translations

v1.0   2013/07/15

a simple translator

Clemens Niederberger

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

English documentation

Contents

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 beamer[1] bundle). Internationalization is also possible with babel[2] 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 polyglossia[3]) language itself and provides expandable retrieving of the translated key. TRANSLATIONS also provides support for language dialects which means package authors can for example distinguish between British, Australian, Canadian and US English.

The first draft of the package was written since I missed an expandable version of translator's `\translate` command. Once I had the package available I began using it in various of my other packages so it got extended to the needs I faced there.

---

[1] on CTAN: beamer   [2] on CTAN: babel   [3] on CTAN: polyglossia

2 License and Requirements

TRANSLATIONS is placed under the terms of the LATEX Project Public License, version 1.3 or later (`http://www.latex-project.org/lppl.txt`). It has the status "maintained."

TRANSLATIONS requires the packages etoolbox[4] and scrlfile (part of the KOMA-Script bundle[5]).

3 Usage

3.1 Available Commands

Below the commands provided by TRANSLATIONS are explained. The symbol ▷ means that the command is expandable, ► means that it isn't.

► `\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.

► `\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.

► `\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.

► `\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.

► `\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.

► `\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.

► `\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.

▷ `\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.

---

[4] on CTAN: etoolbox   [5] on CTAN: koma-script

► \GetTranslationForWarn{<lang>}{<key>}

Fetches and prints the translation of <key> for the language <lang>. Issues a warning if no translation is available.

► \GetTranslationWarn{<key>}

Fetches and prints the translation of <key> for the currently active language (as for example set by babel). Issues a warning if no translation is available.

► \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.

► \ProvideDictionaryFor{<lang>}{<name>}[<date>]

Needs to be in a dictionary file. This command tells TRANSLATIONS that the file indeed is a dictionary and also sets the language for the dictionary which is used by \DeclareDictTranslation.

► \PrintDictionaryFor{<lang>}{<name>}{<pre>}{<mid>}{<post>}

Prints all entries of dictionary <name> in language <lang> in the order the entries have been declared. For every entry the code

<pre><key><mid><translation><post>

is printed. The dictionary must have been loaded of course.

3.2 A Small Example

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

```
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

3.3.1 Basic Structure

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}}
```

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.

3.3.2 The 'fallback' language

If a user has neither loaded babel nor polyglossia TRANSLATIONS will use English as language and translate to English if the translation was provided. If the user *has* loaded one of the language packages but has chosen a language for which no translation is defined the language 'fallback' will be used, i.e., the translation provided with \DeclareTranslationFallback. If no fallback translation is provided either the translation will expand to the literal string.

The following three examples should make this concept clear:

```
1  \documentclass{article}
2  \DeclareTranslation{German}{foo-literal}{bar}
3  \begin{document}
4  \GetTranslation{foo-literal} => `foo-literal'
5  \end{document}
```

```
1  \documentclass{article}
2  \DeclareTranslationFallback{foo-literal}{foo}
3  \DeclareTranslation{German}{foo-literal}{bar}
4  \begin{document}
5  \GetTranslation{foo-literal} => `foo'
6  \end{document}
```

```
1  \documentclass{article}
2  \usepackage[ngerman]{babel}
3  \DeclareTranslation{German}{foo-literal}{bar}
4  \begin{document}
5  \GetTranslation{foo-literal} => `bar'
6  \end{document}
```

3.4 Dictionaries

3.4.1 Own 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\"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.

3.4.2 TRANSLATIONS' Basic Dictionaries

TRANSLATIONS already provides a basic dictionary for the languages English, French, German and Spanish. This dictionary is loaded automatically if the document language is one of these four. If you'd like to contribute and add the basic dictionary in your language this is more than welcome and highly appreciated! The easiest way to do this would be to copy one of the existing files `translations-basic-dictionary-<lang>.trsl` and modify the file accordingly. You can then send me the file via email and I'll add it to TRANSLATIONS.

Table 1 lists all words provided by the basic dictionary for German.

