

TRANSLATIONS

vo.10a 2013/07/12

a simple translator

Clemens NIEDERBERGER

<https://github.com/cgnieder/translations/>
contact@mychemistry.eu

English documentation

Contents			
		3.2 A Small Example	3
		3.3 Usage in Packages	4
		3.4 Dictionaries	5
1 Motivation	1		
2 License and Requirements	2	4 Defined Languages	5
3 Usage	2	5 Implementation	6
3.1 Available Commands	2	Index	17

1 Motivation

This package provides means for package authors to have an easy interface for internationalization of their packages. The functionality of this package is in many parts also covered by the package translator (part of the beamer¹ bundle). Internationalization is also possible with babel² and it's `\addto\captions<language>` mechanism or KOMA-Script's `\providecaptionname`. However, I believe that **TRANSLATIONS** is more flexible than all of these. Unlike translator it detects the used (babel or polyglossia³) language itself and provides expandable retrieving of the translated key. **TRANSLATIONS** also provides support for language dialects which means package authors can distinguish between British, Australian, Canadian and US English, say.

¹ onCTAN:beamer ² onCTAN:babel ³ onCTAN:polyglossia

2 License and Requirements

TRANSLATIONS is placed under the terms of the L^AT_EX Project Public License, version 1.3 or later (<http://www.latex-project.org/lppl.txt>). It has the status “maintained.”

TRANSLATIONS requires the etoolbox⁴ package.

3 Usage

3.1 Available Commands

Below the commands provided by **TRANSLATIONS** are explained. The symbol \triangleright means that the command is expandable, \blacktriangleright means that it isn’t.

- \blacktriangleright **\DeclareLanguage**{<lang>}
Declare a language that can be used by **TRANSLATIONS**. If the language already exists it will be silently redefined. This command can only be used in the preamble.
- \blacktriangleright **\DeclareLanguageAlias**{<lang2>}{<lang1>}
Declares <lang2> to be an alias of <lang1>. If <lang1> doesn’t exist yet a warning will be raised and it will be defined. This command can only be used in the preamble.
- \blacktriangleright **\DeclareLanguageDialect**{<dialect>}{<lang>}
Declares <dialect> to be a dialect of language <lang>. If a translation for <dialect> is provided it is used by the translation macros. If there is none the corresponding translation for <lang> is used instead.
- \blacktriangleright **\NewTranslation**{<lang>}{<key>}{<translation>}
Defines a translation of key <key> for the language <lang>. An error will be raised if a translation of <key> already exists. This command can only be used in the preamble.
- \blacktriangleright **\RenewTranslation**{<lang>}{<key>}{<translation>}
Redefines a translation of key <key> for the language <lang>. An error will be raised if no translation of <key> exists. This command can only be used in the preamble.
- \blacktriangleright **\DeclareTranslation**{<lang>}{<key>}{<translation>}
Defines a translation of key <key> for the language <lang>. No error will be raised if a translation of <key> already exists. This command can only be used in the preamble.
- \blacktriangleright **\DeclareTranslationFallback**{<key>}{<fallback>}
Defines a fallback translation for key <key> that is used in case no translation of <key> for the currently active language has been provided. No error will be raised if a fallback for <key> already exists. This command can only be used in the preamble.

⁴ onCTAN:etoolbox

- ▷ `\GetTranslationFor{<lang>}{<key>}`
Fetches and prints the translation of <key> for the language <lang>. This command is expandable.
- ▷ `\GetTranslation{<key>}`
Fetches and prints the translation of <key> for the currently active language (as for example set by babel). This command is expandable.
- ▶ `\SaveTranslationFor{<cmd>}{<lang>}{<key>}`
Fetches and saves the translation of <key> for the language <lang> in the macro <cmd>.
- ▶ `\SaveTranslation{<cmd>}{<key>}`
Fetches and saves the translation of <key> for the currently active language (as for example set by babel) in the macro <cmd>.
- ▶ `\LoadDictionary{<name>}`
Loads a file named <name>-<lang>.trsl where <lang> corresponds to the lowercase name of the current language as defined with `\DeclareLanguage`. This file should contain the translations for the specified language.
- ▶ `\LoadDictionaryFor{<lang>}{<name>}`
Loads a file named <name>-<lang>.trsl.
- ▶ `\DeclareDictTranslation{<key>}{<translation>}`
This command is to be used in a dictionary file and picks up the language of that file, see section 3.4 for an example.

