

TRANSLATIONS

v1.1a 2013/09/30

Internationalization of L^AT_EX 2_ε Packages

Clemens NIEDERBERGER

<https://github.com/cgnieder/translations/>

contact@mychemistry.eu

Table of Contents

1	Motivation	1			
2	License and Requirements	2			
3	Usage	2			
3.1	Background	2			
3.2	Available Commands	2			
3.3	A Small Example	5			
3.4	Usage in Packages	6			
3.4.1	Basic Structure	6			
3.4.2	The ‘fallback’ language	7			
3.5	Dictionaries	7			
3.5.1	Own Dictionaries	7			
			3.5.2	TRANSLATIONS’ Basic Dictionaries	8
			4	Defined Languages	11
			4.1	Base Languages	11
			4.2	Language Dialects	11
			4.3	Language Aliases	12
			5	Implementation	14
				Bibliography	32
				Index	33

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 bundle). Internationalization is also possible with babel [Bra13] and it’s `\addto\captions⟨language⟩` mechanism or KOMA-Script’s `\providecaptionname` and similar commands. However, I believe that **TRANSLATIONS** is more flexible than all of these. Unlike translator it detects the used (babel or polyglossia [Cha13]) 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.

2 License and Requirements

Permission is granted to copy, distribute and/or modify this software under the terms of the L^AT_EX Project Public License (LPPL), version 1.3 or later (<http://www.latex-project.org/lppl.txt>). The software has the status “maintained.”

TRANSLATIONS requires the packages etoolbox¹ [Leh11] and scrfile (part of the KOMA-Script bundle² [KN12]).

3 Usage

3.1 Background

The **TRANSLATIONS** package enables the author of a package or a class (or a document) to declare translations of key words in different languages 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 (although possible) 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 11), language dialects (see table 3 on page 12), and language aliases (see table 4 on page 12). For the commands declaring or fetching a translation base languages and language aliases are equivalent. Dialects are similar to aliases but there are 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.

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).
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. Commands without the marker aren’t expandable.

1. on CTAN as etoolbox: <http://mirrors.ctan.org/macros/latex/contrib/etoolbox/>

2. on CTAN as koma-script: <http://mirrors.ctan.org/macros/latex/contrib/koma-script/>

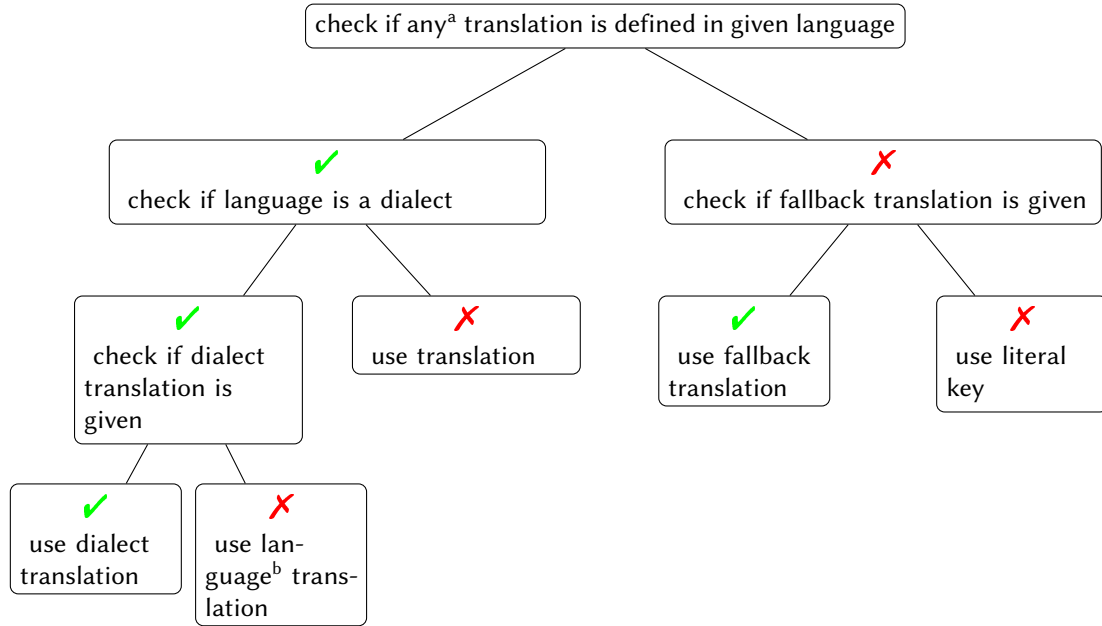


FIGURE 1: Schematic representation of **TRANSLATIONS**' translating mechanism. Notes: ^a except for a possible fallback translation. ^b *i. e.*, the base language of the dialect.

\DeclareLanguage{ $\langle lang \rangle$ }

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. It should never be necessary to use this command as **TRANSLATIONS** already declares loads of languages (section 4). Should you miss one please send me an email and I'll add it to **TRANSLATIONS**.

\DeclareLanguageAlias{ $\langle lang_2 \rangle$ }{ $\langle lang_1 \rangle$ }

Declares $\langle lang_2 \rangle$ to be an alias of $\langle lang_1 \rangle$. If $\langle lang_1 \rangle$ doesn't exist yet a warning will be raised and it will be defined. This command can only be used in the preamble. It should never be necessary to use this command as **TRANSLATIONS** already declares loads of languages (section 4). Should you miss one please send me an email and I'll add it to **TRANSLATIONS**.

\DeclareLanguageDialect{ $\langle dialect \rangle$ }{ $\langle lang \rangle$ }

Declares $\langle dialect \rangle$ to be a dialect of language $\langle lang \rangle$. If a translation for $\langle dialect \rangle$ is provided it is used by the translation macros. If there is none the corresponding translation for $\langle lang \rangle$ is used instead. It should never be necessary to use this command as **TRANSLATIONS** already declares loads of languages (section 4). Should you miss one please send me an email and I'll add it to **TRANSLATIONS**.

\NewTranslation{ $\langle lang \rangle$ }{ $\langle key \rangle$ }{ $\langle translation \rangle$ }

Defines a translation of key $\langle key \rangle$ for the language $\langle lang \rangle$. An error will be raised if a translation of $\langle key \rangle$ 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.

* `\GetLCTranslationFor{⟨lang⟩}{⟨key⟩}`

Introduced in
version 1.1

Fetches and prints the translation of `⟨key⟩` for the language `⟨lang⟩`. This command ensures that the fetched translation is set lowercase. This command is expandable.

* `\GetLCTranslation{⟨key⟩}`

Introduced in
version 1.1

Fetches and prints the translation of `⟨key⟩` for the currently active language (as for example set by babel). This command ensures that the fetched translation is set lowercase. 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 at the cost of expandability.

`\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 at the cost of expandability.

`\GetLCTranslationForWarn{⟨lang⟩}{⟨key⟩}`

Introduced in
version 1.1

Fetches and prints the translation of `⟨key⟩` for the language `⟨lang⟩`. This command ensures that the fetched translation is set lowercase. Issues a warning if no translation is available at the cost of expandability.

