

The hyphsubst package

Heiko Oberdiek
<oberdiek@uni-freiburg.de>

2008/06/09 v0.2

Abstract

A \TeX format file may include alternative hyphenation patterns for a language with a different name. If the naming convention follows `babel`'s rules, then the hyphenation patterns for a language can be replaced by the alternative hyphenation patterns, provided in the format file.

Contents

1	Documentation	1
1.1	In short	1
1.2	Longer version	2
1.3	\LaTeX	2
1.4	plain- \TeX	3
2	Implementation	3
2.1	Reload check and package identification	3
2.2	Package	5
3	Test	6
3.1	Catcode checks for loading	6
3.2	Main tests	7
4	Installation	8
4.1	Download	8
4.2	Bundle installation	8
4.3	Package installation	9
4.4	Refresh file name databases	9
4.5	Some details for the interested	9
5	History	10
	[2008/06/07 v0.1]	10
	[2008/06/09 v0.2]	10
6	Index	10

1 Documentation

1.1 In short

The package is an experimental package that allows the substitution of hyphenation patterns, example:

```
\RequirePackage[ngerman=ngerman-x-20080601]{hyphsubst}
\documentclass{article}
\usepackage[ngerman]{babel}
```

The patterns `ngerman` are replaced by the patterns `ngerman-x-20080601`. The format must contain these patterns and should use the naming scheme of either `babel`'s `language.dat` or `etex.src`'s `language.def`.

1.2 Longer version

Assume the format may contain the following hyphenation patterns (excerpt from `language.dat`):

```
...
ngerman dehyphn.tex
ngerman-x-20071231 dehyphn-x-20071231
ngerman-x-20080601 dehyphn-x-20080601
=ngerman-x-latest % alias for ngerman-x-20080601
...
```

The patterns that contain `-x-` are experimental new patterns for `ngerman`. However, package `babel` does not provide the use of patterns that do not have the same name as the used language (dialect). The `babel` system remembers patterns in macros: `\l@<name>`. ε -TeX's `etex.src` uses `\lang@<name>` instead. In the following we use `babel`'s naming scheme, but `etex.src`'s naming scheme is supported, too.

This package `hyphsubst` solves the problem by redefining the macro `\l@<name>` to use other patterns.

`\HyphSubstLet {<nameA>} {<nameB>}`

`\l@<nameA>` now has the same meaning as `\l@<nameB>`. The patterns for `nameB` must exist. If the patterns for `nameA` exist, then they will be overwritten to use the patterns for `nameB`. Example:

```
\documentclass{article}
\usepackage{hyphsubst}
\HyphSubstLet{ngerman}{ngerman-x-20080601}
\usepackage[ngerman]{babel}
```

Now the patterns `ngerman-x-20080601` are be used.

Or if you want to compare hyphenations:

```
\documentclass{article}
\usepackage{hyphsubst}
% save original patterns for ngerman in ngerman-saved
\HyphSubstLet{ngerman-saved}{ngerman}
\usepackage[ngerman]{babel}
\begin{document}
  We start with the original patterns for ngerman.
  \HyphSubstLet{ngerman}{ngerman-x-latest}%
  Now we are using ngerman-x-latest.
  \HyphSubstLet{ngerman}{ngerman-saved}%
  Again we are using the original patterns.
\end{document}
```

`\HyphSubstIfExists {<name>} {<then>} {<else>}`

Tests if patterns with name `<name>` exist and execute `<then>` in case of success and `<else>` otherwise.

1.3 L^AT_EX

The package can also be loaded before `\documentclass`:

```

\RequirePackage[ngerman=ngerman-x-20080601]{hyphsubst}
\documentclass{article}
...

```

This allows to put the package in a format file.