TABLE 1: All entries of TRANSLATIONS' basic dictionary in German.

key	translation
Abstract	Zusammenfassung
Addresses	Adressen
addresses	Adressen
Address	Adresse
address	Adresse
and	und
Appendix	Anhang
Authors	Autoren
authors	Autoren
Author	Autor
author	Autor
Bibliography	Literaturverzeichnis
cc	Verteiler
Chapters	Kapitel
chapters	Kapitel
Chapter	Kapitel
chapter	Kapitel
Conclusion	Zusammenfassung
conclusion	Zusammenfassung
Contents	Inhaltsverzeichnis
Continuation	Fortsetzung
continuation	Fortsetzung
cont	Forts
encl (plural)	Anlagen
encl (singular)	Anlage
encl	Anlage(n)
Figures	Abbildungen
figures	Abbildungen
Figure	Abbildung

continues

key	translation
figure	Abbildung
From	Von
from	von
Glossary	Glossar
Index	Index
Introduction	Einleitung
introduction	Einleitung
List of Figures and Tables	Abbildungs- und Tabellenverzeichnis
List of Figures	Abbildungsverzeichnis
List of Tables	Tabellenverzeichnis
or	oder
Outline	Gliederung
Overview	Übersicht
Pages	Seiten
pages	Seiten
Page	Seite
page	Seite
Paragraphs	Absätze
paragraphs	Absätze
Paragraph	Absatz
paragraph	Absatz
Parts	Teile
parts	Teile
Part	Teil
part	Teil
Preface	Vorwort
Proofs	Beweise
proofs	Beweise
Proof	Beweis
proof	Beweis
References	Literatur
Related work	Verwandte Arbeiten
Related Work	Verwandte Arbeiten
Sections	Abschnitte
sections	Abschnitte
Section	Abschnitt
section	Abschnitt
See also	Siehe auch
see also	siehe auch
See	Siehe
see	siehe
Sketch of Proofs	Beweisskizzen
Sketch of proofs	Beweisskizzen

continues

key	translation
Sketch of Proof	Beweisskizze
Sketch of proof	Beweisskizze
Subsections	Unterabschnitte
subsections	Unterabschnitte
Subsection	Unterabschnitt
subsection	Unterabschnitt
Summary	Zusammenfassung
Tables	Tabellen
tables	Tabellen
Table	Tabelle
table	Tabelle
To	An
to	an

4 Defined Languages

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 the tables below if the language doesn't already exist. TRANSLATIONS currently has these languages defined, "fallback" being a dummy language used for fallback translations. Table 2 lists all base languages.

TABLE 2: Languages defines by 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. All defined aliases are listed in table 3.

TABLE 3: All language aliases defined by TRANSLATIONS.

alias	language	alias	language
Fallback	fallback	Albanian	albanian

continues

alias	language	alias	language
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	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. They are listed in table 4.

TABLE 4: All dialects defined by TRANSLATIONS.

dialect	language	dialect	language
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{2013/07/15}
32  \def\@trnslt@version{v1.0}
33
34  \ProvidesPackage{translations}[\@trnslt@date\space \@trnslt@version\space a simple
      translator]
35  \RequirePackage{etoolbox,scrlfile}
36
37  % -------------------------------------------------------------------------
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{%
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  % 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 `english'
74          as default language.}}
75    }
76    \ifdef\languagename{}
```

