The flags package

Heiko Oberdiek*

2016/05/16 v0.5

Abstract

Package flags allows the setting and clearing of flags in bit fields and converts the bit field into a decimal number. Currently the bit field is limited to 31 bits.

Contents

1 Documentation

A new powerful package bitset is written by me and supersedes this package:

- The bit range is not restricted to 31 bits, only index numbers are objected to TEX's number limit.
- Many more operations are available.
- No dependency of ε-TEX.

Therefore I consider this package as obsolete and have stopped the development of this package.

*Please report any issues at https://github.com/ho-tex/oberdiek/issues

1.1 User interface

Flag positions are one-based, thus the flag position must be a positive integer. Currently supported range: `1..31`

`\resetflags` {⟨*fname*⟩}

The bit field ⟨*fname*⟩ is cleared. Currently is is also used for initialization, because a `\newflags` macro is not implemented.

`\setflag` {⟨*fname*⟩} {⟨*position*⟩}

The flag at bit position ⟨*position*⟩ is set in the bit field ⟨*fname*⟩.

`\clearflag` {⟨*fname*⟩} {⟨*position*⟩}

The flag at bit position ⟨*position*⟩ is cleared in the bit field ⟨*fname*⟩.

`\printflags` {⟨*fname*⟩}

The bit field ⟨*fname*⟩ is converted to a decimal number. The macro is expandible.

`\extractflag` {⟨*fname*⟩} {⟨*position*⟩}

Extracts the flag setting at bit position ⟨*position*⟩. `\extractflag` expands to `1` if the flag is set and `0` otherwise.

`\queryflag` {⟨*fname*⟩} {⟨*position*⟩} {⟨*set part*⟩} {⟨*clear part*⟩}

It is a wrapper for `\extractflag`. ⟨*set part*⟩ is called if `\extractflag` returns `1`. Otherwise ⟨*clear part*⟩ is executed.

Example. See package bookmark. It uses package flags for its font style options.

1.2 Requirements

- ε-TEX (`\numexpr`)

1.3 ToDo

- Named positions.
- Setting positions by a key-value interface.
- Support for more than 31 bits while maintaining expandibility of `\printflags`.
- Eventually `\newflags`, `\newflagstype`.

2 Implementation

```
1 ⟨*package⟩
2 \NeedsTeXFormat{LaTeX2e}
3 \ProvidesPackage{flags}%
4   [2016/05/16 v0.5 Setting/clearing of flags in bit fields (HO)]%
5 \begingroup\expandafter\expandafter\expandafter\endgroup
```

```
\expandafter\ifx\csname numexpr\endcsname\relax
  \PackageError{flags}{%
    Missing e-TeX, package loading aborted%
  }{%
    This packages makes heavy use of \string\numexpr.%
  }%
  \expandafter\endinput
\fi
```

\resetflags

```
\newcommand*{\resetflags}[1]{%
  \expandafter\let\csname flags@#1\endcsname\@empty
}
```

\printflags Macro \printflags converts the bit field into a decimal number.

```
\newcommand*{\printflags}[1]{%
  \expandafter\@printflags\csname flags@#1\endcsname
}
\def\@printflags#1{%
  \expandafter\@firstofone\expandafter{%
    \number\numexpr
    \ifx#1\@empty
      0%
    \else
      \expandafter\@@printflags#1%
    \fi
  }%
}
\def\@@printflags#1#2\fi{%
  \fi
  #1%
  \ifx\\#2\\%
  \else
    +2*\numexpr\expandafter\@@printflags#2%
  \fi
}
```

\setflag

```
\newcommand*{\setflag}[2]{%
  \ifnum#2>\z@
    \expandafter\@setflag\csname flags@#1\expandafter\endcsname
      \expandafter{\romannumeral\number\numexpr#2-1\relax000}%
  \else
    \PackageError{flags}{Position must be a positive number}\@ehc
  \fi
}
\def\@setflag#1#2{%
  \ifx#1\relax
    \let#1\@empty
  \fi
  \edef#1{%
    \expandafter\@@setflag\expandafter{#1}{#2}%
  }%
}
\def\@@setflag#1#2{%
  \ifx\\#1\\%
    \FLAGS@zero#2\relax
    1%
  \else
    \ifx\\#2\\%
      1\@gobble#1%
    \else
      \@@@setflag#1|#2%
```

```
    \fi
  \fi
}
\def\@@@setflag#1#2|#3#4\fi\fi{%
  \fi\fi
  #1%
  \@@setflag{#2}{#4}%
}
```