`\GetLCTranslationWarn{⟨key⟩}`

Introduced in
version 1.1

Fetches and prints the translation of `⟨key⟩` for the currently active language (as for example set by babel). This command ensures that the fetched translation is set lowercase. Issues a warning if no translation is available at the cost of expandability.

\SaveTranslationFor{ $\langle cmd \rangle$ }{ $\langle lang \rangle$ }{ $\langle key \rangle$ }

Fetches and saves the translation of $\langle key \rangle$ for the language $\langle lang \rangle$ in the macro $\langle cmd \rangle$.

\SaveTranslation{ $\langle cmd \rangle$ }{ $\langle key \rangle$ }

Fetches and saves the translation of $\langle key \rangle$ for the currently active language (as for example set by `babel`) in the macro $\langle cmd \rangle$.

\LoadDictionary{ $\langle name \rangle$ }

Loads a file named $\langle name \rangle$ - $\langle lang \rangle$.`trsl` where $\langle lang \rangle$ corresponds to the lowercase name of the current language as defined with `\DeclareLanguage`. This file should contain the translations for the specified language.

\LoadDictionaryFor{ $\langle lang \rangle$ }{ $\langle name \rangle$ }

Loads a file named $\langle name \rangle$ - $\langle lang \rangle$.`trsl`.

\DeclareDictTranslation{ $\langle key \rangle$ }{ $\langle translation \rangle$ }

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{ $\langle lang \rangle$ }{ $\langle name \rangle$ }[$\langle date \rangle$]

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**{ $\langle lang \rangle$ }{ $\langle name \rangle$ }{ $\langle pre \rangle$ }{ $\langle mid \rangle$ }{ $\langle post \rangle$ }

Introduced in
version 1.0

Prints all entries of dictionary $\langle name \rangle$ in language $\langle lang \rangle$ in the order the entries have been declared. For every entry the code

$\langle pre \rangle \langle key \rangle \langle mid \rangle \langle translation \rangle \langle post \rangle$

is printed. The dictionary must have been loaded of course. This may be of minor use but was used to print table 1.

* **\baseLanguage**{ $\langle lang \rangle$ }

Introduced in
version 1.1

Returns the (internal) base name of the given language, language alias or language dialect. For a base language (see section 4.1) this usually simply is the lowercase version of the name.

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\{"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 fall-back 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

continues

key	translation
encl	Anlage(n)
Figures	Abbildungen
figures	Abbildungen
Figure	Abbildung
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

continues

key	translation
See also	Siehe auch
see also	siehe auch
See	Siehe
see	siehe
Sketch of Proofs	Beweisskizzen
Sketch of proofs	Beweisskizzen
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
Monday	Montag
Tuesday	Dienstag
Wednesday	Mittwoch
Thursday	Donnerstag
Friday	Freitag
Saturday	Samstag
Sunday	Sonntag
January	Januar
February	Februar
March	März
April	April
May	Mai
June	Juni
July	Juli
August	August
September	September
October	Oktober
November	November
December	Dezember

4 Defined Languages

4.1 Base 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. Tables 2, 3 and 4 list *all* language names known to **TRANSLATIONS**. However, they're not sorted alphabetically but listed in the order they have been defined. I tried to make the definitions in an alphabetical order but sometimes rather grouped related language names together.

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	spanglish	spanish	swedish	tamil
telugu	thai	tibetan	turkish	turkmen
ukrainian	vietnamese	welsh	canadien	acadian
american	australian	british	canadian	newzealand
irish	scottish	austrian	hindi	urdu
indonesian	brazil	serbian	croatian	lowersorbian
uppersorbian	swissgerman			

4.2 Language Dialects

TRANSLATIONS also defines a few dialects of the base languages. They are listed in table 3. The decision what is a dialect and what is an alias is not always clear. I am no linguist so I looked up information available on the internet. A language that was described as “standardized register” was always defined as a dialect. For some other languages it seemed to make sense, such as British or Austrian. The decisions are open for debate.

4 Defined Languages

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		

4.3 Language Aliases

To most of the base 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. For a number of languages aliases were defined in order to match babel’s or polyglossia’s names for the languages. Others are defined because there apparently exist more than one name for the same language. The decisions are not consistent. For example it could be argued that “deutsch” is an alias of “German”. I am open to suggestions and improvements. 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
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	frenchle	french

continues

4 Defined Languages

alias	language	alias	language
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 meyalu	bahasa	bahasa meyalu
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

continues

5 Implementation

alias	language	alias	language
Brazil	brazil	brazilian	brazil
Brazilian	brazilian	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	lsorbian	lowersorbian
usorbian	uppersorbian	lower sorbian	lowersorbian
upper sorbian	uppersorbian	Lower Sorbian	lowersorbian
Upper Sorbian	uppersorbian	Spanglish	spanglish
Spanish	spanish	Swedish	swedish
swiss	swissgerman	Swiss	swiss
Swissgerman	swissgerman	swiss german	swissgerman
Swiss German	swiss german	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

```

1 % -----
2 % the TRANSLATIONS package
3 %
4 %   a simple translator
5 %
6 % -----
7 % Clemens Niederberger
8 % Web:   https://github.com/cgnieder/translations

```

5 Implementation

```
9 % E-Mail: contact@mychemistry.eu
10 % -----
11 % Copyright 2012 Clemens Niederberger
12 %
13 % This work may be distributed and/or modified under the
14 % conditions of the LaTeX Project Public License, either version 1.3
15 % of this license or (at your option) any later version.
16 % The latest version of this license is in
17 %   http://www.latex-project.org/lppl.txt
18 % and version 1.3 or later is part of all distributions of LaTeX
19 % version 2005/12/01 or later.
20 %
21 % This work has the LPPL maintenance status 'maintained'.
22 %
23 % The Current Maintainer of this work is Clemens Niederberger.
24 % -----
25 % If you have any ideas, questions, suggestions or bugs to report, please
26 % feel free to contact me.
27 % -----
28 \def\@trnslt@date{2013/09/30}
29 \def\@trnslt@version{v1.1a}
30 \def\@trnslt@info{internationalization of LaTeX2e packages}
31
32 \ProvidesPackage{translations}[\@trnslt@date\space \@trnslt@version\space \
  \@trnslt@info\space (CN)]
33 \RequirePackage{etoolbox,scrfile}
34
35 % -----
36 % message handling
37 \def\@trnslt@error@message{%
38   For details have a look at the 'translations' manual.}
39
40 \def\@trnslt@create@message#1{%
41   \ifstrequal{#1}{Error}
42     {%
43       \lowercase{\csdef{\@trnslt@#1}}##1{%
44         \csuse{Package#1}{translations}{##1}{\@trnslt@error@message}}%
45     }{%
46       \lowercase{\csdef{\@trnslt@#1}}##1{%
47         \csuse{Package#1}{translations}{##1}}%
48     }
49   \@trnslt@create@message{Error}
50   \@trnslt@create@message{Warning}
51   \@trnslt@create@message{WarningNoLine}
52   \@trnslt@create@message{Info}
53
54   \def\@trnslt@err@unknown@lang#1{%
55     \@trnslt@error{Unknown language '#1'}}
56
```