```
77       {\def\languagename{english}}
78     \def\@trnslt@current@language{\languagename}
79     \ifdef\bbl@afterfi{}
80       {\long\def\bbl@afterfi#1\fi{\fi#1}}
81   }
82
83   % -------------------------------------------------------------------------
84   % book keeping: the following macros will be used as `etoolbox' lists that
85   % keep record of defined languages, dialects and aliases
86   \def\@trnslt@languages{}% all languages
87   \def\@trnslt@aliases@pair{}% all aliases and their base
88   \def\@trnslt@aliases@single{}% all aliases
89   \def\@trnslt@dialects@pair{}% all dialects and their base
90   \def\@trnslt@dialects@single{}% all dialects
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        \listadd\@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@languagedialect{#1}{#2}}
120  \@onlypreamble\DeclareLanguageDialect
121
122  \def\@trnslt@declare@languagedialect#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              \listeadd\@trnslt@dialects@single{#1}%
137              \listeadd\@trnslt@dialects@pair{{#1}{\@trnslt@alias{#2}}}%
138            }
139            {%
140              \csdef{@trnslt@dialect@#1}{{#2}{#1}}%
141              \@trnslt@declare@language{#1}%
142              \listeadd\@trnslt@dialects@single{#1}%
143              \listeadd\@trnslt@dialects@pair{{#1}{#2}}%
144            }%
145      }%
146 }
147
148 \def\@trnslt@dialect#1{%
149   \csuse{@trnslt@dialect@#1}}
150
151 \def\@trnslt@dialect@of#1{%
152   \expandafter\expandafter\expandafter
153     \@trnslt@dialect@of@aux
154     \csname @trnslt@dialect@#1\endcsname\@empty
155 }
156 \def\@trnslt@dialect@of@aux#1#2{\ifx\relax#1\@empty\else#1\fi}
157
158 \def\@trnslt@if@dialect#1{%
159   \ifcsundef{@trnslt@dialect@#1}
160     {\expandafter\@secondoftwo}
161     {\expandafter\@firstoftwo}%
162 }
163
164 % #1: alias
165 % #2: language
166 \newrobustcmd*\DeclareLanguageAlias[2]{%
167   \@trnslt@declare@languagealias{#1}{#2}}
168 \@onlypreamble\DeclareLanguageAlias
169
170 \def\@trnslt@declare@languagealias#1#2{%
171   \@trnslt@if@language{#2}
172     {}{%
173       \@trnslt@warn@unknown@lang{#2}%
174       \@trnslt@declare@language{#2}%
175     }%
176   \csletcs{@trnslt@language@#1}{@trnslt@language@#2}%
177   \@trnslt@if@dialect{#2}
178     {\csletcs{@trnslt@dialect@#1}{@trnslt@dialect@#2}}
179     {}%
180   \ifinlist{#1}\@trnslt@aliases@single
```

```
181        {}{%
182          \csdef{@trnslt@alias@#1}{#2}%
183          \listeadd\@trnslt@aliases@pair{{#1}{#2}}%
184          \listeadd\@trnslt@aliases@single{#1}%
185        }%
186    }
187
188    \def\@trnslt@alias#1{%
189      \csuse{@trnslt@alias@#1}}
190
191    \def\@trnslt@if@alias#1{%
192      \ifcsundef{@trnslt@alias@#1}
193        {\expandafter\@secondoftwo}
194        {\expandafter\@firstoftwo}%
195    }
196
197    % dummy language: `fallback':
198    \DeclareLanguage{fallback}
199    \DeclareLanguageAlias{Fallback}{fallback}
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}%
230          % check if the language is a dialect:
231          \@trnslt@if@dialect{#2}
232            {\csdef{@trnslt@word@#1@\@trnslt@dialect{#2}}{#3}}
```

