

Improving “`setspace`”

Markus Kohm*

Version 2023-08-04 v1.0

Package `setspaceenhanced` has started as hack module of the KOMA-Script package `scrhack` years ago to fix an issue when using package `setspace` with other document font sizes than 10 pt, 11 pt or 12 pt. This became necessary because package `setspace` originally only supported these three font sizes and loading the package with a floating point definition of `\@ptsize` even resulted in errors. These two issues has been fixed some years ago. Now, `setspace` uses a static factor for all font sizes but 10 pt, 11 pt, or 12 pt.

Additionally, if you change font size inside the document `setspace` still uses the stretch factor of the document font size instead of using a proper stretch factor for the new font size.

Package `setspaceenhanced` provides improvements for all these limitations and also some additional enhancements.

Contents

1 Why should I use this package instead of <code>setpace</code> at least if I use a KOMA-Script class or KOMA-Script package <code>scrxextend</code> or a similar package?	2
2 Why should I use this package instead of <code>setpace</code> independent from using a KOMA-Script class or KOMA-Script package <code>scrxextend</code> or similar packages?	2
3 How to use <code>setspaceenhanced</code>?	3
4 How does the result of <code>setspaceenhanced</code> differ from <code>setspace</code> even for using the standard font sizes of the standard classes?	5
5 Implementation	6
References	9
Index	10
Change History	10

*Repository and bug reports: <https://github.com/komascript/third-party-enhancements>

1 Why should I use this package instead of **setpace** at least if I use a KOMA-Script class or KOMA-Script package **scrxextend** or a similar package?

From 2006 **KOMA-Script** classes and KOMA-Script package **scrxextend** provide option **fontsize** for setting document font sizes not limited to 10 pt, 11 pt, or 12 pt. Using this option is also not limited to integer font sizes but also supports floating point sizes like 11.5 pt. To support such font sizes, the specification of macro `\@ptsize` has been changed to be no longer either 0, 1 or 2 but to be the font size minus 10 in pt.

By the way, e.g., Ivan Valbusa's package **fontsize** adopted this existing definition of `\@ptsize` from KOMA-Script by using most of the font size code of KOMA-Script.

Unfortunately package **setspace** provides stretch factors for individual font sizes only for 10 pt, 11 pt or 12 pt. For all sizes between these size, it uses the factor of the down rounded integer. For all sizes below 10 pt or 11 pt it uses a static value, 1.25 for `\onehalfspacing` and 1.667 for `\doublespacing`. Package **setspaceenhanced** uses for `\onehalfspacing` and `\doublespacing` a calculation of the stretch factor depending on the selected baseline skip and font size. With this, every font size is supported.

But that's just the tip of the iceberg and even comparatively unimportant. Much more important are the reasons in the following section, which also apply when using a KOMA-Script class or the KOMA-Script package **scrxextend** or a similar package.

2 Why should I use this package instead of **setpace** independent from using a KOMA-Script class or KOMA-Script package **scrxextend** or similar packages?

setspace does not care for the `\baselineskip` selected by `\fontsize`. Instead it sets the stretch factor always depending on the document font size. And using `\singlespacing`, `\onehalfspacing` or `\doublespacing` or even `\setstretch` after switching the font size using `\fontsize` instead of `\Huge`, `\huge`, `\LARGE`, `\Large`, `\large`, `\normalsize`, `\small`, `\footnotesize`, `\scriptsize`, `\tiny`, or another command defined using `\@setsize`, reactivates the last used such font size commd. So using something like

```
\normalsize\fontsize{5pt}{7pt}\selectfont\onehalfspacing
```

result in `\normalsize` with `onehalfspacing`, not 5 pt with `onehalfspacing`!

Package **setspaceenhanced** uses for `\onehalfspacing` and `\doublespacing` a calculation of the stretch factor depending on the selected baseline skip and font size. So using something like

```
\normalsize\fontsize{5pt}{7pt}\onehalfspacing
```

will not set the stretch factor based on `\normalsize` but based on font size 5 pt with baseline skip 7 pt. So this results in real `onehalfspacing` of the 5 pt font.