Package options are interpreted as ‘let’ assignments and passed to macro `\HyphSubstLet`:

```
\usepackage[ngerman=ngerman-x-20080601]{hyphsubst}
```

The part before the equal sign is the first argument for `\HyphSubstLet` and the part after the equal sign forms the second argument:

```
\HyphSubstLet{ngerman}{ngerman-x-20080601}
```

Note, this only works for direct package options. Global options are ignored.

1.4 plain-TeX

The package can be loaded and used with plain-TeX, e.g.:

```

\input hyphsubst.sty
\HyphSubstLet{ngerman}{ngerman-x-latest}

```

2 Implementation

```
1 <(*package>
```

2.1 Reload check and package identification

Reload check, especially if the package is not used with L^AT_EX.

```

2 \begingroup
3 \catcode44 12 % ,
4 \catcode45 12 % -
5 \catcode46 12 % .
6 \catcode58 12 % :
7 \catcode64 11 % @
8 \expandafter\let\expandafter\x\csname ver@hyphsubst.sty\endcsname
9 \ifcase 0%
10 \ifx\x\relax % plain
11 \else
12 \ifx\x\empty % LaTeX
13 \else
14 1%
15 \fi
16 \fi
17 \else
18 \catcode35 6 % #
19 \catcode123 1 % {
20 \catcode125 2 % }
21 \expandafter\ifx\csname PackageInfo\endcsname\relax
22 \def\x#1#2{%
23 \immediate\write-1{Package #1 Info: #2.}%
24 }%
25 \else
26 \def\x#1#2{\PackageInfo{#1}{#2, stopped}}%
27 \fi
28 \x{hyphsubst}{The package is already loaded}%
29 \endgroup
30 \expandafter\endinput
31 \fi
32 \endgroup

```

Package identification:

```
33 \begingroup
34 \catcode35 6 % #
35 \catcode40 12 % (
36 \catcode41 12 % )
37 \catcode44 12 % ,
38 \catcode45 12 % -
39 \catcode46 12 % .
40 \catcode47 12 % /
41 \catcode58 12 % :
42 \catcode64 11 % @
43 \catcode123 1 % {
44 \catcode125 2 % }
45 \expandafter\ifx\csname ProvidesPackage\endcsname\relax
46 \def\x#1#2#3[#4]{\endgroup
47 \immediate\write-1{Package: #3 #4}%
48 \xdef#1{#4}%
49 }%
50 \else
51 \def\x#1#2[#3]{\endgroup
52 #2[{#3}]}%
53 \ifx#1@undefined
54 \xdef#1{#3}%
55 \fi
56 \ifx#1\relax
57 \xdef#1{#3}%
58 \fi
59 }%
60 \fi
61 \expandafter\x\csname ver@hyphsubst.sty\endcsname
62 \ProvidesPackage{hyphsubst}%
63 [2008/06/09 v0.2 Substitute hyphenation patterns (H0)]
64 \begingroup
65 \catcode123 1 % {
66 \catcode125 2 % }
67 \def\x{\endgroup
68 \expandafter\edef\csname HyphSubst@AtEnd\endcsname{%
69 \catcode35 \the\catcode35\relax
70 \catcode64 \the\catcode64\relax
71 \catcode123 \the\catcode123\relax
72 \catcode125 \the\catcode125\relax
73 }%
74 }%
75 \x
76 \catcode35 6 % #
77 \catcode64 11 % @
78 \catcode123 1 % {
79 \catcode125 2 % }
80 \def\TMP@EnsureCode#1#2{%
81 \edef\HyphSubst@AtEnd{%
82 \HyphSubst@AtEnd
83 \catcode#1 \the\catcode#1\relax
84 }%
85 \catcode#1 #2\relax
86 }
87 \TMP@EnsureCode{39}{12}% '
88 \TMP@EnsureCode{46}{12}% .
89 \TMP@EnsureCode{47}{12}% /
90 \TMP@EnsureCode{58}{12}% :
91 \TMP@EnsureCode{61}{12}% =
92 \TMP@EnsureCode{96}{12}% ‘
```

