

# The `stackrel` package

Heiko Oberdiek\*

2016/05/16 v1.3

## Abstract

This package adds an optional argument to `\stackrel` for putting something below the relational symbol and defines `\stackbin` for binary symbols.

## Contents

<b>1</b>	<b>User interface</b>	<b>1</b>
<b>2</b>	<b>Implementation</b>	<b>2</b>
<b>3</b>	<b>Installation</b>	<b>3</b>
3.1	Download . . . . .	3
3.2	Bundle installation . . . . .	3
3.3	Package installation . . . . .	3
3.4	Refresh file name databases . . . . .	3
3.5	Some details for the interested . . . . .	3
<b>4</b>	<b>History</b>	<b>4</b>
	[2006/12/02 v1.0] . . . . .	4
	[2007/05/06 v1.1] . . . . .	4
	[2007/11/11 v1.2] . . . . .	4
	[2016/05/16 v1.3] . . . . .	4
<b>5</b>	<b>Index</b>	<b>5</b>

## 1 User interface

L<sup>A</sup>T<sub>E</sub>X's `\stackrel` allows a superscript above a relational symbol, but pure L<sup>A</sup>T<sub>E</sub>X does not provide a macro for putting a subscript below the symbol. This is supported by  $\mathcal{A}\mathcal{M}\mathcal{S}$  L<sup>A</sup>T<sub>E</sub>X's `\underset` macro that works on both relational and binary symbols. A combination of `\underset` and `\overset` can be used to put sub- and superscripts to the same symbol.

This package `stackrel` extends the syntax of `\stackrel` by adding an optional argument for the subscript position. It follows the syntax of extensible arrows of packages `amsmath` and `mathtools`.

<code>\stackrel</code> [ <i>subscript</i> ] { <i>superscript</i> } { <i>rel</i> }
<code>\stackbin</code> [ <i>subscript</i> ] { <i>superscript</i> } { <i>bin</i> }

Example:

$$\begin{array}{c} A \stackrel{\text{and}}{\stackbin{+}} B \stackrel{!}{\stackrel{=}{\stackrel{x}{\stackrel{\text{and}}{\stackbin{+}}}}} C \\ A + B \stackrel{\text{and}}{\stackbin{x}} C \end{array}$$

---

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

## 2 Implementation

```

1 <*package>
2 \NeedsTeXFormat{LaTeX2e}
3 \ProvidesPackage{stackrel}
4 [2016/05/16 v1.3 Adding subscript option to stackrel (H0)]%

```

Given the original definition of `\stackrel` the addition of the optional argument is straightforward. If an argument is empty, then the corresponding sub- or superscript is suppressed.

Depending on the available resources ( $\varepsilon$ -TeX, pdfTeX) three methods are given for testing emptiness. All tests allow the hash to be used inside the arguments without doubling (for the unlikely case that someone wants to define macros with arguments).

```

\stack@relbin
5 \RequirePackage{etexcmds}[2007/09/09]
6 \ifetex@unexpanded
7   \RequirePackage{pdftexcmds}[2016/05/16]%
8   \begingroup\expandafter\expandafter\expandafter\endgroup
9   \expandafter\ifx\csname pdf@strcmp\endcsname\relax
10    \newcommand*{\stack@relbin}[3][]{%
11      \mathop{#3}\limits
12      \edef\reserved@a{\etex@unexpanded{#1}}%
13      \ifx\reserved@a@empty\else_{#1}\fi
14      \edef\reserved@a{\etex@unexpanded{#2}}%
15      \ifx\reserved@a@empty\else^{#2}\fi
16    \egroup
17  }%
18 \else
19   \newcommand*{\stack@relbin}[3][]{%
20     \mathop{#3}\limits
21     \ifcase\pdf@strcmp{\detokenize{#1}}{-}\else_{#1}\fi
22     \ifcase\pdf@strcmp{\detokenize{#2}}{-}\else^{#2}\fi
23   \egroup
24 }%
25 \fi
26 \else
27   \newcommand*{\stack@relbin}[3][]{%
28     \mathop{#3}\limits
29     \toks@{#1}%
30     \edef\reserved@a{\the\toks@}%
31     \ifx\reserved@a@empty\else_{#1}\fi
32     \toks@{#2}%
33     \edef\reserved@a{\the\toks@}%
34     \ifx\reserved@a@empty\else^{#2}\fi
35   \egroup
36 }%
37 \fi

\stackrel
38 \renewcommand*{\stackrel}{%
39   \mathrel\bgroup\stack@relbin
40 }

\stackbin
41 \newcommand*{\stackbin}{%
42   \mathbin\bgroup\stack@relbin
43 }

44 </package>

```

## 3 Installation

### 3.1 Download

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

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

[CTAN:macros/latex/contrib/oberdiek/stackrel.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 T<sub>E</sub>X 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):

```
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 T<sub>E</sub>X:

```
tex stackrel.dtx
```

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

```
stackrel.sty → tex/latex/oberdiek/stackrel.sty
stackrel.pdf → doc/latex/oberdiek/stackrel.pdf
stackrel.dtx → source/latex/oberdiek/stackrel.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 T<sub>E</sub>X distribution (T<sub>E</sub>X Live, MiK<sub>T</sub><sub>E</sub>X, ...) relies on file name databases, you must refresh these. For example, T<sub>E</sub>X 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{stackrel.dtx}
```

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

---

<sup>1</sup>[CTAN:pkg/stackrel](#)

**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<sup>A</sup>T<sub>E</sub>X:

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

## 4 History

[2006/12/02 v1.0]

- First version.

[2007/05/06 v1.1]

- Uses package `etexcmds`.

[2007/11/11 v1.2]

- Use of package `pdftexcmds` for LuaT<sub>E</sub>X support.

[2016/05/16 v1.3]

- 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	
<code>\@empty</code> .....	13, 15, 31, 34
C	
<code>\csname</code> .....	9
D	
<code>\detokenize</code> .....	21, 22
E	
<code>\endcsname</code> .....	9
<code>\etex@unexpanded</code> .....	12, 14
I	
<code>\ifcase</code> .....	21, 22
<code>\ifetex@unexpanded</code> .....	6
<code>\ifx</code> .....	9, 13, 15, 31, 34
L	
<code>\limits</code> .....	11, 20, 28
M	
<code>\mathbin</code> .....	42
<code>\mathop</code> .....	11, 20, 28
<code>\mathrel</code> .....	39
N	
<code>\NeedsTeXFormat</code> .....	2
<code>\newcommand</code> .....	10, 19, 27, 41
P	
<code>\pdf@strcmp</code> .....	21, 22
<code>\ProvidesPackage</code> .....	3
R	
<code>\renewcommand</code> .....	38
<code>\RequirePackage</code> .....	5, 7
<code>\reserved@a</code> .....	12, 13, 14, 15, 30, 31, 33, 34
S	
<code>\stack@relbin</code> .....	<u>5</u> , 39, 42
<code>\stackbin</code> .....	<u>41</u>
<code>\stackrel</code> .....	1, <u>38</u>
T	
<code>\the</code> .....	30, 33
<code>\toks@</code> .....	29, 30, 32, 33