

The hyphsubst package

Heiko Oberdiek
<heiko.oberdiek at googlemail.com>

2008/06/09 v0.2

Abstract

A T_EX 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	L ^A T _E X	2
1.4	plain T _E X	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	8
4	Installation	9
4.1	Download	9
4.2	Bundle installation	9
4.3	Package installation	9
4.4	Refresh file name databases	9
4.5	Some details for the interested	10
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 T_EX

The package can be loaded and used with plain T_EX, 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\catcode61\catcode48\catcode32=10\relax%
3 \catcode13=5 % ^~M
4 \endlinechar=13 %
5 \catcode35=6 % #
6 \catcode39=12 % '
7 \catcode44=12 % ,
8 \catcode45=12 % -
9 \catcode46=12 % .
10 \catcode58=12 % :
11 \catcode64=11 % @
12 \catcode123=1 % {
13 \catcode125=2 % }
14 \expandafter\let\expandafter\x\csname ver@hyphsubst.sty\endcsname
15 \ifx\x\relax % plain-TeX, first loading
16 \else
17 \def\empty{}%
18 \ifx\x\empty % LaTeX, first loading,
19 % variable is initialized, but \ProvidesPackage not yet seen
20 \else
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 \aftergroup\endinput
30 \fi
31 \fi
32 \endgroup%

```

Package identification:

```

33 \begingroup\catcode61\catcode48\catcode32=10\relax%
34 \catcode13=5 % ^^M
35 \endlinechar=13 %
36 \catcode35=6 % #
37 \catcode39=12 % '
38 \catcode40=12 % (
39 \catcode41=12 % )
40 \catcode44=12 % ,
41 \catcode45=12 % -
42 \catcode46=12 % .
43 \catcode47=12 % /
44 \catcode58=12 % :
45 \catcode64=11 % @
46 \catcode91=12 % [
47 \catcode93=12 % ]
48 \catcode123=1 % {
49 \catcode125=2 % }
50 \expandafter\ifx\csname ProvidesPackage\endcsname\relax
51   \def\x#1#2#3[#4]{\endgroup
52     \immediate\write-1{Package: #3 #4}%
53     \xdef#1{#4}%
54   }%
55 \else
56   \def\x#1#2[#3]{\endgroup
57     #2[{#3}]%
58     \ifx#1@undefined
59       \xdef#1{#3}%
60     \fi
61     \ifx#1\relax
62       \xdef#1{#3}%
63     \fi
64   }%
65 \fi
66 \expandafter\x\csname ver@hyphsubst.sty\endcsname
67 \ProvidesPackage{hyphsubst}%
68 [2008/06/09 v0.2 Substitute hyphenation patterns (HO)]%
69 \begingroup\catcode61\catcode48\catcode32=10\relax%
70 \catcode13=5 % ^^M
71 \endlinechar=13 %
72 \catcode123=1 % {
73 \catcode125=2 % }
74 \catcode64=11 % @
75 \def\x{\endgroup
76   \expandafter\edef\csname HyphSubst@AtEnd\endcsname{
77     \endlinechar=\the\endlinechar\relax
78     \catcode13=\the\catcode13\relax
79     \catcode32=\the\catcode32\relax
80     \catcode35=\the\catcode35\relax
81     \catcode61=\the\catcode61\relax
82     \catcode64=\the\catcode64\relax
83     \catcode123=\the\catcode123\relax
84     \catcode125=\the\catcode125\relax
85   }%
86 }%
87 \x\catcode61\catcode48\catcode32=10\relax%
88 \catcode13=5 % ^^M
89 \endlinechar=13 %
90 \catcode35=6 % #
91 \catcode64=11 % @
92 \catcode123=1 % {
93 \catcode125=2 % }

```

```

94 \def\TMP@EnsureCode#1#2{%
95   \edef\HyphSubst@AtEnd{%
96     \HyphSubst@AtEnd
97     \catcode#1=\the\catcode#1\relax
98   }%
99   \catcode#1=#2\relax
100 }
101 \TMP@EnsureCode{39}{12}% '
102 \TMP@EnsureCode{46}{12}% .
103 \TMP@EnsureCode{47}{12}% /
104 \TMP@EnsureCode{58}{12}% :
105 \TMP@EnsureCode{91}{12}% [
106 \TMP@EnsureCode{93}{12}% ]
107 \TMP@EnsureCode{96}{12}% '
108 \edef\HyphSubst@AtEnd{\HyphSubst@AtEnd\noexpand\endinput}

```

2.2 Package