Quite a number of languages already are defined, either directly or via an alias. So, before you define a language you should take a look at section 4 if the language doesn't already exist.

3.2 A Small Example

This section demonstrates with two short examples how the macros are used. The first example covers the basics: declaring of translations and then retrieving and typesetting them.

3 Usage

```
1 % in the preamble:
2 % \DeclareTranslation{English}{Kueche}{kitchen}
3 % \DeclareTranslation{German}{Kueche}{K\"uche}
4 % \DeclareTranslation{Spanish}{Kueche}{cocina}
5 % \DeclareTranslation{French}{Kueche}{cuisine}
6
7 \GetTranslation{Kueche}
8 \SaveTranslation\kitchen{Kueche}
9 \SaveTranslationFor\cuisine{french}{Kueche}
10
11 \selectlanguage{ngerman}
12 \GetTranslation{Kueche} \kitchen\ \GetTranslationFor{spanish}{Kueche}
13 \cuisine

kitchen
Küche kitchen cocina cuisine
```

The next example demonstrates the use of dialects and how they fall back to the translation for the main language if no extra translation was declared:

```
1 % in the preamble:
2 % \DeclareTranslation{English}{farbe}{color}
3 % \DeclareTranslation{British}{farbe}{colour}
4
5 \GetTranslationFor{English}{farbe} \
6 \GetTranslationFor{British}{farbe} \
7 \GetTranslationFor{American}{farbe}

color
colour
color
```

3.3 Usage in Packages

A typical usage in a package would look as follows:

```
1 \RequirePackage{translations}
2 \DeclareTranslationFallback{mypackage-title}{Nice Title}
3 \DeclareTranslation{English}{mypackage-title}{Nice Title}
4 \DeclareTranslation{French}{mypackage-title}{Beau Titre}
5 \DeclareTranslation{German}{mypackage-title}{Sch\"o}ner
  Titel}
6 ...
7 \def\mypackage@title{\GetTranslation{mypackage-title}}
```

4 Defined Languages

That is, a package defines some unique key for an expression and at least defines a fallback translation. Additionally translations for as many languages as the author wants are defined. A user then may add `\DeclareTranslation{<language>}{<translation>}` if they find their translation missing.

3.4 Dictionaries

A typical dictionary file should look as follows:

```
1 % this is file housing-german.trsl
2 \ProvideDictionaryFor{German}{housing}[<version info>]
3 \DeclareDictTranslation{kitchen (housing)}{K\u"uche}
4 \DeclareDictTranslation{bathroom (housing)}{Bad}
5 \DeclareDictTranslation{living room (housing)}{Wohnzimmer}
6 \DeclareDictTranslation{bedroom (housing)}{Schlafzimmer}
7 ...
8 \endinput
```

The usage is similar to the one in a package: unique keys are given translations, this time for the language the dictionary file is declared for only.

4 Defined Languages

TRANSLATIONS currently has these languages defined, “fallback” being a dummy language used for fallback translations:

fallback, albanian, bulgarian, catalan, croatian, czech, danish,
dutch, english, finnish, french, german, greek, hebrew,
hungarian, icelandic, italian, norwegian, polish, portuges,
romanian, russian, serbocroatian, slovak, slovenian, spanish,
swedish, turkish, ukrainian, canadien, american, australian,
british, canadian, austrian, naustrian, magyar, brazil,
swissgerman

To every one of these languages at least one alias exists, the uppercase variant. This is due to the fact that it is common to write language names uppercased. The defined aliases are these (in parentheses the base language name is given):