```
233        {}%
234      % check if translation already exists:
235        \@trnslt@if@translation{#1}{#2}
236          {}
237          {\csdef{@trnslt@word@#1@\@trnslt@language{#2}}{#3}}%
238      }
239      {\@trnslt@err@unknown@lang{#2}}%
240  }
241
242  \def\@trnslt@if@translation#1#2{%
243    \ifcsundef{@trnslt@word@#1@\@trnslt@language{#2}}
244      {%
245        \@trnslt@if@dialect{#2}
246          {%
247            \ifboolexpe{
248              test {\ifcsundef{@trnslt@word@#1@\@trnslt@dialect{#2}}} or
249              test {\ifcsundef{@trnslt@word@#1@\@trnslt@dialect@of{#2}}}
250            }
251            {\expandafter\@firstoftwo}
252            {\expandafter\@secondoftwo}%
253          }
254          {\expandafter\@secondoftwo}%
255      }
256      {\expandafter\@firstoftwo}%
257  }
258
259  \def\@trnslt@new@translation#1#2#3{%
260    \@trnslt@if@translation{#1}{#2}
261      {\@trnslt@err@already@defined{#1}{#2}}
262      {\@trnslt@declare@translation{#1}{#2}{#3}}}
263
264  \def\@trnslt@renew@translation#1#2#3{%
265    \@trnslt@if@translation{#1}{#2}
266      {\@trnslt@declare@translation{#1}{#2}{#3}}
267      {\@trnslt@err@not@defined{#1}{#2}}}
268
269  % -------------------------------------------------------------------------
270  % \GetTranslationFor and \GetTranslation
271  % these need to be expandable!
272  % #1: language
273  % #2: word
274  \newcommand*\GetTranslationFor[2]{%
275    \@trnslt@checkandget@translation@for{#2}{#1}}
276
277  \newcommand*\GetTranslation[1]{%
278    \@trnslt@checkandget@translation@for{#1}{\@trnslt@current@language}}
279
280  % unexpandable version of the commands that raise a warning if no translation
281  % is available:
282  \newcommand*\GetTranslationForWarn[2]{%
283    \@trnslt@getandwarn@translation@for{#2}{#1}}
284
```

```
285  \newcommand*\GetTranslationWarn[1]{%
286    \@trnslt@getandwarn@translation@for{#1}{\@trnslt@current@language}}
287
288  % #1: word #2: language
289  \def\@trnslt@get@translation@for#1#2{%
290    \@trnslt@if@dialect{#2}
291      {%
292        \ifcsdef{@trnslt@word@#1@\@trnslt@dialect{#2}}
293          {\csuse{@trnslt@word@#1@\@trnslt@dialect{#2}}}
294          {\csuse{@trnslt@word@#1@\@trnslt@dialect@of{#2}}}%
295      }
296      {\csuse{@trnslt@word@#1@\@trnslt@language{#2}}}%
297  }
298
299  \def\@trnslt@checkandget@translation@for#1#2{%
300    \@trnslt@if@translation{#1}{#2}
301      {\@trnslt@get@translation@for{#1}{#2}}
302      {%
303        \@trnslt@if@translation{#1}{fallback}
304          {\csuse{@trnslt@word@#1@fallback}}
305          {\csuse{@trnslt@word@#1@literal}}%
306      }%
307  }
308
309  % this is not expandable!
310  \protected\def\@trnslt@getandwarn@translation@for#1#2{%
311    \@trnslt@if@translation{#1}{#2}
312      {\@trnslt@get@translation@for{#1}{#2}}
313      {%
314        \@trnslt@warning{Translation for `#1' in #2 unknown. You may try to use
315          \string\DeclareTranslation{#2}{#1}{ ... } in your preamble.}%
316        \@trnslt@if@translation{#1}{fallback}
317          {%
318            \@trnslt@info{Using fallback translation for `#1'}%
319            \csuse{@trnslt@word@#1@fallback}
320          }
321          {\csuse{@trnslt@word@#1@literal}}%
322      }%
323  }
324
325  % -------------------------------------------------------------------------
326  % \SaveTranslationFor and \SaveTranslation
327  \newrobustcmd*\SaveTranslationFor[3]{%
328    \@trnslt@save@translation@for{#1}{#3}{#2}}
329
330  \newrobustcmd*\SaveTranslation[2]{%
331    \@trnslt@save@translation@for{#1}{#2}{\@trnslt@current@language}}
332
333  \def\@trnslt@save@translation@for#1#2#3{%
334    \edef#1{%
335      \@trnslt@if@translation{#2}{#3}
336        {\csuse{@trnslt@word@#2@\@trnslt@language{#3}}}
```

