Package 'NVIpackager'

August 29, 2023

Title Tools to facilitate Development of NVIverse Packages

Version 0.4.0.9001

Date 2023-##-##

Description Provides tools to facilitate development of NVIverse packages. You should use 'create_NVIpkg_skeleton' to make a package skeleton in accord with NVIverse standards. For further development and maintenance there are tools for updating documentation and installing development versions. The NVIpackager functions are to a large extent wrappers for devtools and usethis functions. In addition, the templates folder keeps templates for standard files like sections of README and CONTRIBUTING.

License BSD_3_clause + file LICENSE

URL https://github.com/NorwegianVeterinaryInstitute/NVIdevpkg

BugReports https://github.com/NorwegianVeterinaryInstitute/NVIdevpkg/issues

Encoding UTF-8

Language en-GB

LazyData true

Roxygen list(markdown = FALSE)

RoxygenNote 7.2.3

Depends R (>= 4.0.0)

Imports checkmate (>= 2.0.0),
desc,
devtools,
knitr,
remotes,
rmarkdown,
shiny,
stringi,
styler (>= 1.3.2),
usethis (>= 2.1.0),
withr,
NVIrpackages,
R.rsp

Suggests covr,
testthat (>= 3.0.0)

Remotes github::NorwegianVeterinaryInstitute/NVIrpackages

VignetteBuilder knitr, R.rsp

Config/testthat/edition 3

R topics documented:

`create_NVIpkg_skeleton` *Create the package skeleton for NVIverse packages*

Description

Creates the package skeleton in agreement with the conventions developed for NVIverse packages. `create_NVIpkg_skeleton` should be run once after a GitHub repository has been synchronized with the the package directory.

Usage

```
create_NVIpkg_skeleton(
  pkg = stringi::stri_extract_last_words(usethis::proj_path()),
  pkg_path = usethis::proj_path(),
  license_keyword = "BSD_3_clause"
)
```

Arguments

`pkg`	[`character(1)`] The package name. Defaults to `stringi::stri_extract_last_words(usethis::proj_path())`.
`pkg_path`	[`character(1)`] The path to the package directory. Defaults to `usethis::proj_path()`.
`license_keyword`	[`character(1)`] The keyword for the package's license in accord with list of license keywords. Defaults to "BSD_3_clause".

Details

create_NVIpkg_skeleton is a wrapper for several usethis -functions. It sets up the package directory with standard files and standard sub-directories. Modifies the DESCRIPTION, .gitignore, and .Rbuildignore in agreement with standard dependencies for NVIverse.

Standard values are input to DESCRIPTION. For modifying these values, you need to modify set_description_default.

In addition README.Rmd-template, CONTRIBUTING, CODE_OF_CONDUCT, and the vignette Contribute_to_NVIpkg are copied to the package directory and develop.R is copied to "./notes/".

The function accepts a short list of the listed license keywords accepted at Cran in "./share/license/license.db" in R home. If you need any other license than in the short list, submit an issue.

Value

None. Sets up the package directories and writes and modifies several files, see details.

Author(s)

Petter Hopp Petter.Hopp@vetinst.no

Examples

```
## Not run:
# Attach packages and set up with temporary directory
library(NVIpackager)
td <- tempdir()
if (!dir.exists(file.path(td, "NVItest"))) {
  dir.create(file.path(td, "NVItest"))
}
# Create package skeleton in temporary directory
create_NVIpkg_skeleton(pkg = "NVItest", pkg_path = file.path(td, "NVItest"))

## End(Not run)
```

---

document_NVIpkg *Update styling and documentation of a package*

Description

A wrapper around functions for styling, write the help files and update README and CONTRIBUTING.md.