Fallback (fallback), Albanian (albanian), Bulgarian (bulgarian), Catalan
(catalan), Croatian (croatian), Czech (czech), Danish (danish), Dutch
(dutch), Finnish (finnish), francais (french), Francais (francais),
Canadien (canadien), French (french), American (american), Australian

5 Implementation

(australian), British (british), Canadian (canadian), English (english), UKenglish (british), USenglish (american), Austrian (austrian), German (german), germanb (german), ngerman (german), Greek (greek), polutonikogreek (greek), Hebrew (hebrew), Hungarian (hungarian), Magyar (magyar), Icelandic (icelandic), Italian (italian), norsk (norwegian), Norsk (norsk), Norwegian (norwegian), nynorsk (norwegian), Nynorsk (nynorsk), Polish (polish), Brazil (brazil), brazilian (brazil), Brazilian (brazilian), Portuges (portuges), portuguese (portuges), Portuguese (portuguese), Romanian (romanian), Russian (russian), Serbocroatian (serbocroatian), Slovak (slovak), Slovenian (slovenian), Spanish (spanish), Swedish (swedish), Swiss (swissgerman), Swissgerman (swissgerman), Turkish (turkish), Ukrainian (ukrainian)

TRANSLATIONS also defines a few dialects. The language to which the dialect belongs to is given in parentheses:

canadien (french), american (english), australian (english), british (english), canadian (english), austrian (german), naustrian (austrian), magyar (hungarian), brazil (portuges), swissgerman (german)

These languages should cover all languages which are currently covered by babel and polyglossia.

5 Implementation

In the following code the lines 1–30 have been omitted. They only repeat the license statement which has already been mentioned in section 2.

```
31 \def\@trnslt@date{2013/07/12}
32 \def\@trnslt@version{v0.10a}
33
34 \ProvidesPackage{translations}[\@trnslt@date\space \@trnslt@version\space a
    simple translator]
35 \RequirePackage{etoolbox}
36
37 % -----
38
38 % message handling
39 \def\@trnslt@error@message{%
40   For details have a look at the ‘translations’ manual.}
41
42 \def\@trnslt@create@message#1{%
43   \ifstrequal{#1}{Error}
44     {%
45       \lowercase{\csdef{\@trnslt@#1}}##1{%
```

5 Implementation

```

46      \csuse{Package#1}{translations}{##1}{\@trnslt@error@message}}%
47    }{%
48      \lowercase{\csdef{@trnslt@#1}}##1{%
49        \csuse{Package#1}{translations}{##1}}%
50    }}
51    \@trnslt@create@message{Error}
52    \@trnslt@create@message{Warning}
53    \@trnslt@create@message{WarningNoLine}
54    \@trnslt@create@message{Info}
55
56    \def\@trnslt@err@unknown@lang#1{%
57      \@trnslt@error{Unknown language ‘#1’}}
58
59    \def\@trnslt@warn@unknown@lang#1{%
60      \@trnslt@warning{Unknown language ‘#1’}}
61
62    \def\@trnslt@err@already@defined#1#2{%
63      \@trnslt@error{The #2 translation for ‘#1’ is already defined.}}
64
65    \def\@trnslt@err@not@defined#1#2{%
66      \@trnslt@error{The \@trnslt@language{#2} translation for ‘#1’ is not
        defined yet.}}
67
68    % -----
69
69    % check if babel or polyglossia is used
70    \AtEndPreamble{
71      \@ifpackageloaded{babel}{{
72        \@ifpackageloaded{polyglossia}{{
73          {\@trnslt@warning{No language package found. I am going to use ‘
74            english’
75            as default language.}}
76        }
77        \ifdef\language{
78          {\def\language{english}}
79        }
80        \def\@trnslt@current@language{\language}
81        \ifdef\bbbl@afterfi{
82          {\long\def\bbbl@afterfi#1\fi{\fi#1}}
83        }
84      }}
85
86    % book keeping: the following macros will be used as ‘etoolbox’ lists that
87    % keep record of defined languages, dialects and aliases
88    \def\@trnslt@languages{}
89    \def\@trnslt@aliases@pair{}
90    \def\@trnslt@aliases@single{}
91    \def\@trnslt@dialects@pair{}
92    \def\@trnslt@dialects@single{}

