

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	4
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	8
4.4	Refresh file name databases	9
4.5	Some details for the interested	9
5	History	9
	[2008/06/07 v0.1]	9
	[2008/06/09 v0.2]	9
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\relax
54         \xdef#1{#3}%
55       \fi
56     }%
57   \fi
58   \expandafter\x\csname ver@hyphsubst.sty\endcsname
59   \ProvidesPackage{hyphsubst}%
60   [2008/06/09 v0.2 Substitute hyphenation patterns (HO)]
61 \begingroup
62   \catcode123 1 % {
63   \catcode125 2 % }
64   \def\x{\endgroup
65     \expandafter\edef\csname HyphSubst@AtEnd\endcsname{%
66       \catcode35 \the\catcode35\relax
67       \catcode64 \the\catcode64\relax
68       \catcode123 \the\catcode123\relax
69       \catcode125 \the\catcode125\relax
70     }%
71   }%
72 \x
73 \catcode35 6 % #
74 \catcode64 11 % @
75 \catcode123 1 % {
76 \catcode125 2 % }
77 \def\TMP@EnsureCode#1#2{%
78   \edef\HyphSubst@AtEnd{%
79     \HyphSubst@AtEnd
80     \catcode#1 \the\catcode#1\relax
81   }%
82   \catcode#1 #2\relax
83 }
84 \TMP@EnsureCode{39}{12}% '
85 \TMP@EnsureCode{46}{12}% .
86 \TMP@EnsureCode{47}{12}% /
87 \TMP@EnsureCode{58}{12}% :
88 \TMP@EnsureCode{61}{12}% =
89 \TMP@EnsureCode{96}{12}% `
```

2.2 Package

```
90 \begingroup\expandafter\expandafter\expandafter\endgroup
```

```

91 \expandafter\ifx\csname RequirePackage\endcsname\relax
92   \input infwarerr.sty\relax
93 \else
94   \RequirePackage{infwarerr}[2007/09/09]%
95 \fi

```

\HyphSubst@l

```

96 \begingroup\expandafter\expandafter\expandafter\endgroup
97 \expandafter\ifx\csname et@xlang\endcsname\relax
98   \def\HyphSubst@l{10}%
99 \else
100   \def\HyphSubst@l{lang@}%
101 \fi

```

\HyphSubstLet

```

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

```

\HyphSubstIfExists

```

143 \def\HyphSubstIfExists#1{%
144   \begingroup\expandafter\expandafter\expandafter\endgroup
145   \expandafter\ifx\csname\HyphSubst@l#1\endcsname\relax
146     \expandafter\@secondoftwo

```

```

147 \else
148 \expandafter\@firstoftwo
149 \fi
150 }

\@firstoftwo

151 \expandafter\ifx\csname @firstoftwo\endcsname\relax
152 \long\def\@firstoftwo#1#2{#1}%
153 \fi

\@secondoftwo

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

157 \begingroup\expandafter\expandafter\expandafter\endgroup
158 \expandafter\ifx\csname documentclass\endcsname\relax
159 \HyphSubst@AtEnd
160 \expandafter\endinput
161 \fi

162 \DeclareOption*{%
163 \expandafter\HyphSubst@Option\CurrentOption==\relax
164 }
165 \def\HyphSubst@Option#1=#2=#3\relax{%
166 \HyphSubstLet{#1}{#2}%
167 }
168 \ProcessOptions*\relax

169 \HyphSubst@AtEnd
170 \end{package}

```

3 Test

3.1 Catcode checks for loading

```

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

```

```

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

```

3.2 Main tests

```

249 <*test2>
250 \input hyphsubst.sty\relax
251
252 \catcode'\@=11\relax
253 \ifx\et@xlang\@undefined
254   \def\l#1{\csname l@#1\endcsname}%
255 \else
256   \def\l#1{\csname lang@#1\endcsname}%

```

```

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

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
```

¹[ftp://ftp.ctan.org/tex-archive/](http://ftp.ctan.org/tex-archive/)

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		H	
\#	174, 236	\HyphSubst@AtEnd	78, 79, 159, 169
\%	239	\HyphSubst@l	<u>96</u> , 105, 109, 111, 116, 118, 121, 124, 133, 134, 137, 145
\@	175, 232, 252	\HyphSubst@Option	163, 165
\@PackageError	106, 261, 272	\HyphSubstIfExists	<u>2</u> , <u>143</u>
\@PackageInfo	135	\HyphSubstLet	<u>2</u> , <u>102</u> , 166, 266, 268, 274
\@ehc	106, 261, 272	I	
\@firstofone	183, 186	\ifcase	9
\@firstoftwo	148, <u>151</u>	\ifnum	121, 212, 220, 259, 270
\@gobble	180, 188	\ifx	10, 12, 21, 45, 53, 91, 97, 105, 109, 145, 151, 154, 158, 176, 179, 182, 185, 224, 253
\@secondoftwo	146, <u>154</u>	\immediate	23, 47
\@undefined	253	\input	92, 225, 250
\[237	\iterate	193, 195, 197
\\	233	L	
\{	172, 234	\l	254, 256, 267, 269, 270, 275, 276
\}	173, 235	\language	121, 123, 128, 265, 272
\]	238	\lmsg	108, 126, 138
_	240	\LoadCommand	225, 242
A		\loop	191, 207, 218
\advance	213, 221	M	
B		\MessageBreak	112, 117, 119, 127
\body	192, 196	\msg	110, 115, 136
C		N	
\catcode	3, 4, 5, 6, 7, 18, 19, 20, 34, 35, 36, 37, 38, 39, 40, 41, 42, 43, 44, 62, 63, 66, 67, 68, 69, 73, 74, 75, 76, 80, 82, 172, 173, 174, 175, 210, 219, 232, 233, 234, 235, 236, 237, 238, 239, 240, 241, 252	\next	197, 199, 201
\Check	258, 267, 269, 275, 276	\number	118, 124, 137
\count@	177, 206, 210, 212, 213, 217, 219, 220, 221	P	
\countdef	177	\PackageInfo	26
\csname	8, 21, 45, 58, 65, 91, 97, 105, 109, 111, 116, 118, 121, 124, 133, 134, 137, 145, 151, 154, 158, 176, 179, 182, 185, 224, 246, 254, 256, 277	\ProcessOptions	168
\CurrentOption	163	\ProvidesPackage	59
D		R	
\DeclareOption	162	\RangeCatcodeInvalid	216, 228, 229, 230, 231
E		\repeat	191, 203, 214, 222
\empty	12	\RequirePackage	94
\end	247, 277	\RestoreCatcodes	205, 208, 209, 243
\endcsname	8, 21, 45, 58, 65, 91, 97, 105, 109, 111, 116, 118, 121, 124, 133, 134, 137, 145, 151, 154, 158, 176, 179, 182, 185, 224, 246, 254, 256, 277	S	
\endinput	30, 160	\space	128, 272
\et@xlang	253	T	
		\Test	227, 245
		\the	66, 67, 68, 69, 80, 210
		\TMP@EnsureCode	77, 84, 85, 86, 87, 88, 89
		W	
		\write	23, 47
		X	
		\x	8, 10, 12, 22, 26, 28, 46, 51, 58, 64, 72, 104, 122, 141