The \texttt{hyphsubst} package

Heiko Oberdiek
\texttt{<heiko.oberdiek at googlemail.com>}

2008/06/09 v0.2

Abstract

A \LaTeX\ format file may include alternative hyphenation patterns for a
language with a different name. If the naming convention follows \texttt{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{} .................................................. 3
  1.4 \texttt{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 ................................................ 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 Catalogue .................................................... 10

6 History ...................................................... 11
  [2008/06/07 v0.1] ........................................... 11
  [2008/06/09 v0.2] ........................................... 11

7 Index .......................................................... 11

1 Documentation

1.1 In short

The package is an experimental package that allows the substitution of hyphen-
ation patterns, example:
The patterns \texttt{ngerman} are replaced by the patterns \texttt{ngerman-x-20080601}. The format must contain these patterns and should use the naming scheme of either \texttt{babel}'s \texttt{language.dat} or \texttt{etex.src}'s \texttt{language.def}.

1.2 Longer version

Assume the format may contain the following hyphenation patterns (excerpt from \texttt{language.dat}):

\begin{verbatim}
... ngerman dehyphn.tex ngerman-x-20071231 dehyphn-x-20071231 ngerman-x-20080601 dehyphn-x-20080601 =ngerman-x-latest \% alias for ngerman-x-20080601 ...
\end{verbatim}

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

This package \texttt{hyphsubst} solves the problem by redefining the macro \texttt{$\backslash l@\langle$name$\rangle$} to use other patterns.

\begin{verbatim}
\HyphSubstLet {\langle nameA \rangle} {\langle nameB \rangle}
\end{verbatim}

\texttt{$\backslash l@\langle$name$\rangle$} now has the same meaning as \texttt{$\backslash l@\langle$nameB$\rangle$}. The patterns for \texttt{nameB} must exist. If the patterns for \texttt{nameA} exist, then they will be overwritten to use the patterns for \texttt{nameB}. Example:

\begin{verbatim}
\documentclass{article}
\usepackage{hyphsubst}
$\text{\HyphSubstLet{ngerman}{ngerman-x-20080601}}$
$\text{\usepackage[ngerman]{babel}}$
\end{verbatim}

Now the patterns \texttt{ngerman-x-20080601} are be used.

Or if you want to compare hyphenations:

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

\begin{verbatim}
\HyphSubstIfExists {\langle name \rangle} {\langle then \rangle} {\langle else \rangle}
\end{verbatim}

Tests if patterns with name \texttt{name} exist and execute \texttt{then} in case of success and \texttt{else} otherwise.
1.3 \LaTeX

The package can also be loaded before \texttt{\documentclass}:

\begin{verbatim}
\RequirePackage[ngerman=ngerman-x-20080601]{hyphsubst}
\documentclass{article}
\end{verbatim}

This allows to put the package in a format file.

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

\begin{verbatim}
\usepackage[ngerman=ngerman-x-20080601]{hyphsubst}
\end{verbatim}

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

\begin{verbatim}
\HyphSubstLet{ngerman}{ngerman-x-20080601}
\end{verbatim}

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.:

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

2 Implementation

1 (*package)

2.1 Reload check and package identification

Reload check, especially if the package is not used with \LaTeX. 

\begin{verbatim}
\begingroup\catcode61=10 \catcode48=13 \catcode32=10 \relax
\endlinechar=13 %
\catcode35=6 % #
\catcode39=12 % '
\catcode44=12 % ,
\catcode45=12 % -
\catcode46=12 % .
\catcode58=12 % :
\catcode64=11 % @
\catcode123=1 % {
\catcode125=2 % }
\expandafter\let\expandafter\x\csname ver@hyphsubst.sty\endcsname
\ifx\x\relax % plain-TeX, first loading
\else
\expandafter\ifx\csname PackageInfo\endcsname\relax
\immediate\write-1{Package #1 Info: #2.}%
\else
\expandafter\ifx\csname PackageInfo\endcsname\relax
\immediate\write-1{Package #1 Info: #2.}
\fi
\fi
\else
\edef\empty{}%
\expandafter\ifx\csname PackageInfo\endcsname\relax
\immediate\write-1{Package #1 Info: #2, stopped}%
\fi
\fi
\endgroup
\end{verbatim}


Package identification:

2
\begin{group}\catcode61=10 \catcode48=10 \catcode32=10
\endlinechar=13
\catcode35=6 \%
\catcode39=12 \%
\catcode40=12 \%
\catcode41=12 \%
\catcode44=12 \%
\catcode45=12 \%
\catcode46=12 \%
\catcode47=12 \%
\catcode58=12 \%
\catcode64=11 \%
\catcode91=12 \%
\catcode93=12 \%
\catcode123=1 \%
\catcode125=2 \%
\expandafter\ifx\csname ProvidesPackage\endcsname\relax
  \def\x#1#2#3[#4]{\endgroup
  \immediate\write-1{Package: #3 #4}%
  \xdef#1{#4}%
  }%
  \else
    \def\x#1#2[#3]{\endgroup
    #2[{#3}]%
    \ifx#1\@undefined
    \xdef#1{#3}%
    \fi
    \ifx#1\relax
    \xdef#1{#3}%
    \fi
    }%
  \fi
  \expandafter\x\csname ver@hyphsubst.sty\endcsname
\ProvidesPackage{hyphsubst}[
[2008/06/09 v0.2 Substitute hyphenation patterns (HO)]%
\begin{group}\catcode61=10 \catcode48=10 \catcode32=10
\endlinechar=13
\catcode35=6 \%
\catcode39=12 \%
\catcode40=12 \%
\catcode41=12 \%
\catcode44=12 \%
\catcode45=12 \%
\catcode46=12 \%
\catcode47=12 \%
\catcode58=12 \%
\catcode64=11 \%
\catcode91=12 \%
\catcode93=12 \%
\catcode123=1 \%
\catcode125=2 \%
\expandafter\ifx\csname ProvidesPackage\endcsname\relax
  \def\x#1#2#3[#4]{\endgroup
  \immediate\write-1{Package: #3 #4}%
  \xdef#1{#4}%
  }%
  \else
    \def\x#1#2[#3]{\endgroup
    #2[{#3}]%
    \ifx#1\@undefined
    \xdef#1{#3}%
    \fi
    \ifx#1\relax
    \xdef#1{#3}%
    \fi
    }%
  \fi
  \expandafter\x\csname ver@hyphsubst.sty\endcsname
\ProvidesPackage{hyphsubst}[
[2008/06/09 v0.2 Substitute hyphenation patterns (HO)]%
\def\TMP@EnsureCode#1#2{%
\edef\HyphSubst@AtEnd{%
\HyphSubst@AtEnd
\catcode#1=\the\catcode#1\relax
}%
\catcode#1=\relax
}
\TMP@EnsureCode{39}{12}% ' 
\TMP@EnsureCode{46}{12}% . 
\TMP@EnsureCode{47}{12}% / 
\TMP@EnsureCode{58}{12}% : 
\TMP@EnsureCode{91}{12}% [ 
\TMP@EnsureCode{93}{12}% ] 
\TMP@EnsureCode{96}{12}% ` 
\edef\HyphSubst@AtEnd{\HyphSubst@AtEnd\noexpand\endinput}