```

5 Implementation

```

90
91 % -----
92
93 % \DeclareLanguage and \DeclareLanguageAlias
94 % #1: language
95 \newrobustcmd*\DeclareLanguage[1]{%
96   \@trnslt@declare@language{#1}}
97 \onlypreamble\DeclareLanguage
98
99 \def\@trnslt@declare@language#1{%
100   \@trnslt@if@language{#1}
101   {}{%
102     \csdef{\@trnslt@language@#1}{#1}%
103     \listead{\@trnslt@languages{#1}}%
104   }%
105 }
106
107 \def\@trnslt@language#1{%
108   \csuse{\@trnslt@language@#1}}
109
110 \def\@trnslt@if@language#1{%
111   \ifcsundef{\@trnslt@language@#1}
112   {\expandafter\@secondoftwo}
113   {\expandafter\@firstoftwo}%
114 }
115
116 % #1: dialect
117 % #2: language
118 \newrobustcmd*\DeclareLanguageDialect[2]{%
119   \@trnslt@declare@language@dialect{#1}{#2}}
120 \onlypreamble\DeclareLanguageDialect
121
122 \def\@trnslt@declare@language@dialect#1#2{%
123   \@trnslt@if@language{#2}
124   {}{%
125     \@trnslt@warn@unknown@lang{#2}%
126     \@trnslt@declare@language{#2}%
127   }%
128   \@trnslt@if@dialect{#1}
129   {% => ist schon als dialect definiert => irgendwelche weiteren checks?
130   }
131   {%
132     \@trnslt@if@alias{#2}
133     {%
134       \csedef{\@trnslt@dialect@#1}{\@trnslt@alias{#2}}{#1}}%
135       \@trnslt@declare@language{#1}%
136       \listead{\@trnslt@dialects@single{#1}}%

```


5 Implementation

```

136         \listeadadd\@trnslt@dialects@pair{{#1}{\@trnslt@alias{#2}}}%
137     }
138     {%
139         \csdef{@trnslt@dialect@#1}{{#2}{#1}}%
140         \@trnslt@declare@language{#1}%
141         \listeadadd\@trnslt@dialects@single{#1}%
142         \listeadadd\@trnslt@dialects@pair{{#1}{#2}}%
143     }%
144 }%
145 }
146
147 \def\@trnslt@dialect#1{%
148     \csuse{@trnslt@dialect@#1}}
149
150 \def\@trnslt@dialect@of#1{%
151     \expandafter\expandafter\expandafter
152     \@trnslt@dialect@of@aux
153     \csname @trnslt@dialect@#1\endcsname\@empty
154 }
155 \def\@trnslt@dialect@of@aux#1#2{\ifx\relax#1\@empty\else#1\fi}
156
157 \def\@trnslt@if@dialect#1{%
158     \ifcsundef{@trnslt@dialect@#1}
159     {\expandafter\@secondoftwo}
160     {\expandafter\@firstoftwo}%
161 }
162
163 % #1: alias
164 % #2: language
165 \newrobustcmd*{\DeclareLanguageAlias[2]}{%
166     \@trnslt@declare@languagealias{#1}{#2}}
167 \@onlypreamble\DeclareLanguageAlias
168
169 \def\@trnslt@declare@languagealias#1#2{%
170     \@trnslt@if@language{#2}
171     {}{%
172         \@trnslt@warn@unknown@lang{#2}%
173         \@trnslt@declare@language{#2}%
174     }%
175     \csletcs{@trnslt@language@#1}{@trnslt@language@#2}%
176     \@trnslt@if@dialect{#2}
177     {\csletcs{@trnslt@dialect@#1}{@trnslt@dialect@#2}}
178     {}%
179     \ifinlist{#1}\@trnslt@aliases@single
180     {}{%
181         \csdef{@trnslt@alias@#1}{#2}%
182         \listeadadd\@trnslt@aliases@pair{{#1}{#2}}%

