Spacing in ConTEXt

Contents

Introduction

In this document I will collext some remarks about spacing (related) commands but it will happen stepwise. Feel free to contribute.

Hans Hagen
Hasselt NL

1 Line correction

1.1 Wrapping content

We really do out best to make the spacing look as good as possible (or at least consistent) but sometimes TEX needs a bit of help. An example of a helper is the following:

```
\startlinecorrection
    \input ward
\stoplinecorrection
```

The Earth, as a habitat for animal life, is in old age and has a fatal illness. Several, in fact. It would be happening whether humans had ever evolved or not. But our presence is like the effect of an old-age patient who smokes many packs of cigarettes per day—and we humans are the cigarettes.

When we use the same command with some left and/or right margins set, we get this:

```
\startnarrower
    \startlinecorrection
        \input ward
    \stoplinecorrection
\stopnarrower
```

The Earth, as a habitat for animal life, is in old age and has a fatal illness. Several, in fact. It would be happening whether humans had ever evolved or not. But our presence is like the effect of an old-age patient who smokes many packs of cigarettes per day—and we humans are the cigarettes.

Here we do obey the margins inside the correction box but the box itself is still as wide as the current width. A typical case where this happens is:

```
\startitemize
    \startitem an item:
        \startlinecorrection
            \input ward
        \stoplinecorrection
    \stopitem
\stopitemize
```

- an item:

The Earth, as a habitat for animal life, is in old age and has a fatal illness. Several, in fact. It would be happening whether humans had ever evolved or not. But our presence is like the effect of an old-age patient who smokes many packs of cigarettes per day—and we humans are the cigarettes.

Here is a variant:

```
\startitemize
    \startitem a local linecorrection:
        \startlocallinecorrection
            \input ward
        \stoplocallinecorrection
    \stopitem
\stopitemize
```

- a local linecorrection:

  The Earth, as a habitat for animal life, is in old age and has a fatal illness. Several, in fact. It would be happening whether humans had ever evolved or not. But our presence is like the effect of an old-age patient who smokes many packs of cigarettes per day—and we humans are the cigarettes.

Both types of correction wrap their content in a box and make sure that the spacing around it is visually as good as possible. The local variant uses a box fitting the available width taking margins into account and but resetting them inside the box. The normal variant applies the margins inside the box. Which one you use depends on the situation and content.

You can pass an optional argument that indicates the amount of spacing to be added before and after the correction.

```
\startlinecorrection[3*line]
    \input ward
\stoplinecorrection
```

The Earth, as a habitat for animal life, is in old age and has a fatal illness. Several, in fact. It would be happening whether humans had ever evolved or not. But our presence is like the effect of an old-age patient who smokes many packs of cigarettes per day—and we humans are the cigarettes.

Normally you don't need this correction. It is mainly used for correcting spacing around boxed material, like `\framed`:

```
before
\startlinecorrection
    \framed{something inbetween}
\stoplinecorrection
after
```

before

something inbetween

after

Because in this document we have set the whitespace this also gets added around the box. So, in case your spacing around some special content looks bad, consider using these commands.

2 Spacing

2.1 Spaces

When TEX reads its input and transforms content it into a node list spaces are turned into glue nodes of subtype 'spaceskip' or 'xspaceskip'. In pseudo code, this is what happens:

```
if spacefactor >= 2000 and xspaceskip ~= 0 then
    space   = xspaceskip_space
    stretch = xspaceskip_stretch
    shrink  = xspaceskip_shrink
    subtype = xspaceskip
else
    if spaceskip == 0 then
        space   = font_space
        stretch = font_stretch
        shrink  = font_shrink
    else
        space   = spaceskip_space
        stretch = spaceskip_stretch
        shrink  = spaceskip_shrink
    end
    if space_factor >= 2000
        space = space + font_extraspace
    end
    stretch = stretch * space_factor
    shrink  = shrink  * space_factor
    subtype = spaceskip
end
insert_glue(space,stretch,shrink,subtype)
```

