

translations

v1.0a 2013/07/19

internationalization of L^AT_EX 2_ε packages

Clemens NIEDERBERGER

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

English documentation

Contents

| | | | | |
|----------|---------------------------------|----------|---|-----------|
| | | 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 | | |
| 3.4.2 | The 'fallback' language . | 6 | | |
| | | 4 | Defined Languages | 9 |
| | | 4.1 | Base Languages | 9 |
| | | 4.2 | Language Dialects | 10 |
| | | 4.3 | Language Aliases | 10 |
| | | 5 | Implementation | 13 |
| | | | 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 dialects (see table 3 on page 10) language aliases (see table 4 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.

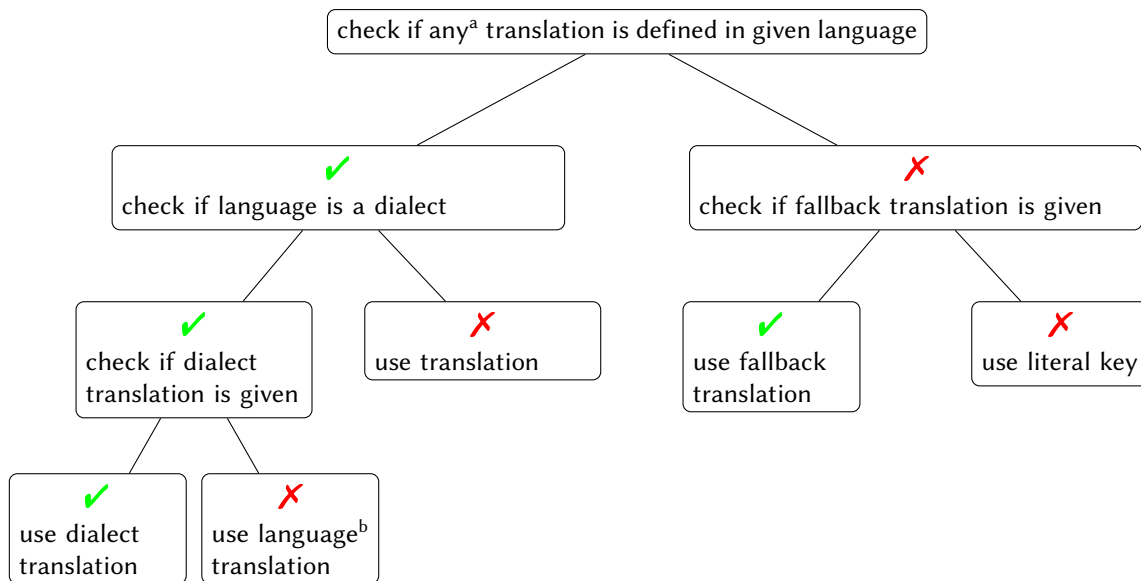


FIGURE 1: Schematic representation of **TRANSLATIONS**’ translating mechanism. Notes: ^a except for a possible fallback translation. ^b i.e., the base language of the dialect.

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

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 |

continues

| | | | | |
|--------------|-------------|----------|----------|--------------|
| 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.

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 |

continues

4 Defined Languages

| alias | language | alias | language |
|------------------|------------------|------------------|---------------|
| 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 |
| francais | french | Francais | francais |
| Canadien | canadien | French | french |
| Acadian | acadian | 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 meyalu | bahasa | bahasa meyalu |

continues

4 Defined Languages

| alias | language | alias | language |
|----------------|---------------|----------------|---------------|
| 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 | piemontese | Piemontese | piemontese |
| piemonteis | piemontese | Piemonteis | piemonteis |
| Pinyin | pinyin | Polish | polish |
| 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 | Isorbian | 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

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{2013/07/19}
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.}
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     }

```

5 Implementation

```

77 \ifdef\language\name{}
78 {\def\language\name{english}}
79 \def\@trns\lt@current\language{\language\name}
80 \ifdef\bb\l@afterfi{}
81 {\long\def\bb\l@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\@trns\lt@languages{}% all languages
88 \def\@trns\lt@aliases@pair{}% all aliases and their base
89 \def\@trns\lt@aliases@single{}% all aliases
90 \def\@trns\lt@dialects@pair{}% all dialects and their base
91 \def\@trns\lt@dialects@single{}% all dialects
92
93 % -----
94 % define \@trns\lt@if@<name> conditionals that don't leave the checked macro as
95 % \relax behind and check for \@trns\lt@<name>@#1. These conditionals should
96 % also be expandable in an \edef-like context. Thanks to e-TeX there's
97 % \ifcsname:
98 \def\@trns\lt@newif#1{%
99   \csdef{\@trns\lt@if@#1}##1{%
100     \ifcsname \@trns\lt@#1@##1\endcsname
101       \expandafter\@firstoftwo
102     \else
103       \expandafter\@secondoftwo
104     \fi
105   }%
106 }
107
108 % -----
109 % \DeclareLanguage
110 % #1: language
111 \newrobustcmd*\DeclareLanguage[1]{%
112   \@trns\lt@declare@language{#1}}
113 \@onlypreamble\DeclareLanguage
114
115 \def\@trns\lt@declare@language#1{%
116   \@trns\lt@if@language{#1}
117   {}{%
118     \csdef{\@trns\lt@language@#1}{#1}%
119     \listadd\@trns\lt@languages{#1}%
120   }%
121 }
122
123 \def\@trns\lt@language#1{%
124   \csuse{\@trns\lt@language@#1}}
125
126 \@trns\lt@newif{language}
127
128 % -----

```

5 Implementation

