

# The hyphsubst package

Heiko Oberdiek\*

2016/05/16 v0.3

## Abstract

A  $\text{\TeX}$  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

<b>1</b>	<b>Documentation</b>	<b>1</b>
1.1	In short	1
1.2	Longer version	2
1.3	$\text{\LaTeX}$	2
1.4	plain $\text{\TeX}$	3
<b>2</b>	<b>Implementation</b>	<b>3</b>
2.1	Reload check and package identification	3
2.2	Package	5
<b>3</b>	<b>Installation</b>	<b>6</b>
3.1	Download	6
3.2	Bundle installation	7
3.3	Package installation	7
3.4	Refresh file name databases	7
3.5	Some details for the interested	7
<b>4</b>	<b>History</b>	<b>7</b>
	[2008/06/07 v0.1]	7
	[2008/06/09 v0.2]	8
	[2016/05/16 v0.3]	8
<b>5</b>	<b>Index</b>	<b>8</b>

## 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`.

---

\*Please report any issues at <https://github.com/ho-tex/oberdiek/issues>

## 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>`.  $\epsilon$ -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<sup>A</sup>T<sub>E</sub>X

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<sup>A</sup>T<sub>E</sub>X.

```
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 [2016/05/16 v0.3 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{1@}%
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\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 Installation

### 3.1 Download

**Package.** This package is available on CTAN<sup>1</sup>:

[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:pkg/tds](#)). Directories with `texmf` in their name are usually organized this way.

---

<sup>1</sup>[CTAN:pkg/hyphsubst](#)

## 3.2 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
```

## 3.3 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
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`.

## 3.4 Refresh file name databases

If your  $\TeX$  distribution ( $\TeX$  Live, MiK $\TeX$ , ...) relies on file name databases, you must refresh these. For example,  $\TeX$  Live users run `texhash` or `mktexlsr`.

## 3.5 Some details for the interested

**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
```

# 4 History

[2008/06/07 v0.1]

- First public version.

[2008/06/09 v0.2]

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

[2016/05/16 v0.3]

- Documentation updates.

## 5 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; plain numbers refer to the code lines where the entry is used.

Symbols	
<code>\@PackageError</code> .....	125
<code>\@PackageInfo</code> .....	154
<code>\@ehc</code> .....	125
<code>\@firstoftwo</code> .....	167, <u>170</u>
<code>\@secondoftwo</code> .....	165, <u>173</u>
<code>\@undefined</code> .....	58
<b>A</b>	
<code>\aftergroup</code> .....	29
<b>C</b>	
<code>\catcode</code> 2, 3, 5, 6, 7, 8, 9, 10, 11, 12, 13, 33, 34, 36, 37, 38, 39, 40, 41, 42, 43, 44, 45, 46, 47, 48, 49, 69, 70, 72, 73, 74, 78, 79, 80, 81, 82, 83, 84, 87, 88, 90, 91, 92, 93, 97, 99	
<code>\csname</code> .....	14, 21, 50, 66, 76, 110, 116, 124, 128, 130, 135, 137, 140, 143, 152, 153, 156, 164, 170, 173, 177
<code>\CurrentOption</code> .....	181
<b>D</b>	
<code>\DeclareOption</code> .....	180
<b>E</b>	
<code>\empty</code> .....	17, 18
<code>\endcsname</code> .....	14, 21, 50, 66, 76, 110, 116, 124, 128, 130, 135, 137, 140, 143, 152, 153, 156, 164, 170, 173, 177
<code>\endinput</code> .....	29, 108
<code>\endlinechar</code> .....	4, 35, 71, 77, 89
<b>H</b>	
<code>\HyphSubst@AtEnd</code> 95, 96, 108, 178, 187	
<code>\HyphSubst@l</code> <u>115</u> , 124, 128, 130, 135, 137, 140, 143, 152, 153, 156, 164	
<code>\HyphSubst@Option</code> .....	181, 183
<code>\HyphSubstIfExists</code> .....	2, <u>162</u>
<code>\HyphSubstLet</code> .....	2, <u>121</u> , 184
<b>I</b>	
<code>\ifnum</code> .....	140
<code>\ifx</code> .... 15, 18, 21, 50, 58, 61, 110, 116, 124, 128, 164, 170, 173, 177	
<code>\immediate</code> .....	23, 52
<code>\input</code> .....	111
<b>L</b>	
<code>\language</code> .....	140, 142, 147
<code>\lmsg</code> .....	127, 145, 157
<b>M</b>	
<code>\MessageBreak</code> .....	131, 136, 138, 146
<code>\msg</code> .....	129, 134, 155
<b>N</b>	
<code>\number</code> .....	137, 143, 156
<b>P</b>	
<code>\PackageInfo</code> .....	26
<code>\ProcessOptions</code> .....	186
<code>\ProvidesPackage</code> .....	19, 67
<b>R</b>	
<code>\RequirePackage</code> .....	113
<b>S</b>	
<code>\space</code> .....	147
<b>T</b>	
<code>\the</code> ... 77, 78, 79, 80, 81, 82, 83, 84, 97	
<code>\TMP@EnsureCode</code> .....	94, 101, 102, 103, 104, 105, 106, 107
<b>W</b>	
<code>\write</code> .....	23, 52
<b>X</b>	
<code>\x</code> .....	14, 15, 18, 22, 26, 28, 51, 56, 66, 75, 87, 123, 141, 160