

translations

v1.0a 2013/07/18

internationalization of L^AT_EX 2_ε packages

Clemens NIEDERBERGER

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

English documentation

Contents

		3.4.2	The ‘fallback’ language	6
		3.5	Dictionaries	6
		3.5.1	Own Dictionaries	6
		3.5.2	TRANSLATIONS’ Basic Dictionaries	7
1	Motivation	1		
2	License and Requirements	2		
3	Usage	2		
	3.1	Background	2	
	3.2	Available Commands	3	
	3.3	A Small Example	4	
	3.4	Usage in Packages	5	
	3.4.1	Basic Structure	5	
		4	Defined Languages	9
		5	Implementation	12
			Index	28

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 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 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 packages etoolbox⁴ and scrfile (part of the KOMA-Script bundle⁵).

3 Usage

3.1 Background

The **TRANSLATIONS** package enables the author of a package or a class (or a document) to declare translations in different languages of key words and fetch these translations in the document depending on the active language as set by babel or polyglossia. Since **TRANSLATIONS** checks which language is active it is generally not necessary to specify the language for which a translation should be fetched manually.

TRANSLATIONS knows of three types of languages: main languages (see table 2 on page 9), language aliases (see table 4 on page 10) and language dialects (see table 3 on page 10). For the commands declaring or fetching a translation base languages and language aliases are equivalent. Dialects are similar to aliases but there are a few important differences. An alias can for example be an alias of a dialect.

Figure 1 shows what happens if **TRANSLATIONS** is asked to fetch a translation for a given key.

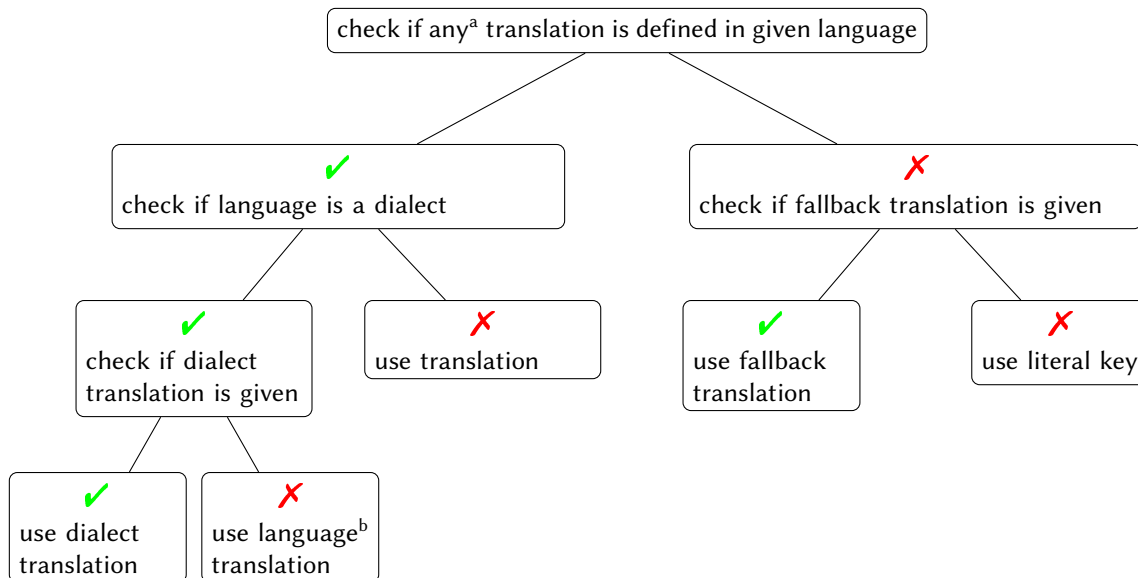


FIGURE 1: Schematic representation of **TRANSLATIONS**’ translating mechanism. Notes: ^a except for a possible fallback translation. ^b i.e., the base language of the dialect.

What happens if you declare a translation? There are four cases:

1. You declare a translation for a base language: this is the normal case where an internal macro is defined which can be fetched by the `\GetTranslation` command (see section 3.2).

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

2. You declare a translation for a language alias: this is the very same as the first case since the same internal macro is defined.
3. You declare a translation for a dialect: this is two-fold. Either a translation for the base language exists so only the translation for the dialect is saved. If the translation for the base language does not exist it is defined to be the same as the one for the dialect.
4. You declare a translation for an alias of a dialect: this is the very same as the third case as again the internal macros are the same.

3.2 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.

★ `\GetTranslationForWarn{<lang>}{<key>}`

Introduced in version 1.0 Fetches and prints the translation of <key> for the language <lang>. Issues a warning if no translation is available.

★ `\GetTranslationWarn{<key>}`

Introduced in version 1.0 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 3.5 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>}`

Introduced in version 1.0 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.

3.3 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.

```

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

3.4 Usage in Packages

3.4.1 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\"oner 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.

3.4.2 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}

```

3.5 Dictionaries

3.5.1 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.

3.5.2 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. Table 2 lists all base languages, "fallback" being a dummy language used for fallback translations. The tables 2, 4 and 3 list *all* language names known to **TRANSLATIONS**. Unfortunately they're not sorted but listed in the order they have been defined. Although I tried to make this alphabetically it will not always be.

If you miss a language or recognize a language that has falsely been declared as an alias but should rather be a dialect or base language itself (or any variation of this theme) please let me know, preferably with a short explanation what's wrong and why.

TABLE 2: Base languages defined by **TRANSLATIONS**, from left to right in the order of definition.

fallback	afrikaans	albanian	amharic	arabic
armenian	asturian	basque	bengali	breton
bulgarian	catalan	coptic	czech	danish
dutch	english	esperanto	estonian	ethiop
farsi	finnish	french	friulan	gaelic
galician	german	greek	hebrew	hindustani
hungarian	icelandic	interlingua	italian	japanese
kannada	latin	lao	latin	latvian
lithuanian	malay	malayalam	maldivian	marathi
mongolian	norwegian	occitan	piedmontese	pinyin
polish	portuges	romanian	romansh	russian
samin	sanskrit	serbocroatian	slovak	slovenian
sorbian	spanish	spanish	swedish	tamil
telugu	thai	tibetan	turkish	turkmen

continues

4 Defined Languages

ukrainian	vietnamese	welsh	asturian-leonese	canadien
acadian	american	australian	british	canadian
newzealand	irish	scottish	austrian	hindi
urdu	indonesian	bahasa melayu	brazil	serbian
croatian	lowersorbian	uppersorbian	swissgerman	

TRANSLATIONS also defines a few dialects of these base languages. They are listed in table 3.