5 Implementation

```

57 \def\@trnslt@warn@unknown@lang#1{%
58   \@trnslt@warning{Unknown language ‘#1’}}
59
60 \def\@trnslt@err@already@defined#1#2{%
61   \@trnslt@error{The #2 translation for ‘#1’ is already defined.}}
62
63 \def\@trnslt@err@not@defined#1#2{%
64   \@trnslt@error{The \@trnslt@language{#2} translation for ‘#1’ is not defined yet
65   .}}
66 % -----
67 % catcode stuff:
68 \DeclareListParser*{\@trnslt@map@tokenlist}{}
69
70 \def\@trnslt@save@catcodes#1{%
71   \@trnslt@map@tokenlist\@trnslt@save@catcode{#1}%
72 }
73 \def\@trnslt@save@catcode#1{%
74   \csedef{\@trnslt@catcode@#1}{\the\catcode‘#1}%
75 }
76
77 \def\@trnslt@restore@catcodes#1{%
78   \@trnslt@map@tokenlist\@trnslt@restore@catcode{#1}%
79 }
80 \protected\def\@trnslt@restore@catcode#1{%
81   \catcode‘#1=\csname \@trnslt@catcode@#1\endcsname
82 }
83
84 \def\@trnslt@set@catcodes#1#2{%
85   \@trnslt@map@tokenlist{\@trnslt@set@catcode{#2}}{#1}%
86 }
87 \def\@trnslt@set@catcode#1#2{%
88   \catcode‘#2=#1\relax
89 }
90
91 % -----
92 % check if babel or polyglossia is used
93 \AtEndPreamble{
94   \@ifpackageloaded{babel}{}{
95     \@ifpackageloaded{polyglossia}{}
96     {\@trnslt@warning{No language package found. I am going to use ‘english’
97       as default language.}}
98   }
99   \ifdef\language{}
100     {\def\language{english}}
101   \def\@trnslt@current@language{\language}
102   \ifdef\bbl@afterfi{}
103     {\long\def\bbl@afterfi#1\fi{\fi#1}}
104 }

```


5 Implementation

```
105
106 % -----
107 % book keeping: the following macros will be used as 'etoolbox' lists that
108 % keep record of defined languages, dialects and aliases
109 \def\@trnslt@languages{}% all languages
110 \def\@trnslt@aliases@pair{}% all aliases and their base
111 \def\@trnslt@aliases@single{}% all aliases
112 \def\@trnslt@dialects@pair{}% all dialects and their base
113 \def\@trnslt@dialects@single{}% all dialects
114
115 % -----
116 % define \@trnslt@if@<name> conditionals that don't leave the checked macro as
117 % \relax behind and check for \@trnslt@<name>@#1. These conditionals should
118 % also be expandable in an \edef-like context. Thanks to e-TeX there's
119 % \ifcsname:
120 \def\@trnslt@new@check#1{%
121   \csdef{@trnslt@if@#1}##1{%
122     \ifcsname @trnslt@#1@##1\endcsname
123       \expandafter\@firstoftwo
124     \else
125       \expandafter\@secondoftwo
126     \fi
127   }%
128 }
129
130 % -----
131 \long\def\@trnslt@get@lowercase#1{%
132   \lowercase\expandafter{\romannumeral-'\Q#1}%
133 }
134
135 % -----
136 % \DeclareLanguage
137 % #1: language
138 \newrobustcmd*\DeclareLanguage[1]{%
139   \@trnslt@declare@language{#1}}
140 \onlypreamble\DeclareLanguage
141
142 \def\@trnslt@declare@language#1{%
143   \@trnslt@if@language{#1}
144   {}{%
145     \csdef{@trnslt@language@#1}{#1}%
146     \listead\@trnslt@languages{#1}%
147   }%
148 }
149
150 \def\@trnslt@language#1{%
151   \csuse{@trnslt@language@#1}}
152
153 \newcommand*\baselanguage[1]{\@trnslt@language{#1}}
```

5 Implementation

```

154
155 % define \@trnslt@if@language{#1}{}{} that actually checks the existence of
156 % \@trnslt@language@#1:
157 \@trnslt@new@check{language}
158
159 % -----
160 % \DeclareLanguageDialect
161 % #1: dialect
162 % #2: language
163 \newrobustcmd*\DeclareLanguageDialect[2]{%
164   \@trnslt@declare@language@dialect{#1}{#2}}
165 \onlypreamble\DeclareLanguageDialect
166
167 \def\@trnslt@declare@language@dialect#1#2{%
168   \@trnslt@if@language{#2}
169   {%
170     \@trnslt@warn@unknown@lang{#2}%
171     \@trnslt@declare@language{#2}%
172   }%
173   \@trnslt@if@dialect{#1}
174   {% => ist schon als dialect definiert => irgendwelche weiteren checks?
175   }
176   {%
177     \@trnslt@if@alias{#2}
178     {%
179       \csedef{\@trnslt@dialect@#1}{\@trnslt@alias{#2}}{#1}}%
180       \@trnslt@declare@language{#1}%
181       \listeaddd\@trnslt@dialects@single{#1}%
182       \listeaddd\@trnslt@dialects@pair{#1}{\@trnslt@alias{#2}}}%
183     }
184     {%
185       \csedef{\@trnslt@dialect@#1}{#2}{#1}}%
186       \@trnslt@declare@language{#1}%
187       \listeaddd\@trnslt@dialects@single{#1}%
188       \listeaddd\@trnslt@dialects@pair{#1}{#2}}}%
189   }%
190 }%
191 }
192
193 \def\@trnslt@dialect#1{%
194   \csuse{\@trnslt@dialect@#1}}
195
196 % this macros fetches the base language for a given dialect, expandably:
197 \def\@trnslt@dialect@of#1{%
198   \@trnslt@if@dialect{#1}
199   {%
200     \expandafter\expandafter\expandafter
201     \@firstoftwo
202     \csname \@trnslt@dialect@#1\endcsname

```

5 Implementation

```

203     }{}%
204 }
205
206 % define \@trnslt@if@dialect{#1}{}{} that actually checks the existence of
207 % \@trnslt@dialect@#1:
208 \@trnslt@new@check{dialect}
209
210 % -----
211 % \DeclareLanguageAlias
212 % #1: alias
213 % #2: language
214 \newrobustcmd*\DeclareLanguageAlias[2]{%
215   \@trnslt@declare@languagealias{#1}{#2}}
216 \onlypreamble\DeclareLanguageAlias
217
218 \def\@trnslt@declare@languagealias#1#2{%
219   \@trnslt@if@language{#2}
220     {}{%
221       \@trnslt@warn@unknown@lang{#2}%
222       \@trnslt@declare@language{#2}%
223     }%
224   \csletcs{@trnslt@language@#1}{@trnslt@language@#2}%
225   \@trnslt@if@dialect{#2}
226     {\csletcs{@trnslt@dialect@#1}{@trnslt@dialect@#2}}
227     {}%
228   \ifinlist{#1}\@trnslt@aliases@single
229     {}{%
230       \csdef{@trnslt@alias@#1}{#2}%
231       \listadd\@trnslt@aliases@pair{{#1}{#2}}%
232       \listadd\@trnslt@aliases@single{#1}%
233     }%
234 }
235
236 \def\@trnslt@alias#1{%
237   \csuse{@trnslt@alias@#1}}
238
239 % define \@trnslt@if@alias{#1}{}{} that actually checks the existence of
240 % \@trnslt@alias@#1:
241 \@trnslt@new@check{alias}
242
243 % -----
244 % dummy language: 'fallback':
245 \DeclareLanguage{fallback}
246 \DeclareLanguageAlias{Fallback}{fallback}
247
248 % -----
249 % \DeclareTranslation, \NewTranslation and \RenewTranslation
250 % #1: language
251 % #2: word

