

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	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 \catcode123 1 % {
9 \catcode125 2 % }
10 \expandafter\let\expandafter\x\csname ver@hyphsubst.sty\endcsname
11 \ifx\x\relax % plain-TeX, first loading
12 \else
13 \def\empty{}%
14 \ifx\x\empty % LaTeX, first loading,
15 % variable is initialized, but \ProvidesPackage not yet seen
16 \else
17 \catcode35 6 % #
18 \expandafter\ifx\csname PackageInfo\endcsname\relax
19 \def\x#1#2{%
20 \immediate\write-1{Package #1 Info: #2.}%
21 }%
22 \else
23 \def\x#1#2{\PackageInfo{#1}{#2, stopped}}%
24 \fi
25 \x{hyphsubst}{The package is already loaded}%
26 \aftergroup\endinput
27 \fi
28 \fi
29 \endgroup

```

Package identification:

```

30 \begingroup
31 \catcode35 6 % #
32 \catcode40 12 % (

```

```

33 \catcode41 12 % )
34 \catcode44 12 % ,
35 \catcode45 12 % -
36 \catcode46 12 % .
37 \catcode47 12 % /
38 \catcode58 12 % :
39 \catcode64 11 % @
40 \catcode91 12 % [
41 \catcode93 12 % ]
42 \catcode123 1 % {
43 \catcode125 2 % }
44 \expandafter\ifx\csname ProvidesPackage\endcsname\relax
45   \def\x#1#2#3[#4]{\endgroup
46     \immediate\write-1{Package: #3 #4}%
47     \xdef#1{#4}%
48   }%
49 \else
50   \def\x#1#2[#3]{\endgroup
51     #2[{#3}]%
52     \ifx#1\@undefined
53       \xdef#1{#3}%
54     \fi
55     \ifx#1\relax
56       \xdef#1{#3}%
57     \fi
58   }%
59 \fi
60 \expandafter\x\csname ver@hyphsubst.sty\endcsname
61 \ProvidesPackage{hyphsubst}%
62 [2008/06/09 v0.2 Substitute hyphenation patterns (H0)]
63 \begingroup
64   \catcode123 1 % {
65   \catcode125 2 % }
66   \def\x{\endgroup
67     \expandafter\edef\csname HyphSubst@AtEnd\endcsname{%
68       \catcode35 \the\catcode35\relax
69       \catcode64 \the\catcode64\relax
70       \catcode123 \the\catcode123\relax
71       \catcode125 \the\catcode125\relax
72     }%
73   }%
74 \x
75 \catcode35 6 % #
76 \catcode64 11 % @
77 \catcode123 1 % {
78 \catcode125 2 % }
79 \def\TMP@EnsureCode#1#2{%
80   \edef\HyphSubst@AtEnd{%
81     \HyphSubst@AtEnd
82     \catcode#1 \the\catcode#1\relax
83   }%
84   \catcode#1 #2\relax
85 }
86 \TMP@EnsureCode{39}{12}% '
87 \TMP@EnsureCode{46}{12}% .
88 \TMP@EnsureCode{47}{12}% /
89 \TMP@EnsureCode{58}{12}% :
90 \TMP@EnsureCode{61}{12}% =
91 \TMP@EnsureCode{96}{12}% ‘

```

2.2 Package

```

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

```

\HyphSubst@l

```

98 \begingroup\expandafter\expandafter\expandafter\endgroup
99 \expandafter\ifx\csname et@xlang\endcsname\relax
100   \def\HyphSubst@l{1@}%
101 \else
102   \def\HyphSubst@l{lang@}%
103 \fi

```

\HyphSubstLet

```

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

```

\HyphSubstIfExists

```

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

```

```

148     \expandafter\@secondoftwo
149   \else
150     \expandafter\@firstoftwo
151   \fi
152 }

\@firstoftwo

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

\@secondoftwo

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

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

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

```

3 Test

3.1 Catcode checks for loading

```

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

```

```

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

```

3.2 Main tests

```

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

```

```

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

[2008/06/07 v0.1]

- [2008/06/09 v0.2]

- ## 6 Index

[illegible]

\msg	112, 117, 138	\RestoreCatcodes ..	207, 210, 211, 245
N		S	
\next	199, 201, 203	\space	130, 274
\number	120, 126, 139	T	
P		\Test	229, 247
\PackageInfo	23	\the	68, 69, 70, 71, 82, 212
\ProcessOptions	170	\TMP@EnsureCode	79, 86, 87, 88, 89, 90, 91
\ProvidesPackage	15, 61	W	
R		\write	20, 46
\RangeCatcodeInvalid		X	
.....	218, 230, 231, 232, 233	\x	10, 11, 14, 19, 23,
\repeat	193, 205, 216, 224		25, 45, 50, 60, 66, 74, 106, 124, 143
\RequirePackage	96		