```

109 \begingroup\expandafter\expandafter\expandafter\endgroup
110 \expandafter\ifx\csname RequirePackage\endcsname\relax
111   \input infwarerr.sty\relax
112 \else
113   \RequirePackage{infwarerr}[2007/09/09]%
114 \fi

```

\HyphSubst@l

```

115 \begingroup\expandafter\expandafter\expandafter\endgroup
116 \expandafter\ifx\csname et@xlang\endcsname\relax
117   \def\HyphSubst@l{10}%
118 \else
119   \def\HyphSubst@l{lang}%
120 \fi

```

\HyphSubstLet

```

121 \def\HyphSubstLet#1#2{%
122   \begingroup
123     \def\x{%
124       \expandafter\ifx\csname\HyphSubst@l#2\endcsname\relax
125         \@PackageError{hyphsubst}{Unknown pattern '#2'}{\@ehc
126       \else
127         \def\lmsg{%
128           \expandafter\ifx\csname\HyphSubst@l#1\endcsname\relax
129             \edef\msg{%
130               New: \expandafter\string\csname\HyphSubst@l#1\endcsname
131               \noexpand\MessageBreak
132             }%
133           \else
134             \edef\msg{%
135               Redefined: \expandafter\string\csname\HyphSubst@l#1\endcsname
136               \noexpand\MessageBreak
137               old value: \number\csname\HyphSubst@l#1\endcsname
138               \noexpand\MessageBreak
139             }%
140             \ifnum\csname\HyphSubst@l#1\endcsname=\language
141               \edef\x{%
142                 \noexpand\language=%
143                 \number\csname\HyphSubst@l#2\endcsname\relax
144               }%
145             \edef\lmsg{%
146               \noexpand\MessageBreak
147               \string\language\noexpand\space updated%
148             }%

```

```

149     \fi
150     \fi
151     \expandafter\global\expandafter\let
152         \csname\HyphSubst@l#1\expandafter\endcsname
153         \csname\HyphSubst@l#2\endcsname
154     \@PackageInfo{hyphsubst}{%
155         \msg
156         new value: \number\csname\HyphSubst@l#1\endcsname
157         \lmsg
158     }%
159     \fi
160 \expandafter\endgroup\x
161 }

```

`\HyphSubstIfExists`

```

162 \def\HyphSubstIfExists#1{%
163     \begingroup\expandafter\expandafter\expandafter\endgroup
164     \expandafter\ifx\csname\HyphSubst@l#1\endcsname\relax
165         \expandafter\@secondoftwo
166     \else
167         \expandafter\@firstoftwo
168     \fi
169 }

```

`\@firstoftwo`

```

170 \expandafter\ifx\csname @firstoftwo\endcsname\relax
171     \long\def\@firstoftwo#1#2{#1}%
172 \fi

```

`\@secondoftwo`

```

173 \expandafter\ifx\csname @secondoftwo\endcsname\relax
174     \long\def\@secondoftwo#1#2{#2}%
175 \fi

176 \begingroup\expandafter\expandafter\expandafter\endgroup
177 \expandafter\ifx\csname documentclass\endcsname\relax
178     \expandafter\HyphSubst@AtEnd
179 \fi%

180 \DeclareOption*{%
181     \expandafter\HyphSubst@Option\CurrentOption==\relax
182 }
183 \def\HyphSubst@Option#1=#2=#3\relax{%
184     \HyphSubstLet{#1}{#2}%
185 }
186 \ProcessOptions*\relax
187 \HyphSubst@AtEnd%
188 \</package>

```

3 Test

3.1 Catcode checks for loading

```

189 (*test1)

190 \catcode'\{=1 %
191 \catcode'\}=2 %
192 \catcode'\#=6 %
193 \catcode'\@=11 %
194 \expandafter\ifx\csname count@\endcsname\relax
195     \countdef\count@=255 %
196 \fi

```