TABLE 3: All dialects defined by **TRANSLATIONS**, from left to right in the order of definition.

dialect	language	dialect	language
canadien	french	acadian	french
american	english	australian	english
british	english	canadian	english
newzealand	english	irish	gaelic
scottish	gaelic	austrian	german
hindi	hindustani	urdu	hindustani
indonesian	malay	brazil	portuges
serbian	serbocroatian	croatian	serbocroatian
lowersorbian	sorbian	uppersorbian	sorbian
swissgerman	german		

To every one of these languages and dialects 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 4.

TABLE 4: All language aliases defined by **TRANSLATIONS**, from left to right in the order of definition.

alias	language	alias	language
Fallback	fallback	Afrikaans	afrikaans
Albanian	albanian	Amharic	amharic
Arabic	arabic	Armenian	armenian
Asturian	asturian	astur-leonese	asturian
Astur-Leonese	astur-leonese	Asturian-Leonese	asturian-leonese
Basque	basque	Bengali	bengali
Breton	breton	Bulgarian	bulgarian
Catalan	catalan	Coptic	coptic
coptic egyptian	coptic	Coptic Egyptian	coptic egyptian
Czech	czech	Danish	danish
Dutch	dutch	Farsi	farsi
Finnish	finnish	français	french
Français	français	Canadien	canadien
French	french	Acadian	acadian

continues

4 Defined Languages

alias	language	alias	language
frenchle	french	American	american
Australian	australian	British	british
Canadian	canadian	English	english
UKenglish	british	USenglish	american
Newzealand	newzealand	Ethiop	ethiop
Esperanto	esperanto	Estonian	estonian
Friulan	friulan	Gaelic	gaelic
Irish	irish	irish gaelic	irish
Irish Gaelic	irish	Scottish	scottish
scottish gaelic	scottish	Scottish Gaelic	scottish
Galician	galician	German	german
germanb	german	ngerman	german
Austrian	austrian	naustrian	austrian
Greek	greek	polutonikogreek	greek
ibygreek	greek	bgreek	greek
Hebrew	hebrew	Hindustani	hindustani
hindi-urdu	hindustani	Hindi-Urdu	hindi-urdu
Hindi	hindi	Urdu	urdu
Hungarian	hungarian	magyar	hungarian
Magyar	magyar	Icelandic	icelandic
Interlingua	interlingua	Italian	italian
Japanese	japanese	Kannada	kannada
Ladin	ladin	Lao	lao
laotian	lao	Laotian	laotian
Latin	latin	Latvian	latvian
lettish	latvian	Lettish	lettish
Lithuanian	lithuanian	Malay	malay
Indonesian	indonesian	indon	indonesian
bahasa meyalu	malay	Bahasa Meyalu	bahasa melayu
bahasa	bahasa melayu	Bahasa	bahasa
bahasai	bahasa	bahasam	bahasa
Malayalam	malayalam	Maldivian	maldivian
divehi	maldivian	Divehi	divehi
Marathi	marathi	Mongolian	mongolian
norsk	norwegian	Norsk	norsk
Norwegian	norwegian	nynorsk	norwegian
Nynorsk	nynorsk	Occitan	occitan
lenga d'oc	occitan	langue d'oc	occitan
Piedmontese	piedmontese	piemontese	piedmontese
Piemontese	piemontese	piemonteis	piedmontese
Piemonteis	piemonteis	Pinyin	pinyin
Polish	polish	Brazil	brazil
brazilian	brazil	Brazilian	brazilian

continues

alias	language	alias	language
Portuges	portuges	portuguese	portuges
Portuguese	portuguese	Romanian	romanian
Romansh	romansh	Romansch	romansh
Rumantsh	romansh	Rumantsch	romansh
Romanche	romansh	Russian	russian
Samin	samin	north sami	samin
North Sami	north sami	northern sami	north sami
Northern Sami	northern sami	Sanskrit	sanskrit
serbo-croatian	serbocroatian	Serbo-Croatian	serbocroatian
Serbian	serbian	serbianc	serbian
Croatian	croatian	Slovak	slovak
Slovenian	slovenian	Sorbian	sorbian
Lowersorbian	lowersorbian	Uppersorbian	uppersorbian
Isorbian	lowersorbian	usorbian	uppersorbian
lower sorbian	lowersorbian	upper sorbian	uppersorbian
Lower Sorbian	lowersorbian	Upper Sorbian	uppersorbian
Spanglish	spanglish	Spanish	spanish
Swedish	swedish	Swiss	swissgerman
Swissgerman	swissgerman	Tamil	tamil
Telugu	telugu	Thai	thai
thaicjk	thai	Thaicjk	thaicjk
Tibetan	tibetan	Turkish	turkish
Turkmen	turkmen	Ukrainian	ukrainian
Vietnamese	vietnamese	Welsh	welsh

These languages *should* cover all languages which are currently covered by babel and polyglossia but very likely this is not the case. Should you miss a language please send me an email so I can add it to [TRANSLATIONS](#).