In other words: Package **setspaceenhanced** uses a completely different definition of `\onehalfspacing` and `\doublespacing`. It always uses `\f@size` and `\f@baselineskip` to calculate a factor resulting in real `onehalfspacing` and `doublespacing`. This also means, if you use one of these commands after changing the font size, a new stretch factor is calculated depending on the current font size without changing the font size.

3 How to use setspaceenhanced?

In the document preamble of your document you just can replace

```
\usepackage{setspace}
```

by

```
\usepackage{setspaceenhanced}
```

to load package `setspaceenhanced`. This does still also load package `setspace` but additionally replaces several commands of `setspace` to avoid the issues shown in [section 1](#) and [section 2](#). `setspaceenhanced` also does support the same options as `setspace`. So you can also replace, e.g.

```
\usepackage[onehalfspacing]{setspace}
```

by

```
\usepackage[onehalfspacing]{setspaceenhanced}
```

If you want you can alternatively also load both packages explicitly, either `setspace` before `setspaceenhanced` or—if you want—`setspaceenhanced` before `setspace`. In this case, you should use the same options for both package.

This is also useful, if you use a package, that uses `setspace` itself. In this case, you always should load `setspaceenhanced` *before* the package, that uses `setspace`. Otherwise it is very likely that the initial line spacing is still done with the unchanged commands and settings of `setspace` and therefore the full functionality of `setspaceenhanced` cannot be reached. Only if `setspaceenhanced` is loaded before the first use of `\singlespacing`, `\onehalfspacing`, `\doublespacing`, or `\setstretch` can it be ensured that the enhancements of `setspaceenhanced` are initialized and used correctly.

When using a class that uses `setspace`, the correct operation can be ensured with

```
\AddToHook{package/setspace/after}{\RequirePackage{setspaceenhanced}}
```

even before `\documentclass`. This requires at least L^AT_EX 2020/10/01. For older versions of L^AT_EX you can use

```
\RequirePackage{scrfile}  
\AfterPackage{setspace}{\RequirePackage{setspaceenhanced}}
```

also before `\documentclass`. This would require the KOMA-Script package `scrfile`. In both cases you should also use the same optional argument for `\RequirePackage`, that is used for loading `setspace`.

Package `setspaceenhanced` provides all options and commands of the user interface of `setspace`, see [\[TF22\]](#). Following we document only the differences and enhancements.

There are some options influencing the behavior and result of the examples shown in the section before. All these options are $\langle key \rangle = \langle value \rangle$ options using the new L^AT_EX kernel interface. Therefore the package needs at least L^AT_EX 2022-06-01.

`\spacesetup`

Options can be set as global option via `\documentclass`, as package option via `\usepackage` or using:

```
\spacesetup{\langle options \rangle}
```

The command can be used in the document preamble and also in the document body. In the document body the changes are local to the current group.

Available options:

<code>byselectfont</code> (<i>opt.</i>)	<code>byselectfont=<boolean></code>	<code>initial=false</code> , <code>default=true</code>
<code>\selectfont</code>	In the <code>setspaceenhanced</code> examples in the previous section, the correct factor has only been used, because of using <code>\onehalfspacing</code> after changing the font size, e.g., to <code>\small</code> . If you use <code>\onehalfspacing</code> before changing the font size, the factor is calculated with the previous valid font size, which is the document font size 12pt in all these examples. This behavior can be changed to a more dynamic automatism using option <code>byselectfont</code> or <code>byselectfont=true</code> . This will use the generic L ^A T _E X hook <code>selectfont</code> to reactivate <code>\onehalfspacing</code> or <code>\doublespacing</code> after every <code>\selectfont</code> if the font size has been changed.	
<code>onehalfspacing</code> (<i>opt.</i>)	<code>doublespacing=<real></code>	<code>initial=nan</code> , <code>default=empty</code>
	If this option is used without value, it is the same as <code>setspace</code> 's package option <code>doublespacing</code> or using command <code>\doublespacing</code> . But if you assign a real number ¹ this would be used as the new stretch factor used for <code>doublespacing</code> . This also means, that the default calculation of the factor is deactivated. But a factor of <code>nan</code> would reactivate the calculation of the factor depending on the font size and the baseline skip set for the font size. It is recommended to use the option always without value!	
<code>fontsize</code> (<i>opt.</i>)	<code>fontsize=<boolean></code>	<code>initial=false</code> , <code>default=true</code>
<code>\setstretch</code>	As explained in section 2 , <code>setspace</code> 's <code>\setstretch</code> behaves different after a font size command like <code>\Huge</code> , <code>\huge</code> , <code>\LARGE</code> , <code>\Large</code> , <code>\large</code> , <code>\normalsize</code> , <code>\small</code> , <code>\footnotesize</code> , <code>\scriptsize</code> , <code>\tiny</code> , or another command defined using <code>\@setsize</code> than after using <code>\fontsize{...}{...}\selectfont</code> . With the last the font size will be reset to the previous usage of one of the other or the document font size. For a lot of users this is somehow unexpected. With option <code>fontsize</code> or <code>fontsize=true</code> this is changed and using <code>\setstretch</code> does not reactivate the last used <code>\Huge</code> , <code>\huge</code> , <code>\LARGE</code> , <code>\Large</code> , <code>\large</code> , <code>\normalsize</code> , <code>\small</code> , <code>\footnotesize</code> , <code>\scriptsize</code> , <code>\tiny</code> .	
<code>onehalfspacing</code> (<i>opt.</i>)	<code>onehalfspacing=<real></code>	<code>initial=nan</code> , <code>default=empty</code>
	If this option is used without value, it is the same as <code>setspace</code> 's package option <code>onehalfspacing</code> or using command <code>\onehalfspacing</code> . But if you assign a real number ¹ this would be used as the new stretch factor used for <code>onehalfspacing</code> . This also means, that the default calculation of the factor is deactivated. But a factor of <code>nan</code> would reactivate the calculation of the factor depending on the font size and the baseline skip set for the font size. It is recommended to use the option always without value!	
<code>singlespacing</code> (<i>opt.</i>)	<code>singlespacing=<real></code>	<code>initial=1</code> , <code>default=empty</code>
	If this option is used without value, it is the same as <code>setspace</code> 's package option <code>singlespacing</code> or using command <code>\singlespacing</code> . But if you assign a real number ¹ this would be used as the new stretch factor used for <code>singlespacing</code> . So this is similar to using <code>\SetSingleSpace{<real>}\singlespacing</code> . A factor of <code>nan</code> would activate the calculation of the factor depending on the font size and the baseline skip set for the font size. It is recommended to use the option always without value!	

¹Here are all values allowed, that would be allowed as second argument of L^AT_EX3 function `\fp_set:Nn`

Compatibility Notes:

scrhack If you want to use this package together with package **scrhack** from **KOMA-Script** before version 3.42, you should deactivate the **setspace** hack using **scrhack**' option **setspace=false**. From version 3.42 **scrhack** does not use the old hacks any longer but **setspaceenhanced** and is therefore compatible again.

4 How does the result of **setspaceenhanced** differ from **setspace** even for using the standard font sizes of the standard classes?

For example if you have a document:

```
\documentclass[12pt]{article}
\usepackage{setspace}
\begin{document}
\small\onehalfspacing This is font size
\csname f@size\endcsname pt with
normal baseline skip \csname f@baselineskip\endcsname.
The current stretch factor is \baselinestretch. This
results in a baseline skip of \the\baselineskip.
\end{document}
```

this will result in:

This is font size 10.95pt with normal baseline skip 13.6pt. The current stretch factor is 1.241. This results in a baseline skip of 16.87756pt.

But one moment: $10.95 \text{ pt} \cdot 1.5 = 16.425 \text{ pt}$. So the factor seems to be wrong. It is not real **onehalfspacing** depending on the used font size. It is also not **onehalfspacing** depending on the document font size, because this would need a baseline skip of 18pt. So what is it? It is using the stretch factor of 12pt for the 10.95pt of **\small**.

