The **comment** package[*]

Victor Eijkhout
`victor@eijkhout.net`

October 1999

1 Purpose:

selectively in/exclude pieces of text: the user can define new comment versions, and each is controlled separately. Special comments can be defined where the user specifies the action that is to be taken with each comment line.

This style can be used with plain TEX or LATEX, and probably most other packages too.

2 Usage:

all text included between

```
\comment ... \endcomment
```

or

```
\begin{comment} ... \end{comment}
```

is discarded.

The opening and closing commands should appear on a line of their own. No starting spaces, nothing after it. This environment should work with arbitrary amounts of comment, and the comment can be arbitrary text.

Other '`comment`' environments are defined by and are selected/deselected with

```
\includecomment{versiona}
\excludecoment{versionb}
```

---

[*] This manual corresponds to comment v3.6, dated October 1999.

These environments are used as

```
\versiona ... \endversiona
```

or

```
\begin{versiona} ... \end{versiona}
```

with the opening and closing commands again on a line of their own.

LaTeX users note: for an included comment, the `\begin` and `\end` lines act as if they don't exist. In particular, they don't imply grouping, so assignments &c are not local.

Special comments are defined as

```
\specialcomment{⟨name⟩}{⟨before commands⟩}{⟨after commands⟩}
```

where the second and third arguments are executed before and after each comment block. You can use this for global formatting commands. To keep definitions &c local, you can include `\begingroup` in the '⟨*before commands*⟩' and `\endgroup` in the '⟨*after commands*⟩'. ex:

```
\specialcomment{smalltt}
    {\begingroup\ttfamily\footnotesize}{\endgroup}
```

You do *not* have to do an additional

```
\includecomment{smalltt}
```

To remove '`smalltt`' blocks, give `\excludecomment{smalltt}` after the definition.

Processing comments can apply processing to each line.

```
\processcomment{⟨name⟩}{⟨each-line commands⟩}{⟨before
commands⟩}{⟨after commands⟩}
```

By defining a control sequence

```
\def\Thiscomment##1{...}
```

in the before commands the user can specify what is to be done with each comment line. BUG this does not work quite yet BUG

Trick for short in/exclude macros (such as `\maybe{this snippet}`):

```
\includecomment{cond}
\newcommand{\maybe}[1]{}
\begin{cond}
\renewcommand{\maybe}[1]{#1}
\end{cond}
```

3 Basic approach of the implementation:

to comment something out, scoop up every line in verbatim mode as macro argument, then throw it away. For inclusions, in LaTeX the block is written out to a file `\CommentCutFile` (default “`comment.cut`”), which is then included. In plain TeX (and other formats) both the opening and closing comands are defined as noop.

4 Changes in version 3.1

- updated author’s address
- cleaned up some code
- trailing contents on `\begin{`⟨*env*⟩`}` line is always discarded even if you’ve done `\includecomment{`⟨*env*⟩`}`
- comments no longer define grouping!! you can even

  ```
  \includecomment{env}
  \begin{env}
  \begin{itemize}
  \end{env}
  ```

  Isn’t that something…
- included comments are written to file and input again.

5 Changes in 3.2

- `\specialcomment` brought up to date (thanks to Ivo Welch).

6 Changes in 3.3

- updated author’s address again
- parametrised `\CommentCutFile`

7 Changes in 3.4

- added GNU public license
- added `\processcomment`, because Ivo’s fix (above) brought an inconsistency to light.

8 Changes in 3.5

- corrected typo in header.
- changed author email
- corrected `\specialcomment` yet again.
- fixed excludecomment of an earlier defined environment.

9 Changes in 3.6

- The ‘cut’ file is now written more verbatim, using `\meaning`; some people reported having trouble with ISO latin 1, or `umlaute.sty`.
- removed some `\newif` statements. Has this suddenly become `\outer` again?

10 Changes in 3.8

T1 font encoding is now supported. See t1test.tex.