```

5 Implementation

```

183     \listadd\@trnslt@aliases@single{#1}%
184   }%
185 }
186
187 \def\@trnslt@alias#1{%
188   \csuse{\@trnslt@alias@#1}}
189
190 \def\@trnslt@if@alias#1{%
191   \ifcsundef{\@trnslt@alias@#1}
192     {\expandafter\@secondoftwo}
193     {\expandafter\@firstoftwo}%
194 }
195
196 % dummy language: 'fallback':
197 \DeclareLanguage{fallback}
198 \DeclareLanguageAlias{Fallback}{fallback}
199
200 % -----
201
202 % \DeclareTranslation, \NewTranslation and \RenewTranslation
203 % #1: language
204 % #2: word
205 % #3: replacement
206 \newrobustcmd*\DeclareTranslation[3]{%
207   \@trnslt@declare@translation{#2}{#1}{#3}}
208 \onlypreamble\DeclareTranslation
209
210 \newrobustcmd*\DeclareTranslationFallback[2]{%
211   \@trnslt@declare@translation{#1}{fallback}{#2}}
212 \onlypreamble\DeclareTranslationFallback
213
214 \newrobustcmd*\NewTranslation[3]{%
215   \@trnslt@new@translation{#2}{#1}{#3}}
216 \onlypreamble\NewTranslation
217
218 \newrobustcmd*\RenewTranslation[3]{%
219   \@trnslt@renew@translation{#2}{#1}{#3}}
220 \onlypreamble\RenewTranslation
221
222 % #1: word
223 % #2: language
224 % #3: replacement
225 \def\@trnslt@declare@translation#1#2#3{%
226   \@trnslt@if@language{#2}
227   {%
228     % save the <word> as <word>:
229     \csdef{\@trnslt@word@#1@literal}{#1}%

```

5 Implementation

```

229 % check if the language is a dialect:
230 \@trnslt@if@dialect{#2}
231     {\csdef{@trnslt@word@#1@\@trnslt@dialect{#2}}{#3}}
232     {}%
233 % check if translation already exists:
234 \@trnslt@if@translation{#1}{#2}
235     {}
236     {\csdef{@trnslt@word@#1@\@trnslt@language{#2}}{#3}}%
237 }
238 {\@trnslt@err@unknown@lang{#2}}%
239 }
240
241 \def\@trnslt@if@translation#1#2{%
242     \ifcsundef{@trnslt@word@#1@\@trnslt@language{#2}}
243     {%
244         \@trnslt@if@dialect{#2}
245         {%
246             \ifboolexpe{
247                 test {\ifcsundef{@trnslt@word@#1@\@trnslt@dialect{#2}}} or
248                 test {\ifcsundef{@trnslt@word@#1@\@trnslt@dialect@of{#2}}}
249             }
250             {\expandafter\@firstoftwo}
251             {\expandafter\@secondoftwo}%
252         }
253         {\expandafter\@secondoftwo}%
254     }
255     {\expandafter\@firstoftwo}%
256 }
257
258 \def\@trnslt@new@translation#1#2#3{%
259     \@trnslt@if@translation{#1}{#2}
260     {\@trnslt@err@already@defined{#1}{#2}}
261     {\@trnslt@declare@translation{#1}{#2}{#3}}}
262
263 \def\@trnslt@renew@translation#1#2#3{%
264     \@trnslt@if@translation{#1}{#2}
265     {\@trnslt@declare@translation{#1}{#2}{#3}}
266     {\@trnslt@err@not@defined{#1}{#2}}}
267
268 % -----
269
269 % \GetTranslationFor and \GetTranslation
270 % these need to be expandable!
271 % #1: language
272 % #2: word
273 \newcommand*\GetTranslationFor[2]{%
274     \@trnslt@checkandget@translation@for{#2}{#1}}

