Package 'msgtools'

March 2, 2017

Type Package

Title Tools for Developing Diagnostic Messages

Description A number of utilities for developing and maintaining error, warning, and other messages in R packages, including checking for consistency across messages, spell-checking messages, and building message translations into various languages for purposes of localization.

Date 2017-02-28

URL https://github.com/RL10N/msgtools/

BugReports https://github.com/RL10N/msgtools/issues

License MIT + file LICENSE

Version 0.2.7

Depends R (>= 3.2.0)

Imports tools, utils, poio (>= 0.0-3), digest, tibble, hunspell, devtools

Suggests testthat, knitr, rmarkdown

SystemRequirements GNU gettext

VignetteBuilder knitr

RoxygenNote 6.0.1

NeedsCompilation no

Author Thomas J. Leeper [aut, cre]

Maintainer Thomas J. Leeper <thosjleeper@gmail.com>

Repository CRAN

Date/Publication 2017-03-02 08:09:35

R topics documented:

`check_for_gettext` *'gettext' availability*

Description

Check whether 'gettext' is available on your system; and install for Windows.

Usage

```
check_for_gettext()

install_gettext(location)
```

Arguments

`location`	A character string specifying a file path where the 'gettext' binaries should be installed.

Details

`install_gettext` provides a Windows-only installer into the specified directory. That directory should be one included in `PATH` so that the binaries are available.

Value

A logical.

Note

Windows binaries for 'gettext' can also be installed from http://www.stats.ox.ac.uk/pub/Rtools/goodies/gettext-tools.zip or https://github.com/mlocati/gettext-iconv-windows.

See Also

use_localization

Examples

```
## Not run:
  install_gettext()

## End(Not run)
check_for_gettext()
```

`check_translation` *Install translations*

Description

Check and install message translations

Usage

```
check_translation(language, pkg = ".", domain = "R", strictPlural = FALSE,
  verbose = getOption("verbose"))

check_translations(pkg = ".", strictPlural = FALSE,
  verbose = getOption("verbose"))

install_translations(pkg = ".", verbose = getOption("verbose"))
```

Arguments

`language`	A character string specifying a language.
`pkg`	The directory of an R package to extract diagnostic messages from. Path is passed to `as.package`.
`domain`	A character string specifying the "domain" of the messages. Either "R" (the default) or "C". This is case insensitive.
`strictPlural`	A logical indicating whether to compare formats of singular and plural forms in a strict way. See `checkPoFile`.
`verbose`	A logical. Should the function be chatty?

Details

`check_translation` checks a specific translation file. `check_translations` checks all translations in the /po directory. These are just wrappers around `checkPoFile`.

`install_translations` performs a reduced set of the functionality described in `update_pkg_po`.

Value

`check_translation` returns an object of class "check_po_files"; `check_translations` returns a list of such objects. `install_translations` returns a logical TRUE, if successful.

Note

These functions require that gettext is installed on your system.

See Also

use_localization, make_translation

Examples

```
## Not run:
  # check translations
  check_translations()

  # install translations
  install_translations()

## End(Not run)
```

---

dummy_pkg *Dummy package creation*

Description

Create a simple example package in the temporary directory

Usage

```
dummy_pkg(location = tempdir(), messages = TRUE)
```

Arguments

location	A character string containing a path to a directory in which to create the package, by default the location of tempdir.
messages	A logical indicating whether to include some dummy translatable messages. Default TRUE.

Value

A string denoting a path to the root of the package (called “translateme”).

edit_translation *Interactive translation editing*

Description

Edit a translation object in-memory using a simple interface

Usage

```
edit_translation(translation, language, pkg = ".", domain = "R")
```

Arguments

translation	An object of class "po", such as returned by make_translation. In lieu of translation, language and pkg and domain can be specified, which will load the translation from file using read_translation.
language	A character string specifying a language, as either: (1) "ll", an ISO 639 two-letter language code (lowercase), or (2) "ll_CC", an ISO 639 two-letter language code (lowercase) and 'CC' is an ISO 3166 two-letter country code (uppercase).
pkg	The directory of an R package to extract diagnostic messages from. Path is passed to as.package.
domain	A character string specifying the "domain" of the messages. Either "R" (the default) or "C". This is case insensitive.

Value

An object of class "po".

Note

Emacs users may find the gettext PO file editor more comfortable: https://www.gnu.org/software/gettext/manual/gettext.html#PO-Mode.

