

# THE SPBMARK PACKAGE

V1.39      2022/03/04

Customize superscript and subscript

Qu Yi

<https://github.com/texl3/spbmark>

[qljx@foxmail.com](mailto:qljx@foxmail.com)

spbmark provides three commands `\super`, `\sub` and `\supersub` to improve the layout of superscript and subscript which can be adjusted the relative position and format, and can be used in text and math mode.

## Table of Contents

<b>1</b>	<b>Macro package options</b>	<b>1</b>	<b>5</b>	<b>Developer commands</b>	<b>7</b>
<b>2</b>	<b>User commands</b>	<b>2</b>	<b>6</b>	<b>Known issues</b>	<b>7</b>
<b>3</b>	<b>Global control interface</b>	<b>4</b>	<b>Index</b>		<b>8</b>
<b>4</b>	<b>Examples of use</b>	<b>6</b>			

## 1 Macro package options

The following macro package options will redefine the script commands of the  $\TeX$  kernel. If you do not specify the values of boolean options, they default to true.

`text = true|false` Default: false  
`\textsuperscript` and `\textsubscript` are equivalent to the `\super` and `\sub` commands.

`math = true|false` Default: false  
`\sp` and `\sb` are equivalent to the `\super` and `\sub` commands.

`foot = true|false` Default: false  
The format of the footnote mark match the superscript move and format of the `\spbset` global setting.

`both`  
The values of `text` and `math` two options are true at the same time.

**all**

The values of **text**, **math** and **foot** three options are true at the same time.

```
\usepackage[both]{spbmark}
\usepackage[text,foot=true]{spbmark}
```

## 2 User commands

There are currently three commands to set superscript and subscript. Their format can be set temporarily using the optional parameters of the command, or set globally using a key-value list, see section 3.

**\super**[ $\langle kv list \rangle$ ]{ $\langle content \rangle$ }[ $\langle kv list \rangle$ ]

This is a superscript output command. The two  $\langle kv list \rangle$  are equivalent.

**\sub**[ $\langle kv list \rangle$ ]{ $\langle content \rangle$ }[ $\langle kv list \rangle$ ]

This is a subscript output command. The two  $\langle kv list \rangle$  are equivalent.

**\supersub**[ $\langle kv list \rangle$ ]{ $\langle super cont \rangle$ }{ $\langle sub cont \rangle$ }[ $\langle kv list \rangle$ ]

This is a command that outputs both superscript and subscript at the same time. You can also use the shorter command **\spb** instead of it. The two  $\langle kv list \rangle$  are equivalent.

**\defspbstyle**{ $\langle style name \rangle$ }[ $\langle kv list \rangle$ ]

Defines the style of the superscript or subscript used for the **style** option.

**\spbifmath**{ $\langle math code \rangle$ }{ $\langle text code \rangle$ }

In some cases, math or text output modes require different code for format or move. This command can be used when using the **match** option or changing the output mode locally, which should be used in the **move** or **format** options. It can switch the corresponding code according to different output modes.

When the horizontal movement distance of the marker is **non-negative**, since the **\super** and **\sub** commands put the marker in a horizontal box, the horizontal position is staggered when using superscript and subscript **consecutively**, which is a different mechanism than the sequential use of  $\wedge$  and  $\_$  symbols in math mode. The following three commands save the width of the previous marker, use it for negative move of the **hmove** option of the next marker, and provide the corresponding alignment.

It should be noted that the following commands should save the longer width of the marker, that is, the longer width of the marker in the front, the shorter width of the marker in the back. This is due to the fact that horizontal movement uses a negative distance to leave the marker in a zero-width box state.

**\llastwd**

Save the width of the previous marker and provide left alignment.

☆ New

☆ New

`\clastwd`

Save the width of the previous marker and provide center alignment.

