

# letterspaceadjust

v0.0a

Arno Trautmann

[arno.trautmann@gmx.de](mailto:arno.trautmann@gmx.de)

This is the documentation of the package `letterspaceadjust`. It is not documented so far.

*Attention:* This package is under development and everything presented here might be subject to incompatible changes.

If you have any suggestions or comments, just drop me a mail, I'll be happy to get any response! The latest source code is hosted on github: <https://github.com/alt/letterspaceadjust>. Feel free to comment or report bugs there, to fork, pull, etc.

<p>This package is copyright © 2013 Arno L. Trautmann. It may be distributed and/or modified under the conditions of the LaTeX Project Public License, either version 1.3c of this license or (at your option) any later version. This work has the LPPL maintenance status ‘maintained’.</p>
---

# Contents

<b>I</b>	<b>User Documentation</b>	<b>3</b>
1	What is this about?	3
2	How do I use it?	3
2.1	Controlling the parameters . . . . .	3
2.2	Example use . . . . .	4
<b>II</b>	<b>Implementation</b>	<b>5</b>
3	The $\text{\LaTeX}$ package: <code>letterspaceadjust.sty</code>	5
4	Lua Module	6
4.1	setup of variables . . . . .	6
4.2	function implementation . . . . .	7
4.3	<code>textletterspaceadjust</code> . . . . .	7

## Part I

# User Documentation

## 1 What is this about?

This package provides an experimental approach of improving the result of typesetting in very narrow columns. For this, the distance between letters is stretched evenly by a very small amount – in the best case so small that you hardly notice. In the worst case so much that it disturbs the reading. This very implementation is just for testing purposes – please play around with the parameters and report your results so I can see whether this package will be of any use. If so, an official CTAN release will follow when the package is stable and useable.

## 2 How do I use it?

The basic macro of this package is `\letterspaceadjust`. Using this macro, one activates the additional spacing for the rest of the document – until an `\unletterspaceadjust` stops it. These macros work per paragraph, so you cannot (de)activate it for single lines. For this, you can use the macro `\textletterspaceadjust` or `\textlsa{}` which activates spacing for everything inside the braces.

### 2.1 Controlling the parameters

Using `letterspaceadjust` without the `microtype` package is dumb. Therefore, the package requires `microtype` and offers two commands for convenience to change the `microtype` parameters. These are just shorthands and shall help in finding good parameters:

`\lsamicrostretch{}` Changes the amount by which glyphs are extended. Default value in `microtype` is 20 (measured in 1/1000, so 20 means glyphs are extended at max. 2% of their width) However, I find that values up to 40 are useable and improve the result in narrow columns. Try it out. A larger expansion also helps to increase the greyness which is needed when we increase the letter spacing.

`\lsamicroshrink{}` Same for the amount of shrinking. By default, this is also set to 20 and if not changed explicitly, the shrink value is the same as the stretch value. I think that the shrinking should not be increased when using `letterspaceadjust`, so try setting (i. e. keeping) it to 20.

The second way of maybe improving the quality of very narrow columns is to increase the value of `\emergencystretch`. In simple words, this is the amount of additional stretchable material that is added if `TEX` cannot find a good solution of the paragraph. This may lead to very loose columns, but that is exactly what the letterspacing will improve. So a larger `emergencystretch` will help us here. Maybe. If you really want to understand what `\emergencystretch` does, please consult the `TEXbook`.

`\lsaemergency{}` Sets the additional stretchable material. Try values below 1em or so. Use font related units (as em) to be independent of the current font.

Finally, there are macros that directly change what this package does. Mainly this is the amount of spacing that is used between the glyphs. This value should be very small; everything above 0.05em looks too loose and unreadable.

`\lsastretch{}` Sets the amount of stretch to be inserted between the glyphs. Values below 0.05em seem reasonable.

In a final version of this package, there will be of course a key-value interface and also a set of pre-defined values à la “very strong letterspacing”, “light letterspacing” etc. Do not forget to consider the `\textletterspaceadjust-variant` to be used when the linewidth changes during the paragraph. (e.g. pictures in the text)

## 2.2 Example use

For testing, you might want to consider the following example. Remember that you cannot change the amount of microtypes stretching after `\begin{document}`, therefore I cannot show different expansions here.

The examples are typeset with the following setting, from left to right. The amount of font expansion in this document is set more than twice as high as the default value (45 instead of 20) to show more effect.

- without expansion or letterspacing,
- with expansion, but without letterspacing
- without expansion, but with letterspacing
- with expansion and letterspacing

This is a small  
text with  
some words  
that should  
show how  
to use this  
package.

This is a small  
text with  
some words  
that should  
show how  
to use this  
package.

This is a small  
text with some  
words that  
should show  
how to use this  
package.

This is a small  
text with some  
words that  
should show  
how to use this  
package.

The default values correspond to the following settings:

```
\lsaemergency{0.0em}  
\lsastretch{0.02em}  
\lsamicrostretch{20}  
\lsamicroshrink{20}
```

## Part II

# Implementation

### 3 The $\LaTeX$ package: letterspaceadjust.sty

Everything to get stuff working from the  $\TeX$  side. Here, only a .sty file is provided and plain/Con $\TeX$ t users have to find their way. I'll happily support them, though!