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{2013/07/18}
32 \def\@trnslt@version{v1.0a}
33 \def\@trnslt@info{internationalization of LaTeX2e packages}
34
35 \ProvidesPackage{translations}[\@trnslt@date\space \@trnslt@version\space \@trnslt@info
\space (CN)]
36 \RequirePackage{etoolbox,scrfile}
37
38 % -----
39 % message handling
40 \def\@trnslt@error@message{%
41   For details have a look at the `translations' manual.}

```

5 Implementation

```

42
43 \def\@trnslt@create@message#1{%
44   \ifstrequal{#1}{Error}
45     {%
46       \lowercase{\csdef{\@trnslt@#1}}##1{%
47         \csuse{Package#1}{translations}{##1}{\@trnslt@error@message}}%
48     }{%
49       \lowercase{\csdef{\@trnslt@#1}}##1{%
50         \csuse{Package#1}{translations}{##1}}%
51     }}
52 \@trnslt@create@message{Error}
53 \@trnslt@create@message{Warning}
54 \@trnslt@create@message{WarningNoLine}
55 \@trnslt@create@message{Info}
56
57 \def\@trnslt@err@unknown@lang#1{%
58   \@trnslt@error{Unknown language `#1'}}
59
60 \def\@trnslt@warn@unknown@lang#1{%
61   \@trnslt@warning{Unknown language `#1'}}
62
63 \def\@trnslt@err@already@defined#1#2{%
64   \@trnslt@error{The #2 translation for `#1' is already defined.}}
65
66 \def\@trnslt@err@not@defined#1#2{%
67   \@trnslt@error{The \@trnslt@language{#2} translation for `#1' is not defined yet.}}
68
69 % -----
70 % check if babel or polyglossia is used
71 \AtEndPreamble{
72   \@ifpackageloaded{babel}{}{
73     \@ifpackageloaded{polyglossia}{}
74     {\@trnslt@warning{No language package found. I am going to use `english'
75       as default language.}}
76   }
77   \ifdef\language{
78     {\def\language{english}}
79   }
80   \ifdef\bbl@afterfi{
81     {\long\def\bbl@afterfi#1\fi{\fi#1}}
82   }
83
84 % -----
85 % book keeping: the following macros will be used as `etoolbox' lists that
86 % keep record of defined languages, dialects and aliases
87 \def\@trnslt@languages{}% all languages
88 \def\@trnslt@aliases@pair{}% all aliases and their base
89 \def\@trnslt@aliases@single{}% all aliases
90 \def\@trnslt@dialects@pair{}% all dialects and their base
91 \def\@trnslt@dialects@single{}% all dialects
92
93 % -----

```

5 Implementation

```

94 % define \@trnslt@if@<name> conditionals that don't leave the checked macro as
95 % \relax behind and check for \@trnslt@<name>@#1. These conditionals should
96 % also be expandable in an \edef-like context:
97 \def\@trnslt@newif#1{%
98   \csdef{@trnslt@if@#1}##1{%
99     \expandafter\expandafter\expandafter\expandafter
100     \expandafter\expandafter\expandafter
101     \@firstofone
102     {\expandafter\expandafter\expandafter}%
103     \ifcsname @trnslt@#1@##1\endcsname
104     \expandafter\@firstoftwo
105     \else
106     \expandafter\@secondoftwo
107     \fi
108   }%
109 }
110
111 % -----
112 % \DeclareLanguage
113 % #1: language
114 \newrobustcmd*\DeclareLanguage[1]{%
115   \@trnslt@declare@language{#1}}
116 \@onlypreamble\DeclareLanguage
117
118 \def\@trnslt@declare@language#1{%
119   \@trnslt@if@language{#1}
120   {}{%
121     \csdef{@trnslt@language@#1}{#1}%
122     \listead\@trnslt@languages{#1}%
123   }%
124 }
125
126 \def\@trnslt@language#1{%
127   \csuse{@trnslt@language@#1}}
128
129 \@trnslt@newif{language}
130
131 % -----
132 % \DeclareLanguageDialect
133 % #1: dialect
134 % #2: language
135 \newrobustcmd*\DeclareLanguageDialect[2]{%
136   \@trnslt@declare@language@dialect{#1}{#2}}
137 \@onlypreamble\DeclareLanguageDialect
138
139 \def\@trnslt@declare@language@dialect#1#2{%
140   \@trnslt@if@language{#2}
141   {}{%
142     \@trnslt@warn@unknown@lang{#2}%
143     \@trnslt@declare@language{#2}%
144   }%
145   \@trnslt@if@dialect{#1}

```

5 Implementation

```

146   {% => ist schon als dialect definiert => irgendwelche weiteren checks?
147   }
148   {%
149   \@trnslt@if@alias{#2}
150   {%
151   \csedef{@trnslt@dialect@#1}{\@trnslt@alias{#2}}{#1}}%
152   \@trnslt@declare@language{#1}%
153   \listeadadd\@trnslt@dialects@single{#1}%
154   \listeadadd\@trnslt@dialects@pair{#1}{\@trnslt@alias{#2}}}%
155   }
156   {%
157   \csdef{@trnslt@dialect@#1}{#2}{#1}}%
158   \@trnslt@declare@language{#1}%
159   \listeadadd\@trnslt@dialects@single{#1}%
160   \listeadadd\@trnslt@dialects@pair{#1}{#2}}%
161   }%
162   }%
163   }
164
165   \def\@trnslt@dialect#1{%
166   \csuse{@trnslt@dialect@#1}}
167
168   % this macros fetches the base language for a given dialect, expandably:
169   \def\@trnslt@dialect@of#1{%
170   \@trnslt@if@dialect{#1}
171   {%
172   \expandafter\expandafter\expandafter
173   \@firstoftwo
174   \csname @trnslt@dialect@#1\endcsname
175   }{%}%
176   }
177
178   \@trnslt@newif{dialect}
179
180   % -----
181   % \DeclareLanguageAlias
182   % #1: alias
183   % #2: language
184   \newrobustcmd*\DeclareLanguageAlias[2]{%
185   \@trnslt@declare@languagealias{#1}{#2}}
186   \@onlypreamble\DeclareLanguageAlias
187
188   \def\@trnslt@declare@languagealias#1#2{%
189   \@trnslt@if@language{#2}
190   }{%}%
191   \@trnslt@warn@unknown@lang{#2}%
192   \@trnslt@declare@language{#2}%
193   }%
194   \csletcs{@trnslt@language@#1}{@trnslt@language@#2}%
195   \@trnslt@if@dialect{#2}
196   {\csletcs{@trnslt@dialect@#1}{@trnslt@dialect@#2}}
197   }{%}%

```

5 Implementation

