 32	<code>\special</code> . . . . . 16	
<code>\incollectionItem</code> . . . . . 13, 15			<code>\standardcaption</code> . . . . . 10	
<code>\inproceedingsItem</code> . . . . . 13		<b>O</b>		<code>standardcaptions</code> . . . . . 6
installation . . . . . 3		organization . . . . . 15	<code>subfigure</code> . . . . . 26	
institution . . . . . 15			<code>\subparagraph</code> . . . . . 10	
<b>J</b>		<b>P</b>		<code>\subsection</code> . . . . . 10
<code>\JEL</code> . . . . . 9		page . . . . . 17	<code>\subsubsection</code> . . . . . 10	
<code>jff</code> . . . . . 18		pages . . . . . 15		
JFIF . . . . . 17, 18		<code>\paragraph</code> . . . . . 10	<b>T</b>	
<code>jfif</code> . . . . . 18		path . . . . . 8	<code>table</code> . . . . . 16, 24, 25	
Joint Photographic Expert Group . . . . . 18		PCL . . . . . 21	<code>tabular</code> . . . . . 24	
journal . . . . . 15		PDF . . . . . 17, 20–23, 30	Tagged Image File Format . . . . . 20	
JPEG . . . . . 18		<code>pdfTEX</code> . . . . . 30	<code>\techreportItem</code> . . . . . 13	
<code>jpeg</code> . . . . . 18		<code>pdftops</code> . . . . . 22	<code>\textfraction</code> . . . . . 26	
JPEG File Interchange For- mat . . . . . 18		PDFWriter . . . . . 30	<code>thebibliography</code> . . . . . 11, 14	
<code>jpg</code> . . . . . 18		perl . . . . . 17	<code>tif</code> . . . . . 20	
		<code>\phdthesisItem</code> . . . . . 13, 16	TIFF . . . . . 20	
<b>K</b>		Photoshop . . . . . 20	title . . . . . 15	
<code>keyval</code> . . . . . 6, 12, 16		PJL . . . . . 21, 23	<code>\title</code> . . . . . 8	
<code>\Keywords</code> . . . . . 9, 10		PNG . . . . . 17, 18	<code>\topfraction</code> . . . . . 26	
<code>kpsewhich</code> . . . . . 4		Portable Network Graph- ics . . . . . 18	<code>topnumber</code> . . . . . 26	
		PostScript . . . . . 20	<code>\totalnumber</code> . . . . . 26	
<b>L</b>		PostScript . . . . . 20, 21, 30	<code>twoside</code> . . . . . 26	
<code>\label</code> . . . . . 30		preamble . . . . . 6	<code>type</code> . . . . . 15	
		<code>\proceedingsItem</code> . . . . . 13	<code>\typItem</code> . . . . . 2, 11–13, 16	
<b>M</b>		<code>\protect</code> . . . . . 8		
<code>\maketitle</code> . . . . . 8, 9		<code>ps2pdf</code> . . . . . 22	<b>U</b>	
<code>\manualItem</code> . . . . . 13		<code>publisher</code> . . . . . 15	<code>\unpublishedItem</code> . . . . . 13, 16	
<code>\mastersthesisItem</code> . . . . . 13, 16			<code>\url</code> . . . . . 7, 11	
<code>\mbox</code> . . . . . 14		<b>R</b>		<code>url</code> . . . . . 6–8, 11, 15
		<code>\renewcommand</code> . . . . . 26	URLs . . . . . 11, 15	
		rotating . . . . . 27	<code>\usepackage</code> . . . . . 7	



<b>V</b>		<b>Y</b>
\voffset .....	8	year .....
volume .....	15	15
<b>W</b>		
widematrix.tex .....	32	
width .....	17	
Windows Metafile .....	21	
WMF .....	21, 30	
wmf2eps .....	21, 30	