```

129 % \DeclareLanguageDialect
130 % #1: dialect
131 % #2: language
132 \newrobustcmd*\DeclareLanguageDialect[2]{%
133   \@trnslt@declare@languagedialect{#1}{#2}}
134 \@onlypreamble\DeclareLanguageDialect
135
136 \def\@trnslt@declare@languagedialect#1#2{%
137   \@trnslt@if@language{#2}
138     {}{%
139       \@trnslt@warn@unknown@lang{#2}%
140       \@trnslt@declare@language{#2}%
141     }%
142   \@trnslt@if@dialect{#1}
143     {% => ist schon als dialect definiert => irgendwelche weiteren checks?
144     }
145     {%
146       \@trnslt@if@alias{#2}
147         {%
148           \csdef{@trnslt@dialect@#1}{\@trnslt@alias{#2}{#1}}%
149           \@trnslt@declare@language{#1}%
150           \listeadadd\@trnslt@dialects@single{#1}%
151           \listeadadd\@trnslt@dialects@pair{{#1}{\@trnslt@alias{#2}}}%
152         }
153         {%
154           \csdef{@trnslt@dialect@#1}{{#2}{#1}}%
155           \@trnslt@declare@language{#1}%
156           \listeadadd\@trnslt@dialects@single{#1}%
157           \listeadadd\@trnslt@dialects@pair{{#1}{#2}}%
158         }%
159       }%
160     }
161
162 \def\@trnslt@dialect#1{%
163   \csuse{@trnslt@dialect@#1}}
164
165 % this macros fetches the base language for a given dialect, expandably:
166 \def\@trnslt@dialect@of#1{%
167   \@trnslt@if@dialect{#1}
168     {%
169       \expandafter\expandafter\expandafter
170       \@firstoftwo
171       \csname @trnslt@dialect@#1\endcsname
172     }{%
173   }
174
175 \@trnslt@newif{dialect}
176
177 % -----
178 % \DeclareLanguageAlias
179 % #1: alias
180 % #2: language

```

5 Implementation

```

181 \newrobustcmd*\DeclareLanguageAlias[2]{%
182   \@trnslt@declare@languagealias{#1}{#2}}
183 \@onlypreamble\DeclareLanguageAlias
184
185 \def\@trnslt@declare@languagealias#1#2{%
186   \@trnslt@if@language{#2}
187     {}{%
188       \@trnslt@warn@unknown@lang{#2}%
189       \@trnslt@declare@language{#2}%
190     }%
191   \csletcs{@trnslt@language@#1}{@trnslt@language@#2}%
192   \@trnslt@if@dialect{#2}
193     {\csletcs{@trnslt@dialect@#1}{@trnslt@dialect@#2}}
194     {}%
195   \ifinlist{#1}\@trnslt@aliases@single
196     {}{%
197       \csdef{@trnslt@alias@#1}{#2}%
198       \listadd\@trnslt@aliases@pair{{#1}{#2}}%
199       \listadd\@trnslt@aliases@single{#1}%
200     }%
201   }
202
203 \def\@trnslt@alias#1{%
204   \csuse{@trnslt@alias@#1}}
205
206 \@trnslt@newif{alias}
207
208 % -----
209 % dummy language: `fallback':
210 \DeclareLanguage{fallback}
211 \DeclareLanguageAlias{Fallback}{fallback}
212
213 % -----
214 % \DeclareTranslation, \NewTranslation and \RenewTranslation
215 % #1: language
216 % #2: word
217 % #3: replacement
218 \newrobustcmd*\DeclareTranslation[3]{%
219   \@trnslt@declare@translation{#2}{#1}{#3}}
220 \@onlypreamble\DeclareTranslation
221
222 \newrobustcmd*\DeclareTranslationFallback[2]{%
223   \@trnslt@declare@translation{#1}{fallback}{#2}}
224 \@onlypreamble\DeclareTranslationFallback
225
226 \newrobustcmd*\NewTranslation[3]{%
227   \@trnslt@new@translation{#2}{#1}{#3}}
228 \@onlypreamble\NewTranslation
229
230 \newrobustcmd*\RenewTranslation[3]{%
231   \@trnslt@renew@translation{#2}{#1}{#3}}
232 \@onlypreamble\RenewTranslation

```


5 Implementation