```

5 Implementation

```
252 % #3: replacement
253 \newrobustcmd* \DeclareTranslation[3]{%
254   \@trnslt@declare@translation{#2}{#1}{#3}}
255 \onlypreamble \DeclareTranslation
256
257 \newrobustcmd* \DeclareTranslationFallback[2]{%
258   \@trnslt@declare@translation{#1}{fallback}{#2}}
259 \onlypreamble \DeclareTranslationFallback
260
261 \newrobustcmd* \NewTranslation[3]{%
262   \@trnslt@new@translation{#2}{#1}{#3}}
263 \onlypreamble \NewTranslation
264
265 \newrobustcmd* \RenewTranslation[3]{%
266   \@trnslt@renew@translation{#2}{#1}{#3}}
267 \onlypreamble \RenewTranslation
268
269 % #1: word
270 % #2: language
271 % #3: replacement
272 \def\@trnslt@declare@translation#1#2#3{%
273   \@trnslt@if@language{#2}
274   {%
275     \@trnslt@if@dialect{#2}
276     {%
277       \csdef{@trnslt@word@#1@\@trnslt@dialect{#2}}{#3}%
278       \@trnslt@if@word\@trnslt@dialect@of{#1}{#2}
279       {}
280       {\csdef{@trnslt@word@#1@\@trnslt@dialect@of{#2}}{#3}}%
281     }
282     {\csdef{@trnslt@word@#1@\@trnslt@language{#2}}{#3}}%
283     % save the <word> as <word>:
284     \csdef{@trnslt@word@#1@literal}{#1}%
285   }
286   {\@trnslt@err@unknown@lang{#2}}%
287 }
288
289 \def\@trnslt@new@translation#1#2#3{%
290   \@trnslt@if@word\@trnslt@language{#1}{#2}
291   {\@trnslt@err@already@defined{#1}{#2}}
292   {\@trnslt@declare@translation{#1}{#2}{#3}}
293
294 \def\@trnslt@renew@translation#1#2#3{%
295   \@trnslt@if@word\@trnslt@language{#1}{#2}
296   {\@trnslt@declare@translation{#1}{#2}{#3}}
297   {\@trnslt@err@not@defined{#1}{#2}}
298
299 % -----
300 % now let's go through some trouble to check if a translation exists:
```

5 Implementation

```

301 \def\@trnslt@if@word#1#2#3{%
302   \ifcsname @trnslt@word@#2@#1{#3}\endcsname
303     \expandafter\@firstoftwo
304   \else
305     \expandafter\@secondoftwo
306   \fi
307 }%
308
309 \def\@trnslt@if@translation#1#2{%
310   \@trnslt@if@word\@trnslt@language{#1}{#2}
311     {\expandafter\@firstoftwo}
312     {%
313       \@trnslt@if@dialect{#2}
314         {%
315           \@trnslt@if@word\@trnslt@dialect{#1}{#2}
316             {\expandafter\@firstoftwo}
317             {%
318               \@trnslt@if@word\@trnslt@dialect@of{#1}{#2}
319                 {\expandafter\@firstoftwo}
320                 {\expandafter\@secondoftwo}%
321             }
322         }
323     {\expandafter\@secondoftwo}%
324 }%
325 }
326
327 % -----
328 % \GetTranslationFor and \GetTranslation
329 % these need to be expandable!
330 % #1: language
331 % #2: word
332 \newcommand*\GetTranslationFor[2]{%
333   \@trnslt@checkandget@translation@for{#2}{#1}}
334
335 \newcommand*\GetTranslation[1]{%
336   \@trnslt@checkandget@translation@for{#1}{\@trnslt@current@language}}
337
338 \newcommand*\GetLCTranslationFor[2]{%
339   \@trnslt@get@lowercase{\@trnslt@checkandget@translation@for{#2}{#1}}%
340 }
341
342 \newcommand*\GetLCTranslation[1]{%
343   \@trnslt@get@lowercase{%
344     \@trnslt@checkandget@translation@for{#1}{\@trnslt@current@language}%
345   }%
346 }
347
348 % unexpandable version of the commands that raise a warning if no translation
349 % is available:

```

5 Implementation

```

350 \newcommand*\GetTranslationForWarn[2]{%
351   \@trnslt@getandwarn@translation@for{#2}{#1}}
352
353 \newcommand*\GetTranslationWarn[1]{%
354   \@trnslt@getandwarn@translation@for{#1}{\@trnslt@current@language}}
355
356 \newcommand*\GetLCTranslationForWarn[2]{%
357   \@trnslt@getandwarn@lctranslation@for{#2}{#1}}
358
359 \newcommand*\GetLCTranslationWarn[1]{%
360   \@trnslt@getandwarn@lctranslation@for{#1}{\@trnslt@current@language}}
361
362 % #1: word #2: language
363 \def\@trnslt@get@translation@for#1#2{%
364   \@trnslt@if@dialect{#2}
365   {%
366     \ifcsdef{@trnslt@word@#1@\@trnslt@dialect{#2}}
367       {\csuse{@trnslt@word@#1@\@trnslt@dialect{#2}}}
368       {\csuse{@trnslt@word@#1@\@trnslt@dialect@of{#2}}}%
369     }
370   {\csuse{@trnslt@word@#1@\@trnslt@language{#2}}}%
371 }
372
373 \def\@trnslt@checkandget@translation@for#1#2{%
374   \@trnslt@if@translation{#1}{#2}
375   {\@trnslt@get@translation@for{#1}{#2}}
376   {%
377     \@trnslt@if@translation{#1}{fallback}
378     {\csuse{@trnslt@word@#1@fallback}}
379     {\csuse{@trnslt@word@#1@literal}}%
380   }%
381 }
382
383 % this is not expandable!
384 \protected\def\@trnslt@getandwarn@translation@for#1#2{%
385   \@trnslt@if@translation{#1}{#2}
386   {\@trnslt@get@translation@for{#1}{#2}}
387   {%
388     \@trnslt@warning{Translation for ‘#1’ in #2 unknown. You may try to use
389       \string\DeclareTranslation{#2}{#1}{ ... } in your preamble.}%
390     \@trnslt@if@translation{#1}{fallback}
391     {%
392       \@trnslt@info{Using fallback translation for ‘#1’}%
393       \csuse{@trnslt@word@#1@fallback}
394     }
395     {\csuse{@trnslt@word@#1@literal}}%
396   }%
397 }
398

