The ifdraft package

Heiko Oberdiek
<oberdiek@uni-freiburg.de>

2008/08/11 v1.3

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 Download 3
3.2 Bundle installation 3
3.3 Package installation 3
3.4 Refresh file name databases 4
3.5 Some details for the interested 4

4 History **4**
[1999/12/28 v1.0] 4
[2005/10/05 v1.1] 5
[2006/02/20 v1.2] 5
[2008/08/11 v1.3] 5

5 Index **5**

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 contradictionary input. Which option is more important? The result is unpredictable in general:

- article, report, book, scrartcl, scrreprt, scrbook:
  draft, final → final is effective.
  final, draft → final is effective.
  ⇒ final wins, if given.
- memoir:
  draft, final → draft is effective.
  final, draft → draft is effective.
  ⇒ draft wins if given.

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

- ifdraft:
  draft, final → `\ifdraft` selects final clause.
  final, draft → `\ifdraft` selects draft clause.
  ⇒ 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   [2008/08/11 v1.3 Switch for option draft (HO)]
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

```
\newcommand*{\ifoptiondraft}{%
  \if@option@draft
    \expandafter\@firstoftwo
  \else
    \expandafter\@secondoftwo
  \fi
}
```

\ifoptionfinal

```
\newcommand*{\ifoptionfinal}{%
  \if@option@final
    \expandafter\@firstoftwo
  \else
    \expandafter\@secondoftwo
  \fi
}
```

⟨/package⟩

3 Installation

3.1 Download

Package. This package is available on CTAN[1]:

CTAN:macros/latex/contrib/oberdiek/ifdraft.dtx The source file.

CTAN:macros/latex/contrib/oberdiek/ifdraft.pdf Documentation.

Bundle. All the packages of the bundle 'oberdiek' are also available in a TDS compliant ZIP archive. There the packages are already unpacked and the documentation files are generated. The files and directories obey the TDS standard.

CTAN:install/macros/latex/contrib/oberdiek.tds.zip

TDS refers to the standard "A Directory Structure for TEX Files" (CTAN:tds/tds.pdf). Directories with `texmf` in their name are usually organized this way.

3.2 Bundle installation

Unpacking. Unpack the `oberdiek.tds.zip` in the TDS tree (also known as `texmf` tree) of your choice. Example (linux):

```
unzip oberdiek.tds.zip -d ~/texmf
```

Script installation. Check the directory `TDS:scripts/oberdiek/` for scripts that need further installation steps. Package attachfile2 comes with the Perl script `pdfatfi.pl` that should be installed in such a way that it can be called as `pdfatfi`. Example (linux):

```
chmod +x scripts/oberdiek/pdfatfi.pl
cp scripts/oberdiek/pdfatfi.pl /usr/local/bin/
```

3.3 Package installation

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

```
tex ifdraft.dtx
```

[1] ftp://ftp.ctan.org/tex-archive/

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.

3.4 Refresh file name databases

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

3.5 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 .
```

Unpacking with LATEX. The `.dtx` chooses its action depending on the format:

plain-TEX: Run docstrip and extract the files.

LATEX: Generate the documentation.

If you insist on using LATEX for docstrip (really, docstrip does not need LATEX), then inform the autodetect routine about your intention:

```
latex \let\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 pdfLATEX:

```
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, published in newsgroup `de.comp.text.tex`: "Re: auf vorhandensein der option "draft" pruefen"[2]
- LPPL 1.1

[2] Url: http://groups.google.com/group/de.comp.text.tex/msg/ccc1ccc9a8c224e9

[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.

[2008/08/11 v1.3]

- Code is not changed.
- URLs updated.

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
`\if@option@draft` 6, 25
`\if@option@final` 7, 32
`\ifdraft` *1*, 17
`\ifoptiondraft` *1*, 24
`\ifoptionfinal` *1*, 31

N

`\NeedsTeXFormat` 2
`\newcommand` 17, 24, 31
`\newif` 5, 6, 7

P

`\ProcessOptions` 16
`\ProvidesPackage` 3