```

233
234 % #1: word
235 % #2: language
236 % #3: replacement
237 \def\@trnslt@declare@translation#1#2#3{%
238   \@trnslt@if@language{#2}
239   {%
240     \@trnslt@if@dialect{#2}
241     {%
242       \csdef{@trnslt@word@#1@\@trnslt@dialect{#2}}{#3}%
243       \@trnslt@if@word\@trnslt@dialect@of{#1}{#2}
244       {}
245       {\csdef{@trnslt@word@#1@\@trnslt@dialect@of{#2}}{#3}}%
246     }
247     {\csdef{@trnslt@word@#1@\@trnslt@language{#2}}{#3}}%
248     % save the <word> as <word>:
249     \csdef{@trnslt@word@#1@literal}{#1}%
250   }
251   {\@trnslt@err@unknown@lang{#2}}%
252 }
253
254 \def\@trnslt@new@translation#1#2#3{%
255   \@trnslt@if@translation{#1}{#2}
256   {\@trnslt@err@already@defined{#1}{#2}}
257   {\@trnslt@declare@translation{#1}{#2}{#3}}
258
259 \def\@trnslt@renew@translation#1#2#3{%
260   \@trnslt@if@translation{#1}{#2}
261   {\@trnslt@declare@translation{#1}{#2}{#3}}
262   {\@trnslt@err@not@defined{#1}{#2}}
263
264 % -----
265 % now let's go through some trouble to check if a translation exists:
266 \def\@trnslt@if@word#1#2#3{%
267   \ifcsname @trnslt@word@#2@#1{#3}\endcsname
268   \expandafter\@firstoftwo
269   \else
270   \expandafter\@secondoftwo
271   \fi
272 }%
273
274 \def\@trnslt@if@translation#1#2{%
275   \@trnslt@if@word\@trnslt@language{#1}{#2}
276   {\expandafter\@firstoftwo}
277   {%
278     \@trnslt@if@dialect{#2}
279     {%
280       \@trnslt@if@word\@trnslt@dialect{#1}{#2}
281       {\expandafter\@firstoftwo}
282       {%
283         \@trnslt@if@word\@trnslt@dialect@of{#1}{#2}
284         {\expandafter\@firstoftwo}

```

5 Implementation

```

285         {\expandafter\@secondoftwo}%
286     }
287 }
288 {\expandafter\@secondoftwo}%
289 }%
290 }
291
292 % -----
293 % \GetTranslationFor and \GetTranslation
294 % these need to be expandable!
295 % #1: language
296 % #2: word
297 \newcommand*\GetTranslationFor[2]{%
298     \@trnslt@checkandget@translation@for{#2}{#1}}
299
300 \newcommand*\GetTranslation[1]{%
301     \@trnslt@checkandget@translation@for{#1}{\@trnslt@current@language}}
302
303 % unexpandable version of the commands that raise a warning if no translation
304 % is available:
305 \newcommand*\GetTranslationForWarn[2]{%
306     \@trnslt@getandwarn@translation@for{#2}{#1}}
307
308 \newcommand*\GetTranslationWarn[1]{%
309     \@trnslt@getandwarn@translation@for{#1}{\@trnslt@current@language}}
310
311 % #1: word #2: language
312 \def\@trnslt@get@translation@for#1#2{%
313     \@trnslt@if@dialect{#2}
314     {%
315         \ifcsdef{@trnslt@word@#1@\@trnslt@dialect{#2}}
316             {\csuse{@trnslt@word@#1@\@trnslt@dialect{#2}}}
317             {\csuse{@trnslt@word@#1@\@trnslt@dialect@of{#2}}}%
318     }
319     {\csuse{@trnslt@word@#1@\@trnslt@language{#2}}}%
320 }
321
322 \def\@trnslt@checkandget@translation@for#1#2{%
323     \@trnslt@if@translation{#1}{#2}
324     {\@trnslt@get@translation@for{#1}{#2}}
325     {%
326         \@trnslt@if@translation{#1}{fallback}
327         {\csuse{@trnslt@word@#1@fallback}}
328         {\csuse{@trnslt@word@#1@literal}}}%
329     }%
330 }
331
332 % this is not expandable!
333 \protected\def\@trnslt@getandwarn@translation@for#1#2{%
334     \@trnslt@if@translation{#1}{#2}
335     {\@trnslt@get@translation@for{#1}{#2}}
336     {%

```

5 Implementation

