

The lualatex-truncate package

<https://gitlab.com/SFr682k/lualatex-truncate>

Sebastian Friedl
sfr682k@t-online.de

2018/08/22 (v1.1)

Abstract

This package provides a wrapper for using the truncate package¹, thus fixing issues related to Lua_T_EX's hyphenation mechanism.

Contents

Dependencies and other requirements	2
Installation	2
License	2
I The documentation	3
Truncating text to a given width	3
Known limitations	3
II The package code	4
Index	6

¹available at <https://ctan.org/pkg/truncate>

Dependencies and other requirements

lualatex-truncate requires $\LaTeX 2_{\epsilon}$ and depends on the following packages:

iftex

letltxmacro

truncate

Installation

Extract the *package* file first:

1. Run \LaTeX over the file `lualatex-truncate.ins`
2. Move the resulting `.sty` file to `TEXMF/tex/lualatex/lualatex-truncate/`

Then, you can compile the *documentation* yourself by executing

```
lualatex lualatex-truncate-doc.dtx
makeindex -s gind.ist lualatex-truncate-doc.idx
makeindex -s gglo.ist -o lualatex-truncate-doc.gls lualatex-truncate-doc.glo
lualatex lualatex-truncate-doc.dtx
lualatex lualatex-truncate-doc.dtx
```

or just use the precompiled documentation shipped with the source files.

In both cases, copy the files `lualatex-truncate-doc.pdf` and `README.md` to `TEXMF/doc/lualatex/lualatex-truncate/`

License

© 2018 Sebastian Friedl

This work 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.

The latest version of this license is available at <http://www.latex-project.org/lppl.txt> and version 1.3c or later is part of all distributions of \LaTeX version 2008-05-04 or later.

This work has the LPPL maintenace status ‘maintained’.

Current maintainer of this work is Sebastian Friedl.

This work consists of the following files:

- `lualatex-truncate.sty`,
- `lualatex-truncate.ins`,
- `lualatex-truncate-doc.dtx` and
- the derived file `lualatex-truncate.sty`

Part I

The documentation

Don't load the truncate package in your preamble!!

This task is already done for you by `lualatex-truncate`!

Also, all package options stated in this documentation are meant to be passed to `lualatex-truncate`; otherwise they can't be processed by the wrapper mechanism.

Truncating text to a given width²

`\truncate` The package defines a command `\truncate[marker]{width}{text}`.

If the text is too wide to fit in the specified width, then it is truncated, and a continuation marker is shown at the end. The default marker, used when the optional [*marker*] parameter is omitted, is `\,` `\dots`. You can change this default by redefining `\TruncateMarker` (e.g. `\renewcommand\TruncateMarker{\dots}`).

`\TruncateMarker`

`fit` Normally, the text (no matter whether truncated or not) is printed flushed-left in a box with exactly the width specified. The package option `fit` causes the output text to have its natural width, up to a maximum of the specified width.

Normally, the text won't be truncated in the middle of a word, nor at a space specified with a tie (`~`). For example:

```
\truncate{122pt}{This text has been~truncated}
```

gives

```
"This text has ..."
```

`hyphenate` You can use one of the package options `hyphenate`, `breakwords` or `breakall` to allow
`breakwords` truncating in the middle of words. The first two only truncate at hyphenation points;
`breakall` with the only difference that `breakwords` suppresses the hyphen character. On the other hand, `breakall` allows truncation at any character. For example:

```
\truncate{122pt}{This text has been~truncated}
```

gives

```
"This text has been trun- ..." (package option hyphenate)
```

```
"This text has been trun ... " (package option breakwords)
```

```
"This text has been trunc ..." (package option breakall)
```

[All of these options work through (Lua)TeX's hyphenation mechanism.]

Known limitations

- `lualatex-truncate` provides a *wrapper* fixing LuaTeX-related "glitches" in the truncate package. It doesn't (and won't) fix any bugs or "undocumented features" introduced by the truncate package itself.

