

Package ‘roam’

March 29, 2026

Title Remote Objects with Active-Binding Magic

Version 0.1.0

Description Provide helper functions for package developers to create active bindings that looks like data embedded in the package, but are downloaded from remote sources.

License MIT + file LICENSE

Encoding UTF-8

RoxygenNote 7.3.3

Imports rappdirs, utils

Suggests devtools, knitr, rmarkdown, testthat (>= 3.0.0), usethis

VignetteBuilder knitr

Config/testthat/edition 3

NeedsCompilation no

Author Mitchell O'Hara-Wild [aut] (ORCID:
<<https://orcid.org/0000-0001-6729-7695>>),
Yangzhuoran Fin Yang [aut, cre] (ORCID:
<<https://orcid.org/0000-0002-1232-8017>>)

Maintainer Yangzhuoran Fin Yang <yangyangzhuoran@gmail.com>

Repository CRAN

Date/Publication 2026-03-29 16:40:03 UTC

Contents

roam	2
Index	5

 roam

Create and manage roam object and their active bindings

Description

Helper functions for package developers to create active bindings that looks like data embedded in the package, but are downloaded from remote sources.

Usage

```
new_roam(package, name, obtainer, ...)
```

```
roam_update(x)
```

```
roam_install(x, version = NA_character_)
```

```
roam_set_version(version = NA_character_)
```

```
roam_version(package, name)
```

```
roam_delete(x)
```

```
roam_activate(x)
```

```
roam_activate_all(package)
```

Arguments

package	the name of the package as a string.
name	the name of the roam object. Should be the same as the name to which the roam object is assigned.
obtainer	a package writer/roam object creator defined function to download data/object. Should include one argument named <code>version</code> to specify the version number user wants to download. If input "latest", the obtainer function should download the latest version.
...	optional arguments to obtainer, other than <code>version</code> .
x	roam active binding
version	In <code>roam_install()</code> version of the data to install. If "latest", the latest version. In <code>roam_set_version()</code> , the version of the currently downloading data.

Details

Users of the package using roam data object can treat the roam active bindings as if they are regular data embedded in the package. The first time a user calls the roam active binding, they will be prompted to download the data using the obtainer function. The obtainer function defines how

the package developer wants to download or generate data. Once the data are downloaded, they will be cached locally using `rappdirs`. The users can then use the data object as normal.

`new_roam()` creates a roam object using the package writer/roam object creator defined obtainer function. The roam object created using `new_roam()` is not an active binding. The active bindings are not preserved during package installation, so the package developer needs to activate the roam object and turn it into an active binding in the `.onLoad` function using either `roam_activate()` or `roam_activate_all()`.

`roam_activate()` takes one roam object and activates it. `roam_activate_all()` looks through the namespace and activates all the roam objects in the package.

If there are a lot of objects in the package, calling `roam_activate()` on each roam object in `.onLoad` might save some package loading time than calling `roam_activate_all()` once.

`roam_set_version()` allows the package developer to control versioning. The obtainer function takes the version user specifies. Inside the obtainer function, the package developer can allow different download mechanism depending on the user version input, and use `roam_set_version()` to set a (transformed) developer version. For example, the user can specify `roam_install(x, version = "latest")`, and the developer can take the version "latest", find out what is the latest version, download it and set the correct version number by using (e.g.) `roam_set_version("1.2.1")`. If `roam_set_version()` is not called inside of the obtainer, the local version label will be set to NA, regardless of the user input version.

`roam_update()` is a wrapper of `roam_install()` with the default version "latest". To control versioning, the package developer should consider the behaviour of the obtainer corresponding to two special user input versions. One is "latest" from the user calling `roam_update()`, and the other is NA from the user calling `roam_install()` or calling the roam object for the first time.

Value

`new_roam` returns a function with class `roam_object`.

`roam_update` returns the updated local cache of the roam active binding

`roam_install` returns the installed local cache of the roam active binding

`roam_set_version` returns the version invisibly.

`roam_version` returns the version.

All the other functions return invisible NULL.

Functions

- `roam_update()`: Update the local cache of the roam active binding using the package writer/roam object creator defined obtainer function
- `roam_install()`: Install (Download) the local cache of the roam active binding of a specific version using the package writer/roam object creator defined obtainer function
- `roam_set_version()`: For package writers to use inside the obtainer function, save the currently downloading version number.
- `roam_version()`: Find the current version of a roam object in a package.
- `roam_delete()`: Delete the local cache of the roam active binding
- `roam_activate()`: Activate a roam object to an active binding. Used in the `.onLoad` function of a package. The roam object is activated in the environment it is defined.

- `roam_activate_all()`: Activate all the roam objects in the given package. Used in the `.onLoad` function of a package. `roam_activate_all()` looks through every object in the package to find roam objects. If the package has lots of objects, use `roam_activate()` to specify roam objects individually to improve performance.

Examples

```
# Define the roam object
bee <- new_roam(
  "roam", "bee",
  function(version)
    read.csv(
      "https://raw.githubusercontent.com/finyang/roam/master/demo/bee_colonies.csv"
    )
)
# Activation
roam_activate(bee)
if (interactive()) {
  # Download
  roam_install(bee)
  # or in an interactive session, simply
  bee
  # Access
  bee
  # Update
  roam_update(bee)
  # Deleting cache
  roam_delete(bee)
}
```

Index

`.onLoad`, [3](#), [4](#)

`new_roam (roam)`, [2](#)

`roam`, [2](#)

`roam_activate (roam)`, [2](#)

`roam_activate()`, [4](#)

`roam_activate_all (roam)`, [2](#)

`roam_delete (roam)`, [2](#)

`roam_install (roam)`, [2](#)

`roam_install()`, [2](#)

`roam_set_version (roam)`, [2](#)

`roam_set_version()`, [2](#)

`roam_update (roam)`, [2](#)

`roam_version (roam)`, [2](#)