☆ New

`\rlastwd`

Save the width of the previous marker and provide right alignment.

```

A\super{bcd}\sub{e}B \\
A\super{bcd}\sub[hmove=-\llastwd]{e}B \\
A\sub{e}\super{bcd}[hmove=-\llastwd]B \\
A\super{bcd}\sub[hmove=-\clastwd]{e}B \\
A\super{bcd}\sub[hmove=-\rlastwd]{e}B \\

```

---


$$A^{bcd}_eB$$

$$A^{bcd}_eB$$

$$A^{bcd}_eB$$

$$A^{bcd}_eB$$

$$A^{bcd}_eB$$

☆ New

`\spbshortkv{⟨short opt⟩}{⟨key value⟩}`

Converts any existing key-value pair to a shorthand option. The value of key #1 indicates that the shorthand option needs to be assigned a value.

```

\spbshortkv{x}{sphmove=#1}
\defspbstyle{fancy}{sbcmd=\color{blue},mode=math}
\spbshortkv{mysb}{style=fancy}
A\super[x=2pt]{b} \\
A\sub{b}[mysb]

```

---


$$A^b$$

$$A_b$$

The options common to  $\langle kv \text{ list} \rangle$  of the three commands are as follows. They can also be used in  $\langle key\text{-}value \text{ list} \rangle$  of the `\spbset` command. In this situation, in addition to `style` and `mode`, they will be set according to the type of the previous most recent command.

`vmove = {⟨fixed length⟩}`

Default: 0pt

Vertical move of superscript or subscript. Represents the extra vertical distance `vsep` between superscript and subscript in the superscript and subscript commands.

**hmove** = {⟨fixed length⟩}

Default: 0pt

Horizontal move of superscript or subscript. Represents the common move in the superscript and subscript commands. Moved values can be expressed mathematically:

```
A\super{b}[vmove=0.4ex+4pt/2,hmove=2pt*3-11.5pt] \\  
A\supersub[vsep={3pt,6pt-2bp}]{examsuper}{sub}
```

---

<sup>b</sup>  
A  
A<sup>examsuper</sup>  
sub

**cmd** = {⟨format cmds⟩}

The format commands of superscript or subscript. The last command can take a parameter, which accepts superscript or subscript. Represents the format of superscript and subscript in the superscript and subscript commands.

**cmd+** = {⟨format cmds⟩}

Add code to the previous global superscript or subscript format commands.

**height** = {⟨fixed length⟩}

The distance between the superscript or subscript baseline and the previous line. It is recommended that the superscript adjusts the height and the subscript adjusts the depth.

**depth** = {⟨fixed length⟩}

The distance between the superscript or subscript baseline and the next line.

**style** = {⟨style name⟩}

Use the ⟨style name⟩ defined by the `\defspbstyle` command to make it work global or local.

**mode** = text|math|match

Default: match

The mode of superscript or subscript output can be text or math mode. The match option automatically matches output modes according to the current mode.

### 3 Global control interface

`\spbset{⟨key-value list⟩}`

spbmark uses the `\spbset` command to control the global default format of superscript and subscript. These options also apply to ⟨kv list⟩ of the above commands. The values set by it will be overwritten by the optional parameters of the superscript and other commands.

The following list of keys control the format both of superscript or subscript.

**spvmove** = {⟨fixed length⟩}

Default: 0pt

Extra vertical move of the superscript.

`sphmove` = { $\langle fixed\ length \rangle$ } Default: 0pt  
Extra horizontal move of the superscript.

`sbvmove` = { $\langle fixed\ length \rangle$ } Default: 0pt  
Extra vertical move of the subscript.

`sbhmove` = { $\langle fixed\ length \rangle$ } Default: 0pt  
Extra horizontal move of the subscript.

`nohmove`  
Cancel the horizontal move of superscript and subscript at the same time.

`novmove`  
Cancel the vertical move of superscript and subscript at the same time.