```
337        {}%
338    }}
339
340  % -------------------------------------------------------------------------
341  % \LoadDictionary and \LoadDictionaryFor
342  \newrobustcmd*\LoadDictionary[1]{%
343    \@trnslt@load@dictionary@for{#1}{\@trnslt@current@language}}
344  \@onlypreamble\LoadDictionary
345
346  \newrobustcmd*\LoadDictionaryFor[2]{%
347    \@trnslt@load@dictionary@for{#2}{#1}}
348  \@onlypreamble\LoadDictionaryFor
349
350  % #1: name
351  % #2: lang
352  \def\@trnslt@load@dictionary@for#1#2{%
353    \AtBeginDocument{%
354      \InputIfFileExists{#1-\@trnslt@language{#2}.trsl}
355        {\@trnslt@check@dictionary{#1}{#2}}
356        {\@trnslt@warning{dictionary file `#1-\@trnslt@language{#2}.trsl' not
357            found.}}%
358    }}
359
360  \def\@trnslt@check@dictionary#1#2{%
361    \AfterFile{#1-\@trnslt@language{#2}.trsl}
362      {%
363        \ifcsdef{@trnslt@dictionary@#1@\@trnslt@language{#2}}
364          {\@trnslt@info{loading dictionary `#1' for `#2'.}}
365          {%
366            \@trnslt@warning{file `#1-\@trnslt@language{#2}.trsl' does not
367              appear to be a dictionary}%
368          }%
369      }%
370  }
371
372  \def\@trnslt@load@dictionary@silent@for#1#2{%
373    \AtBeginDocument{\InputIfFileExists{#1-\@trnslt@language{#2}.trsl}{}{}}}
374
375  \newrobustcmd*\ProvideDictionaryFor[2]{%
376    \@trnslt@provide@dictionary@for{#1}{#2}}
377  \@onlypreamble\ProvideDictionaryFor
378
379  \def\@trnslt@provide@dictionary@for#1#2{%
380    \def\@trnslt@dictionary@name{#2}%
381    \edef\@trnslt@dictionary@lang{\@trnslt@language{#1}}%
382    % this macro can be used to check if we have a dictionary and will also be
383    % used as a list for the dictionary entries:
384    \csdef{@trnslt@dictionary@\@trnslt@dictionary@name @\@trnslt@dictionary@lang}{}%
385    \@ifnextchar[
386      {\@trnslt@provide@dictionary@version}
387      {
388        \ProvidesFile
```

16

```
389        {#2-\@trnslt@dictionary@lang.trsl}%
390        [(\@trnslt@dictionary@lang\space translation file `#2')]
391      }%
392  }
393
394  \def\@trnslt@provide@dictionary@version[#1]{%
395    \ProvidesFile
396      {\@trnslt@dictionary@name-\@trnslt@dictionary@lang.trsl}%
397      [(\@trnslt@dictionary@lang\space translation file `\@trnslt@dictionary@name') #1]}
398
399  % \@trnslt@dictionary@language
400  \newrobustcmd*\DeclareDictTranslation[2]{%
401    \listcsadd
402      {@trnslt@dictionary@\@trnslt@dictionary@name @\@trnslt@dictionary@lang}
403      {{#1}{#2}}%
404    \@trnslt@declare@translation{#1}{\@trnslt@dictionary@lang}{#2}%
405  }
406  \@onlypreamble\DeclareDictTranslation
407
408  % \PrintDictionaryFor
409  % #1: lang
410  % #2: name
411  % #3: pre
412  % #4: mid
413  % #5: post
414  \newcommand*\PrintDictionaryFor[5]{%
415    \@trnslt@print@dictionary@for{#1}{#2}{#3}{#4}{#5}}
416
417  % #1: lang
418  % #2: name
419  % #3: pre
420  % #4: mid
421  % #5: post
422  \def\@trnslt@print@dictionary@for#1#2#3#4#5{%
423    \forlistcsloop
424      {\@trnslt@print@dictionary@entry{#3}{#4}{#5}}
425      {@trnslt@dictionary@#2@\@trnslt@language{#1}}%
426  }
427
428  % #1: pre
429  % #2: mid
430  % #3: post
431  % #4: {key}{translation}
432  \def\@trnslt@print@dictionary@entry#1#2#3#4{%
433    \@trnslt@print@dictionary@entry@aux{#1}{#2}{#3}#4}
434
435  % #1: pre
436  % #2: mid
437  % #3: post
438  % #4: key
439  % #5: translation
440  \def\@trnslt@print@dictionary@entry@aux#1#2#3#4#5{#1#4#2#5#3}
```