Usage

```
document_NVIpkg(
  pkg = stringi::stri_extract_last_words(usethis::proj_path()),
  pkg_path = usethis::proj_path(),
  style = FALSE,
  manual = "update",
  contributing = FALSE,
  readme = FALSE,
  ...
)
```

Arguments

pkg	[character(1)] The package name. Defaults to stringi::stri_extract_last_words(usethis::proj_path()).
pkg_path	[character(1)] The path to the package directory. Defaults to usethis::proj_path().
style	[logica(1)l] Should the package be styled, defaults to FALSE.
manual	[character(1)] Should a reference manual be included, updated or removed. Defaults to manual = "update" that will update the manual if exists and do nothing if it doesn't exist.
contributing	[logical(1)] Should CONTRIBUTING.md and the vignette "Contribute to NVIpkg" be updated, defaults to FALSE.
readme	[logical(1)] Should README be updated, defaults to FALSE.
...	Other arguments to be passed to styler::style_pkg.

Details

The help files for R-functions will always be generated. Whether styling should be performed and README and CONTRIBUTING.md should be updated are controlled by input arguments.

Default value for styling is scope = "spaces". Input to scope can be any subset of c("spaces", "indention", "line_breaks", "tokens"). For indention, rather use Ctrl+I than scope = "indention". Be careful if using scope = "tokens" as code may be broken.

Value

none. Updated help files for all functions and, depending on argument input, can updated style, CONTRIBUTING.md, and README

Author(s)

Petter Hopp Petter.Hopp@vetinst.no

Examples

```
## Not run:

# Attach packages and set up with temporary directory
library(NVIpackager)
td <- tempdir()
if (!dir.exists(file.path(td, "NVItest"))) {
  dir.create(file.path(td, "NVItest"))
}
if (!dir.exists(file.path(td, "NVItest", "vignettes"))) {
  dir.create(file.path(td, "NVItest", "vignettes"))
}

document_NVIpkg(pkg = "NVItest",
                pkg_path = file.path(td, "NVItest"),
                style = FALSE,
```

```
                contributing = TRUE,
                readme = TRUE)

## End(Not run)
```

`increase_NVIpkg_version` *Increase the package version*

Description

Increase the package version in the `DESCRIPTION` and `NEWS` files. As a default, the help and the reference manual will be styled and updated.

Usage

```
increase_NVIpkg_version(
  pkg = stringi::stri_extract_last_words(usethis::proj_path()),
  pkg_path = usethis::proj_path(),
  type = "develop",
  document = FALSE,
  ...
)
```

Arguments

`pkg`	[`character(1)`] The package name. Defaults to `stringi::stri_extract_last_words(usethis::proj_path())`.
`pkg_path`	[`character(1)`] The path to the package directory. Defaults to `usethis::proj_path()`.
`type`	[`character(1)`] The type of update to perform. Must be one of c("major", "minor" , "fix" , "develop" , "first"). Defaults to "develop".
`document`	[`logical(1)`] Should styling be performed and documentation be updated. Defaults to `FALSE`.
`...`	Other arguments to be passed to `document_NVIpkg`.

Details

When publishing a new release, use `type` one of c("major", "minor", "fix"), depending of which release type it is.

When `type = "first"`, the `NEWS` will be created with the template for the first release. If `create_NVIpkg_skeleton` has been used to create the package, the `DESCRIPTION` and `NEWS` files should already have been created with correct version number and `NEWS` template.

When `type = "develop"`, the `NEWS` will be created with the template for the next release if the previous version was a release. If not, it increased the development version.

Value

None. Modifies `NEWS` and `DESCRIPTION` and eventually updates help.

Author(s)

Petter Hopp Petter.Hopp@vetinst.no

Examples

```
## Not run:
# Modifies \code{NEWS} and \code{DESCRIPTION} with new version number.
# You need to be in a package directory at you PC for the code to work.
pkg <- stringi::stri_extract_last_words(usethis::proj_path())
pkg_path = usethis::proj_path()

NVIpackager::increase_NVIpkg_version(pkg = pkg,
                                     pkg_path = pkg_path,
                                     type = "develop")

# If you publish a new release, you want to ensure that documentation is up to date.
NVIpackager::increase_NVIpkg_version(pkg = pkg,
                                     pkg_path = pkg_path,
                                     type = "minor",
                                     document = TRUE)

## End(Not run)
```

`install_NVIpkg` *Install an NVIverse Package during the Development Phase*

Description

Installation of an NVIverse package from github or from local directory. Mainly intended for installation of a package during development to test new code in the package scripts .