With **setspaceenhanced**:

```
\documentclass[12pt]{article}
\usepackage{setspaceenhanced}
\begin{document}
\small\onehalfspacing This is font size
\csname f@size\endcsname pt with
normal baseline skip \csname f@baselineskip\endcsname.
The current stretch factor is \baselinestretch. This
results in a baseline skip of \the\baselineskip.
\end{document}
```

the result changes:

This is font size 10.95pt with normal baseline skip 13.6pt. The current stretch factor is 1.207720046225135. This results in a baseline skip of 16.42496pt.

Here the difference from the correct value 16.425pt is very, very small: 0.00004pt. So you can say, this is really **onehalfspacing** depending on the used font size.

Moreover if you have a document:

```
\documentclass[12pt]{article}
\usepackage{setspace}
\begin{document}
\fontsize{5pt}{7pt}\selectfont\onehalfspacing This is font size
\csname f@size\endcsname pt with
normal baseline skip \csname f@baselineskip\endcsname.
The current stretch factor is \baselinestretch. This
results in a baseline skip of \the\baselineskip.
\end{document}
```

this result in:

This is font size 12pt with normal baseline skip 14.5pt. The current stretch factor is 1.241. This results in a baseline skip of 17.99446pt.

But

```
\documentclass[12pt]{article}
\usepackage[fontsize]{setspaceenhanced}
\begin{document}
\fontsize{5pt}{7pt}\selectfont\onehalfspacing This is font size
\csname f@size\endcsname pt with
normal baseline skip \csname f@baselineskip\endcsname.
The current stretch factor is \baselinestretch. This
results in a baseline skip of \the\baselineskip.
\end{document}
```

results in:

This is font size 5pt with normal baseline skip 7.0pt. The current stretch factor is 1.071428571428571. This results in a baseline skip of 7.49998pt.

In my opinion this is more the expected result. See the previous [section 3](#) for more information about options like `fontsize`.

5 Implementation

We use the new L^AT_EX kernel feature of $\langle key \rangle = \langle value \rangle$ options introduced in [\[TLT22\]](#). So we need at least L^AT_EX 2022-06-01:

```
1 \ifnum 0=\ifcsname IfFormatAtLeastTF\endcsname
2   \IfFormatAtLeastTF{2022-06-01}{1}{0}
3   \else
4     0
5   \fi\relax
6   \PackageError{setspaceenhanced}{LaTeX~kernel~too~old}{
7     The~package~needs~at~least~LaTeX~2022-06-01.\MessageBreak
8     This~error~is~fatal.~Loading~will~be~aborted
9   }
10  \endinput
11 \fi
```

We do not pass any options to `setspace`, because we handle them different. So we can just load the package here:

```
12 \RequirePackage{setspace}
```

`\c_@@_single_fp` (*const.*) The internal constants store the absolute factor for singlespacing, onehalfspacing and doublespacing used to calculate the stretch factors.

```
\c_@@_onehalf_fp (const.)
\c_@@_double_fp (const.)
13 \fp_const:Nn \c_@@_single_fp { 1.2 }
14 \fp_const:Nn \c_@@_onehalf_fp { 1.5 }
15 \fp_const:Nn \c_@@_double_fp { 2.0 }
```

`\@@_single_factor_fp` (*var.*) The internal variables used to store the configured stretch factors for singlespacing, onehalfspacing and doublespacing. If `\nan \@@_linespread` is calculated.

```
\@@_onehalf_factor_fp (var.)
\@@_double_factor_fp (var.)
16 \fp_new:N \@@_single_factor_fp \fp_set:Nn \@@_single_factor_fp { 1.0 }
17 \fp_new:N \@@_onehalf_factor_fp \fp_set_eq:NN \@@_onehalf_factor_fp \c_nan_fp
18 \fp_new:N \@@_double_factor_fp \fp_set_eq:NN \@@_double_factor_fp \c_nan_fp
```

`\@@_linespread_fp` (*var.*) Storage of the current calculated stretch factor and the used constant.

```
\@@_fp (var.)
19 \fp_new:N \@@_linespread_fp \fp_set:Nn \@@_linespread_fp { 1.0 }
```