```
441
442   % --------------------------------------------------------------------------
443   % predefined languages
444   \DeclareLanguage{albanian}
445   \DeclareLanguage{bulgarian}
446   \DeclareLanguage{catalan}
447   \DeclareLanguage{croatian}
448   \DeclareLanguage{czech}
449   \DeclareLanguage{danish}
450   \DeclareLanguage{dutch}
451   \DeclareLanguage{english}
452   \DeclareLanguage{finnish}
453   \DeclareLanguage{french}
454   \DeclareLanguage{german}
455   \DeclareLanguage{greek}
456   \DeclareLanguage{hebrew}
457   \DeclareLanguage{hungarian}
458   \DeclareLanguage{icelandic}
459   \DeclareLanguage{italian}
460   \DeclareLanguage{norwegian}
461   \DeclareLanguage{polish}
462   \DeclareLanguage{portuges}
463   \DeclareLanguage{romanian}
464   \DeclareLanguage{russian}
465   \DeclareLanguage{serbocroatian}
466   \DeclareLanguage{slovak}
467   \DeclareLanguage{slovenian}
468   \DeclareLanguage{spanish}
469   \DeclareLanguage{swedish}
470   \DeclareLanguage{turkish}
471   \DeclareLanguage{ukrainian}
472
473   % --------------------------------------------------------------------------
474   % aliases and dialects:
475   \DeclareLanguageAlias   {Albanian}{albanian}
476   \DeclareLanguageAlias   {Bulgarian}{bulgarian}
477   \DeclareLanguageAlias   {Catalan}{catalan}
478   \DeclareLanguageAlias   {Croatian}{croatian}
479   \DeclareLanguageAlias   {Czech}{czech}
480   \DeclareLanguageAlias   {Danish}{danish}
481   \DeclareLanguageAlias   {Dutch}{dutch}
482   \DeclareLanguageAlias   {Finnish}{finnish}
483   \DeclareLanguageAlias   {francais}{french}
484   \DeclareLanguageAlias   {Francais}{francais}
485   \DeclareLanguageDialect{canadien}{french}
486   \DeclareLanguageAlias   {Canadien}{canadien}
487   \DeclareLanguageAlias   {French}{french}
488   \DeclareLanguageDialect{american}{english}
489   \DeclareLanguageAlias   {American}{american}
490   \DeclareLanguageDialect{australian}{english}
491   \DeclareLanguageAlias   {Australian}{australian}
492   \DeclareLanguageDialect{british}{english}
```

