TRANSLATIONS

v2.0 2022/03/28

Internationalization of LaTeX Packages

Clemens Niederberger

https://github.com/cgnieder/translations

clemens@cnltx.de

Table of Contents

Part I.
Preface

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 [TW21] (which used to be part of the beamer bundle). Internationalization is also possible with babel [Bra22] 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 [Cha21]) 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. Initially the package was part of my exsheets bundle [Nie19]. However, once I had the package available I began using it in various of my other packages and it got extended to the needs I faced there. In the end it made sense to distribute it as package of its own.

2. License

Part II.
Usage

3. 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 two types of languages: main languages (see table 2 on page 15) and language dialects (see table 3 on page 16).

TRANSLATIONS also knows language aliases (see table 4 on page 16). For the commands declaring or fetching a translation languages/dialects and their aliases are equivalent.

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

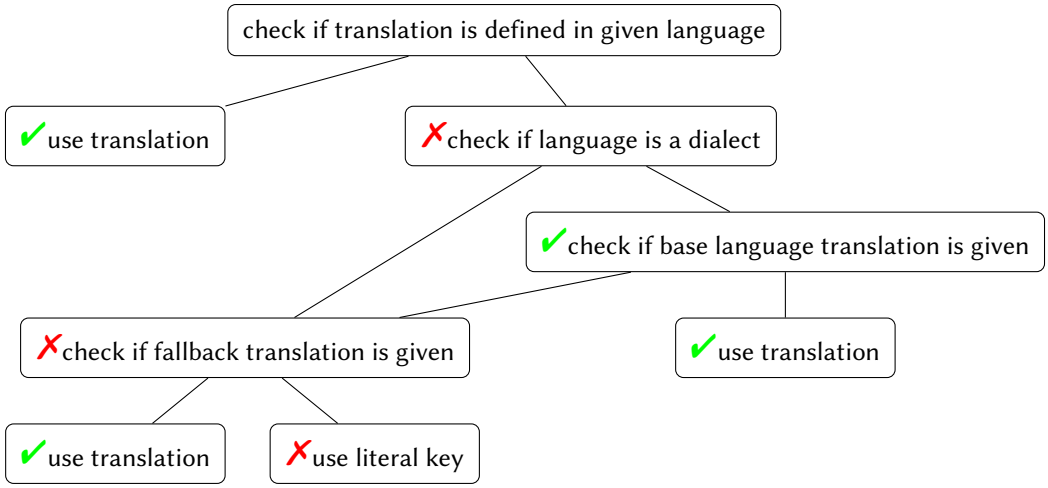


FIGURE 1: Schematic representation of TRANSLATIONS' translating mechansim.

What happens if you declare a translation?

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 4). This translation is also used for dialects of the base language if the dialect translation has not been defined.

2. You declare a translation for a dialect: this is more or less the same, except the translation is not used for any other cases.

3. You declare a translation for a language alias: this is the very same as the other cases, depending on wether its the alias of a base language or a dialect.

This means that if the document language is a base language (*English* for example) and there is a translation for a dialect (*British* for example) but not for the base language then no translation is found. If the situation is the other way around (the document language is *British* and a translation for *English* exists but not for *British*) then the translation for the base language is found and used.

! Beware that if the current language is a language using a non-latin font, a translation is missing for said language, and the fallback translation needs a Latin script font then *nothing* might be printed.

4. 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.

`\DeclareLanguage{`⟨*language*⟩`}`
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 (appendix A). Should you miss one please send me an email and I'll add it to TRANSLATIONS.

`\DeclareLanguageAlias{`⟨*language2*⟩`}{`⟨*language1*⟩`}`
Declares ⟨*language2*⟩ to be an alias of ⟨*language1*⟩. If ⟨*language1*⟩ 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 (appendix A). Should you miss one please send me an email and I'll add it to TRANSLATIONS.

`\DeclareLanguageDialect{`⟨*dialect*⟩`}{`⟨*language*⟩`}`
Declares ⟨*dialect*⟩ to be a dialect of language ⟨*language*⟩. If a translation for ⟨*dialect*⟩ is provided it is used by the translation macros. If there is none the corresponding translation for ⟨*language*⟩ is used instead. It should never be necessary to use this command as TRANSLATIONS already declares loads of languages (appendix A). Should you miss one please send me an email and I'll add it to TRANSLATIONS.

`\NewTranslation{`⟨*language*⟩`}{`⟨*key*⟩`}{`⟨*translation*⟩`}`
Defines a translation of key ⟨*key*⟩ for the language ⟨*language*⟩. An error will be raised if a translation of ⟨*key*⟩ in language `{`⟨*language*⟩`}` already exists. This command can only be used in the preamble.

`\NewTranslationFallback{`⟨*key*⟩`}{`⟨*translation*⟩`}`
Introduced in version v1.4 (2016/06/02)
Defines a fallback translation of key ⟨*key*⟩ for the language ⟨*language*⟩. An error will be raised if a fallback translation of ⟨*key*⟩ already exists. This command can only be used in the preamble.

`\RenewTranslation{`⟨*language*⟩`}{`⟨*key*⟩`}{`⟨*translation*⟩`}`
Redefines a translation of key ⟨*key*⟩ for the language ⟨*language*⟩. An error will be raised if no translation of ⟨*key*⟩ in language ⟨*language*⟩ exists. This command can only be used in the preamble.