\clearflag

```
\newcommand*{\clearflag}[2]{%
  \ifnum#2>\z@
    \expandafter\@clearflag\csname flags@#1\expandafter\endcsname
      \expandafter{\romannumeral\number\numexpr#2-1\relax000}%
  \else
    \PackageError{flags}{Position must be a positive number}\@ehc
  \fi
}
\def\@clearflag#1#2{%
  \ifx#1\relax
    \let#1\@empty
  \fi
  \edef#1{%
    \expandafter\@@clearflag\expandafter{#1}{#2}%
  }%
}
\def\@@clearflag#1#2{%
  \ifx\\#1\\%
  \else
    \ifx\\#2\\%
      0\@gobble#1%
    \else
      \@@@clearflag#1|#2%
    \fi
  \fi
}
\def\@@@clearflag#1#2|#3#4\fi\fi{%
  \fi\fi
  #1%
  \@@clearflag{#2}{#4}%
}

\def\FLAGS@zero#1{%
  \ifx#1\relax
  \else
    0%
    \expandafter\FLAGS@zero
  \fi
}
```

\queryflag

```
\newcommand*{\queryflag}[2]{%
  \ifnum\extractflag{#1}{#2}=\@ne
    \expandafter\@firstoftwo
  \else
    \expandafter\@secondoftwo
  \fi
}
```

\extractflag

```
\newcommand*{\extractflag}[1]{%
  \expandafter\@extractflag\csname flags@#1\endcsname
}
```

```
\def\@extractflag#1#2{%
  \ifx#1\@undefined
    0%
  \else
    \ifx#1\relax
      0%
    \else
      \ifx#1\@empty
        0%
      \else
        \expandafter\expandafter\expandafter\@@extractflag
        \expandafter\expandafter\expandafter{%
        \expandafter#1\expandafter
        }\expandafter{%
          \romannumeral\number\numexpr#2-1\relax000%
        }%
      \fi
    \fi
  \fi
}
\def\@@extractflag#1#2{%
  \ifx\\#1\\%
    0%
  \else
    \ifx\\#2\\%
      \@car#1\@nil
    \else
      \@@@extractflag#1|#2%
    \fi
  \fi
}
\def\@@@extractflag#1#2|#3#4\fi\fi{%
  \fi\fi
  \@@extractflag{#2}{#4}%
}
```

⟨/package⟩

3 Installation

3.1 Download

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

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

CTAN:macros/latex/contrib/oberdiek/flags.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:pkg/tds). 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):

[1] CTAN:pkg/flags

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

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 flags.dtx
```

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

```
flags.sty → tex/latex/oberdiek/flags.sty
flags.pdf → doc/latex/oberdiek/flags.pdf
flags.dtx → source/latex/oberdiek/flags.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 (TEX Live, MiKTEX, ...) relies on file name databases, you must refresh these. For example, TEX Live users run `texhash` or `mktexlsr`.

3.5 Some details for the interested

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{flags.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 flags.dtx
makeindex -s gind.ist flags.idx
pdflatex flags.dtx
makeindex -s gind.ist flags.idx
pdflatex flags.dtx
```

4 History

[2007/02/18 v0.1]

- First version.

[2007/03/07 v0.2]

- Raise an error if ε-TEX is not detected.

[2007/03/31 v0.3]

- `\queryflag` and `\extractflag` added.
- Raise an error if position is not positive in case of `\setflag` and `\clearflag`.

[2007/09/30 v0.4]

- Package is deprecated because of new more powerful package bitset.

[2016/05/16 v0.5]

- Documentation updates.

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; plain numbers refer to the code lines where the entry is used.