
it can be done
TEX
canbedone

Contents

Colofon

Author	Hans Hagen
ConTEXt	2025.12.20 23:15
LuaMetaTEX	2.11.08 | 20251231
Support	www.pragma-ade.com
	contextgarden.net
	ntg-context@ntg.nl

2 Periods


it can be done
TEX
periods

Contents

2.1 Introduction

When the TEX program showed up there were not many fonts that could be used so it came with its own fonts and because the number of slots in the encoding was limited (first to 127, later to 255) there was no space characters. It was not needed anyway because the engine uses a model of glue between words. So, instead of fixed spacing, TEX uses flexible spacing. In addition to what is normally considered a word space, spacing is also determined by the so called space factor of characters preceding spaces. But, especially after abbreviations with periods you might want something different depending on the usage of the period. Here we discuss how that can be done.

2.2 Font related spacing

Spacing is controlled by the amount specified in the font in the so called font dimensions. In ConTEXt these can be accessed via macros:

	normal space	**bold space**	*italic space*	***bolditalic space***
`\interwordspace`	3.49658pt	3.82959pt	3.49658pt	3.82959pt
`\interwordstretch`	1.74829pt	1.9148pt	1.74829pt	1.9148pt
`\interwordshrink`	1.16553pt	1.16553pt	1.27652pt	1.27652pt

The differences in the three components are subtle but often of no concern to the user. Stretch and shrink kicks in when we align the left and right edge, otherwise they are basically ignored. These spacing properties are very specific for TEX fonts, they don't come with for instance OpenType fonts. There we derive the stretch and shrink from the regular font space (Unicode slot U+00020 or ascii value 32).

A user can tweak the interword spacing with `\spaceskip` and `\xspaceskip` which works together with the `\spacefactor` and (character specific)`\sfcode` values. And as it is somewhat hard to explain the details involved I just refer to Chapter 25 (Spacing) of TEX by Topic.

2.3 Spacing after periods

For this manual it's only important to know that the space factors influence the spacing after uppercase letters and punctuation and the later aspect is what this is about.

```
\frenchspacing          This is a t.e.s.t. for periods. Does it work?
\vskip-.8\lineheight
\nonfrenchspacing \blue This is a t.e.s.t. for periods. Does it work?
```

This is a t.e.s.t. for periods. Does it work?
This is a t.e.s.t. for periods. Does it work?

You will notice that the spacing after `t.e.s.t.` is as flexible as the space after `periods.` but what if you don't want that? There are several ways to influence the following space:

```
\frenchspacing           This is a t.e.s.t.\ for periods. Does it work?
\vskip-.8\lineheight
\nonfrenchspacing \green This is a t.e.s.t.\ for periods. Does it work?
```

This is a t.e.s.t. for periods. Does it work?
This is a t.e.s.t. for periods. Does it work?

The `\fsp` macro looks ahead and adapts the space factor:

```
\frenchspacing         This is a t.e.s.t\fsp. for periods. Does it work?
\vskip-.8\lineheight
\nonfrenchspacing \red This is a t.e.s.t\fsp. for periods. Does it work?
```

This is a t.e.s.t. for periods. Does it work?
This is a t.e.s.t. for periods. Does it work?

2.4 Automation

Where the manual (explicit) making sure we get spacing right is quite robust and predictable a user might be willing to delegate the task to ConTEXt, and here is the trick:

```
\setperiodkerning[zerospaceperiods]
\frenchspacing         This is a t.e.s.t. for periods. Does it work?
\vskip-.8\lineheight
\nonfrenchspacing \red This is a t.e.s.t. for periods. Does it work?
```

This is a t.e.s.t. for periods. Does it work?
This is a t.e.s.t. for periods. Does it work?

This features has been present from mid 2017 but I admit that till now I never used it. Reasons are that it makes no sense to adapt existing documents and when a text is for

instance meant for a user group journal too, you cannot expect this automatic feature to be present in the macro package used for typesetting it. But maybe it's time to change that policy. I also admit that I seldom have this situation, probably the only few cases are abbreviations like `e.g.` (for example) and `c.q.` (casu quo).

