`LuaQuotes`:
A package for smart quotation marks

Elijah Z Granet*

25 January 2023
`Version 1.4.0`

Contents

---

*e-mail: `ezg21@cantab.ac.uk`

1 Overview

1.1 Purpose

This package provides a function to automatically have ‘smart quotes’ in LuaLaTeX. By ‘smart quotes’, I refer to the automatic insertion of curved or ‘typographer’s’ quotation marks when the user types straight quotation marks. The below figure illustrates the distinction in English typography:

`User input`	Smart Quotes Off	Smart Quotes On
`"Howdy!"`	”Howdy!”	“Howdy!”
`'Don't!'`	’Don’t!’	‘Don’t!’

2 Smart quotes

2.1 Options

The default option, for English quotation marks, is called by:

```
\usepackage{luaquotes}
```

The French, German, German (guillemets), and Swiss German options, for those languages’ punctuation, are called by the following respective commands.

```
\usepackage[fr]{luaquotes} %French
\usepackage[de]{luaquotes} % German
\usepackage[degm]{luaquotes} % German with guillemets
\usepackage[dech]{luaquotes} %Swiss German
```

2.1.1 English Features

The English features are designed to smartly recognise English punctuation:

`Feature`	User input	Output
Double Quotes	`"Hello!"`	“Hello”
Single Quotes	`'Hello!'`	‘Hello’
Contractions	`Don't`	Don’t

2.1.2 French option

The French option produces the following output, including the extra space around punctuation prescribed by French typography:

`User input`	Output
`"Salut!"`	« Salut! »
`'Salut!'`	‹ Salut! ›

2.1.3 The German option

The German option produces the following output:

`User input`	Output
`"Hallo!"`	„Hallo!“
`'Hallo!'`	‚Hallo!‘

2.1.4 The German (guillemets) option

The German (guillemets) option produces the following output:

`User input`	Output
`"Hallo!"`	»Hallo!«
`'Hallo!'`	›Hallo!‹

2.1.5 The Swiss German option

The Swiss German option produces the following output:

`User input`	Output
`"Hallo!"`	« Hallo! »
`'Hallo!'`	‹ Hallo! ›

2.2 Activation and De-activation

The package automatically activates the smart quotes function at the beginning of the document. To deactivate the smart quotes function within a document, the following commands are used:

```
\dumbquotes %English
\frdumbquotes %French
\dedumbquotes %German
\degmdumbquotes %German with French-style quotes
```

The following commands re-activate the smart quotes function:

```
\smartquotes %English
\frsmartquotes %French
\desmartquotes % German
\degmsmartquotes % German with French-style quotes
```

A limitation on the (de-)activation of the package is that the Lua filters will not deactivate within the same paragraph, so the function can only be changed across paragraphs.

2.3 Monospace

As a general rule, smart quotes are rather undesirable in monospace text, and therefore, within the `\texttt` environment the package does not apply smart quotes. Thus, the same input produces in roman face “Hello World” but in monospace `"Hello World"`.

As the example above shows, the default behaviour of this package forces straight quotes in monospace, and disables TEX quote ligatures (but not other TEX ligatures) to do so, on the assumption that any form of curved quotes are undesirable.

For extended periods of monospaced text called by `\ttfamily`, the activation and de-activation methods above should be used. I considered altering the `\ttfamily` command to always call on `\dumbquotes`, but I thought it best to avoid messing with the command. This may change in future development.

If a user desires to disable the smart quotes for other faces, this is easily done globally by adding the following line to the font’s configuration in `fontspec`:

```
RawFeature={+qtbye}
```

This will, however, result in the font using straight quotes rather than the standard TEX quote ligatures, making it a distinct option than the activation and de-activation options *supra*.

3 Auxiliary Punctuation

The smart quotes feature covers the ‘standard’ usage of quotes, but there are many instances where quotation mark or quote-mark like features are needed outside the automatic formatting. The package provides several commands for this.

3.1 Standalone quotes

The marks in this section are the set of quotation marks used generally in writing and require little explanation. However, it should be noted that the commands `\sqoneright` and `\apost` (which produce identical output) are very useful for aphetic words by which the first syllable is clipped, as in the sequence `'bout` the smart quotes function will incorrectly produce an opening quote instead of the correct closing quote to indicate the elision. Thus, the incorrect result of ‘bout is produced. The solution is to use the code `\apost bout` which produces ’bout.

Name	**UTF-8**	**Command**	**Produces**
Single low quote	`U+201A`	`\desingle`	‚□
Double low quote	`U+201E`	`\dedouble`	„□
Single straight quote	`U+0027`	`\dqone`	'□
Double straight quote	`U+0022`	`\dqtwo`	"□
Left single quote	`U+2018`	`\sqoneleft`	‘□

Right single quote	`U+2019`	`\sqoneright`	□’
Apostrophe	`U+2019`	`\apost`	□’
Left double quote	`U+201C`	`\sqtwoleft`	“□
Right double quote	`U+201D`	`\sqtworight`	□”
Left guillemet [w/ space]	`U+00AB`	`\glmtl`	« □
Right guillemet [w/ space]	`U+00BB`	`\glmtr`	□ »
`degm` left guillemet	`U+00BB`	`\degmtl`	»□
`degm` right guillemet	`U+00AB`	`\degmtr`	□«
Single left guillemet [w/ space]	`U+2039`	`\sglmtl`	‹ □
Single right guillemet [w/ space]	`U+203A`	`\sglmtr`	□ ›

degm single left guillemet	U+203A	\desgmtl	›□
degm single right guillemet	U+2039	\desgmtr	□‹

3.2 Additional symbols

These are quote like symbols which are useful for precise punctuation, since standard smart quotation marks do not work well in their specialised use cases.

Name	**UTF-8**	**Command**	**Produces**
Backtick	U+0060	\bcktck	`□

For typesetting US/Imperial measurements like feet and inches, the correct symbol is a prime and double prime. Many modern typefaces have these symbols, and they are thus useful for typesetting feet and inch measurements (like 6′4″).

Name	**UTF-8**	**Command**	**Produces**
Single Prime	U+2032	\lqprime	□′
Double Prime	U+2033	\lqdoubleprime	□″

For certain Polynesian languages, a letter called the ‘Okina is used; while this appears identical to a left single quotation mark in many fonts, it is encoded differently in Unicode because it is properly a letter, not a punctuation mark.

Name	**UTF-8**	**Command**	**Produces**
Okina	U+022B	\okina	ʻ□

4 Future Development and Localisation

The package's online repository is the best place to report bugs, feature requests, or other contributions, and is located at:
`https://github.com/ezgranet/luaquotes`.

One obvious point of future development is the addition of other language localisations; this is technologically easy in principle but difficult for me to accomplish without assistance because it requires detailed knowledge of typographic conventions for any given language standard, and therefore the aid of contributors and collaborators.

5 Licence

6 Implementation

7 Version History

1.4.0

Added support for Swiss German options, on user request

1.3.0

15 January 2023 fixed single quotes and parentheses

1.2.2

Added the “degm” option

1.2.1

Suspended automatic elision support due to implementation issues

1.1.0

25 October 2022: Added support for elision

1.0.1

30 August 2022: Further corrections to bug preventing non-English usage

1.0.1

28 August 2022: Change bug preventing non-English usage

1.0.0

21 August 2022: Package creation