`spcmd` = { $\langle format\ cmds \rangle$ }  
The format commands of superscript. The last command in the code can take an argument, which is a superscript.

`spcmd+` = { $\langle format\ cmds \rangle$ }  
Add code to the previous global superscript format commands.

`sbcmd` = { $\langle format\ cmds \rangle$ }  
The format commands of subscript. The last command in the code can take an argument, which is a subscript.

`sbcmd+` = { $\langle format\ cmds \rangle$ }  
Add code to the previous global subscript format commands.

`spheight` = { $\langle fixed\ length \rangle$ }  
The distance between the superscript baseline and the previous line.

`spdepth` = { $\langle fixed\ length \rangle$ }  
The distance between the superscript baseline and the next line.

`sbheight` = { $\langle fixed\ length \rangle$ }  
The distance between the subscript baseline and the previous line.

`sbdepth` = { $\langle fixed\ length \rangle$ }  
The distance between the subscript baseline and the next line.

If `sphmove` is positive, the superscript or subscript moves to the right, conversely it moves to the left. The positive direction of vertical offset of subscript and superscript is opposite. For superscript, if `spvmove` is positive, the superscript moves up, conversely it moves down. For subscript, if `spvmove` is positive, the subscript moves down, conversely it moves up.

The following list of keys control the format of superscript **and** subscript.

`spbhmove` = { $\langle fixed\ length \rangle$ } Default: 0pt  
Extra vertical move of the superscript and the subscript.

`spbcmd = {\langle super cmds \rangle, \langle sub cmds \rangle}`

The format commands of superscript and subscript. The first part is in superscript format, and the latter part is in subscript format. They are separated by commas, or only the first part exists.

`spbcmd+ = {\langle super cmds \rangle, \langle sub cmds \rangle}`

Add code to the previous global superscript and subscript format commands.

`spbheight = {\langle fixed length \rangle}`

The distance between the superscript baseline and the previous line.

`spbdepth = {\langle fixed length \rangle}`

The distance between the subscript baseline and the next line.

`vsep = {\langle super move \rangle, \langle sub move \rangle}`

Default: 0.6ex

☆ Changed

The extra vertical distance between superscript and subscript. The front part is the movement amount of the superscript, and the latter part is the movement amount of the subscript. If only the front part means that both are vertically shifted by one-half of the current value.

`halign = l|c|r`

Default: l

The alignment of superscript and subscript, which contains l, c, and r parameters respectively for left, center, and right alignment.

## 4 Examples of use

Here is a list of the three commands, please pay attention to the usage of optional parameters. Note when the horizontal move is negative, the starting point is at the right end of the mark.

```
\defspbstyle{fancy}{cmd=\color{purple}}
\spbset{spbcmd={\spbifmath{\mathtt}{\ttfamily},\color{blue}}}
A\super[vmove=0.2ex,hmove=0.2em,cmd=\textcolor{red}]{exam}B \\\
$\sub[style=fancy,cmd+=\mathsf,mode=math]{exam}B$ \\\
A\supersub[vsep=0.6ex,halign=c]{examsuper}{sub}B \\\
A\super{c}[vmove=5pt,hmove=-5.5pt]B\sub[vmove=5pt,hmove=-5pt]{d}AB
```

```
A examB
AexamB
AexamsuperB
A subB
c
ABAB
d
```

It can also be used with the siunitx package to output superscript and subscript in the unit:

```
\unit[mode=math]{kg.m/s\super[vmove=-1pt]{2}} \\\
\qty[mode=text]{30}{A\supersub[hmove=1pt,cmd=\color{red}]{b}{c}} \\\
```

```
\spbset{sbhmove=2pt}\unit[mode=text]{A\sub{b}}
```

kg m/s<sup>2</sup>

30 A<sub>c</sub><sup>b</sup>

A<sub>b</sub>

spbmark also patches the footer markers for standard document class and KOMA-Script. You can format the footer markers by redefining the `\fnmarkfont` command. Note that extra horizontal move does not work with footnote markers.