We demonstrate the effects in a few examples. You can use `\showmakeup[space]` to visualize spaces.

case 1: `\spacefactor 2000` and `\xspaceskip 100pt`

x x. x

x x. x

`\spacefactor`	≥ 2000
`\xspaceskip`	$\neq 0$
space	xspaceskip_space

stretch	xspaceskip_stretch
shrink	xspaceskip_shrink
subtype	xspaceskip

case 2: `\spacefactor 1000` and `\spaceskip 0pt`

x x. x
x x. x

`\spacefactor`	< 2000
`\spaceskip`	$= 0$
space	font_space
stretch	font_stretch
shrink	font_shrink
subtype	spaceskip

case 3: `\spacefactor 2000` and `\spaceskip 0pt`

x x. x
x x. x

`\spacefactor`	≥ 2000
`\spaceskip`	$= 0$
space	font_space + extraspace_font
stretch	font_stretch
shrink	font_shrink
subtype	spaceskip

case 4: `\spacefactor 1000` and `\spaceskip 100pt`

x x. x
x x. x

`\spacefactor`	< 2000
`\spaceskip`	$\neq 0$
space	font_space
stretch	font_stretch × space_factor
shrink	font_shrink × space_factor
subtype	spaceskip

case 5: `\spacefactor 2000` and `\spaceskip 100pt`

x x. x
x x. x

\spacefactor	≥ 2000
\spaceskip	≠ 0
space	font_space + extraspace_font
stretch	font_stretch × space_factor
shrink	font_shrink × space_factor
subtype	spaceskip

The width of a space relates to the design of a font and therefore the t width of the space, its stretch and its shrink are taken from the font and scaled accordingly. Normally we take the space character in the font as reference. Traditional TEX fonts don't have that character but OPENTYPE fonts have. When there is no space character, in the case of a monospaced font we take the emwidth, otherwise we take half the emwidth. As a last resort we can take the average width of characters. And of even that fails we take half of the font units. But, as mentioned, modern fonts have a space.

In the CONTEXT font loader we use a stretch that is 1/2 of the width of a space and the shrink is 1/3 the width of a space. These values are familiar for those who come from traditional TEX.

As with many variables, you can overload these values when a font is loaded by setting the spacing feature. Here is how this is done:

```
\definefontfeature
  [morespace]
  [spacing=0.50 plus 0.50 minus 0.250]
\definefontfeature
  [lessspace]
  [spacing=0.25 plus 0.25 minus 0.125]
\definefontfeature
  [extramorespace]
  [spacing=0.50 plus 0.50 minus 0.250 extra 2.00]
\definefontfeature
  [extralessspace]
  [spacing=0.25 plus 0.25 minus 0.125 extra 2.00]

\definedfont[Serif*default]
    \inleft{\infofont default}
    \samplefile{klein}\par
    \blank
\definedfont[Serif*default,morespace]
    \inleft{\infofont morespace}
    \samplefile{klein}\par
    \blank
\definedfont[Serif*default,extramorespace]
```

```
    \inleft{\infofont extramorespace}
    \samplefile{klein}\par
    \blank
\definedfont[Serif*default,lessspace]
    \inleft{\infofont lessspace}
    \samplefile{klein}\par
    \blank
\definedfont[Serif*default,extralessspace]
    \inleft{\infofont extralessspace}
    \samplefile{klein}\par
    \blank
```

For demonstration purposes we use a somewhat excessive `extra` specification. By default the extra space is equal to the shrink.

default We don't go into a state of shock when something big and bad happens; it has to be something big and bad *that we do not yet understand*. A state of shock is what results when a gap opens up between events and our initial ability to explain them. When we find ourselves in that position, without a story, without our moorings, a great many people become vulnerable to authority figures telling us to fear one another and relinguish our rights for the greater good.