`\RenewTranslationFallback{`⟨*key*⟩`}{`⟨*translation*⟩`}`
Introduced in version v1.4
Renews a fallback translation. This command can only be used in the preamble.

`\ProvideTranslation{`⟨*language*⟩`}{`⟨*key*⟩`}{`⟨*translation*⟩`}`
Introduced in version v1.2 (2014/01/10)
Provides a translation of key ⟨*key*⟩ for the language ⟨*language*⟩. If a translation of ⟨*key*⟩ in language ⟨*language*⟩ already exists it won't be overwritten and no error will be raised. This command can only be used in the preamble.

`\ProvideTranslationFallback{`⟨*key*⟩`}{`⟨*translation*⟩`}`
Introduced in version v1.4
Provides a fallback translation. This command can only be used in the preamble.

`\DeclareTranslation{`⟨*language*⟩`}{`⟨*key*⟩`}{`⟨*translation*⟩`}`
Defines a translation of key ⟨*key*⟩ for the language ⟨*language*⟩. No error will be raised if a translation of ⟨*key*⟩ already exists. This command can only be used in the preamble.

`\DeclareTranslationFallback{`⟨*key*⟩`}{`⟨*fallback*⟩`}`
Declares a fallback translation. This command can only be used in the preamble.

`\definetranslation{`⟨*language*⟩`}{`⟨*key*⟩`}{`⟨*translation*⟩`}`
Introduced in version v1.4
A version of `\NewTranslation` that *can* be used after begin document.

`\definetranslationfallback{`⟨*key*⟩`}{`⟨*translation*⟩`}`
Introduced in version v1.4
A version of `\NewTranslationFallback` that *can* be used after begin document.

`\redefinetranslation{`⟨*language*⟩`}{`⟨*key*⟩`}{`⟨*translation*⟩`}`
Introduced in version v1.4
A version of `\RenewTranslation` that *can* be used after begin document.

`\redefinetranslationfallback{`⟨*key*⟩`}{`⟨*translation*⟩`}`
Introduced in version v1.4
A version of `\RenewTranslationFallback` that *can* be used after begin document.

`\addtranslation{`⟨*language*⟩`}{`⟨*key*⟩`}{`⟨*translation*⟩`}`
Introduced in version v1.4
A version of `\ProvideTranslation` that *can* be used after begin document.

`\addtranslationfallback{`⟨*key*⟩`}{`⟨*translation*⟩`}`
Introduced in version v1.4
A version of `\ProvideTranslationFallback` that *can* be used after begin document.

`\declaretranslation{`⟨*language*⟩`}{`⟨*key*⟩`}{`⟨*translation*⟩`}`
Introduced in version v1.4
A version of `\DeclareTranslation` that *can* be used after begin document.

`\declaretranslationfallback{`⟨*key*⟩`}{`⟨*translation*⟩`}`
Introduced in version v1.4
A version of `\DeclareTranslationFallback` that *can* be used after begin document.

`*\IfTranslation{`⟨*language*⟩`}{`⟨*key*⟩`}{`⟨*true*⟩`}{`⟨*false*⟩`}`
Introduced in version v1.2d (2015/09/06)
Checks if a translation for ⟨*key*⟩ in language ⟨*language*⟩ is defined or not and either leaves ⟨*true*⟩ or ⟨*false*⟩ in the input stream.

`*\GetTranslationFor{`⟨*language*⟩`}{`⟨*key*⟩`}`
Fetches and prints the translation of ⟨*key*⟩ for the language ⟨*language*⟩. 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{`⟨*language*⟩`}{`⟨*key*⟩`}`
Introduced in version v1.0 (2013/07/16)
Fetches and prints the translation of ⟨*key*⟩ for the language ⟨*language*⟩. Issues a warning if no translation is available at the cost of expandability.

`\GetTranslationWarn{`⟨*key*⟩`}`
Introduced in version v1.0
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.

`\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*⟩`-`⟨*language*⟩`.trsl` where ⟨*language*⟩ corresponds to the lowercase name of the current language as defined with `\DeclareLanguage`. This file should contain the translations for the specified language.

`\LoadDictionaryFor{`⟨*language*⟩`}{`⟨*name*⟩`}`
Loads a file named ⟨*name*⟩`-`⟨*language*⟩`.trsl`.

`\NewDictTranslation{`⟨*key*⟩`}{`⟨*translation*⟩`}`
Introduced in version v0.10 (2013/06/28)
This command is to be used in a dictionary file and picks up the language of that file. Issues an error if either the translation for the ⟨*key*⟩ or the dictionary entry for the ⟨*key*⟩ already exists.

`\RenewDictTranslation{`⟨*key*⟩`}{`⟨*translation*⟩`}`
Introduced in version v0.10
This command is to be used in a dictionary file and picks up the language of that file. Issues an error if either the translation for the ⟨*key*⟩ or the dictionary entry for the ⟨*key*⟩ doesn't exist.