```

337 \@trnslt@warning{Translation for `#1' in #2 unknown. You may try to use
338 \string\DeclareTranslation{#2}{#1}{ ... } in your preamble.}%
339 \@trnslt@if@translation{#1}{fallback}
340 {
341 \@trnslt@info{Using fallback translation for `#1'}%
342 \csuse{@trnslt@word@#1@fallback}
343 }
344 {\csuse{@trnslt@word@#1@literal}}}%
345 }%
346 }
347
348 % -----
349 % \SaveTranslationFor and \SaveTranslation
350 \newrobustcmd*\SaveTranslationFor[3]{%
351 \@trnslt@save@translation@for{#1}{#3}{#2}}
352
353 \newrobustcmd*\SaveTranslation[2]{%
354 \@trnslt@save@translation@for{#1}{#2}{\@trnslt@current@language}}
355
356 \def\@trnslt@save@translation@for#1#2#3{%
357 \edef#1{%
358 \@trnslt@if@translation{#2}{#3}
359 {\csuse{@trnslt@word@#2@\@trnslt@language{#3}}}
360 }%
361 }}
362
363 % -----
364 % \LoadDictionary and \LoadDictionaryFor
365 \newrobustcmd*\LoadDictionary[1]{%
366 \@trnslt@load@dictionary@for{#1}{\@trnslt@current@language}}
367 \@onlypreamble\LoadDictionary
368
369 \newrobustcmd*\LoadDictionaryFor[2]{%
370 \@trnslt@load@dictionary@for{#2}{#1}}
371 \@onlypreamble\LoadDictionaryFor
372
373 % #1: name
374 % #2: lang
375 \def\@trnslt@load@dictionary@for#1#2{%
376 \AtBeginDocument{%
377 \InputIfFileExists{#1-\@trnslt@language{#2}.trsl}
378 {\@trnslt@check@dictionary{#1}{#2}}
379 {\@trnslt@warning{dictionary file `#1-\@trnslt@language{#2}.trsl' not
380 found.}}}%
381 }}
382
383 \def\@trnslt@check@dictionary#1#2{%
384 \AfterFile{#1-\@trnslt@language{#2}.trsl}
385 {
386 \ifcsdef{@trnslt@dictionary@#1@\@trnslt@language{#2}}
387 {\@trnslt@info{loading dictionary `#1' for `#2'.}}
388 {

```

5 Implementation

```

389         \@trnslt@warning{file `#1-\@trnslt@language{#2}.trsl' does not
390         appear to be a dictionary}%
391     }%
392 }%
393 }
394
395 \def\@trnslt@load@dictionary@silent@for#1#2{%
396     \AtBeginDocument{\InputIfFileExists{#1-\@trnslt@language{#2}.trsl}{}}{}%
397
398 \newrobustcmd*\ProvideDictionaryFor[2]{%
399     \@trnslt@provide@dictionary@for{#1}{#2}%
400 \@onlypreamble\ProvideDictionaryFor
401
402 \def\@trnslt@provide@dictionary@for#1#2{%
403     \def\@trnslt@dictionary@name{#2}%
404     \edef\@trnslt@dictionary@lang{\@trnslt@language{#1}}%
405     % this macro can be used to check if we have a dictionary and will also be
406     % used as a list for the dictionary entries:
407     \csdef{\@trnslt@dictionary@\@trnslt@dictionary@name \@trnslt@dictionary@lang}{%
408         \@ifnextchar[
409             {\@trnslt@provide@dictionary@version}
410             {
411                 \ProvidesFile
412                 {#2-\@trnslt@dictionary@lang.trsl}%
413                 [(\@trnslt@dictionary@lang\space translation file `#2')]
414             }%
415         }
416
417 \def\@trnslt@provide@dictionary@version[#1]{%
418     \ProvidesFile
419     {\@trnslt@dictionary@name-\@trnslt@dictionary@lang.trsl}%
420     [(\@trnslt@dictionary@lang\space translation file ` \@trnslt@dictionary@name') #1]}
421
422 % \@trnslt@dictionary@language
423 \newrobustcmd*\DeclareDictTranslation[2]{%
424     \listcsadd
425     {\@trnslt@dictionary@\@trnslt@dictionary@name \@trnslt@dictionary@lang}
426     {{#1}{#2}}%
427     \@trnslt@declare@translation{#1}{\@trnslt@dictionary@lang}{#2}%
428 }
429 \@onlypreamble\DeclareDictTranslation
430
431 % \PrintDictionaryFor
432 % #1: lang
433 % #2: name
434 % #3: pre
435 % #4: mid
436 % #5: post
437 \newcommand*\PrintDictionaryFor[5]{%
438     \@trnslt@print@dictionary@for{#1}{#2}{#3}{#4}{#5}%
439
440 % #1: lang

```

5 Implementation

```

441 % #2: name
442 % #3: pre
443 % #4: mid
444 % #5: post
445 \def\@trnslt@print@dictionary@for#1#2#3#4#5{%
446   \forlistcsloop
447     {\@trnslt@print@dictionary@entry{#3}{#4}{#5}}
448     {\@trnslt@dictionary@#2@\@trnslt@language{#1}}%
449 }
450
451 % #1: pre
452 % #2: mid
453 % #3: post
454 % #4: {key}{translation}
455 \def\@trnslt@print@dictionary@entry#1#2#3#4{%
456   \@trnslt@print@dictionary@entry@aux{#1}{#2}{#3}#4}
457
458 % #1: pre
459 % #2: mid
460 % #3: post
461 % #4: key
462 % #5: translation
463 \def\@trnslt@print@dictionary@entry@aux#1#2#3#4#5{#1#4#2#5#3}
464
465 % -----
466 % predefined languages
467 \DeclareLanguage{afrikaans}
468 \DeclareLanguage{albanian}
469 \DeclareLanguage{amharic}
470 \DeclareLanguage{arabic}
471 \DeclareLanguage{armenian}
472 \DeclareLanguage{asturian}
473 \DeclareLanguage{basque}
474 \DeclareLanguage{bengali}
475 \DeclareLanguage{breton}
476 \DeclareLanguage{bulgarian}
477 \DeclareLanguage{catalan}
478 \DeclareLanguage{coptic}
479 \DeclareLanguage{czech}
480 \DeclareLanguage{danish}
481 \DeclareLanguage{dutch}
482 \DeclareLanguage{english}
483 \DeclareLanguage{esperanto}
484 \DeclareLanguage{estonian}
485 \DeclareLanguage{ethiop}
486 \DeclareLanguage{farsi}
487 \DeclareLanguage{finnish}
488 \DeclareLanguage{french}
489 \DeclareLanguage{friulan}
490 \DeclareLanguage{gaelic}
491 \DeclareLanguage{galician}
492 \DeclareLanguage{german}

```

5 Implementation

