Plain TEX BookMacros

Version 0.0.0

Contents

List of figures

List of tables

1 Introduction

Typesetting a book document with Knuth `tex` (*not* `etex` or `pdftex`) using *only* the plain TEX format is not recommended. The *TEXbook*, for example, is set with the `manmac` macros to avoid tedious repetitions of low level commands and to achieve a consistent layout. The intention of the *BookMacros* is to provide the minimum set of macros and tools to create a book document using traditional Knuth `tex` with the plain TEX format. The goal had been to use as much as possible from plain TEX directly and add only as little as necessary to keep writing a document manageable and consistent. Of course an author is encouraged to create macros for each style markup element of a document. The following features are supported:

- Automatic table of contents and list of figures and tables generation
- A set of headlines
- Displays
- Item, enumeration, and description lists

- Table and figure captions
- Footnote numbering
- Cross references and index

Still there is much work to do, but currently work on this package is stopped. Next steps would have been:

- Page header, roman page numbers in front and back matter, two-sided documents
- Additional fonts
- Using MakeIndex instead of the current solution
  (It is not hard to put the main features of MakeIndex into a script—but then why not *use* MakeIndex . . .)
- Use of `\pdfximage` instead of `\epsfbox` when `pdftex` instead of `tex` is used

2 Elements to by taken from plain TEX

Some features for typesetting a document are already supported by plain TEX in a form that allows direct use in a document. In short everything that is not mentioned in later chapters needs to be taken from plain TEX. Please refer to [1] for a thorough description of plain TEX.

2.1 Title page

Since the style of the initial pages of a book may vary significantly there is no macro support for everything before the table of contents.

2.2 Type area

The default plain TEX type area seems to be based on the letter format. To achieve other formats the dimensions `\hsize` and/or `\vsize` needs to be adjusted. This document uses

```
\advance\hsize by-6mm\advance\vsize by18mm
```

for format A4. This must be done *before* the macro package is included since it stores these values.

2.3 Special characters

	`\thinspace` (distance: ‖)
`	`\`{}`
´	`\'{}`
˜	`\~{}`
•	`$\bullet$`
†	`\dag`
‡	`\ddag`
§	`\S`
¶	`\P`
. . .	`\dots`
“⟨*text*⟩”	``⟨*text*⟩''
£	`{\it\$}`
$	`\$`
%	`\%`
&	`\&`
#	`\#`
_	`\_`
⟨	`$\langle$`
⟩	`$\rangle$`
{	`$\{$`
}	`$\}$`
|	`$\|$`
‘	`\lq` (Left single quote. Can be used for a ligature with itself to produce a left double quote.)
’	`\rq`
ä	`\"a` (In general an *umlaut* is generated by preceding a letter with `\"`)
ß	`\ss`

Only in `\tt` available special characters

`<` `<`
`>` `>`
`|` `|`
`"` `"`
␣ `\char`\␣` (visible space)

2.4 Tables

Tables needs to set as described in the TEXbook, unfortunately. Only table captions are supported by this package, please refer to chapter 9 (Table captions) on page 7.

2.5 Keeps

Keeps are achieved by enclosing something with `\vbox{...}`.

2.6 Floats

Floats are created with

`\topinsert`⟨*vertical mode material*⟩`\endinsert`

([1, p. 115]). If `\midinsert` is used instead of `\topinsert`, the float is placed at the current vertical position if possible ([1, p. 116]). Else it has the same effect as `\topinsert`.

2.7 Minipages

If a narrow text area is needed (e. g. in a table) it can be produced with `\vtop` and a local change of `\hsize`. If two such text areas should be placed side by side, they can be put in a `\hbox` separated a `\hskip`:

```
\hbox{%
 \vtop{\hsize=.5\hsize\advance\hsize by-.5em
  ⟨text ...⟩}%
 \hskip1em%
 \vtop{\hsize=.5\hsize\advance\hsize by-.5em
  ⟨text ...⟩}%
}
```

2.8 Bibliography

The plain TEX macro package `btxmac` provides the functionality to produce a bibliography with a *BibTeX* database. The documentation of `btxmac` is contained in it's source code file and is similar to the usage in LATEX. To use the package it needs to be included with

`\input btxmac`

at begin of the document. A reference from the database is cited with

`\cite[`⟨*optional text*⟩`]{`⟨*database reference*⟩`}`

e. g.

`\cite[p. 116]{knuth:ct:a}`

The bibliography is included with

```
\nocite{*}
\bibliographystyle{plain}
\bibliography{⟨database name⟩}
```

The generated `aux` file needs to be processed with the tool `bibtex`.