```

5 Implementation

```

275
276 \newcommand*\GetTranslation[1]{%
277   \@trnslt@checkandget@translation@for{#1}{\@trnslt@current@language}}
278
279 % #1: word #2: language
280 \def\@trnslt@get@translation@for#1#2{%
281   \@trnslt@if@dialect{#2}
282   {%
283     \ifcsdef{@trnslt@word@#1@\@trnslt@dialect{#2}}
284     {\csuse{@trnslt@word@#1@\@trnslt@dialect{#2}}}
285     {\csuse{@trnslt@word@#1@\@trnslt@dialect@of{#2}}}%
286   }
287   {\csuse{@trnslt@word@#1@\@trnslt@language{#2}}}%
288 }
289
290 \def\@trnslt@checkandget@translation@for#1#2{%
291   \@trnslt@if@translation{#1}{#2}
292   {\@trnslt@get@translation@for{#1}{#2}}
293   {%
294     \@trnslt@if@translation{#1}{fallback}
295     {\csuse{@trnslt@word@#1@fallback}}
296     {\csuse{@trnslt@word@#1@literal}}}%
297   }%
298 }
299
300 % this is not expandable!
301 \protected\def\@trnslt@getandwarn@translation@for#1#2{%
302   \@trnslt@if@translation{#1}{#2}
303   {\@trnslt@get@translation@for{#1}{#2}}
304   {%
305     \@trnslt@warning{Translation for ‘#1’ in #2 unknown. You may try to
306     use
307       \string\DeclareTranslation{#2}{#1}{ ... } in your preamble.}%
308     \@trnslt@if@translation{#1}{fallback}
309     {%
310       \@trnslt@info{Using fallback translation for ‘#1’}%
311       \csuse{@trnslt@word@#1@fallback}
312     }
313     {\csuse{@trnslt@word@#1@literal}}}%
314   }%
315 }
316 % -----
317
318 % \SaveTranslationFor and \SaveTranslation
319 \newrobustcmd*\SaveTranslationFor[3]{%
320   \@trnslt@save@translation@for{#1}{#3}{#2}}

```

5 Implementation

```

320
321 \newrobustcmd*\SaveTranslation[2]{%
322   \@trnslt@save@translation@for{#1}{#2}{\@trnslt@current@language}}
323
324 \def\@trnslt@save@translation@for#1#2#3{%
325   \edef#1{%
326     \@trnslt@if@translation{#2}{#3}
327     {\csuse{\@trnslt@word@#2\@trnslt@language{#3}}}
328     }%
329   }}
330
331 % -----
332
332 % \LoadDictionary and \LoadDictionaryFor
333 \newrobustcmd*\LoadDictionary[1]{%
334   \@trnslt@load@dictionary@for{#1}{\@trnslt@current@language}}
335 \@onlypreamble\LoadDictionary
336
337 \newrobustcmd*\LoadDictionaryFor[2]{%
338   \@trnslt@load@dictionary@for{#2}{#1}}
339 \@onlypreamble\LoadDictionaryFor
340
341 % #1: name
342 % #2: lang
343 \def\@trnslt@load@dictionary@for#1#2{%
344   \AtBeginDocument{%
345     \InputIfFileExists{#1-\@trnslt@language{#2}.trsl}
346     {\@trnslt@info{loading dictionary '#1' for '#2'.}}
347     {\@trnslt@warning{File '#1-\@trnslt@language{#2}.trsl' not found.}}%
348   }}
349
350 \newrobustcmd*\ProvideDictionaryFor[2]{%
351   \@trnslt@provide@dictionary@for{#1}{#2}}
352 \@onlypreamble\ProvideDictionaryFor
353
354 \def\@trnslt@provide@dictionary@for#1#2{%
355   \def\@trnslt@dictionary@name{#2}%
356   \def\@trnslt@dictionary@lang{#1}%
357   \@ifnextchar[
358     {\@trnslt@provide@dictionary@version}
359     {\ProvidesFile{#2-#1.trsl}[{#1 translation file '#2'}]}}
360
361 \def\@trnslt@provide@dictionary@version[#1]{%
362   \ProvidesFile
363     {\@trnslt@dictionary@name-\@trnslt@dictionary@lang.trsl}%
364     [(\@trnslt@dictionary@lang\space translation file '\@
   \@trnslt@dictionary@name') #1]}