## 5 Developer commands

If you need to use the original definitions of `\textsuperscript`, `\textsubscript`, `\sp` and `\sb` after using the `text` or `math` option, then you can use the following commands:

```
\spb@textsuperscript@save{\content}
```

Save the original definition of the `\textsuperscript` command, output superscript in the text mode.

```
\spb@textsubscript@save{\content}
```

Save the original definition of the `\textsubscript` command, output subscript in the text mode.

```
\spb@math@super@save{\content}
```

Save the original definition of the `\sp` command, output superscript in the math mode.

```
\spb@math@sub@save{\content}
```

Save the original definition of the `\sb` command, output subscript in the math mode.

## 6 Known issues

At present, the vertical and horizontal move are effective for the unit commands in the siunitx macro package. However, due to the special mechanism that the decimal point is not recognized correctly because it's converted to a space in the `\unit` command, it's recommended to use pt as the unit of move.

## References

- [Rob16] Will ROBERTSON. realscripts. version 0.3d, Feb. 13, 2016 (or newer).  
URL: <https://www.ctan.org/pkg/realscripts>.
- [Tea22] The L<sup>A</sup>T<sub>E</sub>X Project TEAM. The L<sup>A</sup>T<sub>E</sub>X<sub>3</sub> Interfaces. Feb. 24, 2022 (or newer).  
URL: <https://ctan.org/pkg/l3kernel>.
- [Wri22] Joseph WRIGHT. siunitx. version 3.0.48, Feb. 7, 2022 (or newer).  
URL: <https://www.ctan.org/pkg/siunitx>.

## Index

<b>A</b>		<b>S</b>	
<code>all</code> .....	2	<code>\sb</code> .....	1
<b>B</b>		<code>sbcmd</code> .....	5
<code>both</code> .....	1	<code>sbcmd+</code> .....	5
<b>C</b>		<code>sbdepth</code> .....	5
<code>\clastwd</code> .....	3	<code>sbheight</code> .....	5
<code>cmd</code> .....	4	<code>sbhmove</code> .....	5
<code>cmd+</code> .....	4	<code>sbvmove</code> .....	5
<b>D</b>		<code>\sp</code> .....	1
<code>\defspbstyle</code> .....	2, 4	<code>\spb</code> .....	2
<code>depth</code> .....	4	<code>spbcmd</code> .....	6
<b>F</b>		<code>spbcmd+</code> .....	6
<code>foot</code> .....	1 f.	<code>spbdepth</code> .....	6
<b>H</b>		<code>spbheight</code> .....	6
<code>halign</code> .....	6	<code>spbhmove</code> .....	5
<code>height</code> .....	4	<code>\spbifmath</code> .....	2
<code>hmove</code> .....	2, 4	<code>\spbset</code> .....	1, 3 f.
<b>L</b>		<code>\spbshortkv</code> .....	3
<code>\llastwd</code> .....	2	<code>spcmd</code> .....	5
<b>M</b>		<code>spcmd+</code> .....	5
<code>math</code> .....	1 f.	<code>spdepth</code> .....	5
<code>mode</code> .....	3 f.	<code>spheight</code> .....	5
<b>N</b>		<code>sphmove</code> .....	5
<code>nohmove</code> .....	5	<code>spvmove</code> .....	4 f.
<code>novmove</code> .....	5	<code>style</code> .....	2 ff.
<b>R</b>		<code>\sub</code> .....	1 f.
<code>\rlastwd</code> .....	3	<code>\super</code> .....	1 f.
		<code>\supersub</code> .....	1 f.
		<b>T</b>	
		<code>text</code> .....	1 f.
		<code>\textsubscript</code> .....	1
		<code>\textsuperscript</code> .....	1
		<b>V</b>	
		<code>vmove</code> .....	3
		<code>vsep</code> .....	3, 6