`\ProvideDictTranslation{`⟨*key*⟩`}{`⟨*translation*⟩`}`
Introduced in version v0.10
This command is to be used in a dictionary file and picks up the language of that file. Only defines the translation and adds a corresponding dictionary entry if they don't exist yet. This command is used in the dictionaries that a part of TRANSLATIONS.

`\DeclareDictTranslation{`⟨*key*⟩`}{`⟨*translation*⟩`}`
This command is to be used in a dictionary file and picks up the language of that file, see section 7 for an example. Defines the translation and adds a dictionary entry regardless if they exist or not.

`\ProvideDictionaryFor{`⟨*language*⟩`}{`⟨*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`.

`\PrintDictionary{`⟨*language*⟩`}{`⟨*name*⟩`}{`⟨*pre*⟩`}{`⟨*mid*⟩`}{`⟨*post*⟩`}`
Introduced in version v1.0
Prints all entries of dictionary ⟨*name*⟩ in language ⟨*language*⟩ 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. There is probably only a very limited number of use cases for this command. (It was for example used to print table 1.)

`*\baselanguage{`⟨*language*⟩`}`
Changed in version v1.2a (2014/01/23)
Returns the (internal) base name of the given language, language alias or language dialect. For a dialect this expands to the name of language it is a dialect of. For a base language (see section A.1) this usually simply is the lowercase version of the name:

`\baselanguage{english}` ↦ english
`\baselanguage{English}` ↦ english
`\baselanguage{American}` ↦ english
`\baselanguage{USenglish}` ↦ english

* `\thelanguage{`⟨*language*⟩`}`

☆ New

This is the internal name of a language. For base languages and their aliases this is the same as `\baselanguage`, for dialects it returns the base dialect name:

`\thelanguage{english}` ↦ english
`\thelanguage{English}` ↦ english
`\thelanguage{American}` ↦ american
`\thelanguage{USenglish}` ↦ american

* `\ifcurrentlanguage{`⟨*language*⟩`}{`⟨*true*⟩`}{`⟨*false*⟩`}`

Introduced in version v1.2

Places ⟨*true*⟩ in the input stream if the current language is ⟨*language*⟩. Note: a dialect counts as a language of it's own here. `\ifcurrentlanguage{English}` will for example be ⟨*false*⟩ if the current babel language is `american`.

* `\ifcurrentlang{`⟨*language*⟩`}`

Introduced in version v1.9 (2020/11/08)

The same as `\ifcurrentlanguage` but uses the …`\else`…`\fi` syntax.

* `\ifcurrentbaselanguage{`⟨*language*⟩`}{`⟨*true*⟩`}{`⟨*false*⟩`}`

Introduced in version v1.2

Places ⟨*true*⟩ in the input stream if the current language is ⟨*language*⟩. Note: a dialect does not count as a language of it's own here. If the current babel language is `american` then `\ifcurrentbaselanguage{English}` will be ⟨*true*⟩.

* `\ifcurrentbaselang{`⟨*language*⟩`}`

Introduced in version v1.9

The same as `\ifcurrentbaselanguage` but uses the …`\else`…`\fi` syntax.

5. 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.

```
\declaretranslation{English}{Kueche}{kitchen}
\declaretranslation{German}{Kueche}{K\"uche}
\declaretranslation{Spanish}{Kueche}{cocina}
\declaretranslation{French}{Kueche}{cuisine}

\GetTranslation{Kueche}
\SaveTranslation\kitchen{Kueche}
\SaveTranslationFor\cuisine{french}{Kueche}

\selectlanguage{ngerman}
\GetTranslation{Kueche} \kitchen\ \GetTranslationFor{spanish}{Kueche}
```

```
\cuisine

\IfTranslation{German}{Kueche}{true}{false} \par
\IfTranslation{Danish}{Kueche}{true}{false}
```

kitchen
Küche kitchen cocina cuisine
true
false

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:

```
\declaretranslation{English}{farbe}{color}
\declaretranslation{British}{farbe}{colour}

\GetTranslationFor{English}{farbe}
\GetTranslationFor{British}{farbe}
\GetTranslationFor{American}{farbe}
```

color colour color

6. Usage in Packages

6.1. Basic Structure

A typical usage in a package would look as follows:

```
\RequirePackage{translations}
\DeclareTranslationFallback{mypackage-title}{Nice Title}
\DeclareTranslation{English}{mypackage-title}{Nice Title}
\DeclareTranslation{French}{mypackage-title}{Beau Titre}
\DeclareTranslation{German}{mypackage-title}{Sch\"{o}ner Titel}
...
\newcommand*\mypackage@title{\GetTranslation{mypackage-title}}
```

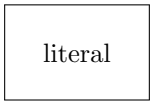
That is, a package defines some unique key for an expression and at least defines a fallback translation. Additionally translations for as many languages as the author wants are defined. A user then may add `\DeclareTranslation{`⟨*language*⟩`}{`⟨*translation*⟩`}` if they find their translation missing.