```

198 \ifinlist{#1}\@trnslt@aliases@single
199 {}{%
200 \csdef{@trnslt@alias@#1}{#2}%
201 \listead{\@trnslt@aliases@pair{#1}{#2}}%
202 \listead{\@trnslt@aliases@single{#1}}%
203 }%
204 }
205
206 \def\@trnslt@alias#1{%
207 \csuse{@trnslt@alias@#1}}
208
209 \@trnslt@newif{alias}
210
211 % -----
212 % dummy language: `fallback':
213 \DeclareLanguage{fallback}
214 \DeclareLanguageAlias{Fallback}{fallback}
215
216 % -----
217 % \DeclareTranslation, \NewTranslation and \RenewTranslation
218 % #1: language
219 % #2: word
220 % #3: replacement
221 \newrobustcmd*\DeclareTranslation[3]{%
222 \@trnslt@declare@translation{#2}{#1}{#3}}
223 \@onlypreamble\DeclareTranslation
224
225 \newrobustcmd*\DeclareTranslationFallback[2]{%
226 \@trnslt@declare@translation{#1}{fallback}{#2}}
227 \@onlypreamble\DeclareTranslationFallback
228
229 \newrobustcmd*\NewTranslation[3]{%
230 \@trnslt@new@translation{#2}{#1}{#3}}
231 \@onlypreamble\NewTranslation
232
233 \newrobustcmd*\RenewTranslation[3]{%
234 \@trnslt@renew@translation{#2}{#1}{#3}}
235 \@onlypreamble\RenewTranslation
236
237 % #1: word
238 % #2: language
239 % #3: replacement
240 \def\@trnslt@declare@translation#1#2#3{%
241 \@trnslt@if@language{#2}
242 {%
243 \@trnslt@if@dialect{#2}
244 {%
245 \csdef{@trnslt@word@#1\@trnslt@dialect{#2}}{#3}%
246 \@trnslt@if@word\@trnslt@dialect@of{#1}{#2}
247 {}
248 {\csdef{@trnslt@word@#1\@trnslt@dialect@of{#2}}{#3}}%
249 }

```


5 Implementation

```

250     {\csdef{@trnslt@word@#1@\@trnslt@language{#2}}{#3}}%
251     % save the <word> as <word>:
252     \csdef{@trnslt@word@#1@literal}{#1}%
253 }
254 {\@trnslt@err@unknown@lang{#2}}%
255 }
256
257 \def\@trnslt@new@translation#1#2#3{%
258   \@trnslt@if@translation{#1}{#2}
259   {\@trnslt@err@already@defined{#1}{#2}}
260   {\@trnslt@declare@translation{#1}{#2}{#3}}}
261
262 \def\@trnslt@renew@translation#1#2#3{%
263   \@trnslt@if@translation{#1}{#2}
264   {\@trnslt@declare@translation{#1}{#2}{#3}}
265   {\@trnslt@err@not@defined{#1}{#2}}}
266
267 % -----
268 % now let's go through some trouble to check if a translation exists:
269 \def\@trnslt@if@word#1#2#3{%
270   \expandafter\expandafter\expandafter\expandafter
271   \expandafter\expandafter\expandafter
272   \@firstofone
273   {\expandafter\expandafter\expandafter}%
274   \ifcsname @trnslt@word@#2@#1{#3}\endcsname
275   \expandafter\@firstoftwo
276   \else
277   \expandafter\@secondoftwo
278   \fi
279 }%
280
281 \def\@trnslt@if@translation#1#2{%
282   \@trnslt@if@word\@trnslt@language{#1}{#2}
283   {\expandafter\@firstoftwo}
284   {%
285     \@trnslt@if@dialect{#2}
286     {%
287       \@trnslt@if@word\@trnslt@dialect{#1}{#2}
288       {\expandafter\@firstoftwo}
289       {%
290         \@trnslt@if@word\@trnslt@dialect@of{#1}{#2}
291         {\expandafter\@firstoftwo}
292         {\expandafter\@secondoftwo}%
293       }
294     }
295     {\expandafter\@secondoftwo}%
296   }%
297 }
298
299 % -----
300 % \GetTranslationFor and \GetTranslation
301 % these need to be expandable!

```

5 Implementation

```

302 % #1: language
303 % #2: word
304 \newcommand*\GetTranslationFor[2]{%
305   \@trnslt@checkandget@translation@for{#2}{#1}}
306
307 \newcommand*\GetTranslation[1]{%
308   \@trnslt@checkandget@translation@for{#1}{\@trnslt@current@language}}
309
310 % unexpandable version of the commands that raise a warning if no translation
311 % is available:
312 \newcommand*\GetTranslationForWarn[2]{%
313   \@trnslt@getandwarn@translation@for{#2}{#1}}
314
315 \newcommand*\GetTranslationWarn[1]{%
316   \@trnslt@getandwarn@translation@for{#1}{\@trnslt@current@language}}
317
318 % #1: word #2: language
319 \def\@trnslt@get@translation@for#1#2{%
320   \@trnslt@if@dialect{#2}
321   {%
322     \ifcsdef{@trnslt@word@#1@\@trnslt@dialect{#2}}
323       {\csuse{@trnslt@word@#1@\@trnslt@dialect{#2}}}
324       {\csuse{@trnslt@word@#1@\@trnslt@dialect@of{#2}}}%
325   }
326   {\csuse{@trnslt@word@#1@\@trnslt@language{#2}}}%
327 }
328
329 \def\@trnslt@checkandget@translation@for#1#2{%
330   \@trnslt@if@translation{#1}{#2}
331   {\@trnslt@get@translation@for{#1}{#2}}
332   {%
333     \@trnslt@if@translation{#1}{fallback}
334     {\csuse{@trnslt@word@#1@fallback}}
335     {\csuse{@trnslt@word@#1@literal}}}%
336   }%
337 }
338
339 % this is not expandable!
340 \protected\def\@trnslt@getandwarn@translation@for#1#2{%
341   \@trnslt@if@translation{#1}{#2}
342   {\@trnslt@get@translation@for{#1}{#2}}
343   {%
344     \@trnslt@warning{Translation for `#1' in #2 unknown. You may try to use
345       \string\DeclareTranslation{#2}{#1}{ ... } in your preamble.}%
346     \@trnslt@if@translation{#1}{fallback}
347     {%
348       \@trnslt@info{Using fallback translation for `#1'}%
349       \csuse{@trnslt@word@#1@fallback}
350     }
351     {\csuse{@trnslt@word@#1@literal}}}%
352   }%
353 }

```

5 Implementation