The Lua file is not found by using a simple `dofile("letterspaceadjust.lua")` call, but we have to use `kpse's find_file`.

```
1 \ProvidesPackage{letterspaceadjust}%
2 [2013/03/10 v0.0b letterspaceadjust package]
3 \input{luatexbase.sty}
4 \directlua{dofile(kpse.find_file("letterspaceadjust.lua"))}
```

Commands for (un-)registering the functions in the callbacks.

```
5 \def\letterspaceadjust{\directlua{luatexbase.add_to_callback(%
6 "pre_linebreak_filter",letterspaceadjust,"letterspaceadjust")}}
7 \def\unletterspaceadjust{
8 \directlua{luatexbase.remove_from_callback("pre_linebreak_filter","letterspaceadjust")}}
```

Synonyms, just to be funny:

```
9 \let\stealsheep\letterspaceadjust %% synonym in honor of Paul
10 \let\unstealsheep\unletterspaceadjust
11 \let\returnsheep\unletterspaceadjust
```

Now the setup for the  $\text{\texttt{text}}$ -versions. We utilize Lua $\TeX$ s attributes to mark all nodes that should be manipulated. The macros should be  $\text{\texttt{long}}$  to allow arbitrary input.

```
12 \newluatexattribute\letterspaceadjustattr
13
14 \long\def\textletterspaceadjust#1{
15 \setluatexattribute\letterspaceadjustattr{42}#1\unsetluatexattribute\letterspaceadjustattr
16 \directlua{
17 if (textletterspaceadjustactive) then else % -- if already active, do nothing
18 luatexbase.add_to_callback(
19 "pre_linebreak_filter",textletterspaceadjust,"textletterspaceadjust")
20 end
21 textletterspaceadjustactive = true % -- set to active
22 }
23 }
24 \let\textlsa\textletterspaceadjust
```

Macros to control the behaviour of the package. First, macros that go to the microtype package:

```
25 \def\lsamicrostretch#1{\microtypesetup{stretch=#1}}
26 \def\lsamicroshrink#1{\microtypesetup{shrink=#1}}
```

Then a macro that just executes  $\text{\texttt{emergencystretch}}=\text{\texttt{#1}}$

```
27 \def\lsaemergency#1{\emergencystretch=#1}
```

Finally, the parameters of the lua function, mainly the amount of stretchability.

```
28 \def\lsastretch#1{%
29   \directlua{
30     lsa_stretch_ind = lsa_stretch_ind + 1
31     letterspace_spec[lsa_stretch_ind] = node.copy(letterspace_spec[1])
32     letterspace_glue.spec = letterspace_spec[lsa_stretch_ind]
33     letterspace_spec[lsa_stretch_ind].stretch = tex.sp"#1"
34   }%
35 }
```

## 4 Lua Module

This file contains all the necessary functions and is the actual work horse of this package. Yet another piece of code by Paul. This is primarily intended for very narrow columns, but may also increase the overall quality of typesetting. Basically, it does nothing else than adding expandable space *between* letters. This way, the amount of stretching between words can be reduced which will, hopefully, result in the greyness to be more equally distributed over the page.

Why the synonym `stealsheep`? Because of a comment of Paul on the `texhax` mailing list: <http://tug.org/pipermail/texhax/2011-October/018374.html>

### 4.1 setup of variables

First, we set up some constants that are used by many of the following functions. These are made global so the code can be manipulated at the document level, too.

```
36 local nodenew = node.new
37 local nodecopy = node.copy
38 local nodeinsertbefore = node.insert_before
39 local nodeinsertafter = node.insert_after
40 local nodeid = node.id
41 local nodetraverseid = node.traverse_id
42
43 Hhead = nodeid("hhead")
44 RULE = nodeid("rule")
45 GLUE = nodeid("glue")
46 WHAT = nodeid("whatsit")
47 COL = node.subtype("pdf_colorstack")
48 GLYPH = nodeid("glyph")
49
50 lsa_stretch_ind = 1
51 letterspace_spec = {}
52 letterspace_spec[lsa_stretch_ind] = nodenew(nodeid("glue_spec"))
53 letterspace_glue = nodenew(nodeid("glue"))
54 local letterspace_pen = nodenew(nodeid("penalty"))
55
56 letterspace_spec.width = tex.sp"0pt"
```

```

57 letterspace_spec[lsa_stretch_ind].stretch = tex.sp"0.02em"
58 letterspace_glue.spec      = letterspace_spec[lsa_stretch_ind]
59 letterspace_pen.penalty    = 10000

```

## 4.2 function implementation

```

60 letterspaceadjust = function(head)
61   for glyph in nodetraverseid(nodeid"glyph", head) do
62     if glyph.prev and (glyph.prev.id == nodeid"glyph" or glyph.prev.id == nodeid"disc" or glyph.p
63       local g = nodecopy(letterspace_glue)
64       nodeinsertbefore(head, glyph, g)
65       nodeinsertbefore(head, g, nodecopy(letterspace_pen))
66     end
67   end
68   return head
69 end

```

## 4.3 textletterspaceadjust

The `\text...`-version of `letterspaceadjust`. Just works, without the need to call `\letterspaceadjust` globally or anything else. Just put the `\textletterspaceadjust` around the part of text you want the function to work on. Might have problems with surrounding spacing, take care!

```

70 textletterspaceadjust = function(head)
71   for glyph in nodetraverseid(nodeid"glyph", head) do
72     if node.has_attribute(glyph, luatexbase.attributes.letterspaceadjustattr) then
73       if glyph.prev and (glyph.prev.id == node.id"glyph" or glyph.prev.id == node.id"disc") then
74         local g = node.copy(letterspace_glue)
75         nodeinsertbefore(head, glyph, g)
76         nodeinsertbefore(head, g, nodecopy(letterspace_pen))
77       end
78     end
79   end
80   luatexbase.remove_from_callback("pre_linebreak_filter", "textletterspaceadjust")
81   return head
82 end

```