6.2. The 'fallback' language

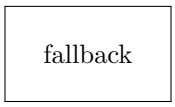
If a user has neither loaded babel nor polyglossia TRANSLATIONS will use English as language and translate to English if the translation was provided. If the user *has* loaded one of the language packages but has chosen a language for which no translation is defined the language 'fallback' will be used,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

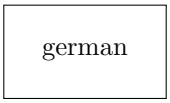
```
\documentclass[margin=5mm]{standalone}
\usepackage{translations}
\DeclareTranslation{German}{literal}{german}
\begin{document}
\GetTranslation{literal} % literal
\end{document}
```




```
\documentclass[margin=5mm]{standalone}
\usepackage{translations}
\DeclareTranslationFallback{literal}{fallback}
\DeclareTranslation{German}{literal}{german}
\begin{document}
\GetTranslation{literal} % fallback
\end{document}
```




```
\documentclass[margin=5mm]{standalone}
\usepackage[ngerman]{babel}
\usepackage{translations}
\DeclareTranslationFallback{literal}{fallback}
\DeclareTranslation{German}{literal}{german}
\begin{document}
\GetTranslation{literal} % german
\end{document}
```



7. Dictionaries

7.1. Background

TRANSLATIONS provides the means to write dictionary files that can be loaded by packages or in a document. Dictionaries can be loaded for the currently active language with `\LoadDictionary` or for a specific language with `\LoadDictionaryFor`.

`\LoadDictionary{`⟨*name*⟩`}`
Tells TRANSLATIONS to load a file named ⟨*name*⟩`-`⟨*language*⟩`.trsl` where ⟨*language*⟩ corresponds of the document language as given by `\languagename` at begin document. This file should contain the translations for the specified language.

`\LoadDictionaryFor{`⟨*language*⟩`}{`⟨*name*⟩`}`
Tells TRANSLATIONS to load a file named ⟨*name*⟩`-`⟨*language*⟩`.trsl` where ⟨*language*⟩ corresponds to the lowercase name of the current language or base language as defined with `\DeclareLanguage`. This file should contain the translations for the specified language. ⟨*language*⟩ can either be a base language or a dialect.

A package could provide dictionary files for its language dependent settings and include the needed one. The basics for creating a dictionary file are explained in section 7.2.

TRANSLATIONS already provides a few basic dictionary files. If the main document language fits to one of the provided files the corresponding basic dictionary is loaded at begin document by TRANSLATIONS, see section 7.3 for more on this.