```

354 % -----
355 % \SaveTranslationFor and \SaveTranslation
356 \newrobustcmd*\SaveTranslationFor[3]{%
357   \@trnslt@save@translation@for{#1}{#3}{#2}}
358
359
360 \newrobustcmd*\SaveTranslation[2]{%
361   \@trnslt@save@translation@for{#1}{#2}{\@trnslt@current@language}}
362
363 \def\@trnslt@save@translation@for#1#2#3{%
364   \edef#1{%
365     \@trnslt@if@translation{#2}{#3}
366     {\csuse{\@trnslt@word@#2@\@trnslt@language{#3}}}
367     }%
368   }}
369
370 % -----
371 % \LoadDictionary and \LoadDictionaryFor
372 \newrobustcmd*\LoadDictionary[1]{%
373   \@trnslt@load@dictionary@for{#1}{\@trnslt@current@language}}
374 \@onlypreamble\LoadDictionary
375
376 \newrobustcmd*\LoadDictionaryFor[2]{%
377   \@trnslt@load@dictionary@for{#2}{#1}}
378 \@onlypreamble\LoadDictionaryFor
379
380 % #1: name
381 % #2: lang
382 \def\@trnslt@load@dictionary@for#1#2{%
383   \AtBeginDocument{%
384     \InputIfFileExists{#1-\@trnslt@language{#2}.trsl}
385     {\@trnslt@check@dictionary{#1}{#2}}
386     {\@trnslt@warning{dictionary file `#1-\@trnslt@language{#2}.trsl' not
387       found.}}%
388   }}
389
390 \def\@trnslt@check@dictionary#1#2{%
391   \AfterFile{#1-\@trnslt@language{#2}.trsl}
392   {%
393     \ifcsdef{\@trnslt@dictionary@#1@\@trnslt@language{#2}}
394     {\@trnslt@info{loading dictionary `#1' for `#2'.}}
395     {%
396       \@trnslt@warning{file `#1-\@trnslt@language{#2}.trsl' does not
397         appear to be a dictionary}%
398     }%
399   }%
400 }
401
402 \def\@trnslt@load@dictionary@silent@for#1#2{%
403   \AtBeginDocument{\InputIfFileExists{#1-\@trnslt@language{#2}.trsl}{}}{}
404
405 \newrobustcmd*\ProvideDictionaryFor[2]{%

```

5 Implementation

```

406 \@trnslt@provide@dictionary@for{#1}{#2}}
407 \@onlypreamble\ProvideDictionaryFor
408
409 \def\@trnslt@provide@dictionary@for#1#2{%
410 \def\@trnslt@dictionary@name{#2}%
411 \edef\@trnslt@dictionary@lang{\@trnslt@language{#1}}%
412 % this macro can be used to check if we have a dictionary and will also be
413 % used as a list for the dictionary entries:
414 \csdef{\@trnslt@dictionary@\@trnslt@dictionary@name \@trnslt@dictionary@lang}{}%
415 \ifnextchar[
416 {\@trnslt@provide@dictionary@version}
417 {
418 \ProvidesFile
419 {#2-\@trnslt@dictionary@lang.trsl}%
420 [(\@trnslt@dictionary@lang\space translation file `#2')]
421 }%
422 }
423
424 \def\@trnslt@provide@dictionary@version[#1]{%
425 \ProvidesFile
426 {\@trnslt@dictionary@name-\@trnslt@dictionary@lang.trsl}%
427 [(\@trnslt@dictionary@lang\space translation file \@trnslt@dictionary@name') #1]}
428
429 % \@trnslt@dictionary@language
430 \newrobustcmd*\DeclareDictTranslation[2]{%
431 \listcsadd
432 {\@trnslt@dictionary@\@trnslt@dictionary@name \@trnslt@dictionary@lang}
433 {{#1}{#2}}%
434 \@trnslt@declare@translation{#1}{\@trnslt@dictionary@lang}{#2}%
435 }
436 \@onlypreamble\DeclareDictTranslation
437
438 % \PrintDictionaryFor
439 % #1: lang
440 % #2: name
441 % #3: pre
442 % #4: mid
443 % #5: post
444 \newcommand*\PrintDictionaryFor[5]{%
445 \@trnslt@print@dictionary@for{#1}{#2}{#3}{#4}{#5}}
446
447 % #1: lang
448 % #2: name
449 % #3: pre
450 % #4: mid
451 % #5: post
452 \def\@trnslt@print@dictionary@for#1#2#3#4#5{%
453 \forlistcsloop
454 {\@trnslt@print@dictionary@entry{#3}{#4}{#5}}
455 {\@trnslt@dictionary@#2@\@trnslt@language{#1}}%
456 }
457

```

5 Implementation

```
458 % #1: pre
459 % #2: mid
460 % #3: post
461 % #4: {key}{translation}
462 \def\@trnslt@print@dictionary@entry#1#2#3#4{%
463   \@trnslt@print@dictionary@entry@aux{#1}{#2}{#3}#4}
464
465 % #1: pre
466 % #2: mid
467 % #3: post
468 % #4: key
469 % #5: translation
470 \def\@trnslt@print@dictionary@entry@aux#1#2#3#4#5{#1#4#2#5#3}
471
472 % -----
473 % predefined languages
474 \DeclareLanguage{afrikaans}
475 \DeclareLanguage{albanian}
476 \DeclareLanguage{amharic}
477 \DeclareLanguage{arabic}
478 \DeclareLanguage{armenian}
479 \DeclareLanguage{asturian}
480 \DeclareLanguage{basque}
481 \DeclareLanguage{bengali}
482 \DeclareLanguage{breton}
483 \DeclareLanguage{bulgarian}
484 \DeclareLanguage{catalan}
485 \DeclareLanguage{coptic}
486 \DeclareLanguage{czech}
487 \DeclareLanguage{danish}
488 \DeclareLanguage{dutch}
489 \DeclareLanguage{english}
490 \DeclareLanguage{esperanto}
491 \DeclareLanguage{estonian}
492 \DeclareLanguage{ethiop}
493 \DeclareLanguage{farsi}
494 \DeclareLanguage{finnish}
495 \DeclareLanguage{french}
496 \DeclareLanguage{friulan}
497 \DeclareLanguage{gaelic}
498 \DeclareLanguage{galician}
499 \DeclareLanguage{german}
500 \DeclareLanguage{greek}
501 \DeclareLanguage{hebrew}
502 \DeclareLanguage{hindustani}
503 \DeclareLanguage{hungarian}
504 \DeclareLanguage{icelandic}
505 \DeclareLanguage{interlingua}
506 \DeclareLanguage{italian}
507 \DeclareLanguage{japanese}
508 \DeclareLanguage{kannada}
509 \DeclareLanguage{ladin}
```

5 Implementation