```
493 \DeclareLanguage{greek}
494 \DeclareLanguage{hebrew}
495 \DeclareLanguage{hindustani}
496 \DeclareLanguage{hungarian}
497 \DeclareLanguage{icelandic}
498 \DeclareLanguage{interlingua}
499 \DeclareLanguage{italian}
500 \DeclareLanguage{japanese}
501 \DeclareLanguage{kannada}
502 \DeclareLanguage{latin}
503 \DeclareLanguage{lao}
504 \DeclareLanguage{latin}
505 \DeclareLanguage{latvian}
506 \DeclareLanguage{lithuanian}
507 \DeclareLanguage{malay}
508 \DeclareLanguage{malayalam}
509 \DeclareLanguage{maldivian}
510 \DeclareLanguage{marathi}
511 \DeclareLanguage{mongolian}
512 % polyglossia seems to support this one but it is unclear which language is
513 % actually meant by it:
514 % \DeclareLanguage{nko}
515 \DeclareLanguage{norwegian}
516 \DeclareLanguage{occitan}
517 \DeclareLanguage{piedmontese}
518 \DeclareLanguage{pinyin}
519 \DeclareLanguage{polish}
520 \DeclareLanguage{portuges}
521 \DeclareLanguage{romanian}
522 \DeclareLanguage{romansh}
523 \DeclareLanguage{russian}
524 \DeclareLanguage{samin}
525 \DeclareLanguage{sanskrit}
526 \DeclareLanguage{serbocroatian}
527 \DeclareLanguage{slovak}
528 \DeclareLanguage{slovenian}
529 \DeclareLanguage{sorbian}
530 % not sure about this: isn't it either a Spanish or English dialect?
531 \DeclareLanguage{spanglish}
532 \DeclareLanguage{spanish}
533 \DeclareLanguage{swedish}
534 % polyglossia seems to support this one but it is unclear which language is
535 % actually meant by it:
536 % \DeclareLanguage{syriac}
537 \DeclareLanguage{tamil}
538 \DeclareLanguage{telugu}
539 \DeclareLanguage{thai}
540 \DeclareLanguage{tibetan}
541 \DeclareLanguage{turkish}
542 \DeclareLanguage{turkmen}
543 \DeclareLanguage{ukrainian}
544 \DeclareLanguage{vietnamese}
```

5 Implementation

```
545 \DeclareLanguage{welsh}
546
547 % -----
548 % aliases and dialects:
549 \DeclareLanguageAlias {Afrikaans}{afrikaans}
550 \DeclareLanguageAlias {Albanian}{albanian}
551 \DeclareLanguageAlias {Amharic}{amharic}
552 \DeclareLanguageAlias {Arabic}{arabic}
553 \DeclareLanguageAlias {Armenian}{armenian}
554 \DeclareLanguageAlias {Asturian}{asturian}
555 \DeclareLanguageAlias {astur-leonese}{asturian}
556 \DeclareLanguageAlias {Astur-Leonese}{astur-leonese}
557 \DeclareLanguageAlias {asturian-leonese}{asturian}
558 \DeclareLanguageAlias {Asturian-Leonese}{asturian-leonese}
559 \DeclareLanguageAlias {Basque}{basque}
560 \DeclareLanguageAlias {Bengali}{bengali}
561 \DeclareLanguageAlias {Breton}{breton}
562 \DeclareLanguageAlias {Bulgarian}{bulgarian}
563 \DeclareLanguageAlias {Catalan}{catalan}
564 \DeclareLanguageAlias {Coptic}{coptic}
565 \DeclareLanguageAlias {coptic egyptian}{coptic}
566 \DeclareLanguageAlias {Coptic Egyptian}{coptic egyptian}
567 \DeclareLanguageAlias {Czech}{czech}
568 \DeclareLanguageAlias {Danish}{danish}
569 \DeclareLanguageAlias {Dutch}{dutch}
570 \DeclareLanguageAlias {Farsi}{farsi}
571 \DeclareLanguageAlias {Finnish}{finnish}
572 \DeclareLanguageAlias {français}{french}
573 \DeclareLanguageAlias {Français}{français}
574 \DeclareLanguageDialect{canadien}{french}
575 \DeclareLanguageAlias {Canadien}{canadien}
576 \DeclareLanguageAlias {French}{french}
577 \DeclareLanguageDialect{acadian}{french}
578 \DeclareLanguageAlias {Acadian}{acadian}
579 \DeclareLanguageAlias {frenchle}{french}
580 \DeclareLanguageDialect{american}{english}
581 \DeclareLanguageAlias {American}{american}
582 \DeclareLanguageDialect{australian}{english}
583 \DeclareLanguageAlias {Australian}{australian}
584 \DeclareLanguageDialect{british}{english}
585 \DeclareLanguageAlias {British}{british}
586 \DeclareLanguageDialect{canadian}{english}
587 \DeclareLanguageAlias {Canadian}{canadian}
588 \DeclareLanguageAlias {English}{english}
589 \DeclareLanguageAlias {UKenglish}{british}
590 \DeclareLanguageAlias {USenglish}{american}
591 \DeclareLanguageDialect{newzealand}{english}
592 \DeclareLanguageAlias {Newzealand}{newzealand}
593 \DeclareLanguageAlias {Ethiop}{ethiop}
594 \DeclareLanguageAlias {Esperanto}{esperanto}
595 \DeclareLanguageAlias {Estonian}{estonian}
596 \DeclareLanguageAlias {Friulan}{friulan}
```

5 Implementation