`\@@_set_spacing:nn` These commands are used to set the stretch factor for one of the spacings.

```
20 \cs_new:Nn \@@_set_spacing:nn
21 {
22   \tl_if_blank:nF { #1 }
23   {
24     \fp_set:cn { @@_#2_factor_fp } { #1 }
25   }
26   \fp_if_nan:nTF { \tl_use:c { @@_#2_factor_fp } }
27   {
28     \fp_set_eq:Nc \@@_fp { c_@@_#2_fp }
29     \@@_calc_stretch:
30   }
31   {
32     \fp_set_eq:Nc \@@_linespread_fp { @@_#2_factor_fp }
33     \fp_set_eq:NN \@@_fp \c_nan_fp
34   }
35   \setstretch{ \fp_to_decimal:N \@@_linespread_fp }
36 }
```

`\@@_calc_stretch:` This macro is used to (re-)calculate the stretch factor `\@@_calcstretch` if the the current used constant is not `\nan`.

```
37 \cs_new:Nn \@@_calc_stretch:
38 {
39   \fp_if_nan:nF { \@@_fp }
40   {
41     \fp_set:Nn \@@_linespread_fp
42     {
43       \f@size / \dim_to_fp:n { \f@baselineskip } * \@@_fp
44     }
45   }
46 }
```

`\selectfont` We also need to hook into `\selectfont` using the general `selectfont` hook to re-calculate the stretch factor after the font size has been changed and `byselectfont=true`.

```

47 \hook_gput_code:nmn { selectfont } { setspaceenhanced }
48 {%
49   \bool_if:NT \__setspaceenhanced_byselectfont_bool
50   {
51     \cs_if_exist:NT \size@update
52     {
53       \fp_set:Nn \l_tmpa_fp { \f@linespread }
54       \fp_compare:nNnT \__setspaceenhanced_linespread_fp = \l_tmpa_fp
55       {
56         \__setspaceenhanced_calc_stretch:
57         \set@fontsize
58         {
59           \fp_to_decimal:N \__setspaceenhanced_linespread_fp
60         }
61         \f@size \f@baselineskip
62       }
63     }
64   }
65 }

```

`\setstretch` If `fontsize=false`, we use the original functionality of `spacespace`. But with `fontsize=true` we use `\linespread`:

```

66 \renewcommand*{\setstretch}[1]{
67   \bool_if:NTF \@@_fontsize_bool
68   {
69     \linespread{#1}\selectfont
70   }
71   {
72     \def\baselinestretch{#1}%
73     \@currsize
74   }
75 }

```

Almost the end of the package we define all options:

```

76 \DeclareKeys{%
77   singlespacing .code      = \@@_set_spacing:nm {#1} {single},
78   singlespacing .usage    = general,
79   onehalfspacing .code    = \@@_set_spacing:nm {#1} {onehalf},
80   onehalfspacing .usage  = general,
81   doublespacing .code    = \@@_set_spacing:nm {#1} {double},
82   doublespacing .usage   = general,
83   byselectfont .bool_set:N = \@@_byselectfont_bool,
84   byselectfont .default:n = true,
85   byselectfont .initial:n = false,
86   fontsize .bool_set:N   = \@@_fontsize_bool,
87   fontsize .default:n    = true,
88   fontsize .initial:n    = false,
89 }

```


And process them:

```
90 \ProcessKeyOptions\relax
```

`\singlespacing` We have to redefine the user interface commands for singlespacing, onehalfspacing and
`\onehalfspacing` doublespacing.

```
\doublespacing 91 \renewcommand*\singlespacing){
\SetSingleSpace 92  \_setspaceenhanced\_set\_spacing:nn {} {single}
93 }
94 \renewcommand*\onehalfspacing){
95  \_setspaceenhanced\_set\_spacing:nn {} {onehalf}
96 }
97 \renewcommand*\doublespacing){
98  \_setspaceenhanced\_set\_spacing:nn {} {double}
99 }
100 \renewcommand*\SetSingleSpace}[1]{
101  \fp\_set:Nn \@@\_single\_factor\_fp { #1 }
102 }
```

`\setspace@singlespace` **Note:** Defining this internal macro does not work using `setspaceenhanced`. Should we add a test for users and package authors, who do not use `\SetSingleSpace` but redefine the internal macro?