3 Creating new documents

For each document a directory must be created first:

`mkdir -p` ⟨*directory path*⟩
`cd` ⟨*directory path*⟩

Then a symlink to the BookMacro directory is created:

`ln -s` ⟨*path to BookMacro directory*⟩

and a symlink to the file `configure` inside the `BookMacro` link is created:

```
ln -s BookMacro/configure
```

The *makefile* assumes that the file names without extension of the input and output file are the same. The default input file name is `doc.tex`, the default output name is `doc.dvi`. A different name can be configured by creating a file `cfg.mk` which contains the line

`NAM =` ⟨*name*⟩

Now a makefile needs to be generated with '`./configure`'. The document is then generated with '`make`'. '`make clean`' removes all generated files except the makefile and the output document. With the target `realclean` really all generated files are removed.

If a specific hyphenation language should be used, some more settings need to be done in `cfg.mk`.

```
TEX = etex
```

sets the tool (instead of traditional Knuth `tex`) which is used for other hyphenation patterns.

```
FTM = bplain
```

sets the required `bplain` format.

```
TEXFLAGS = -fmt $(FMT)
```

specifies the necessary tool options.

The language is selected with the macro

`\Lang{`⟨*language*⟩`}`

e. g.

```
\Lang{ngerman}
```

Document structure

At the beginning the macros needs to be included:

```
\input BookMacros/BookMacros
```

If cross references are used, it's data needs to be included next:

```
\input xref
```

The title page has to be created manually. A simple approach (for a title without title page) is to type

`\centerline{\seventeenss` ⟨*title*⟩`}`

The table of contents is inserted by the command

```
\Toc
```

Chapters are added with `\Chapt`, sections with `\Sec`, paragraph headings with `\Par`:

`\Chapt{`⟨*title*⟩`}`
`\Sec{`⟨*title*⟩`}`
`\Par{`⟨*title*⟩`}`

If an index is used it is included with the following statements:

```
\UChapt{Index}
\input idx
```

The final statement of each document is

```
\bye
```

4 Special characters

„⟨*text*⟩“ `\Bq` ⟨*text*⟩`` (german quotation marks)

Only in `\tt` available special characters

`{` `\lC` (left curly bracket)
`}` `\rC` (right curly bracket)
`'` `\AQ` (apostrophe quote)
`\` `\RS` (reverse slash)

5 Fonts

Table 5.1 shows additional fonts available in the BookMacros.

Tbl. 5.1. Additional fonts

Name	Description	Size
`\seventeenss`	Sans serif 17	17
`\twelvess`	Sans serif 12	12
`\tenssbx`	**Bold sans serif 10**	10

6 Headings and Table of Contents

Chapter headings

`\FChapt{`⟨*title*⟩`}`
: Unnumbered chapter without TOC entry intended for the preamble of a document. This macro is used by `\Toc`.

`\Chapt{`⟨*title*⟩`}`
: Normal chapter heading. The chapter heading font is defined by the macro `\ChaptFont`.

`\UChapt{`⟨*title*⟩`}`
: Unnumbered chapter with TOC entry intended for the appendix of a document.

Section headings

`\Sec{`⟨*title*⟩`}`
: Normal section heading. The section heading font is defined by the macro `\SecFont`.

`\USec{`⟨*title*⟩`}`
: Unnumbered section without TOC entry.

Paragraph heading

`\Par{`⟨*title*⟩`}`
: Paragraph heading.

Table of Contents

The table of contents is output with the

```
\Toc
```

command. It precedes a heading with the default name “Contents”. This name can be changed by redefining `\TocNam`.

List of tables

A list of tables can be output with the

```
\Lot
```

command. It precedes a heading with the default name “List of tables”. This name can be changed by redefining `\LotNam`.

List of figures

A list of figures can be output with the

```
\Lof
```

command. It precedes a heading with the default name "List of figures". This name can be changed by redefining `\LofNam`.

7 Displays

Displays draw text with one line of output for each input line indented on the left side. Display neither have the keep nor the float function. For keeps `\vbox` is required, for floats e.g. `\midinsert` can be used. Input

```
\BeginDisp
\vbox{% only required for keep function
First line
second line
}%
\End
```

yields

First line
second line

If the display should be set in typewriter text `\tt` is simply appended to `\BeginDisp` (`\BeginDisp` and `\End` form a group):

```
\BeginDisp\tt
...
\End
```

Displays and the following text don't have an indentation of the first line.

For indentation of adjusted text at the left and right side the macros `\BeginNarrow` and `\End` are available:

```
\BeginNarrow
...
\End
```

`\BeginNarrow` and `\End` form a group. Font changes may simply be appended to `\BeginNarrow`.

Indented text blocks and the following text don't have an indentation of the first line.