```

5 Implementation

```

399 % lowercase version for translation with warnings:
400 \protected\def\@trnslt@getandwarn@lctranslation@for#1#2{%
401   \@trnslt@if@translation{#1}{#2}
402     {\@trnslt@get@lowercase{\@trnslt@get@translation@for{#1}{#2}}}
403     {%
404       \@trnslt@warning{Translation for ‘#1’ in #2 unknown. You may try to use
405         \string\DeclareTranslation{#2}{#1}{ ... } in your preamble.}%
406       \@trnslt@if@translation{#1}{fallback}
407       {%
408         \@trnslt@info{Using fallback translation for ‘#1’}%
409         \@trnslt@get@lowercase{\csuse{@trnslt@word@#1@fallback}}%
410       }
411       {\@trnslt@get@lowercase{\csuse{@trnslt@word@#1@literal}}}%
412     }%
413 }
414
415 % -----
416 % \SaveTranslationFor and \SaveTranslation
417 \newrobustcmd*\SaveTranslationFor[3]{%
418   \@trnslt@save@translation@for{#1}{#3}{#2}}
419
420 \newrobustcmd*\SaveTranslation[2]{%
421   \@trnslt@save@translation@for{#1}{#2}{\@trnslt@current@language}}
422
423 \def\@trnslt@save@translation@for#1#2#3{%
424   \edef#1{%
425     \@trnslt@if@translation{#2}{#3}
426     {\csuse{@trnslt@word@#2@\@trnslt@language{#3}}}
427     {}%
428   }}
429
430 % -----
431 % \LoadDictionary and \LoadDictionaryFor
432 \newrobustcmd*\LoadDictionary[1]{%
433   \@trnslt@load@dictionary@for{#1}{\@trnslt@current@language}}
434 \onlypreamble\LoadDictionary
435
436 \newrobustcmd*\LoadDictionaryFor[2]{%
437   \@trnslt@load@dictionary@for{#2}{#1}}
438 \onlypreamble\LoadDictionaryFor
439
440 % #1: name
441 % #2: lang
442 \def\@trnslt@load@dictionary@for#1#2{%
443   \AtBeginDocument{%
444     \InputIfFileExists{#1-\@trnslt@language{#2}.trsl}
445     {\@trnslt@check@dictionary{#1}{#2}}
446     {\@trnslt@warning{dictionary file ‘#1-\@trnslt@language{#2}.trsl’ not
447       found.}}%

```

5 Implementation

```

448 }}
449
450 \def\@trnslt@check@dictionary#1#2{%
451   \AfterFile{#1-\@trnslt@language{#2}.trsl}
452   {%
453     \ifcsdef{@trnslt@dictionary@#1@\@trnslt@language{#2}}
454       {\@trnslt@info{loading dictionary '#1' for '#2'.}}
455       {%
456         \@trnslt@warning{file '#1-\@trnslt@language{#2}.trsl' does not
457           appear to be a dictionary}%
458       }%
459   }%
460 }
461
462 \def\@trnslt@if@dictionary#1#2{%
463   \IfFileExists{#1-#2.trsl}}
464
465 \def\@trnslt@load@dictionary@silent@for#1#2{%
466   \InputIfFileExists{#1-#2.trsl}{}{}}
467
468 \newrobustcmd*\ProvideDictionaryFor[2]{%
469   \@trnslt@provide@dictionary@for{#1}{#2}}
470 \onlypreamble\ProvideDictionaryFor
471
472 \def\@trnslt@provide@dictionary@for#1#2{%
473   \def\@trnslt@dictionary@name{#2}%
474   \edef\@trnslt@dictionary@lang{\@trnslt@language{#1}}%
475   % this macro can be used to check if we have a dictionary and will also be
476   % used as a list for the dictionary entries:
477   \csdef{@trnslt@dictionary@\@trnslt@dictionary@name @\@trnslt@dictionary@lang}{}%
478   \@ifnextchar[
479     {\@trnslt@provide@dictionary@version}
480     {
481       \ProvidesFile
482         {#2-\@trnslt@dictionary@lang.trsl}%
483         [(\@trnslt@dictionary@lang\space translation file '#2')]
484     }%
485 }
486
487 \def\@trnslt@provide@dictionary@version[#1]{%
488   \ProvidesFile
489     {\@trnslt@dictionary@name-\@trnslt@dictionary@lang.trsl}%
490     [(\@trnslt@dictionary@lang\space translation file '\@trnslt@dictionary@name'
491       #1)]
492 % \@trnslt@dictionary@language
493 \newrobustcmd*\DeclareDictTranslation[2]{%
494   \listcsadd
495     {@trnslt@dictionary@\@trnslt@dictionary@name @\@trnslt@dictionary@lang}

```


5 Implementation

```
496     {{#1}{#2}}}%
497     \@trnslt@declare@translation{#1}{\@trnslt@dictionary@lang}{#2}%
498 }
499 \@onlypreamble\DeclareDictTranslation
500
501 % \PrintDictionaryFor
502 % #1: lang
503 % #2: name
504 % #3: pre
505 % #4: mid
506 % #5: post
507 \newcommand*\PrintDictionaryFor[5]{%
508     \@trnslt@print@dictionary@for{#1}{#2}{#3}{#4}{#5}}
509
510 % #1: lang
511 % #2: name
512 % #3: pre
513 % #4: mid
514 % #5: post
515 \def\@trnslt@print@dictionary@for#1#2#3#4#5{%
516     \forlistcsloop
517         {\@trnslt@print@dictionary@entry{#3}{#4}{#5}}
518         {\@trnslt@dictionary@#2@\@trnslt@language{#1}}%
519 }
520
521 % #1: pre
522 % #2: mid
523 % #3: post
524 % #4: {key}{translation}
525 \def\@trnslt@print@dictionary@entry#1#2#3#4{%
526     \@trnslt@print@dictionary@entry@aux{#1}{#2}{#3}{#4}
527
528 % #1: pre
529 % #2: mid
530 % #3: post
531 % #4: key
532 % #5: translation
533 \def\@trnslt@print@dictionary@entry@aux#1#2#3#4#5{#1#4#2#5#3}
534
535 % -----
536 % predefined languages
537 \DeclareLanguage{afrikaans}
538 \DeclareLanguage{albanian}
539 \DeclareLanguage{amharic}
540 \DeclareLanguage{arabic}
541 \DeclareLanguage{armenian}
542 \DeclareLanguage{asturian}
543 \DeclareLanguage{basque}
544 \DeclareLanguage{bengali}
```