2.2 Package
\begingroup\expandafter\expandafter\expandafter\endgroup
\input infrarerr.sty\relax
\else
\RequirePackage{infrarerr}[2007/09/09]%
\fi
\HyphSubst@l
\begingroup\expandafter\expandafter\expandafter\endgroup
\expandafter\ifx\csname RequirePackage\endcsname\relax
\else
\def\HyphSubst@l{\language@}\
\else
\def\HyphSubst@l{lang@}\
\fi
\HyphSubstLet
\def\HyphSubstLet#1#2{%
\begingroup
\def\x{}\
\expandafter\ifx\csname\HyphSubst@l#2\endcsname\relax
\@PackageError{hyphsubst}{Unknown pattern `#2'}\@ehc
\else
\def\lmsg{}\
\expandafter\ifx\csname\HyphSubst@l#1\endcsname\relax
\edef\msg{New: \string\csname\HyphSubst@l#1\endcsname
\noexpand\MessageBreak}
\else
\edef\msg{Redefined: \string\csname\HyphSubst@l#1\endcsname
\noexpand\MessageBreak
old value: \number\csname\HyphSubst@l#1\endcsname
\noexpand\MessageBreak}
\fi
\ifnum\csname\HyphSubst@l#1\endcsname=\language
\edef\x{\noexpand\language=\number\csname\HyphSubst@l#2\endcsname\relax}
\else
\edef\lmsg{\noexpand\MessageBreak
\string\language\noexpand\space updated\noexpand\MessageBreak}
\fi
\endgroup
\edef\msg{%
Redefined: \string\csname\HyphSubst@l#1\endcsname
\noexpand\MessageBreak
old value: \number\csname\HyphSubst@l#1\endcsname
\noexpand\MessageBreak}
\def\x{}\
\edef\msg{%
\noexpand\MessageBreak
\string\language\noexpand\space updated\noexpand\MessageBreak
}\
\fi
\HyphSubstIfExists
\def\HyphSubstIfExists#1{% 
\begingroup\expandafter\expandafter\expandafter\endgroup\expandafter
\ifx\csname\HyphSubst@l#1\endcsname\relax
\else
\expandafter\@firstoftwo
\fi
\@firstoftwo
\expandafter\ifx\csname\HyphSubst@l#1\endcsname\relax
\long\def\@firstoftwo#1\#2(\#1)%
\fi
\@secondoftwo
\expandafter\ifx\csname\HyphSubst@l#1\endcsname\relax
\long\def\@secondoftwo#1#2(#2)%
\fi
\begingroup\expandafter\expandafter\expandafter\endgroup\expandafter
\ifx\csname\HyphSubst@l#1\endcsname\relax
\expandafter\HyphSubst@AtEnd
\fi%
\DeclareOption*{%
\expandafter\HyphSubst@Option\CurrentOption==\relax
\}
\def\HyphSubst@Option#1=#2=#3\relax{%
\HyphSubstLet{#1}{#2}%
\ProcessOptions*\relax
\HyphSubst@AtEnd%
⟨/𭗉𭖺𭖼𭗄𭖺𭗀𭖾⟩

3 Test

3.1 Catcode checks for loading

⟨*𭗍𭖾𭗌𭗍𭟣⟩
\catcode`{=1 %
\catcode`}=2 %
\catcode`#=6 %
\catcode`@=11 %
\expandafter\ifx\csname\HyphSubst@l#1\endcsname\relax
\long\def\@firstoftwo#1\#2(#2)%
\fi
\expandafter\ifx\csname\HyphSubst@l#1\endcsname\relax
\long\def\@secondoftwo#1#2(#2)%
\fi
\countdef\count@=255 %
\expandafter\ifx\csname\HyphSubst@l#1\endcsname\relax
\countdef\count@=255 %
\fi
⟨/𭗉𭖺𭖼𭗄𭖺𭗀𭖾⟩
\long\def\@gobble#1{}% 
\fi 
\expandafter\ifx\csname @firstofone\endcsname\relax 
\long\def\@firstofone#1{#1}% 
\fi 
\expandafter\ifx\csname \endcsname\relax 
\else 
\expandafter\@gobble 
\fi 
\expandafter\@gobble 
\expandafter\@gobble 
\expandafter\@gobble 
\expandafter\@gobble 

\def\loop\#1\repeat{% 
\def\body{#1}% 
\iterate 
\}\% 
\def\iterate{% 
\body 
\let\next\iterate 
\else 
\let\next\relax 
\fi 
\next 
}% 
\let\repeat=\fi 
}% 
\def\RestoreCatcodes{} 
\count@=0 % 
\loop 
\edef\RestoreCatcodes{% 
\RestoreCatcodes 
\catcode\the\count@=\the\catcode\count@\relax 
}% 
\ifnum\count@<255 % 
\advance\count@ 1 % 
\repeat 
}% 
\def\RangeCatcodeInvalid#1#2{% 
\count@=#1\relax 
\loop 
\catcode\count@=15 % 
\ifnum\count@<#2\relax 
\advance\count@ 1 % 
\repeat 
}% 
\def\RangeCatcodeCheck#1#2#3{% 
\count@=#1\relax 
\loop 
\ifnum#3=\catcode\count@ 
\else 
\errmessage{% 
Character \the\count@ with wrong catcode \the\catcode\count@ instead of \number#3% 
}% 
\fi 
\ifnum\count@<#2\relax 
\advance\count@ 1 % 
\repeat 
}% 
\def\space{ } 
\expandafter\ifx\csname \endcsname\relax 
\def\LoadCommand{\input hyphsubst.sty\relax}%
\def\Test{%
\RangeCatcodeInvalid{0}{47}\
\RangeCatcodeInvalid{58}{64}\
\RangeCatcodeInvalid{91}{96}\
\RangeCatcodeInvalid{123}{255}\
catcode`\@=12 %
catcode`\=0 %
catcode`X=14 %
LoadCommand
\RangeCatcodeCheck{0}{36}{15}\
\RangeCatcodeCheck{37}{37}{14}\
\RangeCatcodeCheck{38}{47}{15}\
\RangeCatcodeCheck{48}{57}{12}\
\RangeCatcodeCheck{58}{63}{15}\
\RangeCatcodeCheck{64}{64}{12}\
\RangeCatcodeCheck{65}{90}{11}\
\RangeCatcodeCheck{91}{91}{15}\
\RangeCatcodeCheck{92}{92}{0}\
\RangeCatcodeCheck{93}{96}{15}\
\RangeCatcodeCheck{97}{122}{11}\
\RangeCatcodeCheck{123}{255}{15}\
RestoreCatcodes
\}
\Test\csname @@end\endcsname\end