8 Lists

8.1 Item Lists

An item list has the following structure:

```
\BeginItem
\Item ⟨text⟩
\Paragr ⟨text⟩
\Item ⟨text⟩
\End
```

`\Paragr` starts a new paragraph within an item. Example:

- Item 1
  - Item 1.1
    - Item 1.1.1
    - Item 1.1.2
  - Item 1.2
  - Item 1.3
- Item 2

  Paragr
- Item 3

```
\BeginItem
\Item Item 1
 \BeginItem
 \Item Item 1.1
  \BeginItem
  \Item Item 1.1.1
  \Item Item 1.1.2
  \End
 \Item Item 1.2
 \End
\Item Item 2
\Paragr Paragr
\Item Item 3
\End
```

8.2 Enum Lists

An enum list has the following structure:

```
\BeginEnum
\Item ⟨text⟩
\Paragr ⟨text⟩
\Item ⟨text⟩
\End
```

`\Paragr` starts a new paragraph within an item. Example:

1. Item 1
   a) Item 1.1
      i. Item 1.1.1
      ii. Item 1.1.2
   b) Item 1.2
   c) Item 1.3
2. Item 2

   Paragr
3. Item 3

```
\BeginEnum
\Item Item 1
 \BeginEnum
 \Item Item 1.1
  \BeginEnum
  \Item Item 1.1.1
  \Item Item 1.1.2
  \End
 \Item Item 1.2
 \End
\Item Item 2
\Paragr Paragr
\Item Item 3
\End
```

8.3 Description Lists

A description list has the following structure:

⟨*First tag*⟩
⟨*Descritpion text one*⟩
⟨*Second tag*⟩
⟨*Description text two*⟩
⟨*Further text*⟩

```
\BeginDesc
\Item{⟨First tag⟩} ⟨Descritpion text one⟩
\Item{⟨Second tag⟩} ⟨Description text two⟩
\Paragr ⟨Further text⟩
\End
```

Space after '`}`' does not influence the output. If a new paragraph should be started within an item, `\Paragr` is used.

9 Table captions

Table captions are set with the `\TblCapt` command:

`\TblCapt[`⟨*optional short caption*⟩`]{`⟨*caption text*⟩`}`

If an ⟨*optional short caption*⟩ is given, it is used instead of ⟨*caption text*⟩ in the list of tables. To support flexible placement schemes (centered above the table, on top right to the table, etc.), the user is responsible for the caption placement. In simple cases something like

```
\hbox to ⟨table width⟩{\hfil\TblCapt{⟨caption⟩}\hfil}
\smallskip
```

above the table should be sufficient. The caption is preceded by "**Tbl.**" (followed by the table number). This text can be changed be redefining `\TblNam`.

10 EPS file inclusion

To enable EPS inclusion the `epsf` macros need to be included by placing

```
\input epsf
```

at the begin of the document. The command to include the picture is

```
\EpsfBox{⟨file name⟩}
```

e. g.

```
\EpsfBox{ctanlion.eps}
```

The macro sets `\EpsfXsize` to the width of the picture. So if text should be placed beside the picture it can be done using the `\FigCapt` macro with

```
\vbox{\advance\hsize by-\EpsfXsize\noindent
 \FigCapt[⟨optional short caption⟩]{⟨caption text⟩}}
```

If an ⟨*optional short caption*⟩ is given, it is used instead of ⟨*caption text*⟩ in the list of figures. The caption is preceded by "**Fig.**" (followed by the figure number). This text can be changed be redefining `\FigNam`. Example:

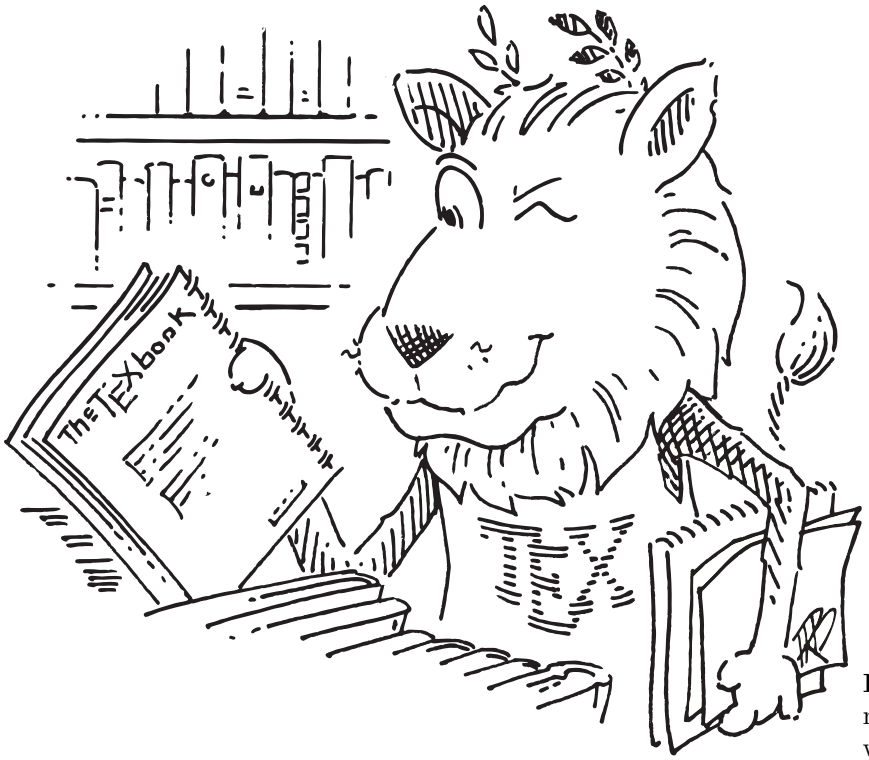

Fig. 10.1. Text that follows normally appears on the right with the same baseline.

Pictures can be scaled by setting `\epsfxsize=`⟨dimen⟩ and/or `\epsfysize=`⟨dimen⟩, e. g.

```
\epsfxsize=5cm
\centerline{\EpsfBox{ctanlion.eps}\FigCapt{Scaled EPS image}}
```

produces:

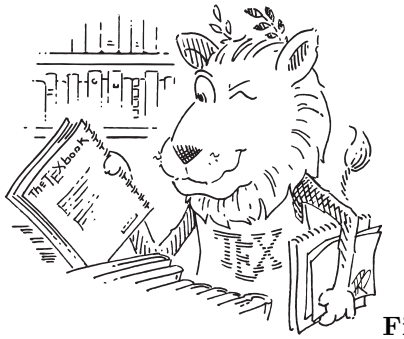

Fig. 10.2. Scaled EPS image.

11 Footnotes

Plain TEX provides a ready to use solution for footnotes*. To have automatic footnote numbering

`\Footnote{`⟨*text*⟩`}`

can be used. The footnote counter[1] is reset when the macro package calls `\supereject` before a headline.[2] Further references to a previous footnote[2] can be marked with `\FootnoteRef`. `\footnote` must not be used in an *insert*. `\vfootnote`[3] may be used instead.

12 Cross references

A reference mark can be placed with

`\Label{`⟨*label*⟩`}`

at a heading, table, figure etc. To reference a label the label data needs to be loaded with

`\Ref{`⟨*label*⟩`}`

After that the following macros are set:

`\RefPgNr`
: Reference page number.

`\RefNr`
: Number of table or figure.

`\RefHdNr`
: Number of heading.

`\RefHd`
: Title of the last heading before the label.

`\RefPg`
: Outputs text:

  ⟨nothing⟩
  : if the label is on the current page,

  " on the previous page"
  : if the label is on the previous page,

  " on the following page"
  : if the label is on the next page, and

  " on page `\RefPgNr`"
  : else.

  (Notice the leading space.)

---

* Like this which is produced with `\footnote{*}{`⟨*text*⟩`}`, see [1, p. 116].

[1] `\FootnoteCtr`

[2] Depends on the `\EjectLevel` setting.

[3] [1, p. 117]

13 Index

Words are added to the index with the `\Idx` macro:

`\index{`⟨*word or symbol*⟩`}`

If the words to add belong to a main category, they are separated by a '`!`':

`\index{`⟨*main word*⟩`!`⟨*word or symbol*⟩`}`

A '`!`' at the begin or end of the argument to `\Idx` it is recognized as itself. If a '`!`' should be placed inside other text it must be doubled. More that two consecutive '`!`' need to be separated by '`{}`':

```
\index{Special characters!{}!!}
```

For special markup additional symbols can be added to the argument, again separated by '`!`' (see table 13.1). If such such a symbol sequence is actually meant to be added to the index it needs to be doubled. The index generator removes such sequences one time from the end of the argument. If it apears a second time it is included into the index. Combinations of such markup symbols are possible:

```
\index{\idx!(!B}
```

Tbl. 13.1. Types of index entries

Markup	Description	Symbol
Bold page number	Text reference contains important information for the subject.	'`B`'
Italic page number	Text reference contains unimportant information for the subject ("*also mentioned there ...*").	'`I`'
Range of page numbers	Subject is descripted on these range of pages.	Use '`(`' for the first page and '`)`' for the last page of the range.

The argument to `\Idx` is written to the file `idx.txt` in the form ⟨*argument*⟩⟨*space*⟩⟨*page number*⟩.

Bibliography

[1] Donald E. Knuth. *The TEXbook*, volume A. Addison-Wesley.

Index