```
597 \DeclareLanguageAlias {Gaelic}{gaelic}
598 \DeclareLanguageDialect{irish}{gaelic}
599 \DeclareLanguageDialect{scottish}{gaelic}
600 \DeclareLanguageAlias {Irish}{irish}
601 \DeclareLanguageAlias {irish gaelic}{irish}
602 \DeclareLanguageAlias {Irish Gaelic}{irish}
603 \DeclareLanguageAlias {Scottish}{scottish}
604 \DeclareLanguageAlias {scottish gaelic}{scottish}
605 \DeclareLanguageAlias {Scottish Gaelic}{scottish}
606 \DeclareLanguageAlias {Galician}{galician}
607 \DeclareLanguageAlias {German}{german}
608 \DeclareLanguageAlias {germanb}{german}
609 \DeclareLanguageAlias {ngerman}{german}
610 \DeclareLanguageDialect{austrian}{german}
611 \DeclareLanguageAlias {Austrian}{austrian}
612 \DeclareLanguageAlias {naustrian}{austrian}
613 \DeclareLanguageAlias {Greek}{greek}
614 \DeclareLanguageAlias {polutonikogreek}{greek}
615 \DeclareLanguageAlias {ibygreek}{greek}
616 \DeclareLanguageAlias {bgreek}{greek}
617 \DeclareLanguageAlias {Hebrew}{hebrew}
618 \DeclareLanguageAlias {Hindustani}{hindustani}
619 \DeclareLanguageAlias {hindi-urdu}{hindustani}
620 \DeclareLanguageAlias {Hindi-Urdu}{hindi-urdu}
621 \DeclareLanguageDialect{hindi}{hindustani}
622 \DeclareLanguageAlias {Hindi}{hindi}
623 \DeclareLanguageDialect{urdu}{hindustani}
624 \DeclareLanguageAlias {Urdu}{urdu}
625 \DeclareLanguageAlias {Hungarian}{hungarian}
626 \DeclareLanguageAlias {magyar}{hungarian}
627 \DeclareLanguageAlias {Magyar}{magyar}
628 \DeclareLanguageAlias {Icelandic}{icelandic}
629 \DeclareLanguageAlias {Interlingua}{interlingua}
630 \DeclareLanguageAlias {Italian}{italian}
631 \DeclareLanguageAlias {Japanese}{japanese}
632 \DeclareLanguageAlias {Kannada}{kannada}
633 \DeclareLanguageAlias {Ladin}{ladin}
634 \DeclareLanguageAlias {Lao}{lao}
635 \DeclareLanguageAlias {laotian}{lao}
636 \DeclareLanguageAlias {Laotian}{laotian}
637 \DeclareLanguageAlias {Latin}{latin}
638 \DeclareLanguageAlias {Latvian}{latvian}
639 \DeclareLanguageAlias {lettish}{latvian}
640 \DeclareLanguageAlias {Lettish}{lettish}
641 \DeclareLanguageAlias {Lithuanian}{lithuanian}
642 % hopefully someone who knows better than me can comment on these
643 \DeclareLanguageAlias {Malay}{malay}
644 \DeclareLanguageDialect{indonesian}{malay}
645 \DeclareLanguageAlias {Indonesian}{indonesian}
646 \DeclareLanguageAlias {indon}{indonesian}
647 \DeclareLanguageAlias {bahasa meyalu}{malay}
648 \DeclareLanguageAlias {Bahasa Meyalu}{bahasa meyalu}
```


5 Implementation