5 Implementation

```
545 \DeclareLanguage{breton}
546 \DeclareLanguage{bulgarian}
547 \DeclareLanguage{catalan}
548 \DeclareLanguage{coptic}
549 \DeclareLanguage{czech}
550 \DeclareLanguage{danish}
551 \DeclareLanguage{dutch}
552 \DeclareLanguage{english}
553 \DeclareLanguage{esperanto}
554 \DeclareLanguage{estonian}
555 \DeclareLanguage{ethiop}
556 \DeclareLanguage{farsi}
557 \DeclareLanguage{finnish}
558 \DeclareLanguage{french}
559 \DeclareLanguage{friulan}
560 \DeclareLanguage{gaelic}
561 \DeclareLanguage{galician}
562 \DeclareLanguage{german}
563 \DeclareLanguage{greek}
564 \DeclareLanguage{hebrew}
565 \DeclareLanguage{hindustani}
566 \DeclareLanguage{hungarian}
567 \DeclareLanguage{icelandic}
568 \DeclareLanguage{interlingua}
569 \DeclareLanguage{italian}
570 \DeclareLanguage{japanese}
571 \DeclareLanguage{kannada}
572 \DeclareLanguage{ladin}
573 \DeclareLanguage{lao}
574 \DeclareLanguage{latin}
575 \DeclareLanguage{latvian}
576 \DeclareLanguage{lithuanian}
577 \DeclareLanguage{malay}
578 \DeclareLanguage{malayalam}
579 \DeclareLanguage{maldivian}
580 \DeclareLanguage{marathi}
581 \DeclareLanguage{mongolian}
582 % polyglossia seems to support this one but it is unclear which language is
583 % actually meant by it:
584 % \DeclareLanguage{nko}
585 \DeclareLanguage{norwegian}
586 \DeclareLanguage{occitan}
587 \DeclareLanguage{piedmontese}
588 \DeclareLanguage{pinyin}
589 \DeclareLanguage{polish}
590 \DeclareLanguage{portuges}
591 \DeclareLanguage{romanian}
592 \DeclareLanguage{romansh}
593 \DeclareLanguage{russian}
```

5 Implementation

```
594 \DeclareLanguage{samin}
595 \DeclareLanguage{sanskrit}
596 \DeclareLanguage{serbocroatian}
597 \DeclareLanguage{slovak}
598 \DeclareLanguage{slovenian}
599 \DeclareLanguage{sorbian}
600 % not sure about this: isn't it either a Spanish or English dialect?
601 \DeclareLanguage{spanglish}
602 \DeclareLanguage{spanish}
603 \DeclareLanguage{swedish}
604 % polyglossia seems to support this one but it is unclear which language is
605 % actually meant by it:
606 % \DeclareLanguage{syriac}
607 \DeclareLanguage{tamil}
608 \DeclareLanguage{telugu}
609 \DeclareLanguage{thai}
610 \DeclareLanguage{tibetan}
611 \DeclareLanguage{turkish}
612 \DeclareLanguage{turkmen}
613 \DeclareLanguage{ukrainian}
614 \DeclareLanguage{vietnamese}
615 \DeclareLanguage{welsh}
616
617 % -----
618 % aliases and dialects:
619 \DeclareLanguageAlias {Afrikaans}{afrikaans}
620 \DeclareLanguageAlias {Albanian}{albanian}
621 \DeclareLanguageAlias {Amharic}{amharic}
622 \DeclareLanguageAlias {Arabic}{arabic}
623 \DeclareLanguageAlias {Armenian}{armenian}
624 \DeclareLanguageAlias {Asturian}{asturian}
625 \DeclareLanguageAlias {astur-leonese}{asturian}
626 \DeclareLanguageAlias {Astur-Leonese}{astur-leonese}
627 \DeclareLanguageAlias {asturian-leonese}{asturian}
628 \DeclareLanguageAlias {Asturian-Leonese}{asturian-leonese}
629 \DeclareLanguageAlias {Basque}{basque}
630 \DeclareLanguageAlias {Bengali}{bengali}
631 \DeclareLanguageAlias {Breton}{breton}
632 \DeclareLanguageAlias {Bulgarian}{bulgarian}
633 \DeclareLanguageAlias {Catalan}{catalan}
634 \DeclareLanguageAlias {Coptic}{coptic}
635 \DeclareLanguageAlias {coptic egyptian}{coptic}
636 \DeclareLanguageAlias {Coptic Egyptian}{coptic egyptian}
637 \DeclareLanguageAlias {Czech}{czech}
638 \DeclareLanguageAlias {Danish}{danish}
639 \DeclareLanguageAlias {Dutch}{dutch}
640 \DeclareLanguageAlias {Farsi}{farsi}
641 \DeclareLanguageAlias {Finnish}{finnish}
642 \DeclareLanguageAlias {français}{french}
```

5 Implementation

```
643 \DeclareLanguageAlias {Francais}{francais}
644 \DeclareLanguageDialect{canadien}{french}
645 \DeclareLanguageAlias {Canadien}{canadien}
646 \DeclareLanguageAlias {French}{french}
647 \DeclareLanguageDialect{acadian}{french}
648 \DeclareLanguageAlias {Acadian}{acadian}
649 \DeclareLanguageAlias {frenchle}{french}
650 \DeclareLanguageDialect{american}{english}
651 \DeclareLanguageAlias {American}{american}
652 \DeclareLanguageDialect{australian}{english}
653 \DeclareLanguageAlias {Australian}{australian}
654 \DeclareLanguageDialect{british}{english}
655 \DeclareLanguageAlias {British}{british}
656 \DeclareLanguageDialect{canadian}{english}
657 \DeclareLanguageAlias {Canadian}{canadian}
658 \DeclareLanguageAlias {English}{english}
659 \DeclareLanguageAlias {UKenglish}{british}
660 \DeclareLanguageAlias {USenglish}{american}
661 \DeclareLanguageDialect{newzealand}{english}
662 \DeclareLanguageAlias {Newzealand}{newzealand}
663 \DeclareLanguageAlias {Ethiop}{ethiop}
664 \DeclareLanguageAlias {Esperanto}{esperanto}
665 \DeclareLanguageAlias {Estonian}{estonian}
666 \DeclareLanguageAlias {Friulan}{friulan}
667 \DeclareLanguageAlias {Gaelic}{gaelic}
668 \DeclareLanguageDialect{irish}{gaelic}
669 \DeclareLanguageDialect{scottish}{gaelic}
670 \DeclareLanguageAlias {Irish}{irish}
671 \DeclareLanguageAlias {irish gaelic}{irish}
672 \DeclareLanguageAlias {Irish Gaelic}{irish}
673 \DeclareLanguageAlias {Scottish}{scottish}
674 \DeclareLanguageAlias {scottish gaelic}{scottish}
675 \DeclareLanguageAlias {Scottish Gaelic}{scottish}
676 \DeclareLanguageAlias {Galician}{galician}
677 \DeclareLanguageAlias {German}{german}
678 \DeclareLanguageAlias {germanb}{german}
679 \DeclareLanguageAlias {ngerman}{german}
680 \DeclareLanguageDialect{austrian}{german}
681 \DeclareLanguageAlias {Austrian}{austrian}
682 \DeclareLanguageAlias {naustrian}{austrian}
683 \DeclareLanguageAlias {Greek}{greek}
684 \DeclareLanguageAlias {polutonikogreek}{greek}
685 \DeclareLanguageAlias {ibygreek}{greek}
686 \DeclareLanguageAlias {bgreek}{greek}
687 \DeclareLanguageAlias {Hebrew}{hebrew}
688 \DeclareLanguageAlias {Hindustani}{hindustani}
689 \DeclareLanguageAlias {hindi-urdu}{hindustani}
690 \DeclareLanguageAlias {Hindi-Urdu}{hindi-urdu}
691 \DeclareLanguageDialect{hindi}{hindustani}
```