7.2. Own Dictionaries

A typical dictionary file should look as follows:

```
% this is file housing-german.trsl
\ProvideDictionaryFor{German}{housing}[<version info>]
\ProvideDictTranslation{kitchen (housing)}{K\"uche}
\ProvideDictTranslation{bathroom (housing)}{Bad}
\ProvideDictTranslation{living room (housing)}{Wohnzimmer}
\ProvideDictTranslation{bedroom (housing)}{Schlafzimmer}
...
\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. Translations can be declared by one of the following commands:

`\NewDictTranslation{⟨key⟩}{⟨translation⟩}`

Introduced in version v0.10

This command is to be used in a dictionary file and picks up the language of that file. Issues an error if either the translation for the ⟨*key*⟩ or the dictionary entry for the ⟨*key*⟩ already exists.

`\RenewDictTranslation{⟨key⟩}{⟨translation⟩}`

Introduced in version v0.10

This command is to be used in a dictionary file and picks up the language of that file. Issues an error if either the translation for the ⟨*key*⟩ or the dictionary entry for the ⟨*key*⟩ doesn't exist.

`\ProvideDictTranslation{⟨key⟩}{⟨translation⟩}`

Introduced in version v0.10

This command is to be used in a dictionary file and picks up the language of that file. Only defines the translation and adds a corresponding dictionary entry if they don't exist yet. This command is used in the dictionaries that a part of TRANSLATIONS.

`\DeclareDictTranslation{⟨key⟩}{⟨translation⟩}`

This command is to be used in a dictionary file and picks up the language of that file, see section 7 for an example. Defines the translation and adds a dictionary entry regardless if they exist or not.

Every dictionary file *must* contain the declaration `\ProvideDictionaryFor`:

`\ProvideDictionaryFor{⟨language⟩}{⟨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 `\NewDictTranslation` or similar commands.

7.3. TRANSLATIONS' Basic Dictionaries

TRANSLATIONS already provides a basic dictionary for the languages

- Brazilian (since version 1.9),
- Catalan (since version 1.5),
- English,
- Dutch (since version 1.5),
- French,
- German,
- Polish (since version 1.12), and
- Spanish.

The corresponding dictionary[1] is loaded automatically if the document language is one of these languages.

1. Or dictionaries if more than one of these languages are loaded in a document. This works since v0.18.

! *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-`⟨*language*⟩`.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
`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
`figure`	Abbildung
`From`	Von

continues

key	translation
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
`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

Part III.
Appendix

A. Defined Languages

A.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	azerbaijani	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	korean	ladin	lao
latin	latvian	lithuanian	macedonian	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	british	australian	american	acadian
canadien	canadian	newzealand	nynorsk	irish
scottish	austrian	hindi	urdu	indonesian
brazil	serbian	croatian	lowersorbian	uppersorbian
swissgerman	swissfrench	swissitalian	swissromansh	

A.2. Language Dialects

TRANSLATIONS also defines a number of 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
british	english	australian	english
american	english	acadian	french
canadien	french	canadian	english
newzealand	english	nynorsk	norwegian
irish	gaelic	scottish	gaelic
austrian	german	hindi	hindustani
urdu	hindustani	indonesian	malay
brazil	portuges	serbian	serbocroatian
croatian	serbocroatian	lowersorbian	sorbian
uppersorbian	sorbian	swissgerman	german
swissfrench	french	swissitalian	italian
swissromansh	romansh		

A.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	asturian	asturian-leonese	asturian

continues

alias	language	alias	language
Asturian-Leonese	asturian	Azerbaijani	azerbaijani
Basque	basque	Bengali	bengali
Breton	breton	Bulgarian	bulgarian
Catalan	catalan	Coptic	coptic
copticegyptian	coptic	CopticEgyptian	coptic
Czech	czech	Danish	danish
Dutch	dutch	Farsi	farsi
Finnish	finnish	francais	french
Francais	french	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	irishgaelic	irish
IrishGaelic	irish	Scottish	scottish
scottishgaelic	scottish	ScottishGaelic	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	hindustani
Hindi	hindi	Urdu	urdu
Hungarian	hungarian	magyar	hungarian
Magyar	hungarian	Icelandic	icelandic
Interlingua	interlingua	Italian	italian
Japanese	japanese	Kannada	kannada
Korean	korean	Ladin	ladin
Lao	lao	laotian	lao
Laotian	lao	Latin	latin
Latvian	latvian	lettish	latvian
Lettish	latvian	Lithuanian	lithuanian
Macedonian	macedonian	Malay	malay
bahasamalaysia	malay	BahasaMalaysia	malay
bahasamelayu	malay	BahasaMelayu	malay
bahasa	malay	Bahasa	malay

continues

alias	language	alias	language
bahasai	malay	Bahasai	malay
bahasam	malay	Bahasam	malay
Indonesian	indonesian	indon	indonesian
Malayalam	malayalam	Maldivian	maldivian
divehi	maldivian	Divehi	maldivian
Marathi	marathi	Mongolian	mongolian
norsk	norwegian	Norsk	norwegian
Norwegian	norwegian	Nynorsk	nynorsk
Occitan	occitan	lengad'oc	occitan
langued'oc	occitan	Piedmontese	piedmontese