2.2 Package

```
93 \begingroup\expandafter\expandafter\expandafter\endgroup
94 \expandafter\ifx\csname RequirePackage\endcsname\relax
95   \input infwarerr.sty\relax
96 \else
97   \RequirePackage{infwarerr}[2007/09/09]%
98 \fi

\HyphSubst@l

99 \begingroup\expandafter\expandafter\expandafter\endgroup
100 \expandafter\ifx\csname et@xlang\endcsname\relax
101   \def\HyphSubst@l{l}%
102 \else
103   \def\HyphSubst@l{lang}%
104 \fi

\HyphSubstLet

105 \def\HyphSubstLet#1#2{%
106   \begingroup
107     \def\x{%
108       \expandafter\ifx\csname\HyphSubst@l#2\endcsname\relax
109         \@PackageError{hyphsubst}{Unknown pattern ‘#2’}\@ehc
110       \else
111         \def\lmsg{%
112           \expandafter\ifx\csname\HyphSubst@l#1\endcsname\relax
113             \edef\msg{%
114               New: \expandafter\string\csname\HyphSubst@l#1\endcsname
115               \noexpand\MessageBreak
116             }%
117           \else
118             \edef\msg{%
119               Redefined: \expandafter\string\csname\HyphSubst@l#1\endcsname
120               \noexpand\MessageBreak
121               old value: \number\csname\HyphSubst@l#1\endcsname
122               \noexpand\MessageBreak
123             }%
124             \ifnum\csname\HyphSubst@l#1\endcsname=\language
125               \edef\x{%
126                 \noexpand\language=%
127                 \number\csname\HyphSubst@l#2\endcsname\relax
128               }%
129               \edef\lmsg{%
130                 \noexpand\MessageBreak
131                 \string\language\noexpand\space updated%
132               }%
133             \fi
134           \fi
135           \expandafter\global\expandafter\let
136             \csname\HyphSubst@l#1\endcsname\expandafter\endcsname
137             \csname\HyphSubst@l#2\endcsname
138           \@PackageInfo{hyphsubst}{%
139             \msg
140             new value: \number\csname\HyphSubst@l#1\endcsname
141             \lmsg
142           }%
143         \fi
144       \expandafter\endgroup\x
145 }
```

```
\HyphSubstIfExists

146 \def\HyphSubstIfExists#1{%
147   \begingroup\expandafter\expandafter\expandafter\endgroup
```

```

148 \expandafter\ifx\csname\HyphSubst@l#1\endcsname\relax
149 \expandafter\@secondoftwo
150 \else
151 \expandafter\@firstoftwo
152 \fi
153 }

\@firstoftwo

154 \expandafter\ifx\csname @firstoftwo\endcsname\relax
155 \long\def\@firstoftwo#1#2{#1}%
156 \fi

\@secondoftwo

157 \expandafter\ifx\csname @secondoftwo\endcsname\relax
158 \long\def\@secondoftwo#1#2{#2}%
159 \fi

160 \begingroup\expandafter\expandafter\expandafter\endgroup
161 \expandafter\ifx\csname documentclass\endcsname\relax
162 \HyphSubst@AtEnd
163 \expandafter\endinput
164 \fi

165 \DeclareOption*{%
166 \expandafter\HyphSubst@Option\CurrentOption==\relax
167 }
168 \def\HyphSubst@Option#1=#2=#3\relax{%
169 \HyphSubstLet{#1}{#2}%
170 }
171 \ProcessOptions*\relax

172 \HyphSubst@AtEnd
173 \end{package}