Author(s)

Thomas J. Leeper

See Also

make_translation

Examples

```
## Not run:
  # setup pkg for localization
  pkg <- dummy_pkg()
  use_localization(pkg = pkg)

  # generate translation in memory
  (tran <- make_translation("es", translator = "Some Person <example@examle.com>", pkg = pkg))

  # edit translations
  tran2 <- edit_translation(tran, pkg = pkg)
  # write to disk
  write_translation(tran2, pkg = pkg)

## End(Not run)
```

`get_messages` *Extract diagnostic messages*

Description

Extracts diagnostic messages from a package using `xgettext`

Usage

```
get_messages(pkg = ".")
```

Arguments

pkg	The directory of an R package to extract diagnostic messages from. Path is passed to `as.package`.

Details

Extracts diagnostic messages from a package.

Value

A "tibble" (data frame) containing messages, pluralized messages, and the file location of each message.

Author(s)

Thomas J. Leeper

See Also

`spell_check_msgs`, `get_message_distances`

Examples

```
pkg <- dummy_pkg()

get_messages(pkg = pkg)
```

`get_message_distances` *Compare message distance*

Description

Compare the string distance of messages to check for near-duplicates

Usage

```
get_message_distances(pkg = ".")
```

Arguments

pkg	The directory of an R package to extract diagnostic messages from. Path is passed to `as.package`.

Details

Compares the generalized Levenshtein (edit) distance between pairs of messages using `adist`, returning the data frame of messages from `get_messages` with additional columns corresponding to the pairwise distance

Value

A "tibble" (data frame) containing messages, pluralized messages, and the file location of each message.

Author(s)

Thomas J. Leeper

See Also

`get_messages`, `spell_check_msgs`

Examples

```
pkg <- dummy_pkg()

# get message distances
dist <- get_message_distances(pkg = pkg)
```

`msgtools` *Tools for developing diagnostic messages*

Description

This package implements a number utilities for developing and maintaining error, warning, and other diagnostic messages in R packages, including checking for consistency across messages, spell-checking messages, and managing internationalization and location of messages (i.e., translations into various languages).

Usage

```
use_localization(charset = "UTF-8", pkg = ".", domain = "R",
  verbose = getOption("verbose"))
```

Arguments

`charset`	A character string specifying the character set of the translation template file.
`pkg`	The directory of an R package to extract diagnostic messages from. Path is passed to `as.package`.
`domain`	A character string specifying the "domain" of the messages. Either "R" (the default) or "C". This is case insensitive.
`verbose`	A logical. Should the function be chatty?

Details

`use_localization` (alias: `use_l10n`) is the high-level, devtools-style function to create a '`/po`' directory and initialize a translation template (.pot) file.

`make_template` provides a lower-level interface for this internationalization step and `make_translation` handles localization. `install_translations` installs the translations for use.

`get_messages` returns a tibble (data frame) of messages and their file locations. `spell_check_msgs` returns a subset of messages with possible misspelled words and suggested replacements.

Note

Most of the functionality of this package requires '`gettext`'. Use `check_for_gettext` to see if it is available on your system; a Windows installer is provided in `install_gettext`.

Author(s)

Thomas J. Leeper

References

GNU gettext Manual

See Also

make_template, make_translation

Examples

```
# create example package
pkg <- dummy_pkg()

# get messages in memory
get_messages(pkg = pkg)
# spell check messages
spell_check_msgs(pkg = pkg)

# setup pkg for localization
use_localization(pkg = pkg)

# make a translation
tran <- make_translation("es", translator = "Some Person <example@examle.com>", pkg = pkg)
write_translation(tran, pkg = pkg)

# install translation(s)
check_translations(pkg = pkg) # check for errors before install
install_translations(pkg = pkg)
```

---

read_translation *Handle message translations (.po files)*

Description

Read, write, and generate translations of diagnostic messages

Usage

```
read_translation(language, pkg = ".", domain = "R")

write_translation(translation, pkg = ".", verbose = getOption("verbose"))

make_translation(language, translator, team = " ", pkg = ".",
  domain = "R", verbose = getOption("verbose"))

sync_translations(pkg = ".", verbose = getOption("verbose"))
```

Arguments

language	A character string specifying a language, as either: (1) "ll", an ISO 639 two-letter language code (lowercase), or (2) "ll_CC", an ISO 639 two-letter language code (lowercase) and 'CC' is an ISO 3166 two-letter country code (uppercase).

pkg	The directory of an R package to extract diagnostic messages from. Path is passed to as.package.
domain	A character string specifying the "domain" of the messages. Either "R" (the default) or "C". This is case insensitive.
translation	An object of class "po" containing a message translation.
verbose	A logical. Should the function be chatty?
translator	A character string the name and email of a translation of the form First Last <email@example.com>.
team	Optionally, a character string specifying contact information for a "translation team".