5 Implementation

```
692 \DeclareLanguageAlias {Hindi}{hindi}
693 \DeclareLanguageDialect{urdu}{hindustani}
694 \DeclareLanguageAlias {Urdu}{urdu}
695 \DeclareLanguageAlias {Hungarian}{hungarian}
696 \DeclareLanguageAlias {magyar}{hungarian}
697 \DeclareLanguageAlias {Magyar}{magyar}
698 \DeclareLanguageAlias {Icelandic}{icelandic}
699 \DeclareLanguageAlias {Interlingua}{interlingua}
700 \DeclareLanguageAlias {Italian}{italian}
701 \DeclareLanguageAlias {Japanese}{japanese}
702 \DeclareLanguageAlias {Kannada}{kannada}
703 \DeclareLanguageAlias {Ladin}{ladin}
704 \DeclareLanguageAlias {Lao}{lao}
705 \DeclareLanguageAlias {laotian}{lao}
706 \DeclareLanguageAlias {Laotian}{laotian}
707 \DeclareLanguageAlias {Latin}{latin}
708 \DeclareLanguageAlias {Latvian}{latvian}
709 \DeclareLanguageAlias {lettish}{latvian}
710 \DeclareLanguageAlias {Lettish}{lettish}
711 \DeclareLanguageAlias {Lithuanian}{lithuanian}
712 % hopefully someone who knows better than me can comment on these
713 \DeclareLanguageAlias {Malay}{malay}
714 \DeclareLanguageDialect{indonesian}{malay}
715 \DeclareLanguageAlias {Indonesian}{indonesian}
716 \DeclareLanguageAlias {indon}{indonesian}
717 \DeclareLanguageAlias {bahasa meyalu}{malay}
718 \DeclareLanguageAlias {Bahasa Meyalu}{bahasa meyalu}
719 \DeclareLanguageAlias {bahasa}{bahasa meyalu}
720 \DeclareLanguageAlias {Bahasa}{bahasa}
721 \DeclareLanguageAlias {bahasai}{bahasa}
722 \DeclareLanguageAlias {bahasam}{bahasa}
723 \DeclareLanguageAlias {Malayalam}{malayalam}
724 \DeclareLanguageAlias {Maldivian}{maldivian}
725 \DeclareLanguageAlias {divehi}{maldivian}
726 \DeclareLanguageAlias {Divehi}{divehi}
727 \DeclareLanguageAlias {Marathi}{marathi}
728 \DeclareLanguageAlias {Mongolian}{mongolian}
729 % \DeclareLanguageAlias {Syriac}{syriac}
730 \DeclareLanguageAlias {norsk}{norwegian}
731 \DeclareLanguageAlias {Norsk}{norsk}
732 \DeclareLanguageAlias {Norwegian}{norwegian}
733 \DeclareLanguageAlias {nynorsk}{norwegian}
734 \DeclareLanguageAlias {Nynorsk}{nynorsk}
735 \DeclareLanguageAlias {Occitan}{occitan}
736 \DeclareLanguageAlias {lenga d'oc}{occitan}
737 \DeclareLanguageAlias {langue d'oc}{occitan}
738 \DeclareLanguageAlias {Piedmontese}{piedmontese}
739 \DeclareLanguageAlias {piemontese}{piedmontese}
740 \DeclareLanguageAlias {Piemontese}{piemontese}
```

5 Implementation

```
741 \DeclareLanguageAlias {piemonteis}{piedmontese}
742 \DeclareLanguageAlias {Piemonteis}{piemonteis}
743 \DeclareLanguageAlias {Pinyin}{pinyin}
744 \DeclareLanguageAlias {Polish}{polish}
745 \DeclareLanguageDialect{brazil}{portuges}
746 \DeclareLanguageAlias {Brazil}{brazil}
747 \DeclareLanguageAlias {brazilian}{brazil}
748 \DeclareLanguageAlias {Brazilian}{brazilian}
749 \DeclareLanguageAlias {Portuges}{portuges}
750 \DeclareLanguageAlias {portuguese}{portuges}
751 \DeclareLanguageAlias {Portuguese}{portuguese}
752 \DeclareLanguageAlias {Romanian}{romanian}
753 \DeclareLanguageAlias {Romansh}{romansh}
754 \DeclareLanguageAlias {Romansch}{romansh}
755 \DeclareLanguageAlias {Rumantsh}{romansh}
756 \DeclareLanguageAlias {Rumantsch}{romansh}
757 \DeclareLanguageAlias {Romanche}{romansh}
758 \DeclareLanguageAlias {Russian}{russian}
759 \DeclareLanguageAlias {Samin}{samin}
760 \DeclareLanguageAlias {north sami}{samin}
761 \DeclareLanguageAlias {North Sami}{north sami}
762 \DeclareLanguageAlias {northern sami}{north sami}
763 \DeclareLanguageAlias {Northern Sami}{northern sami}
764 \DeclareLanguageAlias {Sanskrit}{sanskrit}
765 % this one isn't needed, or is it?
766 % \DeclareLanguageAlias {Serbocroatian}{serbocroatian}
767 \DeclareLanguageAlias {serbo-croatian}{serbocroatian}
768 \DeclareLanguageAlias {Serbo-Croatian}{serbocroatian}
769 \DeclareLanguageDialect{serbian}{serbocroatian}
770 \DeclareLanguageAlias {Serbian}{serbian}
771 \DeclareLanguageAlias {serbianc}{serbian}
772 \DeclareLanguageDialect{croatian}{serbocroatian}
773 \DeclareLanguageAlias {Croatian}{croatian}
774 \DeclareLanguageAlias {Slovak}{slovak}
775 \DeclareLanguageAlias {Slovenian}{slovenian}
776 \DeclareLanguageAlias {Sorbian}{sorbian}
777 \DeclareLanguageDialect{lowersorbian}{sorbian}
778 \DeclareLanguageDialect{uppersorbian}{sorbian}
779 \DeclareLanguageAlias {Lowersorbian}{lowersorbian}
780 \DeclareLanguageAlias {Uppersorbian}{uppersorbian}
781 \DeclareLanguageAlias {lsorbian}{lowersorbian}
782 \DeclareLanguageAlias {usorbian}{uppersorbian}
783 \DeclareLanguageAlias {lower sorbian}{lowersorbian}
784 \DeclareLanguageAlias {upper sorbian}{uppersorbian}
785 \DeclareLanguageAlias {Lower Sorbian}{lowersorbian}
786 \DeclareLanguageAlias {Upper Sorbian}{uppersorbian}
787 \DeclareLanguageAlias {Spanglish}{spanglish}
788 \DeclareLanguageAlias {Spanish}{spanish}
789 \DeclareLanguageAlias {Swedish}{swedish}
```