3.2 Main tests
\input hyphsubst.sty\relax
\catcode`\@=11\relax
\ifx\et@xlang@undefined
  \def\l#1{\csname l@#1\endcsname}%
\else
  \def\l#1{\csname lang@#1\endcsname}%
\fi
\def\Check#1#2{\
  \ifnum#1=#2\relax
  \else
    \@PackageError{test}{Wrong number: #1 <> #2}\@ehc
  \fi
}\
\language=0\relax
\HyphSubstLet{ZeroSaved}{ngerman}\
\Check{\l{USenglish}}{0}%
\HyphSubstLet{USenglish}{ngerman}\
\Check{\l{USenglish}}{\l{ngerman}}\
\ifnum\l{USenglish}>0 %
  \else
    \@PackageError{test}{\string\language\space is not updated}\@ehc
  \fi
\HyphSubstLet{german}{ngerman}\
\Check{\l{german}}{\l{ngerman}}\
\Check{\l{USenglish}}{\l{ngerman}}
\csname @@end\endcsname\end\fi
\}
4 Installation

4.1 Download

Package. This package is available on CTAN:\footnote{ftp://ftp.ctan.org/tex-archive/}:

\verb|CTAN:macros/latex/contrib/oberdiek/hyphsubst.dtx| The source file.
\verb|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.

\verb|CTAN:install/macros/latex/contrib/oberdiek.tds.zip| \n
TDS refers to the standard “A Directory Structure for \TeX{} Files” (\texttt{CTAN:tds/tds.pdf}). Directories with \texttt{texmf} in their name are usually organized this way.

4.2 Bundle installation

Unpacking. Unpack the \verb|oberdiek.tds.zip| in the TDS tree (also known as \texttt{texmf} tree) of your choice. Example (linux):

\verbatim
unzip oberdiek.tds.zip -d ~/texmf
\endverbatim

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

\verbatim
chmod +x scripts/oberdiek/pdfatfi.pl
cp scripts/oberdiek/pdfatfi.pl /usr/local/bin/
\endverbatim

4.3 Package installation

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

\verbatim
tex hyphsubst.dtx
\endverbatim

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

\begin{verbatim}
hyphsubst.sty    \rightarrow \texttt{tex/generic/oberdiek/hyphsubst.sty}
hyphsubst.pdf   \rightarrow \texttt{doc/latex/oberdiek/hyphsubst.pdf}
test/hyphsubst-test1.tex \rightarrow \texttt{doc/latex/oberdiek/test/hyphsubst-test1.tex}
test/hyphsubst-test2.tex \rightarrow \texttt{doc/latex/oberdiek/test/hyphsubst-test2.tex}
hyphsubst.dtx    \rightarrow \texttt{source/latex/oberdiek/hyphsubst.dtx}
\end{verbatim}

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

4.4 Refresh file name databases

If your \TeX{} distribution (\texttt{tex\TeX{}}, \texttt{mik\TeX{}}, …) relies on file name databases, you must refresh these. For example, \texttt{tex\TeX{}} users run \texttt{texhash} or \texttt{mktexlsr}.