```

5 Implementation

```
365 % \@trnslt@dictionary@language
366 \newrobustcmd*\DeclareDictTranslation[2]{%
367   \@trnslt@declare@translation{#1}{\@trnslt@dictionary@lang}{#2}}
368 \onlypreamble\DeclareDictTranslation
369
370 % -----
371
372 % predefined languages
373 \DeclareLanguage{albanian}
374 \DeclareLanguage{bulgarian}
375 \DeclareLanguage{catalan}
376 \DeclareLanguage{croatian}
377 \DeclareLanguage{czech}
378 \DeclareLanguage{danish}
379 \DeclareLanguage{dutch}
380 \DeclareLanguage{english}
381 \DeclareLanguage{finnish}
382 \DeclareLanguage{french}
383 \DeclareLanguage{german}
384 \DeclareLanguage{greek}
385 \DeclareLanguage{hebrew}
386 \DeclareLanguage{hungarian}
387 \DeclareLanguage{icelandic}
388 \DeclareLanguage{italian}
389 \DeclareLanguage{norwegian}
390 \DeclareLanguage{polish}
391 \DeclareLanguage{portuges}
392 \DeclareLanguage{romanian}
393 \DeclareLanguage{russian}
394 \DeclareLanguage{serbocroatian}
395 \DeclareLanguage{slovak}
396 \DeclareLanguage{slovenian}
397 \DeclareLanguage{spanish}
398 \DeclareLanguage{swedish}
399 \DeclareLanguage{turkish}
400 \DeclareLanguage{ukrainian}
401
402 \DeclareLanguageAlias {Albanian}{albanian}
403 \DeclareLanguageAlias {Bulgarian}{bulgarian}
404 \DeclareLanguageAlias {Catalan}{catalan}
405 \DeclareLanguageAlias {Croatian}{croatian}
406 \DeclareLanguageAlias {Czech}{czech}
407 \DeclareLanguageAlias {Danish}{danish}
408 \DeclareLanguageAlias {Dutch}{dutch}
409 \DeclareLanguageAlias {Finnish}{finnish}
410 \DeclareLanguageAlias {français}{french}
```

5 Implementation

```
411 \DeclareLanguageAlias {Francais}{français}
412 \DeclareLanguageDialect{canadien}{french}
413 \DeclareLanguageAlias {Canadien}{canadien}
414 \DeclareLanguageAlias {French}{french}
415 \DeclareLanguageDialect{american}{english}
416 \DeclareLanguageAlias {American}{american}
417 \DeclareLanguageDialect{australian}{english}
418 \DeclareLanguageAlias {Australian}{australian}
419 \DeclareLanguageDialect{british}{english}
420 \DeclareLanguageAlias {British}{british}
421 \DeclareLanguageDialect{canadian}{english}
422 \DeclareLanguageAlias {Canadian}{canadian}
423 \DeclareLanguageAlias {English}{english}
424 \DeclareLanguageAlias {UKenglish}{british}
425 \DeclareLanguageAlias {USenglish}{american}
426 \DeclareLanguageDialect{austrian}{german}
427 \DeclareLanguageAlias {Austrian}{austrian}
428 \DeclareLanguageAlias {German}{german}
429 \DeclareLanguageAlias {germanb}{german}
430 \DeclareLanguageDialect{naustrian}{austrian}
431 \DeclareLanguageAlias {ngerman}{german}
432 \DeclareLanguageAlias {Greek}{greek}
433 \DeclareLanguageAlias {polutonikogreek}{greek}
434 \DeclareLanguageAlias {Hebrew}{hebrew}
435 \DeclareLanguageAlias {Hungarian}{hungarian}
436 \DeclareLanguageDialect{magyar}{hungarian}
437 \DeclareLanguageAlias {Magyar}{magyar}
438 \DeclareLanguageAlias {Icelandic}{icelandic}
439 \DeclareLanguageAlias {Italian}{italian}
440 \DeclareLanguageAlias {norsk}{norwegian}
441 \DeclareLanguageAlias {Norsk}{norsk}
442 \DeclareLanguageAlias {Norwegian}{norwegian}
443 \DeclareLanguageAlias {nynorsk}{norwegian}
444 \DeclareLanguageAlias {Nynorsk}{nynorsk}
445 \DeclareLanguageAlias {Polish}{polish}
446 \DeclareLanguageDialect{brazil}{portuges}
447 \DeclareLanguageAlias {Brazil}{brazil}
448 \DeclareLanguageAlias {brazilian}{brazil}
449 \DeclareLanguageAlias {Brazilian}{brazilian}
450 \DeclareLanguageAlias {Portuges}{portuges}
451 \DeclareLanguageAlias {portuguese}{portuges}
452 \DeclareLanguageAlias {Portuguese}{portuguese}
453 \DeclareLanguageAlias {Romanian}{romanian}
454 \DeclareLanguageAlias {Russian}{russian}
455 \DeclareLanguageAlias {Serbocroatian}{serbocroatian}
456 \DeclareLanguageAlias {Slovak}{slovak}
457 \DeclareLanguageAlias {Slovenian}{slovenian}
```

