

# The `stackrel` package

Heiko Oberdiek  
<oberdiek@uni-freiburg.de>

2006/12/02 v1.0

## 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
<b>5</b>	<b>Index</b>	<b>4</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:

A `\stackbin[\text{and}]{+} B` `\stackrel[x]{!}{=}` C  
 $A + B \underset{x}{\stackrel{!}{=}} C$

## 2 Implementation

```

1 \*package>
2 \NeedsTeXFormat{LaTeX2e}
3 \ProvidesPackage{stackrel}
4 [2006/12/02 v1.0 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 \begingroup\expandafter\expandafter\expandafter\endgroup
6 \expandafter\ifx\csname unexpanded\endcsname\relax
7   \newcommand*{\stack@relbin}[3][]{%
8     \mathop{#3}\limits
9     \toks@{#1}%
10    \edef\reserved@a{\the\toks@}%
11    \ifx\reserved@a\@empty\else_{#1}\fi
12    \toks@{#2}%
13    \edef\reserved@a{\the\toks@}%
14    \ifx\reserved@a\@empty\else^{#2}\fi
15    \egroup
16  }%
17 \else
18   \begingroup\expandafter\expandafter\expandafter\endgroup
19   \expandafter\ifx\csname pdfstrcmp\endcsname\relax
20     \newcommand*{\stack@relbin}[3][]{%
21       \mathop{#3}\limits
22       \edef\reserved@a{\unexpanded{#1}}%
23       \ifx\reserved@a\@empty\else_{#1}\fi
24       \edef\reserved@a{\unexpanded{#2}}%
25       \ifx\reserved@a\@empty\else^{#2}\fi
26       \egroup
27     }%
28   \else
29     \newcommand*{\stack@relbin}[3][]{%
30       \mathop{#3}\limits
31       \ifcase\pdfstrcmp{\detokenize{#1}}{-}\else_{#1}\fi
32       \ifcase\pdfstrcmp{\detokenize{#2}}{-}\else^{#2}\fi
33       \egroup
34     }%
35   \fi
36 \fi

\stackrel

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

\stackbin

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

43 \</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:macros/latex/contrib/oberdiek/oberdiek-tds.zip](#)

### 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 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 TeX distribution (TeX, mikTeX, ...) relies on file name databases, you must refresh these. For example, TeX 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 stackrel.pdf unpack_files output .
```

---

<sup>1</sup>[ftp://ftp.ctan.org/tex-archive/](http://ftp.ctan.org/tex-archive/)

**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.

**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.

## 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	N
\@empty ..... 11, 14, 23, 25	\NeedsTeXFormat ..... 2
	\newcommand ..... 7, 20, 29, 40
C	P
\csname ..... 6, 19	\pdfstrcmp ..... 31, 32
	\ProvidesPackage ..... 3
D	R
\detokenize ..... 31, 32	\renewcommand ..... 37
	\reserved@a 10, 11, 13, 14, 22, 23, 24, 25
E	S
\endcsname ..... 6, 19	\stack@relbin ..... 5, 38, 41
	\stackbin ..... 40
I	\stackrel ..... 1, 37
\ifcase ..... 31, 32	
\ifx ..... 6, 11, 14, 19, 23, 25	
L	T
\limits ..... 8, 21, 30	\the ..... 10, 13
	\toks@ ..... 9, 10, 12, 13
M	U
\mathbin ..... 41	\unexpanded ..... 22, 24
\mathop ..... 8, 21, 30	
\mathrel ..... 38	