piemontese	piedmontese	Piemontese	piedmontese
piemonteis	piedmontese	Piemonteis	piedmontese
Pinyin	pinyin	Polish	polish
Brazil	brazil	brazilian	brazil
Brazilian	brazil	Portuges	portuges
portuguese	portuges	Portuguese	portuges
Romanian	romanian	Romansh	romansh
Romansch	romansh	Rumantsh	romansh
Rumantsch	romansh	Romanche	romansh
Russian	russian	Samin	samin
northsami	samin	NorthSami	samin
northernsami	samin	NorthernSami	samin
Sanskrit	sanskrit	Serbocroatian	serbocroatian
serbo-croatian	serbocroatian	Serbo-Croatian	serbocroatian
Serbian	serbian	serbianc	serbian
Croatian	croatian	Slovak	slovak
Slovenian	slovenian	slovene	slovenian
Slovene	slovenian	Sorbian	sorbian
Lowersorbian	lowersorbian	Uppersorbian	uppersorbian
lsorbian	lowersorbian	usorbian	uppersorbian
lowersorbian	lowersorbian	uppersorbian	uppersorbian
lowerSorbian	lowersorbian	upperSorbian	uppersorbian
LowerSorbian	lowersorbian	UpperSorbian	uppersorbian
Spanglish	spanglish	Spanish	spanish
Swedish	swedish	Swissgerman	swissgerman
swissgerman	swissgerman	SwissGerman	swissgerman
Swissfrench	swissfrench	swissfrench	swissfrench
SwissFrench	swissfrench	Swissitalian	swissitalian
swissitalian	swissitalian	SwissItalian	swissitalian
Swissromansh	swissromansh	swissromansh	swissromansh
SwissRomansh	swissromansh	swiss	swissgerman

continues

alias	language	alias	language
Swiss	swissgerman	Tamil	tamil
Telugu	telugu	Thai	thai
thaicjk	thai	Thaicjk	thai
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.

B. Code

☆ New

This section lists all expl3 functions and variables of TRANSLATIONS, possibly for use in other packages. The symbol * means that the command is expandable. Commands without the marker aren't expandable.

B.1. Package

*\c_trnslt_date_tl
Holds the current release date: 2022/03/28

*\c_trnslt_version_major_number_tl
Holds the current major version number: 2

*\c_trnslt_version_minor_number_tl
Holds the current minor version number: 0

*\c_trnslt_version_subrelease_tl
Holds the current subrelease version (possibly empty):

*\c_trnslt_version_number_tl
Holds the current version number: 2.0

*\c_trnslt_version_tl
Holds the current version: 2.0

*\c_trnslt_info_tl
Holds the package info text: internationalization of LaTeX packages

B.2. Variables

\g_trnslt_languages_seq
A sequence variable that holds all defined base languages.

`\g_trnslt_aliases_seq`
A sequence variable that holds all defined aliases of languages.

`\g_trnslt_aliases_prop`
A property list variable that maps aliases to their base.

`\g_trnslt_dialects_seq`
A sequence variable that holds all defined dialects.

`\g_trnslt_dialects_prop`
A property list variable that maps dialects to their base.

`\g_trnslt_translations_prop`
A property list variable that holds all defined translations. The key always has the form ⟨*word*⟩|⟨*language*⟩.

* `\l_trnslt_dictionary_name_tl`
A token list variable that holds the name of a dictionary within a dictionary.

* `\l_trnslt_dictionary_language_tl`
A token list variable that holds the language of a dictionary within a dictionary.

`\l_trnslt_loaded_languages_clist`
A clist variable that holds the languages from

`\bbl@loaded` (from package babel [Bra22]), or
`\xpg@loaded` (from package polyglossia [Cha21]), or
`\@tracklang@languages` and `\@tracklang@dialects` (from package tracklang [Tal20]),

or a combined set thereof, *after* begin document.

B.3. Defining Functions

`\trnslt_language_declare:n` {⟨*language*⟩}
Defines language ⟨*language*⟩.

`\trnslt_language_new:n` {⟨*language*⟩}
Defines language ⟨*language*⟩. Raises an error if ⟨*language*⟩ already is defined.

`\trnslt_dialect_declare:nn` {⟨*dialect*⟩} {⟨*language*⟩}
Defines dialect as new language and as dialect of ⟨*language*⟩.

`\trnslt_alias_declare:nn` {⟨*alias*⟩} {⟨*language*⟩}
Defines language ⟨*alias*⟩ and defines it to be an alias of ⟨*language*⟩.

`\trnslt_translation_declare:nnn` {⟨*word*⟩} {⟨*language*⟩} {⟨*replacement*⟩}
Defines the translation of ⟨*word*⟩ in ⟨*language*⟩ to be ⟨*replacement*⟩.

`\trnslt_translation_new:nnn` {⟨*word*⟩} {⟨*language*⟩} {⟨*replacement*⟩}
Defines the translation of ⟨*word*⟩ in ⟨*language*⟩ to be ⟨*replacement*⟩. Raises an error if the translation of ⟨*word*⟩ in ⟨*language*⟩ already is defined.

`\trnslt_translation_renew:nnn` {⟨*word*⟩} {⟨*language*⟩} {⟨*replacement*⟩}
Redefines the translation of ⟨*word*⟩ in ⟨*language*⟩ to be ⟨*replacement*⟩. Raises an error if the translation of ⟨*word*⟩ in ⟨*language*⟩ already is not defined, yet.

`\trnslt_translation_provide:nnn` {⟨*word*⟩} {⟨*language*⟩} {⟨*replacement*⟩}
Defines the translation of ⟨*word*⟩ in ⟨*language*⟩ to be ⟨*replacement*⟩ only if the translation of ⟨*word*⟩ in ⟨*language*⟩ already is not defined, yet.

`\trnslt_translations_declare:nn` {⟨*word*⟩} {⟨*list of key/value pairs*⟩}
Declares the translations of ⟨*word*⟩ for several languages at once. Translations should be given as a csv list of ⟨*language*⟩ = ⟨*replacement*⟩ pairs.

`\trnslt_language_translations_declare:nn` {⟨*language*⟩} {⟨*list of key/value pairs*⟩}
Declares the translations of several words in ⟨*language*⟩ at once. Translations should be given as a csv list of ⟨*word*⟩ = ⟨*replacement*⟩ pairs.

`\trnslt_dictionary_translation_declare:nn` {⟨*word*⟩} {⟨*replacement*⟩}