```

3 Test

3.1 Catcode checks for loading

```

174 \test1\
175 \catcode'\{=1 %
176 \catcode'\}=2 %
177 \catcode'\#=6 %
178 \catcode'\@=11 %
179 \expandafter\ifx\csname count@\endcsname\relax
180 \countdef\count@=255 %
181 \fi
182 \expandafter\ifx\csname @gobble\endcsname\relax
183 \long\def\@gobble#1{}%
184 \fi
185 \expandafter\ifx\csname @firstofone\endcsname\relax
186 \long\def\@firstofone#1{#1}%
187 \fi
188 \expandafter\ifx\csname loop\endcsname\relax
189 \expandafter\@firstofone
190 \else
191 \expandafter\@gobble
192 \fi
193 {%
194 \def\loop#1\repeat{%
195 \def\body{#1}%
196 \iterate
197 }%

```

```

198 \def\iterate{%
199   \body
200   \let\next\iterate
201   \else
202   \let\next\relax
203   \fi
204   \next
205 }%
206 \let\repeat=\fi
207 }%
208 \def\RestoreCatcodes{}
209 \count@=0 %
210 \loop
211   \edef\RestoreCatcodes{%
212     \RestoreCatcodes
213     \catcode\the\count@=\the\catcode\count@\relax
214   }%
215   \ifnum\count@<255 %
216     \advance\count@ 1 %
217   \repeat
218
219 \def\RangeCatcodeInvalid#1#2{%
220   \count@=#1\relax
221   \loop
222     \catcode\count@=15 %
223     \ifnum\count@<#2\relax
224       \advance\count@ 1 %
225     \repeat
226 }
227 \expandafter\ifx\csname LoadCommand\endcsname\relax
228   \def\LoadCommand{\input hyphsubst.sty\relax}%
229 \fi
230 \def\Test{%
231   \RangeCatcodeInvalid{0}{47}%
232   \RangeCatcodeInvalid{58}{64}%
233   \RangeCatcodeInvalid{91}{96}%
234   \RangeCatcodeInvalid{123}{255}%
235   \catcode'\@=12 %
236   \catcode'\=0 %
237   \catcode'\{=1 %
238   \catcode'\}=2 %
239   \catcode'\#=6 %
240   \catcode'\[=12 %
241   \catcode'\]=12 %
242   \catcode'\%=14 %
243   \catcode'\ =10 %
244   \catcode13=5 %
245   \LoadCommand
246   \RestoreCatcodes
247 }
248 \Test
249 \csname @@end\endcsname
250 \end
251 \test1)

```

3.2 Main tests

```

252 (*test2)
253 \input hyphsubst.sty\relax
254
255 \catcode'\@=11\relax
256 \ifx\et@xlang@undefined
257   \def\l#1{\csname l@#1\endcsname}%

```

```

258 \else
259 \def\l#1{\csname lang@#1\endcsname}%
260 \fi
261 \def\Check#1#2{%
262 \ifnum#1=#2\relax
263 \else
264 \@PackageError{test}{Wrong number: #1 <> #2}\@ehc
265 \fi
266 }
267
268 \language=0\relax
269 \HyphSubstLet{ZeroSaved}{ngerman}
270 \Check{\l{USenglish}}{0}%
271 \HyphSubstLet{USenglish}{ngerman}
272 \Check{\l{USenglish}}{\l{ngerman}}
273 \ifnum\l{USenglish}>0 %
274 \else
275 \@PackageError{test}{\string\language\space is not updated}\@ehc
276 \fi
277 \HyphSubstLet{german}{ngerman}
278 \Check{\l{german}}{\l{ngerman}}
279 \Check{\l{USenglish}}{\l{ngerman}}
280 \csname @@end\endcsname\end
281 </test2>

```

4 Installation

4.1 Download

Package. This package is available on CTAN¹:

[CTAN:macros/latex/contrib/oberdiek/hyphsubst.dtx](#) The source file.

[CTAN:macros/latex/contrib/oberdiek/hyphsubst.pdf](#) Documentation.

Bundle. All the packages of the bundle ‘oberdiek’ are also available in a TDS compliant ZIP archive. There the packages are already unpacked and the documentation files are generated. The files and directories obey the TDS standard.