```
493  \DeclareLanguageAlias   {British}{british}
494  \DeclareLanguageDialect{canadian}{english}
495  \DeclareLanguageAlias   {Canadian}{canadian}
496  \DeclareLanguageAlias   {English}{english}
497  \DeclareLanguageAlias   {UKenglish}{british}
498  \DeclareLanguageAlias   {USenglish}{american}
499  \DeclareLanguageDialect{austrian}{german}
500  \DeclareLanguageAlias   {Austrian}{austrian}
501  \DeclareLanguageAlias   {German}{german}
502  \DeclareLanguageAlias   {germanb}{german}
503  \DeclareLanguageDialect{naustrian}{austrian}
504  \DeclareLanguageAlias   {ngerman}{german}
505  \DeclareLanguageAlias   {Greek}{greek}
506  \DeclareLanguageAlias   {polutonikogreek}{greek}
507  \DeclareLanguageAlias   {Hebrew}{hebrew}
508  \DeclareLanguageAlias   {Hungarian}{hungarian}
509  \DeclareLanguageDialect{magyar}{hungarian}
510  \DeclareLanguageAlias   {Magyar}{magyar}
511  \DeclareLanguageAlias   {Icelandic}{icelandic}
512  \DeclareLanguageAlias   {Italian}{italian}
513  \DeclareLanguageAlias   {norsk}{norwegian}
514  \DeclareLanguageAlias   {Norsk}{norsk}
515  \DeclareLanguageAlias   {Norwegian}{norwegian}
516  \DeclareLanguageAlias   {nynorsk}{norwegian}
517  \DeclareLanguageAlias   {Nynorsk}{nynorsk}
518  \DeclareLanguageAlias   {Polish}{polish}
519  \DeclareLanguageDialect{brazil}{portuges}
520  \DeclareLanguageAlias   {Brazil}{brazil}
521  \DeclareLanguageAlias   {brazilian}{brazil}
522  \DeclareLanguageAlias   {Brazilian}{brazilian}
523  \DeclareLanguageAlias   {Portuges}{portuges}
524  \DeclareLanguageAlias   {portuguese}{portuges}
525  \DeclareLanguageAlias   {Portuguese}{portuguese}
526  \DeclareLanguageAlias   {Romanian}{romanian}
527  \DeclareLanguageAlias   {Russian}{russian}
528  \DeclareLanguageAlias   {Serbocroatian}{serbocroatian}
529  \DeclareLanguageAlias   {Slovak}{slovak}
530  \DeclareLanguageAlias   {Slovenian}{slovenian}
531  \DeclareLanguageAlias   {Spanish}{spanish}
532  \DeclareLanguageAlias   {Swedish}{swedish}
533  \DeclareLanguageDialect{swissgerman}{german}
534  % this maybe should be a language of it's own:
535  \DeclareLanguageAlias   {Swiss}{swissgerman}
536  \DeclareLanguageAlias   {Swissgerman}{swissgerman}
537  \DeclareLanguageAlias   {Turkish}{turkish}
538  \DeclareLanguageAlias   {Ukrainian}{ukrainian}
539
540  % -------------------------------------------------------------------------
541  % load basic dictionary if available
542  \AtBeginDocument{%
543    \@trnslt@load@dictionary@silent@for
544      {translations-basic-dictionary}
```

```
545        {\@trnslt@current@language}%
546    }
547
548    \endinput
549
550    % --------------------------------------------------------------------------
551    % HISTORY:
552    2012/09/30 v0.2beta - first version (as part of the `exsheets' bundle)
553    2012/10/05 v0.2      - \LoadDictionary and \LoadDictionaryFor added and loads of
554                           languages defined.
555    2013/03/10 v0.8      - basic dictionaries for English, German, French and Spanish
556                          - new command \DeclareDictTranslation
557    2013/04/04 v0.8a     - bug fix in \DeclareDictTranslation
558    2013/04/07 v0.9      - slightly improved messages
559    2013/04/08 v0.9a     - changed fallback warning into info
560                          - synchronized version number with `exsheets' until now but
561                            won't any more
562    2013/06/22 v0.9b     - added Swiss
563    2013/06/28 v0.10     - declaring aliases of dialects now works as expected
564                          - declarings dialects of an alias now correctly declares
565                            the dialect to the correct base language
566                          - corrected a few erroneous language declarations
567    2013/07/12 v0.10a    - \GetTranslation gets two-folded fallback: use
568                            fallback-translation if no translation for the current
569                            language has been defined; use literal string if /no/
570                            language is used - this should never happen but /will/
571                            happen if neither `babel' nor `polyglossia' have been
572                            loaded, i.e., no language has been chosen /and/ the
573                            package writer did not provide an English translation
574    2013/07/15 v1.0      - removed from `exsheets' bundle - `translations' should
575                            be a package of it's own
576                          - load basic dictionary automatically if available
577                          - rudimentary check in \LoadDictionary if loaded file is a
578                            dictionary
579                          - new command \PrintDictionaryFor
```

Index

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