```

649 \DeclareLanguageAlias {bahasa}{bahasa meyalu}
650 \DeclareLanguageAlias {Bahasa}{bahasa}
651 \DeclareLanguageAlias {bahasai}{bahasa}
652 \DeclareLanguageAlias {bahasam}{bahasa}
653 \DeclareLanguageAlias {Malayalam}{malayalam}
654 \DeclareLanguageAlias {Maldivian}{maldivian}
655 \DeclareLanguageAlias {divehi}{maldivian}
656 \DeclareLanguageAlias {Divehi}{divehi}
657 \DeclareLanguageAlias {Marathi}{marathi}
658 \DeclareLanguageAlias {Mongolian}{mongolian}
659 % \DeclareLanguageAlias {Syriac}{syriac}
660 \DeclareLanguageAlias {norsk}{norwegian}
661 \DeclareLanguageAlias {Norsk}{norsk}
662 \DeclareLanguageAlias {Norwegian}{norwegian}
663 \DeclareLanguageAlias {nynorsk}{norwegian}
664 \DeclareLanguageAlias {Nynorsk}{nynorsk}
665 \DeclareLanguageAlias {Occitan}{occitan}
666 \DeclareLanguageAlias {lenga d'oc}{occitan}
667 \DeclareLanguageAlias {langue d'oc}{occitan}
668 \DeclareLanguageAlias {Piedmontese}{piedmontese}
669 \DeclareLanguageAlias {piemontese}{piedmontese}
670 \DeclareLanguageAlias {Piemontese}{piemontese}
671 \DeclareLanguageAlias {piemonteis}{piedmontese}
672 \DeclareLanguageAlias {Piemonteis}{piemonteis}
673 \DeclareLanguageAlias {Pinyin}{pinyin}
674 \DeclareLanguageAlias {Polish}{polish}
675 \DeclareLanguageDialect{brazil}{portuges}
676 \DeclareLanguageAlias {Brazil}{brazil}
677 \DeclareLanguageAlias {brazilian}{brazil}
678 \DeclareLanguageAlias {Brazilian}{brazilian}
679 \DeclareLanguageAlias {Portuges}{portuges}
680 \DeclareLanguageAlias {portuguese}{portuges}
681 \DeclareLanguageAlias {Portuguese}{portuguese}
682 \DeclareLanguageAlias {Romanian}{romanian}
683 \DeclareLanguageAlias {Romansh}{romansh}
684 \DeclareLanguageAlias {Romansch}{romansh}
685 \DeclareLanguageAlias {Rumantsh}{romansh}
686 \DeclareLanguageAlias {Rumantsch}{romansh}
687 \DeclareLanguageAlias {Romanche}{romansh}
688 \DeclareLanguageAlias {Russian}{russian}
689 \DeclareLanguageAlias {Samin}{samin}
690 \DeclareLanguageAlias {north sami}{samin}
691 \DeclareLanguageAlias {North Sami}{north sami}
692 \DeclareLanguageAlias {northern sami}{north sami}
693 \DeclareLanguageAlias {Northern Sami}{northern sami}
694 \DeclareLanguageAlias {Sanskrit}{sanskrit}
695 % this one isn't needed, or is it?
696 % \DeclareLanguageAlias {Serbocroatian}{serbocroatian}
697 \DeclareLanguageAlias {serbo-croatian}{serbocroatian}
698 \DeclareLanguageAlias {Serbo-Croatian}{serbocroatian}
699 \DeclareLanguageDialect{serbian}{serbocroatian}
700 \DeclareLanguageAlias {Serbian}{serbian}

```

5 Implementation