5 Implementation

```

458 \DeclareLanguageAlias {Spanish}{spanish}
459 \DeclareLanguageAlias {Swedish}{swedish}
460 \DeclareLanguageDialect{swissgerman}{german}
461 % this maybe should be a language of it's own:
462 \DeclareLanguageAlias {Swiss}{swissgerman}
463 \DeclareLanguageAlias {Swissgerman}{swissgerman}
464 \DeclareLanguageAlias {Turkish}{turkish}
465 \DeclareLanguageAlias {Ukrainian}{ukrainian}
466
467 \endinput
468
469 % HISTORY:
470 2012/09/30 v0.2beta - first version (as part of the 'exsheets' bundle)
471 2012/10/05 v0.2    - \LoadDictionary and \LoadDictionaryFor added and
    loads of
472                      languages defined.
473 2013/03/10 v0.8    - basic dictionaries for English, German, French and
    Spanish
474                      - new command \DeclareDictTranslation
475 2013/04/04 v0.8a   - bug fix in \DeclareDictTranslation
476 2013/04/07 v0.9    - slightly improved messages
477 2013/04/08 v0.9a   - changed fallback warning into info
478                      - synchronized version number with 'exsheets' until now
    but
479                      won't any more
480 2013/06/22 v0.9b   - added Swiss
481 2013/06/28 v0.10   - declaring aliases of dialects now works as expected
482                      - declarings dialects of an alias now correctly
    declares
483                      the dialect to the correct base language
484                      - corrected a few erroneous language declarations
485 2013/07/12 v0.10a  - \GetTranslation gets two-folded fallback: use
486                      fallback-translation if no translation for the
    current
487                      language has been defined; use literal string if /no/
488                      language is used - this should never happen but /will
    /
489                      happen if neither 'babel' nor 'polyglossia' have been
490                      loaded, i.e., no language has been chosen /and/ the
491                      package writer did not provide an English translation

```


Index

Section titles are indicated **bold**, packages sans serif, commands \brown and options yellow.

B	G	P
babel 1, 3	\GetTranslation 3	polyglossia 1, 6
beamer 1	\GetTranslationFor 3	\providecaptionname 1
C	I	R
\captions<language> 1	Implementation 6	\RenewTranslation 2
D	L	Requirements 2
\DeclareDictTranslation 3	Languages 5f.	S
\DeclareLanguage 2f.	\LoadDictionary 3	\SaveTranslation 3
\DeclareLanguageAlias .. 2	\LoadDictionaryFor 3	\SaveTranslationFor 3
\DeclareLanguageDialect 2	M	T
\DeclareTranslation...2, 5	Motivation 1	translator 1
\DeclareTranslationFallback 2	N	U
E	\NewTranslation 2	Usage 2–5
etoolbox 2		