There are a few predefined period kerning variants and you can define more if you want:

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

\setperiodkerning[zerospaceperiods]
\frenchspacing         How about c.q. and e.g. within a sentence?
\vskip-.8\lineheight
\nonfrenchspacing \red How about c.q. and e.g. within a sentence?
```

How about c.q. and e.g. within a sentence?
How about c.q. and e.g. within a sentence?

Of course one needs to keep an eye on the results because one never knows if the heuristics are flawed. And if needed it can be improved.

2.5 Todo

more spacing related features

2.5 Colofon

Author	Hans Hagen
ConTEXt	2025.12.20 23:15
LuaMetaTEX	2.11.08 | 20251231
Support	www.pragma-ade.com
	contextgarden.net
	ntg-context@ntg.nl

3 Clipping


it can be done
TEX
clipping

Contents

3.1 Clipping

The feature below is rather pdf specific. Inside a graphic group we can register text as clipping path and apply that to what follows. We show a few approaches:

```
\newbox      \MyBox
\newdimension\MyDimenA
\newdimension\MyDimenB

\setbox\MyBox\hbox{\bfd\setstrut\strut\starteffect[clip]JUST SOME TEXT
  \stopeffect}

\MyDimenA\wd  \MyBox
\MyDimenB\htdp\MyBox

\startgraphicgroup
    \startoverlay
      {\box\MyBox}
      {\externalfigure[hacker.jpg][width=\MyDimenA,height=\MyDimenB]}
    \stopoverlay
\stopgraphicgroup
```

We get:

Here we use a helper:

```
\startclipeffect
    {\hbox to 9cm{\hss\bfd\setstrut\strut
       \starteffect[clip]JUST SOME TEXT\stopeffect\hss}}
    {\externalfigure[hacker.jpg][width=9cm,height=1cm]}
\stopclipeffect
```

We get:

A bit easier is this:

```
\defineoverlay
  [hacker]
  [\overlayfigure{hacker.jpg}]

\startgraphicgroup
\framed
  [background={foreground,hacker},align={middle,lohi},width=.8tw]
  {\starteffect[clip]\samplefile{tufte}\stopeffect}
\stopgraphicgroup
```

As you see, we can use more text:

We thrive in information-thick worlds because of our marvelous and everyday capacity to select, edit, single out, structure, highlight, group, pair, merge, harmonize, synthesize, focus, organize, condense, reduce, boil down, choose, categorize, catalog, classify, list, abstract, scan, look into, idealize, isolate, discriminate, distinguish, screen, pigeonhole, pick over, sort, integrate, blend, inspect, filter, lump, skip, smooth, chunk, average, approximate, cluster, aggregate, outline, summarize, itemize, review, dip into, flip through, browse, glance into, leaf through, skim, refine, enumerate, glean, synopsize, winnow the wheat from the chaff and separate the sheep from the goats.

Finally we introduce:

```
\defineoverlay
  [hacker]
  [\overlayfigure{hacker.jpg}]

\startclipframed
  [background={foreground,hacker},align={middle,lohi},width=.8tw]
  \samplefile{ward}
\stopclipframed
```

This hides most of the ugly hackery and gives:

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.

3.1 Colofon

Author	Hans Hagen
ConTEXt	2025.12.20 23:15
LuaMetaTEX	2.11.08 | 20251231
Support	www.pragma-ade.com
	contextgarden.net
	ntg-context@ntg.nl

4 External


it can be done
TEX
external files

Contents

4.1 Introduction

When referring to other documents there are two situations:

- The file is just referred to and we can refer to pages of destinations but no checking takes place. In fact, these external files can have be produced by any application. A reference is rather explicit, to a page in some other document or just to to the document as a whole.

- The file produced by ConTEXt and we can pick up information from the utility file: references, lists (which also have references) and index entries. References can be explicit and for instance involve chapter names and numbers, item counters etc.

A complication is that we don't really know in advance where (especially) references come from. Say that we have these files:

`component-1`	real content, text
`component-2`	loads `component-2-1` and `component-2-1`
`component-2-1`	real content, text
`component-2-2`	real content, text
`component-3`	real content, text
`component-4`	real content, text
`product-1`	loads `component-1` and `component-2`
`product-2`	loads `component-3` and `component-4`
`product-3`	loads `component-1`, `component-2` and `component-3`

In principle we can run each of the component independently. The same is true for the products. When we want to show for instance an item number of an itemize in `component-1` and one from `component-3` and these have the same reference but prefixed by component we can be explicit like in `component-3::myitem`. The information has to be picked up from that components utility file and a hyperlink will then refer to that component file. This assumes that the file is present.

But what if we want to refer to a product where that component is used? Then we need to link to the product and also pick up the number from that products utility file. Add to that nested components, components shared between products and it will become clear that there is not one solution. In order for ConTEXt to deal with this at all, information has to be provided where something comes from. And keep in mind that users have different ways to use references, prefixes and organize files. Also keep in mind that actually a number can be prefixed by for instance a section number and that page numbers for that item are different in a component and that component used in a product.

An external file is registered with:

```
\usefile [tag] [filename] [description]
```

From the perspective of ConTEXt external lists, registers and references are all kind of special: they refer to sections and pages in another document, which means that we need to prepare these external resources for not referring to for instance pages in the current document but to pages in the original. As a result all mechanisms that then use that information have to deal with that and not mix up external and internal properties.
.

4.2 Registers

External registers can either be merged, in which case we need a prefix in front of page numbers, or they can be placed individual. In any case we need to make sure that we have the original page numbers. An external register is defined with:

```
\useregister [tag] [filename] [description] [register] [prefix]
```

If the file is not registered then it will be done automatically. The register data will be preloaded and used on demand.