[CTAN:install/macros/latex/contrib/oberdiek.tds.zip](#)

TDS refers to the standard “A Directory Structure for T_EX Files” ([CTAN:tds/tds.pdf](#)). Directories with `texmf` in their name are usually organized this way.

4.2 Bundle installation

Unpacking. Unpack the `oberdiek.tds.zip` in the TDS tree (also known as `texmf` tree) of your choice. Example (linux):

```
unzip oberdiek.tds.zip -d ~/texmf
```

Script installation. Check the directory `TDS:scripts/oberdiek/` for scripts that need further installation steps. Package `attachfile2` comes with the Perl script `pdfatfi.pl` that should be installed in such a way that it can be called as `pdfatfi`. Example (linux):

```
chmod +x scripts/oberdiek/pdfatfi.pl
cp scripts/oberdiek/pdfatfi.pl /usr/local/bin/
```

¹<http://ftp.ctan.org/tex-archive/>

4.3 Package installation

Unpacking. The `.dtx` file is a self-extracting `docstrip` archive. The files are extracted by running the `.dtx` through plain- \TeX :

```
tex hyphsubst.dtx
```

TDS. Now the different files must be moved into the different directories in your installation TDS tree (also known as `texmf` tree):

```
hyphsubst.sty      → tex/generic/oberdiek/hyphsubst.sty
hyphsubst.pdf      → doc/latex/oberdiek/hyphsubst.pdf
test/hyphsubst-test1.tex → doc/latex/oberdiek/test/hyphsubst-test1.tex
test/hyphsubst-test2.tex → doc/latex/oberdiek/test/hyphsubst-test2.tex
hyphsubst.dtx      → source/latex/oberdiek/hyphsubst.dtx
```

If you have a `docstrip.cfg` that configures and enables `docstrip`'s TDS installing feature, then some files can already be in the right place, see the documentation of `docstrip`.

4.4 Refresh file name databases

If your \TeX distribution (`te \TeX` , `mik \TeX` , ...) relies on file name databases, you must refresh these. For example, `te \TeX` users run `texhash` or `mktextlsr`.

4.5 Some details for the interested

Attached source. The PDF documentation on CTAN also includes the `.dtx` source file. It can be extracted by AcrobatReader 6 or higher. Another option is `pdftk`, e.g. unpack the file into the current directory:

```
pdftk hyphsubst.pdf unpack_files output .
```

Unpacking with \LaTeX . The `.dtx` chooses its action depending on the format:

plain- \TeX : Run `docstrip` and extract the files.

\LaTeX : Generate the documentation.

If you insist on using \LaTeX for `docstrip` (really, `docstrip` does not need \LaTeX), then inform the autodetect routine about your intention:

```
latex \let\install=y\input{hyphsubst.dtx}
```

Do not forget to quote the argument according to the demands of your shell.

Generating the documentation. You can use both the `.dtx` or the `.drv` to generate the documentation. The process can be configured by the configuration file `ltxdoc.cfg`. For instance, put this line into this file, if you want to have A4 as paper format:

```
\PassOptionsToClass{a4paper}{article}
```

An example follows how to generate the documentation with `pdf \LaTeX` :

```
pdflatex hyphsubst.dtx
makeindex -s gind.ist hyphsubst.idx
pdflatex hyphsubst.dtx
makeindex -s gind.ist hyphsubst.idx
pdflatex hyphsubst.dtx
```

5 History

[2008/06/07 v0.1]

- First public version.

[2008/06/09 v0.2]

- Support for ε -TeX's `language.def` added.
- Fix for undefined `\lmsg`.

6 Index

Numbers written in *italic* refer to the page where the corresponding entry is described; numbers underlined refer to the code line of the definition; numbers in roman refer to the code lines where the entry is used.