1 ftp://ftp.ctan.org/tex-archive/
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:

\begin{verbatim}
pdftk hyphsubst.pdf unpack_files output .
\end{verbatim}

**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:

\begin{verbatim}
\texttt{latex \let\install=y\input{hyphsubst.dtx}}
\end{verbatim}

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:

\begin{verbatim}
\PassOptionsToClass{a4paper}{article}
\end{verbatim}

An example follows how to generate the documentation with pdflatex:

\begin{verbatim}
pdflatex hyphsubst.dtx
makeindex -s gind.ist hyphsubst.idx
pdflatex hyphsubst.dtx
makeindex -s gind.ist hyphsubst.idx
pdflatex hyphsubst.dtx
\end{verbatim}

5 Catalogue

The following XML file can be used as source for the \TeX Catalogue. The elements `caption` and `description` are imported from the original XML file from the Catalogue. The name of the XML file in the Catalogue is `hyphsubst.xml`.

```xml
<entry datestamp='$Date$' modifier='$Author$' id='hyphsubst'>
  <name>hyphsubst</name>
  <caption>Substitute hyphenation patterns.</caption>
  <authorref id='auth:oberdiek'/>
  <copyright owner='Heiko Oberdiek' year='2008'/>
  <license type='lppl1.3'/>
  <version number='0.2'/>
  <description>
    A \TeX format file may include alternative hyphenation patterns for a language with a different name. If the naming convention follows `<xref refid='babel'>babel’</xref>` rules, then the hyphenation patterns for a language can be replaced by the alternative hyphenation patterns, provided in the format file.
    <p/>
    The package is part of the `<xref refid='oberdiek'>oberdiek</xref>` bundle.
  </description>
```
6 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.

7 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

<table>
<thead>
<tr>
<th>Symbol</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>#</td>
<td>192</td>
</tr>
<tr>
<td>%</td>
<td>268</td>
</tr>
<tr>
<td>@</td>
<td>193, 266, 291</td>
</tr>
<tr>
<td>\PackageError</td>
<td>125, 300, 311</td>
</tr>
<tr>
<td>\PackageInfo</td>
<td>154</td>
</tr>
<tr>
<td>@</td>
<td>125, 300, 311</td>
</tr>
<tr>
<td>@ehc</td>
<td>201, 204</td>
</tr>
<tr>
<td>@firstoftwo</td>
<td>167, 170</td>
</tr>
<tr>
<td>@gobble</td>
<td>198, 206</td>
</tr>
<tr>
<td>@secondoftwo</td>
<td>165, 173</td>
</tr>
<tr>
<td>@undefined</td>
<td>58, 292</td>
</tr>
<tr>
<td>@{}</td>
<td>267</td>
</tr>
<tr>
<td>@{}</td>
<td>190</td>
</tr>
<tr>
<td>@{}</td>
<td>191</td>
</tr>
</tbody>
</table>

A

\advance 231, 239, 254
\aftergroup 29

B

\body 210, 214

C
\catcode 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, 190, 191, 192, 193, 228,
237, 245, 249, 266, 267, 268, 291
\Check 297, 306, 308, 314, 315

D
\count@ 195, 224, 228, 230, 231, 235, 237, 238,
239, 243, 245, 248, 249, 253, 254
\countdef 195
\csname 14, 21, 50, 66,
76, 110, 116, 124, 128, 130, 135,
137, 140, 143, 152, 153, 156,
164, 170, 173, 177, 194, 197,
200, 203, 258, 285, 293, 295, 316
\CurrentOption 181
\DeclareOption 180
\empty 17, 18
\end 286, 316
\endcsname 14, 21, 50, 66,
76, 110, 116, 124, 128, 130, 135,
137, 140, 143, 152, 153, 156,
164, 170, 173, 177, 194, 197,
200, 203, 258, 285, 293, 295, 316
\endinput 29, 108
\endlinechar 4, 35, 71, 77, 89
\errmessage 247
\et@xlang 292

H
\HyphSubst@AtEnd 95, 96, 108, 178, 187
\HyphSubst@l 115, 124, 128, 130, 135,
137, 140, 143, 152, 153, 156, 164
\HyphSubst@Option 181, 183
\HyphSubstIfExists 2, 162
\HyphSubstLet 2, 121, 184, 305, 307, 313