Declares the translation of ⟨*word*⟩ in the current dictionary language to be ⟨*replacement*⟩. Should only be used inside of a dictionary file.

`\trnslt_dictionary_translation_new:nn` {⟨*word*⟩} {⟨*replacement*⟩}
Declares the translation of ⟨*word*⟩ in the current dictionary language to be ⟨*replacement*⟩. Raises an error if the tranlation already exists. Should only be used inside of a dictionary file.

`\trnslt_dictionary_translation_renew:nn` {⟨*word*⟩} {⟨*replacement*⟩}
Declares the translation of ⟨*word*⟩ in the current dictionary language to be ⟨*replacement*⟩. Raises an error if the tranlation does not exist, yet. Should only be used inside of a dictionary file.

`\trnslt_dictionary_translation_provide:nn` {⟨*word*⟩} {⟨*replacement*⟩}
Declares the translation of ⟨*word*⟩ in the current dictionary language to be ⟨*replacement*⟩ only if the tranlation does not exist, yet. Should only be used inside of a dictionary file.

B.4. Utility Commands

* `\trnslt_translate:nn` {⟨*word*⟩} {⟨*language*⟩}
Gets the translation of ⟨*word*⟩ in ⟨*language*⟩ as described for `\GetTranslation`.

`\trnslt_translation_get:Nnn` ⟨*tl var*⟩ {⟨*word*⟩} {⟨*language*⟩}
Gets the translation of ⟨*word*⟩ in ⟨*language*⟩ and assigns it to ⟨*tl var*⟩.

* `\trnslt_language:n` {⟨*language or alias*⟩}
Get the base name of a language.

* `\trnslt_dialect:n` {⟨*dialect or alias*⟩}
  Gets the base name of a dialect.

* `\trnslt_language_base:n` {⟨*language or dialect or alias thereof*⟩}
  Returns the base language of the input.

`\trnslt_load_dictionary:n` {⟨*name*⟩}
  Tells TRANSLATIONS to load the dictionary ⟨*name*⟩ for the document language. *Can only be used before begin document.*

`\trnslt_load_dictionary:nn` {⟨*name*⟩} {⟨*language*⟩}
  Tells TRANSLATIONS to load the dictionary ⟨*name*⟩ for ⟨*language*⟩. *Can only be used before begin document.*

`\trnslt_show_translations:`
  Writes all defined translations to the log and the terminal.

`\trnslt_show_dictionary:nn` {⟨*name*⟩} {⟨*language*⟩}
  Writes all defined translations of dictionary ⟨*name*⟩ in ⟨*language*⟩ to the log and the terminal. *For this the corresponding dictionary must have been loaded before. This means it only works after begin document.*

`\trnslt_map_languages:n` {⟨*code*⟩}
  Maps over all base languages. Within ⟨*code*⟩ `#2` refers to the language name, and `#1` to the index number.

`\trnslt_map_dialects:n` {⟨*code*⟩}
  Maps over all dialects. Within ⟨*code*⟩ `#2` refers to the dialect, `#3` to the base language, and `#1` to the index number.

`\trnslt_map_aliases:n` {⟨*code*⟩}
  Maps over all dialects. Within ⟨*code*⟩ `#2` refers to the alias, `#3` to the base language, and `#1` to the index number.

`\trnslt_map_dictionary:nnn` {⟨*name*⟩} {⟨*language*⟩} {⟨*code*⟩}
  Maps over all entries of dictionary ⟨*name*⟩ in ⟨*language*⟩. Within ⟨*code*⟩ `#2` refers to the word, `#3` to its translation, and `#1` to the index number. *For this the corresponding dictionary must have been loaded before. This means it only works after begin document.*

B.5. Conditionals

In this section *TF* means that a `T`, `F`, and a `TF` variant exists. For each expandable conditional also a predicate version exists.

* `\trnslt_language_if_exist:n`*TF* {⟨*language*⟩} {⟨*true*⟩} {⟨*false*⟩}
  Tests if the language ⟨*language*⟩ is defined or not and either leaves ⟨*true*⟩ or ⟨*false*⟩ in the input stream.

* `\trnslt_dialect_if_exist:nTF` {⟨*dialect*⟩} {⟨*true*⟩} {⟨*false*⟩}
  Tests if the language ⟨*dialect*⟩ is a dialect or not and either leaves ⟨*true*⟩ or ⟨*false*⟩ in the input stream.

* `\trnslt_alias_if_exist:nTF` {⟨*alias*⟩} {⟨*true*⟩} {⟨*false*⟩}
  Tests if the language ⟨*alias*⟩ is an alias or not and either leaves ⟨*true*⟩ or ⟨*false*⟩ in the input stream.

* `\trnslt_if_language:nTF` {⟨*language*⟩} {⟨*true*⟩} {⟨*false*⟩}
  Tests if the current language is ⟨*language*⟩ or not and either leaves ⟨*true*⟩ or ⟨*false*⟩ in the input stream.

* `\trnslt_if_language_base:nTF` {⟨*language*⟩} {⟨*true*⟩} {⟨*false*⟩}
  Tests if the current base language is ⟨*language*⟩ or not and either leaves ⟨*true*⟩ or ⟨*false*⟩ in the input stream.

* `\trnslt_translation_if_exist:nnTF` {⟨*word*⟩} {⟨*language*⟩} {⟨*true*⟩} {⟨*false*⟩}
  Tests if the translation of ⟨*word*⟩ in ⟨*language*⟩ has been defined or not and either leaves ⟨*true*⟩ or ⟨*false*⟩ in the input stream.

C. Version History

v0.2beta (2012/09/30)

- first version (as part of the exsheets bundle)

v0.2 (2012/10/05)

- `\LoadDictionary` and `\LoadDictionaryFor` added and loads of languages defined.

v0.8 (2013/03/10)

- basic dictionaries for English, German, French and Spanish
- new command `\DeclareDictTranslation`

v0.8a (2013/04/04)

- bug fix in `\DeclareDictTranslation`

v0.9 (2013/04/07)

- slightly improved messages

v0.9a (2013/04/08)

- changed fallback warning into info
- synchronized version number with exsheets until now but won't any more

v0.9b (2013/06/22)

- added Swiss

v0.10 (2013/06/28)

- declaring aliases of dialects now works as expected
- declarings dialects of an alias now correctly declares the dialect to the correct base language
- corrected a few erroneous language declarations

v0.10a (2013/07/12)