```

701 \DeclareLanguageAlias {serbianc}{serbian}
702 \DeclareLanguageDialect{croatian}{serbocroatian}
703 \DeclareLanguageAlias {Croatian}{croatian}
704 \DeclareLanguageAlias {Slovak}{slovak}
705 \DeclareLanguageAlias {Slovenian}{slovenian}
706 \DeclareLanguageAlias {Sorbian}{sorbian}
707 \DeclareLanguageDialect{lowersorbian}{sorbian}
708 \DeclareLanguageDialect{uppersorbian}{sorbian}
709 \DeclareLanguageAlias {Lowersorbian}{lowersorbian}
710 \DeclareLanguageAlias {Uppersorbian}{uppersorbian}
711 \DeclareLanguageAlias {lsorbian}{lowersorbian}
712 \DeclareLanguageAlias {usorbian}{uppersorbian}
713 \DeclareLanguageAlias {lower sorbian}{lowersorbian}
714 \DeclareLanguageAlias {upper sorbian}{uppersorbian}
715 \DeclareLanguageAlias {Lower Sorbian}{lowersorbian}
716 \DeclareLanguageAlias {Upper Sorbian}{uppersorbian}
717 \DeclareLanguageAlias {Spanglish}{spanglish}
718 \DeclareLanguageAlias {Spanish}{spanish}
719 \DeclareLanguageAlias {Swedish}{swedish}
720 \DeclareLanguageDialect{swissgerman}{german}
721 % this is to be discussed: swiss could also be an alias of french, italian or
722 % romansh:
723 \DeclareLanguageAlias {swiss}{swissgerman}
724 \DeclareLanguageAlias {Swiss}{swiss}
725 \DeclareLanguageAlias {Swissgerman}{swissgerman}
726 \DeclareLanguageAlias {swiss german}{swissgerman}
727 \DeclareLanguageAlias {Swiss German}{swiss german}
728 \DeclareLanguageAlias {Tamil}{tamil}
729 \DeclareLanguageAlias {Telugu}{telugu}
730 \DeclareLanguageAlias {Thai}{thai}
731 \DeclareLanguageAlias {thaicjk}{thai}
732 \DeclareLanguageAlias {Thaicjk}{thaicjk}
733 \DeclareLanguageAlias {Tibetan}{tibetan}
734 \DeclareLanguageAlias {Turkish}{turkish}
735 \DeclareLanguageAlias {Turkmen}{turkmen}
736 \DeclareLanguageAlias {Ukrainian}{ukrainian}
737 \DeclareLanguageAlias {Vietnamese}{vietnamese}
738 \DeclareLanguageAlias {Welsh}{welsh}
739
740 % -----
741 % load basic dictionary if available
742 \AtBeginDocument{%
743   \@trnsLt@load@dictionary@silent@for
744     {translations-basic-dictionary}
745     {\@trnsLt@current@language}%
746 }
747
748 \endinput
749
750 % -----
751 % HISTORY:
752 2012/09/30 v0.2beta - first version (as part of the `exsheets' bundle)

```

5 Implementation

| | | |
|-----|-------------------|---|
| 753 | 2012/10/05 v0.2 | - \LoadDictionary and \LoadDictionaryFor added and loads of languages defined. |
| 754 | | |
| 755 | 2013/03/10 v0.8 | - basic dictionaries for English, German, French and Spanish |
| 756 | | - new command \DeclareDictTranslation |
| 757 | 2013/04/04 v0.8a | - bug fix in \DeclareDictTranslation |
| 758 | 2013/04/07 v0.9 | - slightly improved messages |
| 759 | 2013/04/08 v0.9a | - changed fallback warning into info |
| 760 | | - synchronized version number with 'exsheets' until now but won't any more |
| 761 | | |
| 762 | 2013/06/22 v0.9b | - added Swiss |
| 763 | 2013/06/28 v0.10 | - declaring aliases of dialects now works as expected |
| 764 | | - declaring dialects of an alias now correctly declares the dialect to the correct base language |
| 765 | | - corrected a few erroneous language declarations |
| 766 | | |
| 767 | 2013/07/12 v0.10a | - \GetTranslation gets two-folded fallback: use fallback-translation if no translation for the current language has been defined; use literal string if /no/ language is used - this should never happen but /will/ happen if neither 'babel' nor 'polyglossia' have been loaded, i.e., no language has been chosen /and/ the package writer did not provide an English translation |
| 768 | | |
| 769 | | |
| 770 | | |
| 771 | | |
| 772 | | |
| 773 | | |
| 774 | 2013/07/16 v1.0 | - removed from 'exsheets' bundle - 'translations' should be a package of it's own |
| 775 | | |
| 776 | | - load basic dictionary automatically if available |
| 777 | | - rudimentary check in \LoadDictionary if loaded file is a dictionary |
| 778 | | |
| 779 | | - new command \PrintDictionaryFor |
| 780 | | - redefined conditionals; they still seemed to make trouble in some cases |
| 781 | | |
| 782 | 2013/07/19 v1.0a | - added /loads/ of languages, noe the list of babel and polyglossia languages hopefully is complete |
| 783 | | |
| 784 | | - a few languages had falsely been declared as dialect instead of an alias |
| 785 | | |

Index

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

| | | | |
|-----------------------------|---------|------------------------|---|
| B | | | |
| babel | 1–4, 6 | | |
| beamer | 1 | | |
| C | | | |
| \captions<language> | 1 | | |
| D | | | |
| \DeclareDictTranslation | 4 | | |
| \DeclareLanguage | 3f. | | |
| \DeclareLanguageAlias | 3 | | |
| \DeclareLanguageDialect | 3 | | |
| \DeclareTranslation | 3, 5 | | |
| \DeclareTranslationFallback | 3, 6 | | |
| Dictionaries | 6–9 | | |
| basic | 7 | | |
| own | 6 | | |
| E | | | |
| etoolbox | 2 | | |
| G | | | |
| \GetTranslation | 2 f. | | |
| | | \GetTranslationFor | 3 |
| | | \GetTranslationForWarn | 4 |
| | | \GetTranslationWarn | 4 |
| I | | | |
| Implementation | 13 | | |
| L | | | |
| Languages | 9–12 | | |
| \LoadDictionary | 4 | | |
| \LoadDictionaryFor | 4 | | |
| M | | | |
| Motivation | 1 | | |
| N | | | |
| \NewTranslation | 3 | | |
| P | | | |
| polyglossia | 1 f., 6 | | |
| \PrintDictionaryFor | 4 | | |
| \providecaptionname | 1 | | |
| \ProvideDictionaryFor | 4 | | |
| 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 | | |