

The lualatex-truncate package

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

Sebastian Friedl
sfr682k@t-online.de

2018/06/03 (v1.0)

Abstract

This package provides a wrapper for using the truncate package with Lua¹_{La}T_EX

Contents

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

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

Dependencies and other requirements

lualatex-truncate requires \TeX 2 ϵ and depends on the following packages:

iftex

letltxmacro

truncate

Installation

Extract the *package* file first:

1. Run \TeX 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 \TeX 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 \TeX 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

Read this section first

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

This task is already done for you when loading lualatex-truncate!

Also, all package options stated in this documentation are meant to be passed to lualatex-truncate, so they can be processed by the wrapper mechanism.

Truncating text to a given width²

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`

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.

`fit`

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`
`breakwords`
`breakall`

You can use one of the package options `hyphenate`, `breakwords` or `breakall` to allow truncating in the middle of words. The first two only truncate at hyphenation points; 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)T_EX's hyphenation mechanism.]

²This section has been adapted from truncate's documentation:
<http://mirrors.ctan.org/macros/latex/contrib/truncate/truncate.pdf>

Known limitations

- lualatex-truncate provides a *wrapper* to enable using the truncate package together with Lua \TeX . It doesn't (and won't) fix any bugs or “undocumented features” introduced by the truncate package.
- lualatex-truncate applies some “hacking” to the hyphenation character. This might cause some problems if the hyphenation character of the currently used language is different from “\‑”.

Part II

The package code

Initialization

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

```
1 \ProvidesPackage{lualatex-truncate}%  
2   [2018/06/03 v1.0 LuaLaTeX wrapper for the truncate package]  
3 \NeedsTeXFormat{LaTeX2e}
```

Internal variables

This internal variable is used to remember whether “hacking” the hyphenation character (i.e. `\prehyphenchar`) is required:

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

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:

```
6 \DeclareOption{breakwords}{%  
7   \@lualtxtruncate@hackhyphentrue  
8   \PassOptionsToPackage{breakwords}{truncate}}  
9 \DeclareOption{breakall}{%  
10  \@lualtxtruncate@hackhyphentrue  
11  \PassOptionsToPackage{breakall}{truncate}}
```

Pass all other options (hyphenate and fit) directly to the truncate package:

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

Process all options passed to lualatex-truncate:

```
13 \ProcessOptions
```

Loading the truncate package

Load the truncate package. Required package options have already been passed on.

```
14 \RequirePackage{truncate}
```

If the document is not compiled via Lua[®]TeX...

Don't wrap anything if the document isn't compiled by Lua[®]TeX:

```
15 \RequirePackage{iftex}
16 \ifLuaTeX\else%
17   \PackageWarningNoLine{lua®latex-truncate}%
18     {Whoops! You don't want to use LuaLaTeX?}%
19   \endinput%
20 \fi%
```

Constructing the wrapper

lua[®]ltxtruncate@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:

```
21 \RequirePackage{letltxmacro}
22 \LetLtxMacro\lua®ltxtruncate@truncate\truncate
```

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

```
23 \renewcommand{\truncate}[3][\TruncateMarker]{%
24   \if@lua®ltxtruncate@hackhyphen%
25     \prehyphenchar=0%
26   \fi%
27   \lua®ltxtruncate@truncate[#1]{#2}{#3}%
28   \if@lua®ltxtruncate@hackhyphen%
29     \prehyphenchar=\%-
30   \fi%
31 }
```

Famous last words

Enough code for this package. `\endinput`.

```
32 \endinput
```

Index

B		L	
breakall option	3,4	\lualtxtruncate@truncate	5
breakwords option	3,4		
F		T	
fit option	3,4	\truncate	5
H		\TruncateMarker	3
hyphenate option	3,4		