Symbols	
<code>\#</code>	177, 239
<code>\%</code>	242
<code>\@</code>	178, 235, 255
<code>\@PackageError</code>	109, 264, 275
<code>\@PackageInfo</code>	138
<code>\@ehc</code>	109, 264, 275
<code>\@firstofone</code>	186, 189
<code>\@firstoftwo</code>	151, <u>154</u>
<code>\@gobble</code>	183, 191
<code>\@secondoftwo</code>	149, <u>157</u>
<code>\@undefined</code>	53, 256
<code>\[</code>	240
<code>\]</code>	236
<code>\{</code>	175, 237
<code>\}</code>	176, 238
<code>\]</code>	241
<code>_</code>	243
A	
<code>\advance</code>	216, 224
B	
<code>\body</code>	195, 199
C	
<code>\catcode</code>	3, 4, 5, 6, 7, 18, 19, 20, 34, 35, 36, 37, 38, 39, 40, 41, 42, 43, 44, 65, 66, 69, 70, 71, 72, 76, 77, 78, 79, 83, 85, 175, 176, 177, 178, 213, 222, 235, 236, 237, 238, 239, 240, 241, 242, 243, 244, 255
<code>\Check</code>	261, 270, 272, 278, 279
<code>\count@</code>	180, 209, 213, 215, 216, 220, 222, 223, 224
<code>\countdef</code>	180
<code>\csname</code>	8, 21, 45, 61, 68, 94, 100, 108, 112, 114, 119, 121, 124, 127, 136, 137, 140, 148, 154, 157, 161, 179, 182, 185, 188, 227, 249, 257, 259, 280
<code>\CurrentOption</code>	166
D	
<code>\DeclareOption</code>	165
E	
<code>\empty</code>	12
<code>\end</code>	250, 280
<code>\endcsname</code>	8, 21, 45, 61, 68, 94, 100, 108, 112, 114, 119, 121, 124, 127, 136, 137, 140, 148, 154, 157, 161, 179, 182, 185, 188, 227, 249, 257, 259, 280
<code>\endinput</code>	30, 163
<code>\et@xlang</code>	256
H	
<code>\HyphSubst@AtEnd</code>	81, 82, 162, 172
<code>\HyphSubst@l</code> <u>99</u> , 108, 112, 114, 119, 121, 124, 127, 136, 137, 140, 148	
<code>\HyphSubst@Option</code>	166, 168
<code>\HyphSubstIfExists</code>	2, <u>146</u>
<code>\HyphSubstLet</code> <u>2</u> , <u>105</u> , 169, 269, 271, 277	
I	
<code>\ifcase</code>	9
<code>\ifnum</code>	124, 215, 223, 262, 273
<code>\ifx</code>	10, 12, 21, 45, 53, 56, 94, 100, 108, 112, 148, 154, 157, 161, 179, 182, 185, 188, 227, 256
<code>\immediate</code>	23, 47
<code>\input</code>	95, 228, 253
<code>\iterate</code>	196, 198, 200
L	
<code>\l</code> ...	257, 259, 270, 272, 273, 278, 279
<code>\language</code>	124, 126, 131, 268, 275
<code>\lmsg</code>	111, 129, 141
<code>\LoadCommand</code>	228, 245
<code>\loop</code>	194, 210, 221
M	
<code>\MessageBreak</code>	115, 120, 122, 130
<code>\msg</code>	113, 118, 139

N		S	
\next	200, 202, 204	\space	131, 275
\number	121, 127, 140		
P		T	
\PackageInfo	26	\Test	230, 248
\ProcessOptions	171	\the	69, 70, 71, 72, 83, 213
\ProvidesPackage	62	\TMP@EnsureCode	80, 87, 88, 89, 90, 91, 92
R		W	
\RangeCatcodeInvalid	219, 231, 232, 233, 234	\write	23, 47
\repeat	194, 206, 217, 225		
\RequirePackage	97	X	
\RestoreCatcodes	208, 211, 212, 246	\x	8, 10, 12, 22, 26, 28, 46, 51, 61, 67, 75, 107, 125, 144