

# 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

<b>1 Documentation</b>	<b>1</b>
1.1 User interface	2
1.2 Requirements	2
1.3 ToDo	2
<b>2 Implementation</b>	<b>2</b>
<b>3 Installation</b>	<b>5</b>
3.1 Download	5
3.2 Bundle installation	5
3.3 Package installation	6
3.4 Refresh file name databases	6
3.5 Some details for the interested	6
<b>4 History</b>	<b>6</b>
[2007/02/18 v0.1]	6
[2007/03/07 v0.2]	7
[2007/03/31 v0.3]	7
[2007/09/30 v0.4]	7
[2016/05/16 v0.5]	7
<b>5 Index</b>	<b>7</b>

## 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  $\text{T}_{\text{E}}\text{X}$ 's number limit.
- Many more operations are available.
- No dependency of  $\varepsilon\text{-T}_{\text{E}}\text{X}$ .

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  $\langle fname \rangle$  is cleared. Currently is also used for initialization, because a `\newflags` macro is not implemented.

`\setflag {⟨fname⟩} {⟨position⟩}`

The flag at bit position  $\langle position \rangle$  is set in the bit field  $\langle fname \rangle$ .

`\clearflag {⟨fname⟩} {⟨position⟩}`

The flag at bit position  $\langle position \rangle$  is cleared in the bit field  $\langle fname \rangle$ .

`\printflags {⟨fname⟩}`

The bit field  $\langle fname \rangle$  is converted to a decimal number. The macro is expandible.

`\extractflag {⟨fname⟩} {⟨position⟩}`

Extracts the flag setting at bit position  $\langle position \rangle$ . `\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`.  $\langle set part \rangle$  is called if `\extractflag` returns 1. Otherwise  $\langle clear part \rangle$  is executed.

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

## 1.2 Requirements

- $\varepsilon$ -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 (H0)]%
5 \begingroup\expandafter\expandafter\expandafter\endgroup
```

```

6 \expandafter\ifx\csname numexpr\endcsname\relax
7   \PackageError{flags}{%
8     Missing e-TeX, package loading aborted%
9   }{%
10     This packages makes heavy use of \string\numexpr.%
11   }%
12 \expandafter\endinput
13 \fi

\resetflags
14 \newcommand*\resetflags[1]{%
15   \expandafter\let\csname flags@#1\endcsname\@empty
16 }

\printflags Macro \printflags converts the bit field into a decimal number.
17 \newcommand*\printflags[1]{%
18   \expandafter\@printflags\csname flags@#1\endcsname
19 }
20 \def\@printflags#1{%
21   \expandafter\@firstofone\expandafter{%
22     \number\numexpr
23     \ifx#1\@empty
24       0%
25     \else
26       \expandafter\@@printflags#1%
27     \fi
28   }%
29 }
30 \def\@@printflags#1#2\fi{%
31   \fi
32   #1%
33   \ifx\#2\%
34   \else
35     +2*\numexpr\expandafter\@@printflags#2%
36   \fi
37 }

\setflag
38 \newcommand*\setflag[2]{%
39   \ifnum#2>\z@
40     \expandafter\@setflag\csname flags@#1\endcsname
41     \expandafter{\romannumeral\number\numexpr#2-1\relax000}%
42   \else
43     \PackageError{flags}{Position must be a positive number}\@ehc
44   \fi
45 }
46 \def\@setflag#1#2{%
47   \ifx#1\relax
48     \let#1\@empty
49   \fi
50   \edef#1{%
51     \expandafter\@@setflag\expandafter{#1}{#2}%
52   }%
53 }
54 \def\@@setflag#1#2{%
55   \ifx\#1\%
56     \FLAGS@zero#2\relax
57     1%
58   \else
59     \ifx\#2\%
60       1\@gobble#1%
61     \else
62       \@@@setflag#1#2%

```

```

63     \fi
64     \fi
65 }
66 \def\@@@setflag#1#2|#3#4\fi\fi{%
67     \fi\fi
68     #1%
69     \@@setflag{#2}{#4}%
70 }

\clearflag
71 \newcommand*\clearflag[2]{%
72     \ifnum#2>\z@
73         \expandafter\@clearflag\csname flags@#1\expandafter\endcsname
74         \expandafter{\romannumeral\number\numexpr#2-1\relax000}%
75     \else
76         \PackageError{flags}{Position must be a positive number}\@ehc
77     \fi
78 }
79 \def\@clearflag#1#2{%
80     \ifx#1\relax
81         \let#1\@empty
82     \fi
83     \edef#1{%
84         \expandafter\@clearflag\expandafter{#1}{#2}%
85     }%
86 }
87 \def\@@clearflag#1#2{%
88     \ifx\#1\%
89     \else
90         \ifx\#2\%
91             0\@gobble#1%
92         \else
93             \@@clearflag#1|#2%
94         \fi
95     \fi
96 }
97 \def\@@@clearflag#1#2|#3#4\fi\fi{%
98     \fi\fi
99     #1%
100     \@@clearflag{#2}{#4}%
101 }