morespace We don't go into a state of shock when something big and bad happens; it has to be something big and bad *that we do not yet understand*. A state of shock is what results when a gap opens up between events and our initial ability to explain them. When we find ourselves in that position, without a story, without our moorings, a great many people become vulnerable to authority figures telling us to fear one another and relinguish our rights for the greater good.

extramorespace We don't go into a state of shock when something big and bad happens; it has to be something big and bad *that we do not yet understand*. A state of shock is what results when a gap opens up between events and our initial ability to explain them. When we find ourselves in that position, without a story, without our moorings, a great many people become vulnerable to authority figures telling us to fear one another and relinguish our rights for the greater good.

lessspace We don't go into a state of shock when something big and bad happens; it has to be something big and bad *that we do not yet understand*. A state of shock is what results when a gap opens up between events and our initial ability to explain them. When we find ourselves in that position, without a story, without our moorings, a great many people become vulnerable to authority figures telling us to fear one another and relinguish our rights for the greater good.

extralessspace We don't go into a state of shock when something big and bad happens; it has to be something big and bad *that we do not yet understand*. A state of shock is what

results when a gap opens up between events and our initial ability to explain them. When we find ourselves in that position, without a story, without our moorings, a great many people become vulnerable to authority figures telling us to fear one another and relinquish our rights for the greater good.

2.2 Expansion

Spaces become glue that can shrink or stretch. In the worst case words will come too close, or the gap will be large. Even worse is that this can lead to successive lines in a paragraph looking different with respect to spacing. A solution for this is to use font expansion, although the benefits are often less than some users want (you) to believe.

This mechanism is enabled with `\setupalign`. There are two variants (`hz` and `fullhz`) and a reset (`nohz`). In figure 2.1 we use the following font definition:

```
\definefont[testfont][Normal*default,quality @ 9pt]
```

We use `\showfontexpansion` to view the effective expansion factors of each glyph. When `fullhz` is used fontkerns also can get expanded. Zero values are not shown. The font kern factors are shown below the character factors. They can be neglected and one can even wonder if they need a treatment especially because kerns are also used for relative positioning, accent anchoring and cursive attachments.

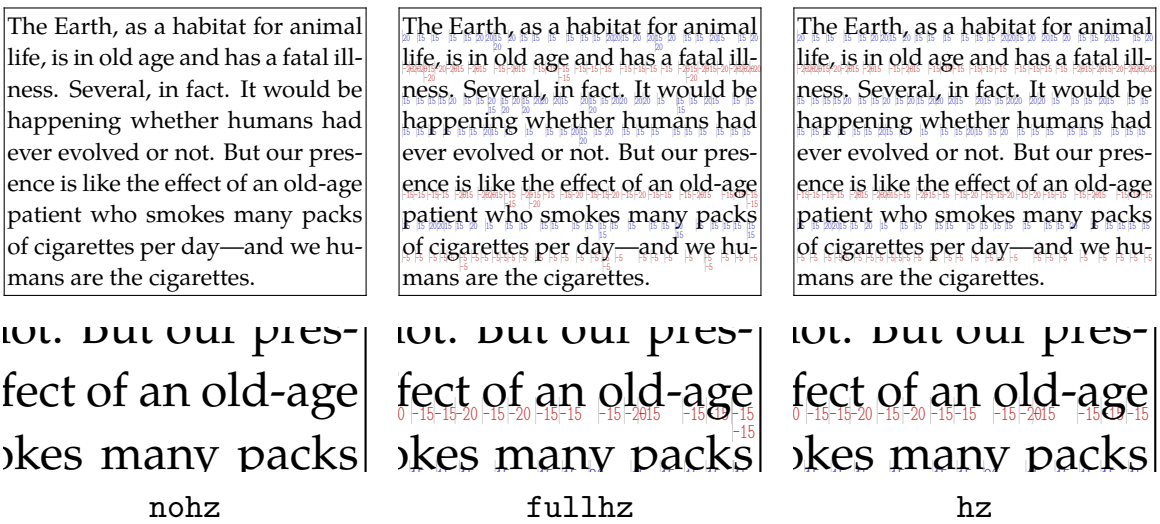


Figure 2.1 The two expansion methods in action.

