

# Package: drat (via r-universe)

October 22, 2024

**Type** Package

**Title** 'Drat' R Archive Template

**Version** 0.2.5

**Date** 2024-10-21

**Depends** R (>= 3.6.0)

**Imports** utils

**Suggests** git2r, simplermardown

**VignetteBuilder** simplermardown

**Description** Creation and use of R Repositories via helper functions to insert packages into a repository, and to add repository information to the current R session. Two primary types of repositories are support: gh-pages at GitHub, as well as local repositories on either the same machine or a local network. Drat is a recursive acronym: Drat R Archive Template.

**License** GPL (>= 2)

**URL** <https://github.com/eddelbuettel/drat>,  
<https://dirk.eddelbuettel.com/code/drat.html>

**BugReports** <https://github.com/eddelbuettel/drat/issues>

**Encoding** UTF-8

**RoxygenNote** 6.0.1

**NeedsCompilation** no

**Author** Dirk Eddelbuettel [aut, cre]

(<<https://orcid.org/0000-0001-6419-907X>>), Felix Ernst [ctb]

(<<https://orcid.org/0000-0001-5064-0928>>), Neal Fultz [ctb],

Colin Gillespie [ctb]

(<<https://orcid.org/0000-0003-1787-0275>>), Christoph Stepper

[ctb], Thomas Leeper [ctb]

(<<https://orcid.org/0000-0003-4097-6326>>), Mark van der Loo

[ctb] (<<https://orcid.org/0000-0002-9807-4686>>), Jan Gorecki

[ctb], Joey Reid [ctb], Arne Johannes Holmin [ctb]

**Maintainer** Dirk Eddelbuettel <edd@debian.org>

**Repository** CRAN

**Date/Publication** 2024-10-21 20:10:02 UTC

## Contents

|                               |   |
|-------------------------------|---|
| drat-package . . . . .        | 2 |
| addRepo . . . . .             | 3 |
| archivePackages . . . . .     | 4 |
| getPackageInfo . . . . .      | 5 |
| identifyPackageType . . . . . | 6 |
| initRepo . . . . .            | 6 |
| insertPackage . . . . .       | 7 |
| pruneRepo . . . . .           | 9 |

**Index** **11**

---

|              |   |
|--------------|---|
| drat-package | <i>Easy-to-use package repository creation and access</i> |
|--------------|---|

---

## Description

The drat package permits user to create and use ad-hoc package repositories. It takes advantage of GitHub accounts and ‘gh-pages’ branches which automatically become web-accessible and can be used to provide a repository. Alternatively, custom repository paths and addresses can be used.

## Details

Given a user account on GitHub, say, ‘eddelbuettel’, and a repository ‘drat’, we can infer an top-level repository URL as such as <https://eddelbuettel.github.io/drat/> by supplying only the username (as the rest is inferred by defaults). This allows us to create easily useable, identifiable and shareable per-user repositories—without the user having to create and administer a webserver anywhere.

Two higher level functions then allow both insertion of (source or binary) packages, as well as addition of a given drat repository to an R session so that package in the repository can be accesses.

## Author(s)

Dirk Eddelbuettel

Maintainer: Dirk Eddelbuettel <edd@debian.org>

## References

The R Installation and Administration manual has more and details about repository creation

**See Also**

[update.packages](#), [available.packages](#), [install.packages](#)

**Examples**

```
## Not run:
drat::addRepo("eddelbuettel") # adds the repo of GitHub user 'eddelbuettel'

## End(Not run)
```

---

|         |   |
|---------|---|
| addRepo | <i>Add a (drat) repository to the current session</i> |
|---------|---|

---

**Description**

R can use multiple archives: CRAN, BioConductor and Omegahat have been supported for years. It is equally easy to add local archives from the same machine, or local network, or university / company network as well as other publically available repositories. This function aids in the process, and defaults to adding a ‘drat’ archive at GitHub under the given account.

**Usage**

```
addRepo(account, alturl)

add(...)
```

**Arguments**

|         |  |
|---------|--|
| account | Character vector with one or more GitHub account for which a ‘drat’ archive is to be added.  |
| alturl  | Alternative repo specification with a complete url string. If ‘alturl’ is provided, a single ‘account’ must be provided as well. For file-based access, the URL format has to follow the <code>file:/some/path/</code> format starting with ‘file’ followed by a single colon. |
| ...     | For the aliases variant, a catch-all collection of parameters.   |

**Details**

This function retrieves the current set of repositories (see `getOption("repos")` for the current values) and adds (or overwrites) the entry for the given ‘account’. For non-GitHub repositories an alternative URL can be specified as ‘alturl’ (and assigned to ‘account’ as well).