```
510 \DeclareLanguage{lao}
511 \DeclareLanguage{latin}
512 \DeclareLanguage{latvian}
513 \DeclareLanguage{lithuanian}
514 \DeclareLanguage{malay}
515 \DeclareLanguage{malayalam}
516 \DeclareLanguage{maldivian}
517 \DeclareLanguage{marathi}
518 \DeclareLanguage{mongolian}
519 % polyglossia seems to support this one but it is unclear which language is
520 % actually meant by it:
521 % \DeclareLanguage{nko}
522 \DeclareLanguage{norwegian}
523 \DeclareLanguage{occitan}
524 \DeclareLanguage{piedmontese}
525 \DeclareLanguage{pinyin}
526 \DeclareLanguage{polish}
527 \DeclareLanguage{portuges}
528 \DeclareLanguage{romanian}
529 \DeclareLanguage{romansh}
530 \DeclareLanguage{russian}
531 \DeclareLanguage{samin}
532 \DeclareLanguage{sanskrit}
533 \DeclareLanguage{serbocroatian}
534 \DeclareLanguage{slovak}
535 \DeclareLanguage{slovenian}
536 \DeclareLanguage{sorbian}
537 % not sure about this: isn't it either a Spanish or English dialect?
538 \DeclareLanguage{spanglish}
539 \DeclareLanguage{spanish}
540 \DeclareLanguage{swedish}
541 % polyglossia seems to support this one but it is unclear which language is
542 % actually meant by it:
543 % \DeclareLanguage{syriac}
544 \DeclareLanguage{tamil}
545 \DeclareLanguage{telugu}
546 \DeclareLanguage{thai}
547 \DeclareLanguage{tibetan}
548 \DeclareLanguage{turkish}
549 \DeclareLanguage{turkmen}
550 \DeclareLanguage{ukrainian}
551 \DeclareLanguage{vietnamese}
552 \DeclareLanguage{welsh}
553
554 % -----
555 % aliases and dialects:
556 \DeclareLanguageAlias {Afrikaans}{afrikaans}
557 \DeclareLanguageAlias {Albanian}{albanian}
558 \DeclareLanguageAlias {Amharic}{amharic}
559 \DeclareLanguageAlias {Arabic}{arabic}
560 \DeclareLanguageAlias {Armenian}{armenian}
561 \DeclareLanguageAlias {Asturian}{asturian}
```

5 Implementation

```
562 \DeclareLanguageAlias {astur-leonese}{asturian}
563 \DeclareLanguageAlias {Astur-Leonese}{astur-leonese}
564 \DeclareLanguageAlias {Asturian-Leonese}{asturian-leonese}
565 \DeclareLanguageAlias {Basque}{basque}
566 \DeclareLanguageAlias {Bengali}{bengali}
567 \DeclareLanguageAlias {Breton}{breton}
568 \DeclareLanguageAlias {Bulgarian}{bulgarian}
569 \DeclareLanguageAlias {Catalan}{catalan}
570 \DeclareLanguageAlias {Coptic}{coptic}
571 \DeclareLanguageAlias {coptic egyptian}{coptic}
572 \DeclareLanguageAlias {Coptic Egyptian}{coptic egyptian}
573 \DeclareLanguageAlias {Czech}{czech}
574 \DeclareLanguageAlias {Danish}{danish}
575 \DeclareLanguageAlias {Dutch}{dutch}
576 \DeclareLanguageAlias {Farsi}{farsi}
577 \DeclareLanguageAlias {Finnish}{finnish}
578 \DeclareLanguageAlias {français}{french}
579 \DeclareLanguageAlias {Français}{français}
580 \DeclareLanguageDialect{canadien}{french}
581 \DeclareLanguageAlias {Canadien}{canadien}
582 \DeclareLanguageAlias {French}{french}
583 \DeclareLanguageDialect{acadian}{french}
584 \DeclareLanguageAlias {Acadian}{acadian}
585 \DeclareLanguageAlias {frenchle}{french}
586 \DeclareLanguageDialect{american}{english}
587 \DeclareLanguageAlias {American}{american}
588 \DeclareLanguageDialect{australian}{english}
589 \DeclareLanguageAlias {Australian}{australian}
590 \DeclareLanguageDialect{british}{english}
591 \DeclareLanguageAlias {British}{british}
592 \DeclareLanguageDialect{canadian}{english}
593 \DeclareLanguageAlias {Canadian}{canadian}
594 \DeclareLanguageAlias {English}{english}
595 \DeclareLanguageAlias {UKenglish}{british}
596 \DeclareLanguageAlias {USenglish}{american}
597 \DeclareLanguageDialect{newzealand}{english}
598 \DeclareLanguageAlias {Newzealand}{newzealand}
599 \DeclareLanguageAlias {Ethiop}{ethiop}
600 \DeclareLanguageAlias {Esperanto}{esperanto}
601 \DeclareLanguageAlias {Estonian}{estonian}
602 \DeclareLanguageAlias {Friulan}{friulan}
603 \DeclareLanguageAlias {Gaelic}{gaelic}
604 \DeclareLanguageDialect{irish}{gaelic}
605 \DeclareLanguageDialect{scottish}{gaelic}
606 \DeclareLanguageAlias {Irish}{irish}
607 \DeclareLanguageAlias {irish gaelic}{irish}
608 \DeclareLanguageAlias {Irish Gaelic}{irish}
609 \DeclareLanguageAlias {Scottish}{scottish}
610 \DeclareLanguageAlias {scottish gaelic}{scottish}
611 \DeclareLanguageAlias {Scottish Gaelic}{scottish}
612 \DeclareLanguageAlias {Galician}{galician}
613 \DeclareLanguageAlias {German}{german}
```

5 Implementation