Usage

```
install_NVIpkg(
  pkg = stringi::stri_extract_last_words(usethis::proj_path()),
  pkg_path = usethis::proj_path(),
  lib = R.home(),
  rsource,
  username = "NorwegianVeterinaryInstitute",
  ...
)
```

Arguments

`pkg`	[character(1)] The package name. Defaults to `stringi::stri_extract_last_words(usethis::proj_path())`.
`pkg_path`	[character(1)] The path to the package directory. Defaults to `usethis::proj_path()`.
`lib`	[character(1)] The library directory where to install. Defaults to `R.home()`.

rsource	[character(1)] Source of package. Must be one of c("github", "local").
username	[character(1)] The github username where the repository is found. Defaults to "Norwegian-VeterinaryInstitute".
...	Other arguments to be passed to remotes::install_github, or devtools::install.

Details

When rsource = "github", the default is to install the latest version in the main branch at the NorwegianVeterinaryInstitute package repository. During package development one will usually change the username to your own.

The repository can be changed by giving more arguments that will be passed to remotes::install_github, see remotes::install_github, for full description of the arguments repo, username, and ref.

When rsource = "local", it installs the package from a local copy of the package repository. It defaults to install the active branch. Use rsource = "local" to test new code during development.

For installing the latest released versions of NVIverse packages, use remotes::install_github, or NVIbatch::use_NVIverse.

Value

None. Installs a package.

Author(s)

Petter Hopp Petter.Hopp@vetinst.no

Examples

```
## Not run:
# Installs a package from local PC
# You need to be in a package directory at you PC for the code to work.
pkg <- stringi::stri_extract_last_words(usethis::proj_path())
pkg_path = usethis::proj_path()

NVIpackager::install_NVIpkg(pkg = pkg, pkg_path = pkg_path, rsource = "local")

## End(Not run)
```

---

set_description_default

Set default Values for the DESCRIPTION *file*

Description

Set default Values for the DESCRIPTION file. The values are saved as the option usethis.description.

Usage

```
set_description_default(
  pkg = stringi::stri_extract_last_words(usethis::proj_path()),
  license_keyword = "BSD_3_clause"
)
```

Arguments

pkg	[character(1)] The package name. Defaults to `stringi::stri_extract_last_words(usethis::proj_path())`.
license_keyword	[character(1)] The keyword for the package's license in accord with list of license keywords. Defaults to "BSD_3_clause".

Details

These options are used when creating the package skeleton using `create_NVIpkg_skeleton`. The following standard values are given:

- "Petter Hopp" with role c("aut", "cre")
- "Norwegian Veterinary Institute" with role = "cph"
- registertype (categories for locations and addresses)
- URL: "https://github.com/NorwegianVeterinaryInstitute/NVIpkg"
- BugReports: "https://github.com/NorwegianVeterinaryInstitute/NVIpkg/issues"
- License: license_keyword "+ file LICENSE",
- Language: "en-GB",
- Depends:= "R (>= 4.0.0)"

The function accepts a short list of the listed license keywords accepted at Cran in "./share/license/license.db" in R home. If you need any other license than in the short list, submit an issue.

Value

None. Writes the options to the option `usethis.description`.

Author(s)

Petter Hopp Petter.Hopp@vetinst.no

Examples

```
## Not run:
library(NVIpackager)
set_description_default()

## End(Not run)


# set up standards for DESCRIPTION file ----
```

update_contributing *Update* CONTRIBUTING.md

Description

Update CONTRIBUTING.md and the vignette "Contribute_to_NVIpkg.Rmd" from the template in NVIpackager. If the files don't exist, they are created.