²This section has been adapted from truncate's documentation:

<http://mirrors.ctan.org/macros/latex/contrib/truncate/truncate.pdf>

Part II

The package code

Initialization

Identify the package and require $\LaTeX 2_{\epsilon}$:

```
1 \ProvidesPackage{lualatex-truncate}%  
2   [2018/08/22 v1.1 LuaLaTeX wrapper for the truncate package]  
3 \NeedsTeXFormat{LaTeX2e}
```

Internal variables

This variable is used to remember whether messing with the hyphen char is required:

```
4 \newif\if@lualtxtruncate@hackhyphen  
5 \@lualtxtruncate@hackhyphenfalse
```

Also, enable remembering whether the `breakall` option has been used:

```
6 \newif\if@lualtxtruncate@breakall  
7 \@lualtxtruncate@breakallfalse
```

Package options

The `breakwords` and `breakall` options require the hyphenation character to be hidden. Therefore, we remember to “hack” the hyphen character before passing these options to the “original” truncate package:

```
8 \DeclareOption{breakwords}{%  
9   \@lualtxtruncate@hackhyphentrue%  
10  \PassOptionsToPackage{breakwords}{truncate}}
```

In case of the `breakall` option, we also need to remember the option’s usage:

```
11 \DeclareOption{breakall}{%  
12   \@lualtxtruncate@breakalltrue%  
13   \@lualtxtruncate@hackhyphentrue%  
14   \PassOptionsToPackage{breakall}{truncate}}
```

Pass all other options (i.e. `hyphenate` and `fit`) directly to the truncate package:

```
15 \DeclareOption*{\PassOptionsToPackage{\CurrentOption}{truncate}}
```

Process all options passed to `lualatex-truncate`:

```
16 \ProcessOptions
```

Load the truncate package

Require the truncate package. All package options have already been passed on. If truncate has been loaded already, a error message is printed to the log.

```
17 \@ifpackageloaded{truncate}{%  
18   \PackageError{lualatex-truncate}%  
19     {Don't load the truncate package when using lualatex-truncate!}%  
20     {Don't load the truncate package and pass all package options  
21       to lualatex-truncate! Otherwise, the wrapper mechanism won't  
22       be able to understand what you want.}%  
23 }{\RequirePackage{truncate}}
```

If the document is not compiled via Lua_{La}T_EX...

Don't wrap anything if the document isn't compiled by Lua_{La}T_EX:

```
24 \RequirePackage{iftex}
25 \ifLuaTeX\else%
26   \PackageWarningNoLine{lualatex-truncate}%
27     {Whoops! You don't want to use LuaLaTeX?}%
28   \endinput%
29 \fi%
```

Constructing the wrapper

`\lualtxtruncate@truncate` Remember the original `\truncate` command's definition. Since `\truncate` provides an optional argument, `\LetLtxMacro` has to be used instead of the primitive `\let` command:

```
30 \RequirePackage{letltxmacro}
31 \LetLtxMacro\lualtxtruncate@truncate\truncate
```

`\truncate` Construct the wrapper by redefining the `\truncate` command and – if necessary – modifying `\prehyphenchar`.

`\prehyphenchar`'s value is stored inside the counter `\lualtxtruncate@tmp`. Also, the current language must be set to US english, if the `breakall` option has been used.

```
32 \renewcommand{\truncate}[3][\TruncateMarker]{{% two braces -> scope changes
33   \if@lualtxtruncate@breakall%
34     \language\z@% force US english
35   \fi%
36   \newcount\lualtxtruncate@tmp%
37   \lualtxtruncate@tmp=\the\prehyphenchar% remember current hyphen char
38   \if@lualtxtruncate@hackhyphen%
39     \prehyphenchar=0% invisible hyphen
40   \fi%
41   \lualtxtruncate@truncate[#1]{#2}{#3}% run the original \truncate macro
42   \if@lualtxtruncate@hackhyphen%
43     \prehyphenchar=\the\lualtxtruncate@tmp% restore hyphen
44   \fi%
45 }}
```

Famous last words

Enough code for this package. `\endinput`.

```
46 \endinput
```

Change History

1.0	General: Initial release	4	hyphenation character	5
1.1	\truncate: Remember previous		General: BUGFIX : visible hyphen char with breakall option and non-english language	4

Index

	B		L	
breakall option	3-5		\lua!txtruncate@truncate	5
breakwords option	3,4			
	F		T	
fit option	3,4		\truncate	5
	H		\TruncateMarker	3
hyphenate option	3,4			