An aliased function `add` is also available, but not exported via `NAMESPACE` to not clobber a possibly unrelated function; use it via `drat:::add()`.

**Value**

The altered set of repositories

**Author(s)**

Dirk Eddebuettel

**Examples**

```
## Not run:
addRepo("drat") # adds GitHub repo via default URL
addRepo(c("eddebuettel", "ghrr")) # ditto but adds two repos at once

addRepo("LocalRepo", "file:/nas/R/repo") # adds local file-based repo,
# assumes you can read /nas/R/repo

## End(Not run)
```

archivePackages

*Move older copies of packages to an archive***Description**

The function moves older versions of packages into a CRAN-style archive folder.

**Usage**

```
archivePackages(repopath = getOption("dratRepo", "~/git/drat"),
  type = c("source", "binary", "mac.binary", "mac.binary.big-sur-x86_64",
    "mac.binary.big-sur-arm64", "mac.binary.el-capitan", "mac.binary.mavericks",
    "win.binary", "both"), pkg, version = getRversion())

archivePackagesForAllRversions(repopath = getOption("dratRepo", "~/git/drat"),
  type = c("source", "binary", "mac.binary", "mac.binary.big-sur-x86_64",
    "mac.binary.big-sur-arm64", "mac.binary.el-capitan", "mac.binary.mavericks",
    "win.binary", "both"), pkg)
```

**Arguments**

|          |   |
|----------|---|
| repopath | Character variable with the path to the repo; defaults to the value of the “dratRepo” option with “~/git/drat” as fallback  |
| type     | Character variable for the type of repository, so far “source”, “binary”, “win.binary”, “mac.binary”, “mac.binary.big-sur-x86_64”, “mac.binary.big-sur-arm64”, “mac.binary.mavericks”, “mac.binary.el-capitan” or “both”                  |
| pkg      | Optional character variable specifying a package name(s), whose older versions should be archived. If missing (the default), archiving is performed on all packages.  |
| version  | R version information in the format X.Y or X.Y.Z. Only used, if archiving binary packages. (default: version = getRversion()). If version = NA, all available R versions will be used. If version = NULL, this defaults to getRversion(). |

**Details**

This function is still undergoing development and polish and may change in subsequent versions.

**Author(s)**

Thomas J. Leeper

**Examples**

```
## Not run:  
archivePackages() # archive all older package versions  
archivePackages(pkg = "drat") # archive older copies of just one package  
  
## End(Not run)
```

---

|                |  |
|----------------|--|
| getPackageInfo | <i>Get information from a binary package</i> |
|----------------|--|

---

**Description**

This function returns the compile-time information added to the DESCRIPTION file in the package.

**Usage**

```
getPackageInfo(file, OSflavour = character())
```

**Arguments**

|           |  |
|-----------|--|
| file      | the fully qualified path of the package  |
| OSflavour | an optional string naming the OSflavour, which is otherwise read as the second element of the 'Built' field of the file. For packages that do not need compilation on macOS for R >= 4.3 the 'Built' field is empty in the DESCRIPTION in a binary file (tgz), in which case it can be useful to set the OSflavour e.g. by the value of R.Version()\$platform, so that <a href="#">insertPackages</a> inserts the binary into the appropriate sub folder (under bin/maxosx). |

**Value**

A named vector with several components

**Note**

This is an internal function, use ::: to access it from outside the internal package code.

**Author(s)**

Dirk Eddelbuettel

---

identifyPackageType     *Identifies the package type from a filename*

---

**Description**

This function identifies the package type from a filename.

**Usage**

```
identifyPackageType(file, pkginfo = getPackageInfo(file))
```

**Arguments**

file                    An R package in source or binary format,  
pkginfo                information on the R package referenced by file

**Details**

The returned string is suitable for write\_PACKAGES().

**Value**

string Type of the supplied package.

**Note**

This is an internal function, use ::: to access it from outside the internal package code.

**Author(s)**

Jan Schulz and Dirk Eddelbuettel

---

initRepo                *Intialize a git repo for drat*

---

**Description**

This helper function creates a new repository, creates and checks out the default GitHub Pages location (either the 'gh-pages' branch or directory 'docs') and fills it with the required new paths.