5 Implementation

```

790 \DeclareLanguageDialect{swissgerman}{german}
791 % this is to be discussed: swiss could also be an alias of french, italian or
792 % romansh:
793 \DeclareLanguageAlias {swiss}{swissgerman}
794 \DeclareLanguageAlias {Swiss}{swiss}
795 \DeclareLanguageAlias {Swissgerman}{swissgerman}
796 \DeclareLanguageAlias {swiss german}{swissgerman}
797 \DeclareLanguageAlias {Swiss German}{swiss german}
798 \DeclareLanguageAlias {Tamil}{tamil}
799 \DeclareLanguageAlias {Telugu}{telugu}
800 \DeclareLanguageAlias {Thai}{thai}
801 \DeclareLanguageAlias {thaicjk}{thai}
802 \DeclareLanguageAlias {Thaicjk}{thaicjk}
803 \DeclareLanguageAlias {Tibetan}{tibetan}
804 \DeclareLanguageAlias {Turkish}{turkish}
805 \DeclareLanguageAlias {Turkmen}{turkmen}
806 \DeclareLanguageAlias {Ukrainian}{ukrainian}
807 \DeclareLanguageAlias {Vietnamese}{vietnamese}
808 \DeclareLanguageAlias {Welsh}{welsh}
809
810 % -----
811 % load basic dictionary if available
812 \AtBeginDocument{%
813   \@trnslt@if@dialect{\@trnslt@current@language}
814   {%
815     \@trnslt@if@dictionary
816     {translations-basic-dictionary}
817     {\@trnslt@language{\@trnslt@current@language}}%
818     {%
819       \@trnslt@load@dictionary@silent@for
820       {translations-basic-dictionary}
821       {\@trnslt@language{\@trnslt@current@language}}%
822     }%
823     {%
824       \@trnslt@load@dictionary@silent@for
825       {translations-basic-dictionary}
826       {\@trnslt@dialect@of{\@trnslt@current@language}}%
827     }
828   }
829   {%
830     \@trnslt@load@dictionary@silent@for
831     {translations-basic-dictionary}
832     {\@trnslt@language{\@trnslt@current@language}}%
833   }%
834 }
835
836 \endinput
837
838 % -----

```

Bibliography

839 % HISTORY:

- 840 2012/09/30 v0.2beta - first version (as part of the 'exsheets' bundle)
- 841 2012/10/05 v0.2 - `\LoadDictionary` and `\LoadDictionaryFor` added and loads of
- 842 languages defined.
- 843 2013/03/10 v0.8 - basic dictionaries for English, German, French and Spanish
- 844 - new command `\DeclareDictTranslation`
- 845 2013/04/04 v0.8a - bug fix in `\DeclareDictTranslation`
- 846 2013/04/07 v0.9 - slightly improved messages
- 847 2013/04/08 v0.9a - changed fallback warning into info
- 848 - synchronized version number with 'exsheets' until now but
- 849 won't any more
- 850 2013/06/22 v0.9b - added Swiss
- 851 2013/06/28 v0.10 - declaring aliases of dialects now works as expected
- 852 - declarings dialects of an alias now correctly declares
- 853 the dialect to the correct base language
- 854 - corrected a few erroneous language declarations
- 855 2013/07/12 v0.10a - `\GetTranslation` gets two-folded fallback: use
- 856 fallback-translation if no translation for the current
- 857 language has been defined; use literal string if /no/
- 858 language is used - this should never happen but /will/
- 859 happen if neither 'babel' nor 'polyglossia' have been
- 860 loaded, i.e., no language has been chosen /and/ the
- 861 package writer did not provide an English translation
- 862 2013/07/16 v1.0 - removed from 'exsheets' bundle - 'translations' should
- 863 be a package of it's own
- 864 - load basic dictionary automatically if available
- 865 - rudimentary check in `\LoadDictionary` if loaded file is a
- 866 dictionary
- 867 - new command `\PrintDictionaryFor`
- 868 - redefined conditionals; they still seemed to make
- 869 trouble in some cases
- 870 2013/08/05 v1.1 - added /loads/ of languages, noe the list of babel and
- 871 polyglossia languages hopefully is complete
- 872 - a few languages had falsely been declared as dialect
- 873 instead of an alias
- 874 - added weekday names and month names to basic dictionary
- 875 - new command `\baselanguage`
- 876 - new commands `\GetLCTranslation`, `\GetLCTranslationFor`,
- 877 `\GetLCTranslationWarn` and `\GetLCTranslationForWarn`
- 878 - load basic dictionary also for dialects and if it
- 879 doesn't exist load it for the corresponding base
- 880 language instead
- 881 2013/09/30 v1.1a - Bug fix in `\NewTranslation` und `\RenewTranslation`

Bibliography

- [Bra13] Johannes Braams, current maintainer: Javier Bezos. babel. version 3.9f, May 16, 2013.
URL: <http://mirror.ctan.org/macros/latex/required/babel>.

- [Cha13] François Charette, current maintainer: Arthur Reutenauer.
polyglossia. version 1.33.4, June 27, 2013.
URL: <http://mirror.ctan.org/macros/latex/contrib/polyglossia>.
- [KN12] Markus Kohm and Frank Neukahm. KOMA-Script. version 3.11b, July 29, 2012.
URL: <http://mirror.ctan.org/macros/latex/contrib/koma-script>.
- [Leh11] Philipp Lehman. etoolbox. version 2.1, Jan. 3, 2011.
URL: <http://mirror.ctan.org/macros/latex/contrib/etoolbox>.

Index

B	E	N
babel (package) 1 f., 4 f., 7, 12, 14	etoolbox (package) 2	Neukahm, Frank 2
<code>\baselanguage</code> 5	G	<code>\NewTranslation</code> 3, 20, 32
Bezos, Javier 1	<code>\GetLCTranslation</code> 4	Niederberger, Clemens 1
Braams, Johannes 1	<code>\GetLCTranslationFor</code> 4	P
C	<code>\GetLCTranslationForWarn</code> 4	polyglossia (package) . . 1 f., 7, 12, 14
Charette, François 1	<code>\GetLCTranslationWarn</code> 4	<code>\PrintDictionaryFor</code> 5
CTAN 2	<code>\GetTranslation</code> 2, 4–7, 21, 32	<code>\ProvideDictionaryFor</code> . . . 5, 7, 24
D	<code>\GetTranslationFor</code> 4, 6, 21	R
<code>\DeclareDictTranslation</code> 5, 7,	<code>\GetTranslationForWarn</code> 4	<code>\RenewTranslation</code> 4, 20, 32
24 f., 32	<code>\GetTranslationWarn</code> 4	Reutenauer, Arthur 1
<code>\DeclareLanguage</code> . 3, 5, 17, 19, 25 ff.	K	S
<code>\DeclareLanguageAlias</code> 3, 19,	Kohm, Markus 2	<code>\SaveTranslation</code> 5 f., 23
27–31	KOMA-Script (package) 2	<code>\SaveTranslationFor</code> 5 f., 23
<code>\DeclareLanguageDialect</code> . . . 3, 18,	L	T
28–31	Lehman, Philipp 2	translator (package) 1
<code>\DeclareTranslation</code> 4, 6 f., 20,	<code>\LoadDictionary</code> 5, 23, 32	
22 f.	<code>\LoadDictionaryFor</code> 5, 23, 32	
<code>\DeclareTranslationFallback</code> . . 4,	LPPL 2	
6 f., 20		