```

614 \DeclareLanguageAlias {germanb}{german}
615 \DeclareLanguageAlias {ngerman}{german}
616 \DeclareLanguageDialect{austrian}{german}
617 \DeclareLanguageAlias {Austrian}{austrian}
618 \DeclareLanguageAlias {naustrian}{austrian}
619 \DeclareLanguageAlias {Greek}{greek}
620 \DeclareLanguageAlias {polutonikogreek}{greek}
621 \DeclareLanguageAlias {ibygreek}{greek}
622 \DeclareLanguageAlias {bgreek}{greek}
623 \DeclareLanguageAlias {Hebrew}{hebrew}
624 \DeclareLanguageAlias {Hindustani}{hindustani}
625 \DeclareLanguageAlias {hindi-urdu}{hindustani}
626 \DeclareLanguageAlias {Hindi-Urdu}{hindi-urdu}
627 \DeclareLanguageDialect{hindi}{hindustani}
628 \DeclareLanguageAlias {Hindi}{hindi}
629 \DeclareLanguageDialect{urdu}{hindustani}
630 \DeclareLanguageAlias {Urdu}{urdu}
631 \DeclareLanguageAlias {Hungarian}{hungarian}
632 \DeclareLanguageAlias {magyar}{hungarian}
633 \DeclareLanguageAlias {Magyar}{magyar}
634 \DeclareLanguageAlias {Icelandic}{icelandic}
635 \DeclareLanguageAlias {Interlingua}{interlingua}
636 \DeclareLanguageAlias {Italian}{italian}
637 \DeclareLanguageAlias {Japanese}{japanese}
638 \DeclareLanguageAlias {Kannada}{kannada}
639 \DeclareLanguageAlias {Ladin}{ladin}
640 \DeclareLanguageAlias {Lao}{lao}
641 \DeclareLanguageAlias {laotian}{lao}
642 \DeclareLanguageAlias {Laotian}{laotian}
643 \DeclareLanguageAlias {Latin}{latin}
644 \DeclareLanguageAlias {Latvian}{latvian}
645 \DeclareLanguageAlias {lettish}{latvian}
646 \DeclareLanguageAlias {Lettish}{lettish}
647 \DeclareLanguageAlias {Lithuanian}{lithuanian}
648 % hopefully someone who knows netter than me can comment on these
649 \DeclareLanguageAlias {Malay}{malay}
650 \DeclareLanguageDialect{indonesian}{malay}
651 \DeclareLanguageAlias {Indonesian}{indonesian}
652 \DeclareLanguageAlias {indon}{indonesian}
653 \DeclareLanguageAlias {bahasa meyalu}{malay}
654 \DeclareLanguageAlias {Bahasa Meyalu}{bahasa melayu}
655 \DeclareLanguageAlias {bahasa}{bahasa melayu}
656 \DeclareLanguageAlias {Bahasa}{bahasa}
657 \DeclareLanguageAlias {bahasai}{bahasa}
658 \DeclareLanguageAlias {bahasam}{bahasa}
659 \DeclareLanguageAlias {Malayalam}{malayalam}
660 \DeclareLanguageAlias {Maldivian}{maldivian}
661 \DeclareLanguageAlias {divehi}{maldivian}
662 \DeclareLanguageAlias {Divehi}{divehi}
663 \DeclareLanguageAlias {Marathi}{marathi}
664 \DeclareLanguageAlias {Mongolian}{mongolian}
665 % \DeclareLanguageAlias {Syriac}{syriac}

```


5 Implementation

```

666 \DeclareLanguageAlias {norsk}{norwegian}
667 \DeclareLanguageAlias {Norsk}{norsk}
668 \DeclareLanguageAlias {Norwegian}{norwegian}
669 \DeclareLanguageAlias {nynorsk}{norwegian}
670 \DeclareLanguageAlias {Nynorsk}{nynorsk}
671 \DeclareLanguageAlias {Occitan}{occitan}
672 \DeclareLanguageAlias {lenga d'oc}{occitan}
673 \DeclareLanguageAlias {langue d'oc}{occitan}
674 \DeclareLanguageAlias {Piedmontese}{piedmontese}
675 \DeclareLanguageAlias {piemontese}{piedmontese}
676 \DeclareLanguageAlias {Piemontese}{piemontese}
677 \DeclareLanguageAlias {piemonteis}{piedmontese}
678 \DeclareLanguageAlias {Piemonteis}{piemonteis}
679 \DeclareLanguageAlias {Pinyin}{pinyin}
680 \DeclareLanguageAlias {Polish}{polish}
681 \DeclareLanguageDialect{brazil}{portuges}
682 \DeclareLanguageAlias {Brazil}{brazil}
683 \DeclareLanguageAlias {brazilian}{brazil}
684 \DeclareLanguageAlias {Brazilian}{brazilian}
685 \DeclareLanguageAlias {Portuges}{portuges}
686 \DeclareLanguageAlias {portuguese}{portuges}
687 \DeclareLanguageAlias {Portuguese}{portuguese}
688 \DeclareLanguageAlias {Romanian}{romanian}
689 \DeclareLanguageAlias {Romansh}{romansh}
690 \DeclareLanguageAlias {Romansch}{romansh}
691 \DeclareLanguageAlias {Rumantsh}{romansh}
692 \DeclareLanguageAlias {Rumantsch}{romansh}
693 \DeclareLanguageAlias {Romanche}{romansh}
694 \DeclareLanguageAlias {Russian}{russian}
695 \DeclareLanguageAlias {Samin}{samin}
696 \DeclareLanguageAlias {north sami}{samin}
697 \DeclareLanguageAlias {North Sami}{north sami}
698 \DeclareLanguageAlias {northern sami}{north sami}
699 \DeclareLanguageAlias {Northern Sami}{northern sami}
700 \DeclareLanguageAlias {Sanskrit}{sanskrit}
701 % this one isn't needed, is it:
702 % \DeclareLanguageAlias {Serbocroatian}{serbocroatian}
703 \DeclareLanguageAlias {serbo-croatian}{serbocroatian}
704 \DeclareLanguageAlias {Serbo-Croatian}{serbocroatian}
705 \DeclareLanguageDialect{serbian}{serbocroatian}
706 \DeclareLanguageAlias {Serbian}{serbian}
707 \DeclareLanguageAlias {serbianc}{serbian}
708 \DeclareLanguageDialect{croatian}{serbocroatian}
709 \DeclareLanguageAlias {Croatian}{croatian}
710 \DeclareLanguageAlias {Slovak}{slovak}
711 \DeclareLanguageAlias {Slovenian}{slovenian}
712 \DeclareLanguageAlias {Sorbian}{sorbian}
713 \DeclareLanguageDialect{lowersorbian}{sorbian}
714 \DeclareLanguageDialect{uppersorbian}{sorbian}
715 \DeclareLanguageAlias {Lowersorbian}{lowersorbian}
716 \DeclareLanguageAlias {Uppersorbian}{uppersorbian}
717 \DeclareLanguageAlias {lsorbian}{lowersorbian}

```

5 Implementation