- `\GetTranslation` gets two-fold 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 we all know that it *will* happen eventually

v1.0 (2013/07/16)

- removed from exsheets bundle – TRANSLATIONS should be a package of it's own
- load basic dictionary automatically if available
- rudimentary check in `\LoadDictionary` if loaded file is a dictionary
- new command `\PrintDictionaryFor`
- redefined conditionals; they still seemed to make trouble in some cases

v1.1 (2013/08/05)

- added *loads* of languages, now the list of babel and polyglossia languages hopefully is complete
- a few languages had falsely been declared as dialect instead of an alias
- added weekday names and month names to basic dictionary
- new command `\baselanguage`
- new commands `\GetLCTranslation`, `\GetLCTranslationFor`, `\GetLCTranslationWarn` and `\GetLCTranslationForWarn`
- load basic dictionary also for dialects and if it doesn't exist load it for the corresponding base language instead

v1.1a (2013/09/30)

- Bug fix in `\NewTranslation` und `\RenewTranslation`

v1.2 (2014/01/10)

- `\ifcurrentlanguage`, `\ifcurrentbaselanguage`
- require cnltx-base
- change the "no language package" warning into an info

- \ProvideTranslation
- \NewDictTranslation, \RenewDictTranslation, \ProvideDictTranslation
- translations in dictionaries are provided
- \baselanguagename

v1.2a (2014/01/23)

- add \detokenize so that translation keys with non-ascii chars can safely be used
- fix bug in dialect declaration
- rename \baselanguagename into \baselanguage
- drop earlier \baselanguage (it was and still is available as \@trnslt@language)

v1.2b (2015/07/09)

- add Korean

v1.2c (2015/08/29)

- fix bug in \@trnslt@save@translation@for

v1.2d (2015/09/06)

- add alias slovene for slovenian
- add user command for \@trnslt@if@translation

v1.2e (2015/11/07)

- Some fixes to the French translations in the basic dictionary, thanks to Denis Bitouzé
- add Macedonian language

v1.3 (2016/04/19)

- \LoadDictionary first checks if a dictionary for the dialect exists and loads it if it does; else it looks for a dictionary of the base language and loads that instead
- New command \LoadDictionaryForDialect which only loads the dictionary for a specified dialect – this doesn't check if a dictionary for a base language exists

v1.4 (2016/06/02)

- new commands
  - \declaretranslation
  - \newtranslation
  - \renewtranslation
  - \providetranslation

  which can be used after begin document
- fallback versions for each definition command

v1.4a (2016/06/02)

- rename new commands: they conflict with the translator package!

v1.5 (2017/03/03)

- remove dependency on cnltx-base
- add Dutch dictionary (thanks to kwikwi)
- add Catalan dictionary (thanks to kwikwi)

v1.5a (2017/03/05)

- correct typos in Dutch dictionary

v1.5b (2017/04/24)

- correct typos in Malaysian language names

v1.5c (2017/05/06)

- add Azerbaijani

v1.6 (2017/05/16)

- allow translations to contain `\par`
- improve perfomance by replacing list checks

v1.6a (2017/07/03)

- fix issue #6 (bug in `\@trnslt@save@translation@for`)

v1.7 (2017/07/06)

- return translation values in `\unexpanded`

v1.7a (2017/08/31)

- add missing dependency on pdftexcmds

v1.8 (2020/02/28)

- load basic dictionary for each loaded language (if available)

v1.8a (2020/04/26)

- fix problem with loading of dictionaries in some circumstances

v1.8b (2020/04/28)

- fix issue #9

v1.9 (2020/11/08)

- add \ifcurrentlang{⟨*lang*⟩}...\else...\fi
- add \ifcurrentbaselang{⟨*lang*⟩}...\else...\fi
- add Brazilian basic dictionary

v1.10 (2021/01/16)

- remove scrlfile dependency

v1.10a (2021/01/16)

- correct bug from last update

v1.11 (2022/01/04)

- fix issue #18
- undo v1.7, fixes issue #12
- make nynorsk a dialect of norsk, fixes issue #19
- don't define the literal string as command but output it directly, fixes issue #13
- correct some Dutch translations

v1.12 (2022/02/05)

- add Polish translations, thanks to Jacub Kaczor

v2.0 (2022/03/28)

- package re-written in expl3
- new commands:
  - \trnslt_show_translations:
  - \trnslt_show_dictionary:nn
- new commands:
  - \trnslt_map_languages:n
  - \trnslt_map_dialects:n
  - \trnslt_map_aliases:n
  - \trnslt_map_dictionary:nnn
- new commands:
  - \trnslt_translations_declare:nn
  - \trnslt_language_translations_declare:nn
- removed commands:
  - \PrintDictionaryFor

- \LoadDictionaryForDialect
- \GetLCTranslationFor
- \GetLCTranslation
- \GetLCTranslationForWarn
- \GetLCTranslationWarn

D. References

[Bra22] Johannes Braams, current maintainer: Javier Bezos.
babel. version 3.70, Jan. 26, 2022 (or newer).
URL: https://www.ctan.org/pkg/babel/.

[Cha21] François Charette, current maintainer: Arthur Reutenauer.
polyglossia. version 1.53, Apr. 12, 2021 (or newer).
URL: https://www.ctan.org/pkg/polyglossia/.

[Nie19] Clemens Niederberger. exsheets. version 0.21k, Sept. 30, 2019 (or newer).
URL: https://ctan.org/pkg/exsheets.

[Tal20] Nicola L. C. Talbot. tracklang. version 1.5, June 30, 2020 (or newer).
URL: https://www.ctan.org/pkg/tracklang/.

[TW21] Till Tantau and Joseph Wright.
translator. version 1.12d, May 31, 2021 (or newer).
URL: https://www.ctan.org/pkg/translator/.

E. Index

T

W