

The ifdraft package

Heiko Oberdiek
<oberdiek@uni-freiburg.de>

2006/02/20 v1.2

Abstract

The package provides an interface for selecting code depending on the options `draft` and `final`.

Contents

| | | |
|----------|---|----------|
| 1 | Usage | 1 |
| 1.1 | Package loading | 1 |
| 1.2 | User macros | 1 |
| 2 | Implementation | 2 |
| 3 | Installation | 3 |
| 3.1 | Some details for the interested | 3 |
| 4 | History | 4 |
| | [1999/12/28 v1.0] | 4 |
| | [2005/10/05 v1.1] | 4 |
| | [2006/02/20 v1.2] | 4 |
| 5 | Index | 4 |

1 Usage

1.1 Package loading

In order to detect the global class options `draft` and `final`, load this package somewhere after `\documentclass` without options:

```
\usepackage{ifdraft}
```

1.2 User macros

```
\ifdraft {\<draft case>} {\<final case>}  
\ifoptiondraft {\<option draft is given>} {\<option draft is not given>}  
\ifoptionfinal {\<option final is given>} {\<option final is not given>}
```

If none of the options `draft` or `final` is used, then this package assumes `final` as default setting for `\ifdraft`. All classes that are known to me behave this way. (Otherwise you can find out with `\ifoptiondraft` and `\ifoptionfinal`, whether none of the options is set.)

If either `draft` or `final` is used, `\ifdraft` is sufficient to distinguish between these cases.

Both options `draft` and `final` should not be used at the same time. This is contradictory input. Which option is more important? The result is unpredictable in general:

- `article`, `report`, `book`, `scrartcl`, `scrreprt`, `scrbook`:
`draft`, `final` \rightarrow `final` is effective.
`final`, `draft` \rightarrow `final` is effective.
 \Rightarrow `final` wins, if given.
- `memoir`:
`draft`, `final` \rightarrow `draft` is effective.
`final`, `draft` \rightarrow `draft` is effective.
 \Rightarrow `draft` wins if given.

These classes evaluate the options in declaration order. Because the declaration order of these options in this package is not really interesting, this package evaluates the options in the order specified in the calling commands:

- `ifdraft`:
`draft`, `final` \rightarrow `\ifdraft` selects `final` clause.
`final`, `draft` \rightarrow `\ifdraft` selects `draft` clause.
 \Rightarrow latest given option wins.

Thus you know with `\ifdraft` the latest given option and you can emulate the behaviour of the different classes with the help of `\ifoptiondraft` and `\ifoptionfinal`.

Summary: `\ifdraft` is sufficient to deal with the normal use cases: one or none out of `draft` and `final`.

2 Implementation

```

1 \*package>
Package identification.
2 \NeedsTeXFormat{LaTeX2e}
3 \ProvidesPackage{ifdraft}%
4   [2006/02/20 v1.2 Switch for option draft (H0)]
5 \newif\if@draft
6 \newif\if@option@draft
7 \newif\if@option@final
8 \DeclareOption{draft}{%
9   \@drafttrue
10  \@option@drafttrue
11 }
12 \DeclareOption{final}{%
13   \@draftfalse
14   \@option@finaltrue
15 }
16 \ProcessOptions*\relax

\ifdraft
17 \newcommand*{\ifdraft}{%
18   \if@draft
19     \expandafter\@firstoftwo
20   \else
21     \expandafter\@secondoftwo
22   \fi
23 }

\ifoptiondraft
24 \newcommand*{\ifoptiondraft}{%

```

```

25 \if@option@draft
26   \expandafter\@firstoftwo
27 \else
28   \expandafter\@secondoftwo
29 \fi
30 }

\ifoptionfinal

31 \newcommand*\ifoptionfinal{%
32   \if@option@final
33     \expandafter\@firstoftwo
34   \else
35     \expandafter\@secondoftwo
36   \fi
37 }

38 </package>

```

3 Installation

CTAN. This package is available on CTAN¹:

[CTAN:macros/latex/contrib/oberdiek/ifdraft.dtx](#) The source file.