102 \def\FLAGS@zero#1{%
103     \ifx#1\relax
104     \else
105         0%
106         \expandafter\FLAGS@zero
107     \fi
108 }

\queryflag
109 \newcommand*\queryflag[2]{%
110     \ifnum\extractflag{#1}{#2}=\@ne
111         \expandafter\@firstoftwo
112     \else
113         \expandafter\@secondoftwo
114     \fi
115 }

\extractflag
116 \newcommand*\extractflag[1]{%
117     \expandafter\@extractflag\csname flags@#1\endcsname
118 }

```

```

119 \def\@extractflag#1#2{%
120   \ifx#1\@undefined
121     0%
122   \else
123     \ifx#1\relax
124       0%
125     \else
126       \ifx#1\@empty
127         0%
128       \else
129         \expandafter\expandafter\expandafter\@extractflag
130         \expandafter\expandafter\expandafter{%
131         \expandafter#1\expandafter
132         }\expandafter{%
133         \romannumeral\number\numexpr#2-1\relax000%
134         }%
135       \fi
136     \fi
137   \fi
138 }
139 \def\@@extractflag#1#2{%
140   \ifx\|#1\|
141     0%
142   \else
143     \ifx\|#2\|
144       \@car#1\@nil
145     \else
146       \@@extractflag#1|#2%
147     \fi
148   \fi
149 }
150 \def\@@@extractflag#1#2|#3#4\fi\fi{%
151   \fi\fi
152   \@@extractflag{#2}{#4}%
153 }
154 \end{package}

```

## 3 Installation

### 3.1 Download

**Package.** This package is available on CTAN<sup>1</sup>:

[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  $\text{\TeX}$  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):

---

<sup>1</sup>[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 L<sup>A</sup>T<sub>E</sub>X.** The `.dtx` chooses its action depending on the format:

**plain T<sub>E</sub>X:** Run `docstrip` and extract the files.

**L<sup>A</sup>T<sub>E</sub>X:** Generate the documentation.

If you insist on using L<sup>A</sup>T<sub>E</sub>X for `docstrip` (really, `docstrip` does not need L<sup>A</sup>T<sub>E</sub>X), 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  $\varepsilon$ -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.

Symbols	F
<code>\@@@clearflag</code> ..... 93, 97	<code>\FLAGS@zero</code> ..... 56, 102, 106
<code>\@@@extractflag</code> ..... 146, 150	
<code>\@@@setflag</code> ..... 62, 66	I
<code>\@clearflag</code> ..... 84, 87, 100	<code>\ifnum</code> ..... 39, 72, 110
<code>\@@extractflag</code> ..... 129, 139, 152	<code>\ifx</code> ..... 6, 23, 33, 47, 55, 59, 80,
<code>\@printflags</code> ..... 26, 30, 35	88, 90, 103, 120, 123, 126, 140, 143
<code>\@@setflag</code> ..... 51, 54, 69	
<code>\@car</code> ..... 144	N
<code>\@clearflag</code> ..... 73, 79	<code>\NeedsTeXFormat</code> ..... 2
<code>\@ehc</code> ..... 43, 76	<code>\newcommand</code> ... 14, 17, 38, 71, 109, 116
<code>\@empty</code> ..... 15, 23, 48, 81, 126	<code>\number</code> ..... 22, 41, 74, 133
<code>\@extractflag</code> ..... 117, 119	<code>\numexpr</code> ..... 10, 22, 35, 41, 74, 133
<code>\@firstofone</code> ..... 21	
<code>\@firstoftwo</code> ..... 111	P
<code>\@gobble</code> ..... 60, 91	<code>\PackageError</code> ..... 7, 43, 76
<code>\@ne</code> ..... 110	<code>\printflags</code> ..... 2, <u>17</u>
<code>\@nil</code> ..... 144	<code>\ProvidesPackage</code> ..... 3
<code>\@printflags</code> ..... 18, 20	
<code>\@secondoftwo</code> ..... 113	Q
<code>\@setflag</code> ..... 40, 46	<code>\queryflag</code> ..... 2, <u>109</u>
<code>\@undefined</code> ..... 120	
<code>\@</code> ..... 33, 55, 59, 88, 90, 140, 143	R
	<code>\resetflags</code> ..... 2, <u>14</u>
C	<code>\romannumeral</code> ..... 41, 74, 133
<code>\clearflag</code> ..... 2, <u>71</u>	
<code>\csname</code> ..... 6, 15, 18, 40, 73, 117	S
	<code>\setflag</code> ..... 2, <u>38</u>
E	
<code>\endcsname</code> ..... 6, 15, 18, 40, 73, 117	Z
<code>\endinput</code> ..... 12	<code>\z@</code> ..... 39, 72
<code>\extractflag</code> ..... 2, 110, <u>116</u>	