```
\placeregister[tag]
\placeregister[index,tag]
```

In the second case it will be merged. In order not to get a prefix in the first case, one can define an extra instance (with `\useregister`) without prefix.

4.3 Lists

External lists are specifically placed and because a list is ordered by page number we don’t merge. Again we need to specify what we use:

```
\uselist [tag] [filename] [list]
```

A list is placed with one of:

```
\uselist[workbook][somefile][section]
\placelist[workbook]
```

4.4 References

External references can come from lists but can also be independent. Contrary to lists and registers we need to use prefixes (namespaces) to access them. This has to do with the fact that they are not guaranteed unique.

```
\usereferences[somefile]
```

4.5 Todo

- check what done now with component or product
- check :: and :::
- check component and product in lists
- check component and product in initial numbers
- check index page ranges

4.5 Colofon

Author	Hans Hagen
ConTEXt	2025.12.20 23:15
LuaMetaTEX	2.11.08 | 20251231
Support	www.pragma-ade.com
	contextgarden.net
	ntg-context@ntg.nl

5 Balancing


it can be done
TEX
balancing

Contents

5.1 Balancing

In the (new) column set implementation we use the balancing features of LuaMetaTEX. When we balance, we collect all the content and then run a break pass similar to the line break routine. It might not be obvious at first sight, but when we want to enforce a new column we face the problem that we don't know where we are yet, so we somehow need to communicate our wish to the builder.

We actually have four cases to deal with: go to a next slot (in a column), go to a next column (in a page or spread), go to a next page (in a spread), and go to a next spread. We communicate to the builder using `\balanceboundary` commands (that take two integers) and when the builder encounters these it will trigger a callback (that gets these integers passed). That callback then has to force moving on using a `\penalty` of (at least) 10000. Of course the amount of moves depend on the case as well as where we are. For instance if we have 3 slots in a column and are in slot 2 we need to move two slot up when we want to go to a next column.

When Mikael was converting lecture notes to use column sets, which for instance gives nice options for adding graphics in various places, a subtle problem surfaced. We could strip down the problematic case to this:

```
\startcolumnset[example]
    %dorecurse{3}{\samplefile{ward}\par}
    \dorecurse{5}{\samplefile{ward}\par} Bla bla bla
    \blank[line]
    \startsection[title=Foo]
        \samplefile{ward}
    \stopsection
\stopcolumnset
```

The uncommented case have a full first column, then an empty one, and a section in the third. The commented case gave a half filled first column and a section in the second. Removing the `\blank` made the issue disappear. The easiest way to demonstrate what happens is using a similar situation in the par builder.

```
{\darkred\dontleavehmode\vrule width \hsize height 3pt}%
\break
```

```
\dontleavehmode\vrule width \hsize height 2pt
```

```
{\darkblue\dontleavehmode\vrule width \hsize height 3pt}%
\space\space\space\space\space
\break
\dontleavehmode\vrule width \hsize height 2pt
```

```
{\darkgreen\dontleavehmode\vrule width {\hsize - 1sp} height 3pt}
\break
\dontleavehmode\vrule width \hsize height 2pt
```

```
{\darkyellow\dontleavehmode\vrule width {\hsize - 2pt} height 3pt}
\break
\dontleavehmode\vrule width \hsize height 2pt
```

```
{\darkgray\dontleavehmode\vrule width {\hsize + 2pt} height 3pt}
\break
\dontleavehmode\vrule width \hsize height 2pt
```

When you compare these cases you can deduce that space is responsible for an additional line when we are filled up. This is also happening in the balancer: the `\blank` became glue that triggers a trial break and as we are filled up we go to the next slot in a balancing shape (similar to a line in a par shape). And because the par builder does that before we signal the column break we are ahead (already moved to a next column). One could imagine the boundary to remove the skips preceding it but we can only do that when we're sure that we are in vertical mode, and triggering that has the same side effects.

When looking at this and experimenting a bit we came to the conclusion that dealing with this in the engine or callback was pretty unreliable. We then realized that as ConTEXt users never `\vskip` but use `\blank` instead, we could actually just do `\blank[back]` before inserting the signals. This is pretty safe because we don't want spacing there anyway. Because removal now happens in a predictable place at a moment before we start balancing it also works okay. After all, the macro package knows what it is dealing with while the engine just has to gamble. In the end the solution was rather trivial but sometimes wasting time on experiments is needed to come to that.

5.1 Colofon

Author	Hans Hagen
ConTEXt	2025.12.20 23:15
LuaMetaTEX	2.11.08 | 20251231
Support	www.pragma-ade.com
	contextgarden.net
	ntg-context@ntg.nl