Details

read_translation and write_translation provide basic input and output functionality for translation (.po) files. If called from with an R package directory, the locations of these fies are identified automatically. read_po provides a lower-level interface for reading a specific file.

The behavior of make_translation depends on context. If the requested translation already exists, it is updated against the template (.pot) file and loaded into memory. If the translation does not already exist, the function creates a "po" translation object from the message template (.pot) file (if one does not exist, it is created). The language and translator arguments are mandatory in the latter case and only used to update values in an existing file if they differ from the existing values. language must be an allowed language code (see language_codes); the "Plural-Forms" metadata field is generated automatically from the language value (see plural_forms).

sync_translations() updates the template file (via sync_template and then updates existing translation files against it.

edit_translation is a very simple interactive interface for editing a translation object in memory.

Value

make_translation and read_translation reutrn an object of class "po". write_translation returns the path to the file, invisibly.

Note

These functions require that gettext is installed on your system.

Author(s)

Thomas J. Leeper

See Also

make_template, use_localization, edit_translation

Examples

```
## Not run:
  # create example package
  pkg <- dummy_pkg()

  # setup pkg for localization
  use_localization(pkg)

  # generate Portugal Portugese translation in memory
  make_translation("pt_PT", translator = "Some Person <example@example.com>")
  # generate Spanish translation in memory
  (tran <- make_translation("es", translator = "Some Person <example@example.com>"))
  # write to disk
  write_translation(tran)

## End(Not run)
```

`spell_check_msgs` *Spell check diagnostic messages*

Description

Extracts diagnostic messages from a package and checks them for spelling errors.

Usage

```
spell_check_msgs(dict = "en_US", pkg = ".")
```

Arguments

`dict`	A `dictionary` object.
`pkg`	The directory of an R package to extract diagnostic messages from. Path is passed to `as.package`.

Details

Extracts diagnostic messages from a package and processes them using `hunspell` to check for possible spelling mistakes.

Value

A data frame containing each message with a potentially misspelled word, the misspelled word, and the source code file in which the message is located, and (for each word) a list of suggested corrections. If no words are misspelled, a data frame with zero rows.

Author(s)

Thomas J. Leeper

See Also

get_messages, get_message_distances

Examples

```
## Not run:
  # Not run since it takes too long
  pkg <- extract_example_pkg()
  spell_check_msgs(pkg)

## End(Not run)
```

---

templates *Handle message templates (.pot files)*

Description

Read, write, and generate .pot diagnostic message templates

Usage

```
make_template(charset = "UTF-8", pkg = ".", domain = "R")

read_template(pkg = ".", domain = "R")

write_template(template, pkg = ".", verbose = getOption("verbose"))

template_exists(pkg = ".", domain = "R")

sync_template(charset = "UTF-8", pkg = ".", domain = "R",
  verbose = getOption("verbose"))

template_current(template, pkg = ".", domain = "R")
```

Arguments

charset	A character string specifying the character set of the translation template file.
pkg	The directory of an R package to extract diagnostic messages from. Path is passed to as.package.
domain	A character string specifying the "domain" of the messages. Either "R" (the default) or "C". This is case insensitive.
template	An object of class "po" containing a message translation template.
verbose	A logical. Should the function be chatty?

Details

`read_template` and `write_template` provide basic input and output functionality for translation template (.pot) files. If called from with an R package directory, the locations of these fies are identified automatically.

`make_template` generates a new template in memory, without writing it to disk. `sync_template` makes a new template and writes it to disk or, if a template file already exists, overwrites it.

`sync_template()` updates the template file. `sync_translations` further updates translation files against that template.

Value

`make_template` and `read_template` reutrn an R6 object of class "po". `write_template` returns the path to the file, invisibly.

Author(s)

Thomas J. Leeper

See Also

`get_messages` to read messages into memory without creating a template file, `use_localization` to setup a package for localization (including generation of a template file)

Examples

```
pkg <- dummy_pkg()

# check for existing template
try(template_exists(pkg = pkg))

# generate an in-memory template
pot <- make_template(pkg = pkg)
write_template(pot, pkg = pkg)
```

Index