Internationalization of R packages with *r18r*

Gergely Daroczi daroczig@rx.studio Rx Studio Inc. CTO
useR! 2022 June 22, 2022 Dissemination of Information

Rx Studio

> vignette(
+ topic = 'why',
+ package = 'r18r')

Base R has been using GNU gettext since around 2005, which is extremely powerful along with the related R utils (e.g. xgettext), but it lacks support for some of the advanced features — such as providing comments for the translator, which is useful when the translation is being done by other than the programmer:

```
#. This is a comment
msgid "message to be translated"
msgstr "translated message"
```

Rx Studio has faced this problem when outsourcing the translation of its R backend and Angular frontend to 6 languages, and created helper functions to find and extract terms to be translated from R packages (either in source code or in the PK/PD modeling and LaTeX templates) and HTML/TypeScript files, contextual metadata to be shared with the translators for higher translation quality.

> ??r18r

Why the package name? *i18n* is the standard shorthand for internationalization (starting with "i", having 18 chars, ending with "n"), but it sounds like an Apple product, and an R package really needs many more R letters — so we changed both the starting and ending letters to "r".

How to pronounce? Well, after looking up the 18th letter of the English alphabet, "rrr" sounds tempting, but since the "1" in the logo looks like the letter "l" and "8" can be read out as "ate", "r18r" should be pronounced just like the English "relator" word: "a person who relates; narrator".

> install.packages('r18r')

Since the **r18r** package has been just open-sourced and we are still finishing up the API changes due to generalization of the already existing internal features, it is currently only available on GitHub, and can be installed via the **remotes** package:

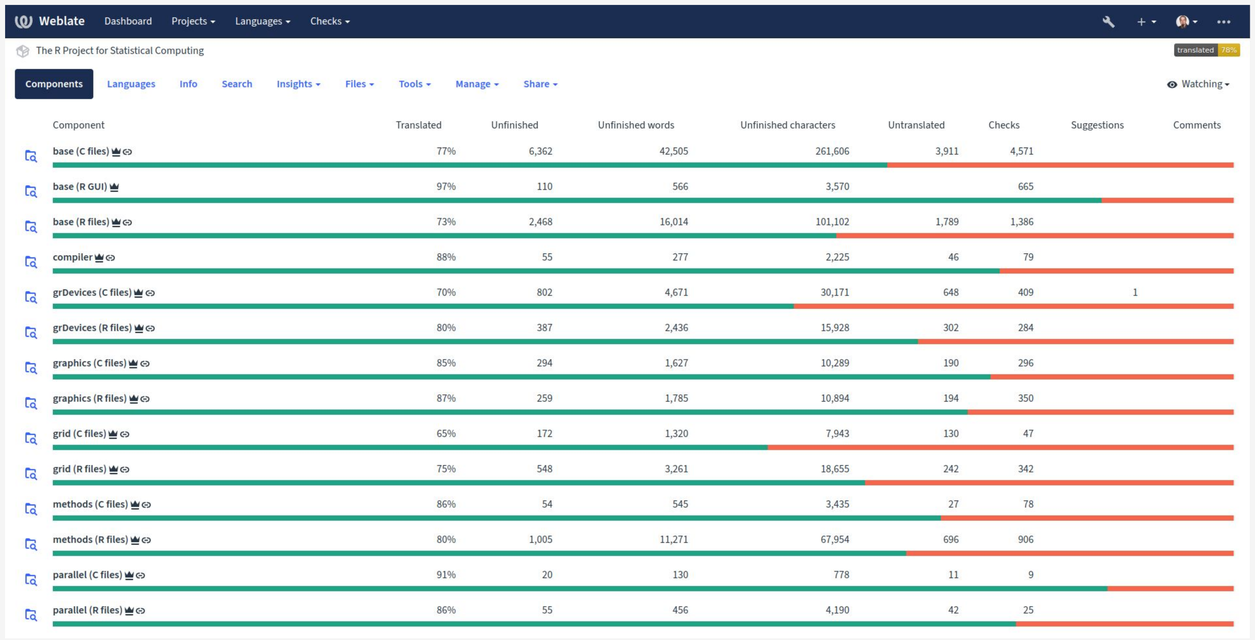
```
> library(remotes)
> install_github("rx-stud-io/r18r")
```

A CRAN release is expected later this year after initial feedback from the community.

> help(package = 'r18r')

The **r18r** package was designed to be imported by other R packages with the intention of translating their runtime messages, and helpers to generate PO or POT files, including comments and guidance for the translators, to be uploaded to translator services or hosted e.g. via Weblate (see on the right). The list of such helpers are:

- Find and parse PO files, e.g. with `po_read`
- Import a set of PO files into a **r18r** namespace with `translations_import`
- List the supported languages in a namespace, and get/set default language
- List all translations for a given language in a namespace, or look-up specific terms via `translations` and `translate`
- Mark terms to be translated and extracted into PO files via `translatable` or `T`
- Extracts terms and generate POT and PO files from R source files (and potentially other formats as well, such as R Markdown or Sweave) with `translations_generate`
- Helper functions for R package developers doing environment lookups, so that the `ns` argument is automatically guessed based on the calling package name.



Component	Translated	Unfinished	Unfinished words	Unfinished characters	Untranslated	Checks	Suggestions	Comments
base (C files)	77%	6,362	42,505	261,606	3,911	4,571		
base (R GUI)	97%	110	566	3,570		665		
base (R files)	73%	2,468	16,014	101,102	1,789	1,386		
compiler	88%	55	277	2,225	46	79		
grDevices (C files)	70%	802	4,671	30,171	648	409	1	
grDevices (R files)	80%	387	2,436	15,928	302	284		
graphics (C files)	85%	294	1,627	10,289	190	296		
graphics (R files)	87%	259	1,785	10,894	194	350		
grid (C files)	65%	172	1,320	7,943	130	47		
grid (R files)	75%	548	3,261	18,655	242	342		
methods (C files)	86%	54	545	3,435	27	78		
methods (R files)	80%	1,005	11,271	67,954	696	906		
parallel (C files)	91%	20	130	778	11	9		
parallel (R files)	86%	55	456	4,190	42	25		

> demo(package = 'r18r')

The following script loads the bundled translations files of the **r18r** package, sets the default language, and do a few lookups:

```
> library(r18n)
> translations_import(
+   po_folder('r18r'), ns = 'r18r')
> translate_set_language(
+   language = 'en', ns = 'r18r')
> translate_get_language('r18r')
[1] "en"
> translate(
+    'Text to be translated',
+    ns = 'r18r')
[1] "Text to be translated"
> translate(
+    'Text to be translated',
+    language = 'hu',
+    ns = 'r18r')
[1] "Fordítandó szöveg"
```

#' @importFrom r18r ...

When using **r18r** inside an R package:

- Use the `T` helper to mark terms
- Run `translations_generate`
- Use `translations_import` and `translate_set_language` in your `.onLoad` function (no need for `ns`)
- Use `translate` (without `ns`)

> edit(translations)

Unfortunately, the real work actually only starts after finishing all the above steps, as this package can only help with managing the translations, but not providing those.

One option is to use machine translations, or outsource to professionals, or crowdsource relying on community members. As an example, visit https://translate.rx.studio using Weblate hosting the translation files of this package, base R and the recommended packages as well (see screenshot above).