Usage

```
update_contributing(
  pkg = stringi::stri_extract_last_words(usethis::proj_path()),
  pkg_path = usethis::proj_path()
)
```

Arguments

pkg	[character(1)] The package name. Defaults to stringi::stri_extract_last_words(usethis::proj_path()).
pkg_path	[character(1)] The path to the package directory. Defaults to usethis::proj_path().

Details

The template is found in NVIpackager. Any change in the text must be done in the template.

Value

None. Writes the vignette "Contribute_to_NVIpkg.Rmd" and the file CONTRIBUTING.md.

Author(s)

Petter Hopp Petter.Hopp@vetinst.no

Examples

```
## Not run:
# Attach packages and set up with temporary directory
library(NVIpackager)
td <- tempdir()
if (!dir.exists(file.path(td, "NVItest"))) {
  dir.create(file.path(td, "NVItest"))
}
if (!dir.exists(file.path(td, "NVItest", "vignettes"))) {
  dir.create(file.path(td, "NVItest", "vignettes"))
}

use_contributing(pkg = "NVItest",
                 pkg_path = file.path(td, "NVItest"))

## End(Not run)
```

update_develop *Update* develop.R

Description

Update develop.R from the template in NVIpackager. If the file already exist, it is overwritten.

Usage

```
update_develop(
  pkg = stringi::stri_extract_last_words(usethis::proj_path()),
  pkg_path = usethis::proj_path()
)
```

Arguments

pkg	[character(1)] The package name. Defaults to stringi::stri_extract_last_words(usethis::proj_path()).
pkg_path	[character(1)] The path to the package directory. Defaults to usethis::proj_path().

Details

"develop.R" is a script comprising code for creating, documenting, testing and installing a package during development and maintenance. All code should be written without reference to a specific package, so that the script can be used without modification in development and maintenance of all NVIverse packages.

The template is found in NVIpackager. Any change in the script should be done in the template. Thereby, it is easier to keep the script updated in all packages.

Value

None. Writes the R script "develop.R" to "./notes/".

Author(s)

Petter Hopp Petter.Hopp@vetinst.no

Examples

```
## Not run:
# Updates the script develop.R
update_develop()

## End(Not run)
```

update_license *Update copyright year in the license*

Description

Update copyright year in the LICENSE file. The copyright years are given as a range from the first year to the current year.

Usage

```
update_license(
  pkg = stringi::stri_extract_last_words(usethis::proj_path()),
  pkg_path = usethis::proj_path(),
  copyright_owner = "Norwegian Veterinary Institute"
)
```

Arguments

pkg	[character(1)] The package name. Defaults to `stringi::stri_extract_last_words(usethis::proj_path())`.
pkg_path	[character(1)] The path to the package directory. Defaults to `usethis::proj_path()`.
copyright_owner	[character(1)] The copyright owner in the copyright statement. Defaults to "Norwegian Veterinary Institute".

Details

The copyright years will only be updated for the given copyright owner This to avoid that copyright for other copyright owners are updated if more than one.

Value

None. Writes the LICENSE file.

Author(s)

Petter Hopp Petter.Hopp@vetinst.no

Examples

```
## Not run:
# Used when are in the package directory
update_license()

## End(Not run)
```

update_logo *Update logo*

Description

Update the logo to be used in the README file.

Usage

```
update_logo(
  pkg = stringi::stri_extract_last_words(usethis::proj_path()),
  pkg_path = usethis::proj_path()
)
```

Arguments

pkg	[character(1)] The package name. Defaults to stringi::stri_extract_last_words(usethis::proj_path()).
pkg_path	[character(1)] The path to the package directory. Defaults to usethis::proj_path().

Details

If the package logo has been created, the logo is copied from NVIrpackages to the package directory "./man/figures/". If the logo already exists, the logo is overwritten.

Value

None. Copies the the logo file to the directory "./man/figures/".