[CTAN:macros/latex/contrib/oberdiek/ifdraft.pdf](#) Documentation.

Unpacking. The `.dtx` file is a self-extracting `docstrip` archive. The files are extracted by running the `.dtx` through plain-`TEX`:

```
tex ifdraft.dtx
```

TDS. Now the different files must be moved into the different directories in your installation TDS tree (also known as `texmf` tree):

```

ifdraft.sty → tex/latex/oberdiek/ifdraft.sty
ifdraft.pdf → doc/latex/oberdiek/ifdraft.pdf
ifdraft.dtx → source/latex/oberdiek/ifdraft.dtx

```

If you have a `docstrip.cfg` that configures and enables `docstrip`'s TDS installing feature, then some files can already be in the right place, see the documentation of `docstrip`.

Refresh file databases. If your `TEX` distribution (`teTEX`, `mikTEX`, ...) rely on file databases, you must refresh these. For example, `teTEX` users run `texhash` or `mktexlsr`.

3.1 Some details for the interested

Attached source. The PDF documentation on CTAN also includes the `.dtx` source file. It can be extracted by AcrobatReader 6 or higher. Another option is `pdftk`, e.g. unpack the file into the current directory:

```
pdftk ifdraft.pdf unpack_files output .
```

¹<http://ftp.ctan.org/tex-archive/>

Unpacking with L^AT_EX. The .dtx chooses its action depending on the format:

plain-T_EX: Run docstrip and extract the files.

L^AT_EX: Generate the documentation.

If you insist on using L^AT_EX for docstrip (really, docstrip does not need L^AT_EX), then inform the autodetect routine about your intension:

```
latex \install=y\input{ifdraft.dtx}
```

Do not forget to quote the argument according to the demands of your shell.

Generating the documentation. You can use both the .dtx or the .drv to generate the documentation. The process can be configured by the configuration file ltxdoc.cfg. For instance, put this line into this file, if you want to have A4 as paper format:

```
\PassOptionsToClass{a4paper}{article}
```

An example follows how to generate the documentation with pdfL^AT_EX:

```
pdflatex ifdraft.dtx
makeindex -s gind.ist ifdraft.idx
pdflatex ifdraft.dtx
makeindex -s gind.ist ifdraft.idx
pdflatex ifdraft.dtx
```

4 History

[1999/12/28 v1.0]

- First public release.
- LPPL 1.1

[2005/10/05 v1.1]

- \ifoptiondraft and \ifoptionfinal added.
- \ProcessOptions changed to \ProcessOptions*. (Order of given class options matters instead of the order of option declaration in this package.)
- LPPL 1.3

[2006/02/20 v1.2]

- DTX framework.

5 Index

Numbers written in *italic* refer to the page where the corresponding entry is described; numbers underlined refer to the code line of the definition; numbers in *roman* refer to the code lines where the entry is used.

| Symbols | |
|------------------------------|------------|
| \@draftfalse | 13 |
| \@drafttrue | 9 |
| \@firstoftwo | 19, 26, 33 |
| \@option@drafttrue | 10 |
| \@option@finaltrue | 14 |
| \@secondoftwo | 21, 28, 35 |
| | D |
| \DeclareOption | 8, 12 |
| | I |
| \if@draft | 5, 18 |

| | | | |
|-------------------------------------|----------|-------------------------------------|------------|
| <code>\if@option@draft</code> | 6, 25 | <code>\newcommand</code> | 17, 24, 31 |
| <code>\if@option@final</code> | 7, 32 | <code>\newif</code> | 5, 6, 7 |
| <code>\ifdraft</code> | 1, 17 | | |
| <code>\ifoptiondraft</code> | 1, 24 | | |
| <code>\ifoptionfinal</code> | 1, 31 | | |
| | | P | |
| | N | <code>\ProcessOptions</code> | 16 |
| <code>\NeedsTeXFormat</code> | 2 | <code>\ProvidesPackage</code> | 3 |