

The dirtytalk Package

M. Klammler*

November 21, 2010

Abstract

This document explains the usage of the dirtytalk package for L^AT_EX 2 ϵ . It can be used to typeset “quotations” more comfortable. The source code is also explained.

Contents

1 Usage	2
1.1 Loading	2
1.2 Macros	2
1.3 Options	2
2 Source Code Documentation	3
2.1 Identification	3
2.2 Preliminary Declarations	3
2.3 Options	4
2.4 More Declarations	4
3 Copyright Statement	5

*moritz.klammler@student.kit.edu
(If you felt sad when Littlefoot’s mother died in the original *Land Before Time* you’ll know what the ∂ means. . .)

Who should use this package? Everybody who feels typesetting quotes in L^AT_EX could be more comfortable.

Who should *really* use this package? Everybody who typesets German texts using `\glqq{}Hallo Welt\grqq{}` for a simple „Hallo Welt“ or—even worse—those who already started writing a text using the same glyphs for nested quotations and soon after find themselves in the embarrassing position of replacing them while being no `regex` whiz.

1 Usage

1.1 Loading

Put `\usepackage{dirtytalk}` in the preamble of your document.

1.2 Macros

`\say` Loaded the package you can typeset quotations by `\say{hello world}` to get “hello world”. Nested quotations are recognized as in:

Alice said: <code>\say{Why did you do that, Bob?}</code>	Alice said: “Why did you do that, Bob?”
And Bob answered: <code>\say{What do you mean with \say{that}, dear Alice?}</code>	And Bob answered: “What do you mean with ‘that’, dear Alice?”

1.3 Options

The package can be loaded with a couple of options to alter the glyphs being used. These are given in the `key=value` style.

key	default value	glyph
<code>left</code>	‘ ‘	“
<code>right</code>	’ ’	”
<code>leftsub</code>	‘	‘
<code>rightsub</code>	’	’

Table 1: Options for the `dirtytalk` package.

So my fellow German citizens might load the package with

```

\usepackage[%
  left      = \glqq,%
  right     = \grqq,%
  leftsub   = \glq,%
  rightsub  = \grq%
]{dirtytalk}

```

to get German glyphs. Please note that the package does *not* check your document language to alter the symbols used. It depends on you as the author to specify those you want.

Happy L^AT_EXing!

2 Source Code Documentation

This section shows and explains the entire code of the package. (It is intended for L^AT_EX programmers only.)

2.1 Identification

```

1 \NeedsTeXFormat{LaTeX2e}
2 \ProvidesPackage{dirtytalk}[2010/11/21 A package making "quoting" easier]

```

2.2 Preliminary Declarations

We use options from the `key=value` type so we need to load a package supporting this:

```

3 \RequirePackage{kvoptions}

```

Define the glyphs that will enclose the phrases. We initially define them for English which will probably be the most requested standard. If the package is loaded with options, the following macros will be redefined.

`\dirtytalk@lqq` Outer opening (left) quotation mark glyph.

```

4 \newcommand{\dirtytalk@lqq}{‘ ’}

```

`\dirtytalk@rqq` Outer closing (right) quotation mark glyph.

```

5 \newcommand{\dirtytalk@rqq}{’ ’}

```

`\dirtytalk@lq` Inner opening (left) quotation mark glyph.

```

6 \newcommand{\dirtytalk@lq}{‘}

```

`\dirtytalk@rq` Inner closing (right) quotation mark glyph.

```
7 \newcommand{\dirtytalk@rq}{'}
```

2.3 Options

We allow the author to specify own quotation mark glyphs instead of those defined above by passing them as options.

```
8 \DeclareStringOption{left}
9 \DeclareStringOption{right}
10 \DeclareStringOption{leftsub}
11 \DeclareStringOption{rightsub}
```

```
12 \ProcessKeyvalOptions*
```

2.4 More Declarations

After processing the options (in case of the `kvoptions`' string options that means a macro has been defined holding the key value) we check if they are empty and, if not so, redefine the default glyphs. For this task we need the `ifthen` package.

```
13 \RequirePackage{ifthen}

14 \ifthenelse%
15   {\equal{\dirtytalk@left}{}}%
16   {}%
17   {\renewcommand{\dirtytalk@lqq}{\dirtytalk@left}}

18 \ifthenelse%
19   {\equal{\dirtytalk@right}{}}%
20   {}%
21   {\renewcommand{\dirtytalk@rq}{\dirtytalk@right}}

22 \ifthenelse%
23   {\equal{\dirtytalk@leftsub}{}}%
24   {}%
25   {\renewcommand{\dirtytalk@lq}{\dirtytalk@leftsub}}

26 \ifthenelse%
27   {\equal{\dirtytalk@rightsub}{}}%
28   {}%
29   {\renewcommand{\dirtytalk@rq}{\dirtytalk@rightsub}}
```

And we define a counter that represents the depth to which a quotation is nested. However, we will only distinguish between “not nested” and “nested” (`dirtytalk@qdepth > 1`). Since there is no convention for sub-sub-quotes. (At least, I don't know any.)

```
30 \newcounter{dirtytalk@qdepth}
```

Using this counter, we can now determine which glyph we have to use.

`\dirtytalk@lsymb` We define one macro for opening quotes:

```
31 \newcommand%
32   {\dirtytalk@lsymb}%
33   {\ifthenelse%
34     {\value{dirtytalk@qdepth}>1}%
35     {\dirtytalk@lq}%
36     {\dirtytalk@lqq}%
37   }%
```

`\dirtytalk@rsymb` And one for closing quotes:

```
38 \newcommand{\dirtytalk@rsymb}%
39   {\ifthenelse%
40     {\value{dirtytalk@qdepth}>1}%
41     {\dirtytalk@rq}%
42     {\dirtytalk@rqq}%
43   }
```

`\say` And so finally we can declare the only macro this package provides to the author. On opening quotes we increment the `dirtytalk@qdepth` counter and decrement it on closing ones. The appropriate glyph is obtained from `\dirtytalk@lsymb` or `\dirtytalk@rsymb` respectively.

```
44 \providecommand{\say}[1]%
45   {%
46     \addtocounter{dirtytalk@qdepth}{1}%
47     \dirtytalk@lsymb%
48     #1%
49     \dirtytalk@rsymb%
50     \addtocounter{dirtytalk@qdepth}{-1}%
51   }
```

3 Copyright Statement

I, the copyright holder of this work, release the source code of the `dirtytalk` package as well as this documentation into the **public domain**. This applies worldwide.

In some countries this may not be legally possible; if so: I grant anyone the right to use this work for any purpose, without any conditions, unless such conditions are required by law.