Author(s)

Petter Hopp Petter.Hopp@vetinst.no

Examples

```
## Not run:
# Attach packages and set up with temporary directory
library(NVIpackager)
update_logo()

## End(Not run)
```

update_news *Update* NEWS

Description

Create or update the NEWS file. It will include a template for new sections if relevant, and update the version number.

Usage

```
update_news(
  pkg = stringi::stri_extract_last_words(usethis::proj_path()),
  pkg_path = usethis::proj_path(),
  template = NULL
)
```

Arguments

pkg	[character(1)] The package name. Defaults to stringi::stri_extract_last_words(usethis::proj_path()).
pkg_path	[character(1)] The path to the package directory. Defaults to usethis::proj_path().
template	[character(1)] The template to add to the NEWS file. Must be one of c("first", "develop"). Defaults to NULL.

Details

Default is template = NULL that will only update the version number. Use template = "first" to make the NEWS template for the first release of the package. Use template = "develop" to make the NEWS template for the successive releases of the package.

Value

None. Creates or updates the NEWS file in the package directory.

Author(s)

Petter Hopp Petter.Hopp@vetinst.no

Examples

```
## Not run:
# Create the first NEWS file
library(NVIpackager)
update_news(template = "first")

## End(Not run)
```

update_readme *Update* README.md

Description

Update README.md from README.Rmd and child templates in NVIpackager. If the file don't exist, it is created.

Usage

```
update_readme(
  pkg = stringi::stri_extract_last_words(usethis::proj_path()),
  pkg_path = usethis::proj_path()
)
```

Arguments

pkg	[character(1)] The package name. Defaults to stringi::stri_extract_last_words(usethis::proj_path()).
pkg_path	[character(1)] The path to the package directory. Defaults to usethis::proj_path().

Details

The child templates are found in NVIpackager. Any change in this text must be done in the templates.

Value

None. Writes the file README.md.

Author(s)

Petter Hopp Petter.Hopp@vetinst.no

Examples

```
## Not run:
# Attach packages and set up with temporary directory
library(NVIpackager)
td <- tempdir()
if (!dir.exists(file.path(td, "NVItest"))) {
  dir.create(file.path(td, "NVItest"))
}

update_readme(pkg = "NVItest",
              pkg_path = file.path(td, "NVItest"))

## End(Not run)
```

`update_reference_manual` *Update reference manual*

Description

Update the PDF reference manual and save it together with the vignettes.

Usage

```
update_reference_manual(
  pkg = stringi::stri_extract_last_words(usethis::proj_path()),
  pkg_path = usethis::proj_path(),
  manual = "update"
)
```

Arguments

`pkg`	[`character(1)`] The package name. Defaults to `stringi::stri_extract_last_words(usethis::proj_path())`.
`pkg_path`	[`character(1)`] The path to the package directory. Defaults to `usethis::proj_path()`.
`manual`	[`character(1)`] Must be one of c("include", "update", "remove"), see details. Defaults to "update".

Details

The default is `manual = "update"` that will update the reference manual if it exists, but will do nothing if it doesn't exist. Use `manual = "include"` to make the reference manual for the first time. For both `manual = c("include", "update")`, the `DESCRIPTION` and `.gitignore` will be checked and modified if necessary and the "pkgname.pdf.asis" will be created in addition to the reference manual "pkgname.pdf".

Use `manual = "remove"` to remove the reference manual. This will remove the reference manual "pkgname.pdf" and the file "pkgname.pdf.asis" and modify `.gitignore`. Any changes that have been done to the `DESCRIPTION` will not be reversed as these may also be used by other processes. If these should be reversed, it must be done manually.

Value

None. Creates or updates the PDF reference manual in directory "./vignettes/".

Author(s)

Petter Hopp Petter.Hopp@vetinst.no

Examples

```
## Not run:
# Update the reference manual if it already exists
library(NVIpackager)
update_reference_manual(manual = "update")

## End(Not run)
```

Index