```

197 \expandafter\ifx\csname @gobble\endcsname\relax
198   \long\def\@gobble#1{}%
199 \fi
200 \expandafter\ifx\csname @firstofone\endcsname\relax
201   \long\def\@firstofone#1{#1}%
202 \fi
203 \expandafter\ifx\csname loop\endcsname\relax
204   \expandafter\@firstofone
205 \else
206   \expandafter\@gobble
207 \fi
208 {%
209   \def\loop#1\repeat{%
210     \def\body{#1}%
211     \iterate
212   }%
213   \def\iterate{%
214     \body
215     \let\next\iterate
216   \else
217     \let\next\relax
218   \fi
219   \next
220 }%
221 \let\repeat=\fi
222 }%
223 \def\RestoreCatcodes{}
224 \count@=0 %
225 \loop
226   \edef\RestoreCatcodes{%
227     \RestoreCatcodes
228     \catcode\the\count@=\the\catcode\count@\relax
229   }%
230 \ifnum\count@<255 %
231   \advance\count@ 1 %
232 \repeat
233
234 \def\RangeCatcodeInvalid#1#2{%
235   \count@=#1\relax
236   \loop
237     \catcode\count@=15 %
238     \ifnum\count@<#2\relax
239       \advance\count@ 1 %
240     \repeat
241 }
242 \def\RangeCatcodeCheck#1#2#3{%
243   \count@=#1\relax
244   \loop
245     \ifnum#3=\catcode\count@
246   \else
247     \errmessage{%
248       Character \the\count@\space
249       with wrong catcode \the\catcode\count@\space
250       instead of \number#3%
251     }%
252   \fi
253   \ifnum\count@<#2\relax
254     \advance\count@ 1 %
255   \repeat
256 }
257 \def\space{ }
258 \expandafter\ifx\csname LoadCommand\endcsname\relax

```

```

259 \def\LoadCommand{\input hyphsubst.sty\relax}%
260 \fi
261 \def\Test{%
262   \RangeCatcodeInvalid{0}{47}%
263   \RangeCatcodeInvalid{58}{64}%
264   \RangeCatcodeInvalid{91}{96}%
265   \RangeCatcodeInvalid{123}{255}%
266   \catcode'\@=12 %
267   \catcode'\=0 %
268   \catcode'\%=14 %
269   \LoadCommand
270   \RangeCatcodeCheck{0}{36}{15}%
271   \RangeCatcodeCheck{37}{37}{14}%
272   \RangeCatcodeCheck{38}{47}{15}%
273   \RangeCatcodeCheck{48}{57}{12}%
274   \RangeCatcodeCheck{58}{63}{15}%
275   \RangeCatcodeCheck{64}{64}{12}%
276   \RangeCatcodeCheck{65}{90}{11}%
277   \RangeCatcodeCheck{91}{91}{15}%
278   \RangeCatcodeCheck{92}{92}{0}%
279   \RangeCatcodeCheck{93}{96}{15}%
280   \RangeCatcodeCheck{97}{122}{11}%
281   \RangeCatcodeCheck{123}{255}{15}%
282   \RestoreCatcodes
283 }
284 \Test
285 \csname @@end\endcsname
286 \end
287 </test1>

```

3.2 Main tests

```

288 (*test2)
289 \input hyphsubst.sty\relax
290
291 \catcode'\@=11\relax
292 \ifx\et@xlang@undefined
293   \def\l#1{\csname l@#1\endcsname}%
294 \else
295   \def\l#1{\csname lang@#1\endcsname}%
296 \fi
297 \def\Check#1#2{%
298   \ifnum#1=#2\relax
299   \else
300     \@PackageError{test}{Wrong number: #1 <> #2}\@ehc
301   \fi
302 }
303
304 \language=0\relax
305 \HyphSubstLet{ZeroSaved}{ngerman}
306 \Check{\l{USenglish}}{0}%
307 \HyphSubstLet{USenglish}{ngerman}
308 \Check{\l{USenglish}}{\l{ngerman}}
309 \ifnum\l{USenglish}>0 %
310 \else
311   \@PackageError{test}{\string\language\space is not updated}\@ehc
312 \fi
313 \HyphSubstLet{german}{ngerman}
314 \Check{\l{german}}{\l{ngerman}}
315 \Check{\l{USenglish}}{\l{ngerman}}
316 \csname @@end\endcsname\end
317 </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/
```

4.3 Package installation

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

```
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):

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

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 T_EX distribution (teT_EX, miK_TE_X, ...) relies on file name databases, you must refresh these. For example, teT_EX users run `texhash` or `mktextlsr`.

¹<http://ftp.ctan.org/tex-archive/>

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:

Unpacking with L^AT_EX. The .dtx chooses its action depending on the format:

L^AT_EX: Generate the documentation.

$$\text{latex } \backslash \text{let} \backslash \text{install} = \text{y} \backslash \text{input} \{ \text{hyphsubst.dtx} \}$$

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:

An example follows how to generate the documentation with pdfL^AT_EX:

5 History

- First public version.

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

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

10