2.3 Looseness

The `\looseness` parameter can be used to let the par builder add more lines, but that condition is only met when the demand is reasonable. So we need stretch and often also tolerance to achieve it.

```
\looseness=1 ... text ... \par
```

This setting is reset afterwards. Because `framed` does some grouping deep down, we need either to use it in there like this:

```
\framed
  [align={normal,verytolerant,stretch},strut=no]
  {\looseness1 ... \par}
```

which is somewhat clumsy, or we can do:

```
\framed
  [align={normal,verytolerant,stretch,2*more}]
  {...}
```

This is demonstrated in figure 2.2.

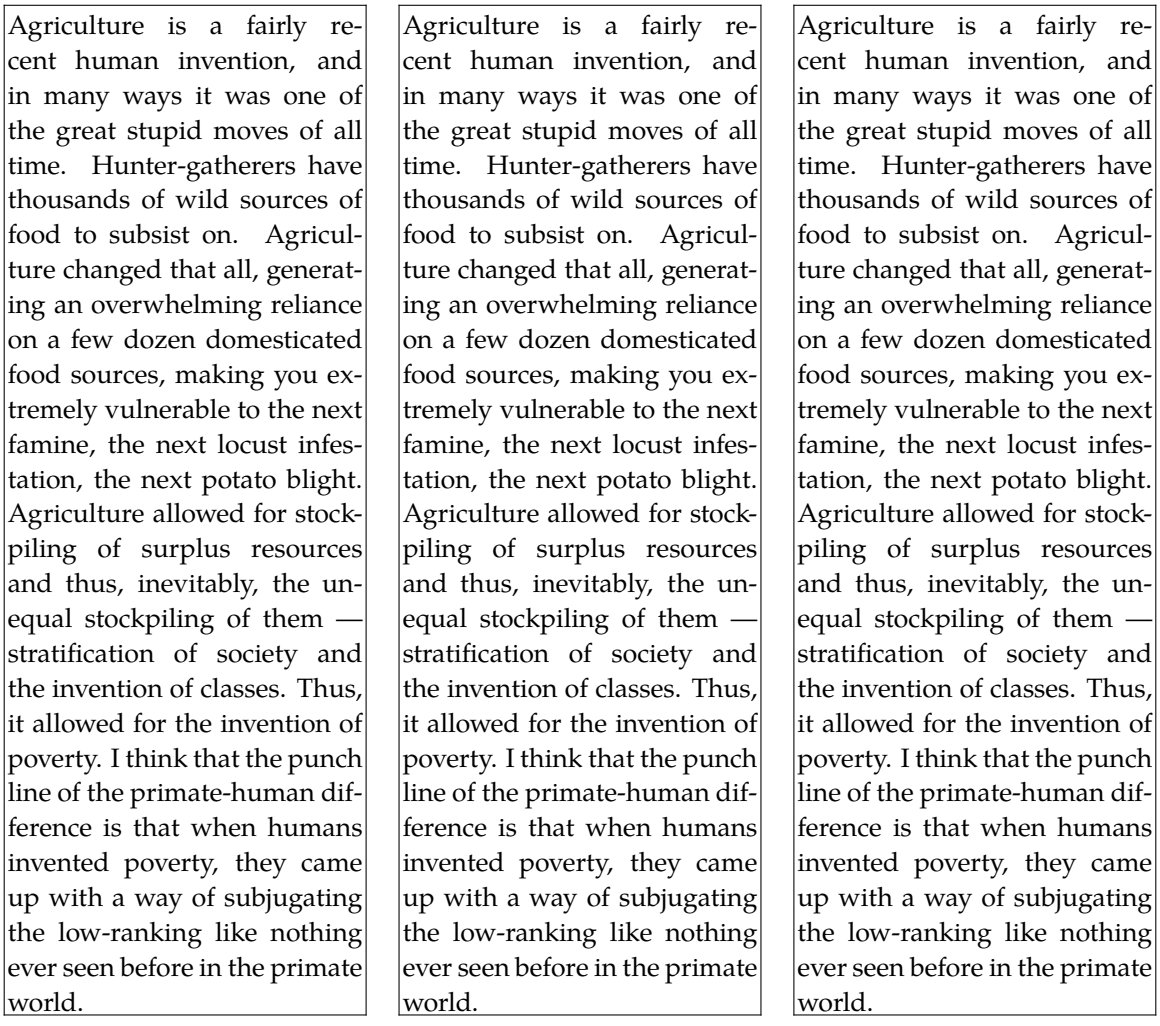


Figure 2.2 Looseness in action.

3 Periods in abbreviations

When you use so called non french spacing you get more spacing after punctuation (as determined by the `sfcode` of the punctuation character) . However, when you use periods as delimiters for abbreviations, that period is not the end of a sentence and you want normal interword spacing instead. One way to achieve this is to add a backslash after the period but in an automated workflow where the source is not coming from TEX but for instance in XML format, you can't do that. The `\setupspacing` command can be used to set one of:

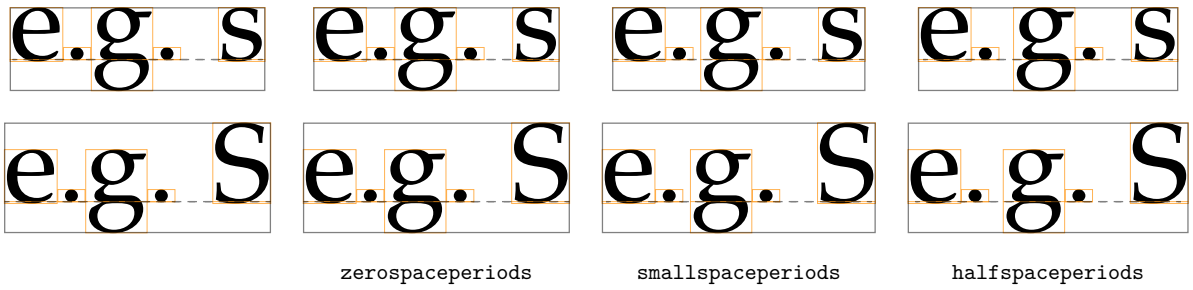
`fixed`	bla bla e.g. some more
`packed`	bla bla e.g. some more
`broad`	bla bla e.g. some more

The `packed` case is similar to `fixed` but has slightly larger (some 5%) spacing after punctuation which (at least historically) avoids some side effects with hyphenation and dashes. We default to `broad` anyway.

The next examples demonstrate what the `\setperiodkerning` does when it gets an option passed. Its counterpart is `\resetperiodkerning`.

		bla bla e.g. some more bla bla e.g. Some more
`zerospaceperiods`	0	bla bla e.g. some more bla bla e.g. Some more
`smallspaceperiods`	.25	bla bla e.g. some more bla bla e.g. Some more
`halfspaceperiods`	.50	bla bla e.g. some more bla bla e.g. Some more

Next we enlarge the affected bit of text so that you can see that the last two options also affects the space after the periods that bind the characters.



Defining more options is easy, we only specify the factor that determines mid periods. When `factor` is zero, only the final period is looked at.

```
\defineperiodkerning [zerospaceperiods]  [factor=0]
\defineperiodkerning [smallspaceperiods] [factor=.25]
\defineperiodkerning [halfspaceperiods]  [factor=.5]
```

This mechanism has been present for a while but I forgot about it. When cleaning up code it was decided to add it to the core. Maybe more options and features are needed but so far there has never been demand for this so ...