```
718 \DeclareLanguageAlias {usorbian}{uppersorbian}
719 \DeclareLanguageAlias {lower sorbian}{lowersorbian}
720 \DeclareLanguageAlias {upper sorbian}{uppersorbian}
721 \DeclareLanguageAlias {Lower Sorbian}{lowersorbian}
722 \DeclareLanguageAlias {Upper Sorbian}{uppersorbian}
723 \DeclareLanguageAlias {Spanglish}{spanglish}
724 \DeclareLanguageAlias {Spanish}{spanish}
725 \DeclareLanguageAlias {Swedish}{swedish}
726 \DeclareLanguageDialect{swissgerman}{german}
727 % this maybe should be a language of it's own:
728 \DeclareLanguageAlias {Swiss}{swissgerman}
729 \DeclareLanguageAlias {Swissgerman}{swissgerman}
730 \DeclareLanguageAlias {Tamil}{tamil}
731 \DeclareLanguageAlias {Telugu}{telugu}
732 \DeclareLanguageAlias {Thai}{thai}
733 \DeclareLanguageAlias {thaicjk}{thai}
734 \DeclareLanguageAlias {Thaicjk}{thaicjk}
735 \DeclareLanguageAlias {Tibetan}{tibetan}
736 \DeclareLanguageAlias {Turkish}{turkish}
737 \DeclareLanguageAlias {Turkmen}{turkmen}
738 \DeclareLanguageAlias {Ukrainian}{ukrainian}
739 \DeclareLanguageAlias {Vietnamese}{vietnamese}
740 \DeclareLanguageAlias {Welsh}{welsh}
741
742 % -----
743 % load basic dictionary if available
744 \AtBeginDocument{%
745   \@trnslt@load@dictionary@silent@for
746     {translations-basic-dictionary}
747     {\@trnslt@current@language}%
748 }
749
750 \endinput
751
752 % -----
753 % HISTORY:
754 2012/09/30 v0.2beta - first version (as part of the `exsheets' bundle)
755 2012/10/05 v0.2    - \LoadDictionary and \LoadDictionaryFor added and loads of
756                   languages defined.
757 2013/03/10 v0.8    - basic dictionaries for English, German, French and Spanish
758                   - new command \DeclareDictTranslation
759 2013/04/04 v0.8a   - bug fix in \DeclareDictTranslation
760 2013/04/07 v0.9    - slightly improved messages
761 2013/04/08 v0.9a   - changed fallback warning into info
762                   - synchronized version number with `exsheets' until now but
763                   won't any more
764 2013/06/22 v0.9b   - added Swiss
765 2013/06/28 v0.10   - declaring aliases of dialects now works as expected
766                   - declarings dialects of an alias now correctly declares
767                   the dialect to the correct base language
768                   - corrected a few erroneous language declarations
769 2013/07/12 v0.10a  - \GetTranslation gets two-folded fallback: use
```

5 Implementation

770 fallback-translation if no translation for the current
771 language has been defined; use literal string if /no/
772 language is used - this should never happen but /will/
773 happen if neither `babel' nor `polyglossia' have been
774 loaded, i.e., no language has been chosen /and/ the
775 package writer did not provide an English translation
776 2013/07/16 v1.0 - removed from `exsheets' bundle - `translations' should
777 be a package of it's own
778 - load basic dictionary automatically if available
779 - rudimentary check in \LoadDictionary if loaded file is a
780 dictionary
781 - new command \PrintDictionaryFor
782 - redefined conditionals; they still seemed to make
783 trouble in some cases
784 2013/07/18 v1.0a - added /loads/ of languages, noe the list of babel and
785 polyglossia languages hopefully is complete
786 - a few languages had falsely been declared as dialect
787 instead of an alias

Index

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

B			
babel	1–4, 6	\GetTranslationFor	3
beamer	1	\GetTranslationForWarn	4
		\GetTranslationWarn	4
C		I	
\captions<language>	1	Implementation	12
D		L	
\DeclareDictTranslation	4	Languages	9–12
\DeclareLanguage	3f.	\LoadDictionary	4
\DeclareLanguageAlias	3	\LoadDictionaryFor	4
\DeclareLanguageDialect	3	M	
\DeclareTranslation	3, 5	Motivation	1
\DeclareTranslationFallback	3, 6	N	
Dictionaries	6–9	\NewTranslation	3
basic	7	P	
own	6	polyglossia	1f., 6
E		\PrintDictionaryFor	4
etoolbox	2	\providecaptionname	1
G		\ProvideDictionaryFor	4
\GetTranslation	2f.		
		R	
		\RenewTranslation	3
		Requirements	2
		S	
		\SaveTranslation	4
		\SaveTranslationFor	4
		scrfile	2
		T	
		\translate	1
		translator	1
		U	
		Usage	2–9
		Background	2
		Commands	3
		Dictionaries	6
		Example	4
		Packages	5
		fallback	6
		structure	5