`spacesetup` User interface to not need to use `\SetKeys`:

```
103 \newcommand*\spacesetup}{\SetKeys[setspaceenhanced]}
```

References

- [Car+22] David Carlisle et al. *setspace* — *Set space between lines*. Version 6.7b. Provides support for setting the spacing between lines in a document. Package options include `singlespacing`, `onehalfspacing`, and `doublespacing`. Alternatively the spacing can be changed as required with the `\singlespacing`, `\onehalfspacing`, and `\doublespacing` commands. Other size spacings also available. Dec. 4, 2022. URL: <https://ctan.org/pkg/setspace> (visited on 07/25/2023).
- [Koh23a] Markus Kohm. *KOMA-Script* — *A bundle of versatile classes and packages*. Version 3.41. The KOMA-Script bundle provides replacements for the `article`, `report`, and `book` classes with emphasis on typography and versatility. There is also a letter class. July 7, 2023. URL: <https://ctan.org/pkg/koma-script> (visited on 07/14/2023).
- [Koh23b] Markus Kohm. *scrfile* — *Installation control (not only) for KOMA-Script packages*. Version 3.41. The package provides hooks for the execution of commands before or after loading files, classes or packages independent from the L^AT_EX kernel version. July 7, 2023. URL: <https://ctan.org/pkg/scrfile> (visited on 07/19/2023).
- [TF22] Geoffrey Tobin and Robin Fairbairns. *The setspace Package*. Dec. 4, 2022. URL: <https://ctan.org/tex-archive/macros/latex/contrib/setspace> (visited on 07/25/2023).

[TLT22] The L^AT_EX Project Team. “Issue 35.” In: *L^AT_EX News* (June 2022). URL: <http://mirrors.ctan.org/macros/latex/base/ltnews35.pdf> (visited on 07/14/2023).

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		<code>\singlespacing</code> 91	<code>\SetSingleSpace</code> 91
<code>\@@_double_factor_fp</code>		<code>\spacesetup</code> 3 , 103	<code>\setstretch</code> 2 , 4 , 66
<i>(var.)</i> 16	Constants:	<code>\singlespacing</code> 91	<code>\singlespacing</code> (<i>opt.</i>) 4
<code>\@@_fp</code> (<i>var.</i>) 19	<code>\c_@@_double_fp</code> 13	<code>\spacesetup</code> 3 , 103	
<code>\@@_linespread_fp</code> (<i>var.</i>) 19	<code>\c_@@_onehalf_fp</code> 13		
<code>\@@_onehalf_factor_fp</code>	<code>\c_@@_single_fp</code> 13		
<i>(var.)</i> 16			
<code>\@@_single_factor_fp</code>	D		T
<i>(var.)</i> 16	<code>\doublespacing</code> 91		TeX macros (internal):
			<code>\@@_calc_stretch:</code> . . . 37
B			<code>\@@_set_spacing:nn</code> . . 20
<code>byselectfont</code> (<i>opt.</i>) 4	F		<code>\setspace@singlespace</code>
	<code>fontsize</code> (<i>opt.</i>) 4	 103
	O		V
C	<code>\onehalfspacing</code> 91		Variable:
<code>\c_@@_double_fp</code> (<i>const.</i>) 13	<code>onehalfspacing</code> (<i>opt.</i>) . . 4 , 4		<code>\@@_double_factor_fp</code>
<code>\c_@@_onehalf_fp</code> (<i>const.</i>) 13	Options:	 16
<code>\c_@@_single_fp</code> (<i>const.</i>) 13	<code>byselectfont</code> 4		<code>\@@_fp</code> 19
Commands:	<code>fontsize</code> 4		<code>\@@_linespread_fp</code> . . 19
<code>\doublespacing</code> 91	<code>onehalfspacing</code> 4 , 4		<code>\@@_onehalf_factor_fp</code>
<code>\onehalfspacing</code> 91	<code>singlespacing</code> 4	 16
<code>\selectfont</code> 4 , 47			<code>\@@_single_factor_fp</code>
<code>\SetSingleSpace</code> 91	S	 16
<code>\setstretch</code> 2 , 4 , 66	<code>\selectfont</code> 4 , 47		

Change History

v0.1 – 2023/06/01

 General: new KOMA-Script spin-off [1](#)