**Usage**

```
initRepo(name = "drat", basepath = getOption("dratDirectory", "~/git"),  
          location = getOption("dratBranch", "gh-pages"))
```

**Arguments**

|          |  |
|----------|--|
| name     | A character variable with the name the new repository, the default is “drat”.  |
| basepath | A character variable with path to the directory in which the new repository is to be created. The default value is “~/git” and can be overridden via option ‘dratDirectory’.                                       |
| location | A character variable with the GitHub Pages location: either “gh-pages” indicating a branch of that name, or “docs/” directory in the main branch. The default value can be overridden via the “dratBranch” option. |

**Details**

Currently only ‘src/contrib’ for source repositories is supported by this function. The `insertPackage()` function knows to deal with binaries for different architectures.

The function also installs a top-level `index.html` file to ensure external tests against the repository (as for example done by CRAN if you list the repository as an ‘Additional\_repositories’ in a package) do not return a ‘404’ error.

**Value**

The function is invoked for its side-effects and only returns NULL invisibly.

**Author(s)**

Dirk Eddelbuettel

---

|               |  |
|---------------|--|
| insertPackage | <i>Insert a package source or binary file into a drat repository</i> |
|---------------|--|

---

**Description**

R can use multiple archives: CRAN, BioConductor and Omegahat have been supported for years. It is equally easy to add local archives from the same machine, or local network, or university / company network as well as other publically available repositories. This function aids in the process, and defaults to inserting a given source archive into a given repository.

**Usage**

```
insertPackage(file, repodir = getOption("dratRepo", "~/git/drat"),
  commit = FALSE, pullfirst = FALSE, action = c("none", "archive",
  "prune"), location = getOption("dratBranch", "gh-pages"),
  OSflavour = character(), ...)
```

```
insertPackages(file, ...)
```

```
insert(...)
```

**Arguments**

|           |   |
|-----------|---|
| file      | One or more R package(s) in source or binary format   |
| reporDir  | A local directory corresponding to the repository top-level directory.  |
| commit    | Either boolean toggle to select automatic git operations ‘add’, ‘commit’, and ‘push’ or, alternatively, a character variable can be used to specify a commit message; this also implies the ‘TRUE’ values in other contexts.  |
| pullfirst | Boolean toggle to call <code>git pull</code> before inserting the package.  |
| action    | A character string containing one of: “none” (the default; add the new package into the repo, effectively masking previous versions), “archive” (place any previous versions into a package-specific archive folder, creating such an archive if it does not already exist), or “prune” (calling <code>pruneRepo</code> ).  |
| location  | A character variable with the GitHub Pages location: either “gh-pages” indicating a branch of that name, or “docs/” directory in the main branch. The default value can be overridden via the “dratBranch” option.  |
| OSflavour | an optional string naming the OSflavour, which is otherwise read as the second element of the ‘Built’ field of the file. For packages that do not need compilation on macOS for R >= 4.3 the ‘Built’ field is empty in the DESCRIPTION in a binary file (tgz), in which case it can be useful to set the OSflavour e.g. by the value of <code>R.Version()\$platform</code> , so that <code>insertPackages</code> inserts the binary into the appropriate sub folder (under bin/maxosx). |
| ...       | For <code>insert</code> the aliases variant, a catch-all collection of parameters. For <code>insertPackage</code> arguments passed to <code>write_PACKAGES</code> currently include <code>latestOnly</code> , for which the default value is set here to <code>FALSE</code> . See <code>write_PACKAGES</code> .   |

**Details**

This function inserts the given (source or binary) package file into the given (local) package repository and updates the index. By setting the `commit` option to `TRUE`, one can then push to a remote git code repository. If the `git2r` package is installed, it is used for the interaction with the git repository; otherwise the `git` shell command is used.

An aliased function `insert` is also available, but not exported via `NAMESPACE` to not clobber a possibly unrelated function; use it via `drat:::insert()`.

The function also checks for a top-level `index.html` file to ensure external tests against the repository (as for example done by CRAN if you list the repository as an ‘Additional\_repositories’ in a package) do not return a ‘404’ error. If missing, a simple one-line example is shown.

**Value**

NULL is returned.

**Options**

Set using [options](#)

`dratRepo` Path to git repo. Defaults to `~/git/drat`

`dratBranch` The git branch to store packages on. Defaults to `gh-pages`

**Author(s)**

Dirk Eddelbuettel

**Examples**

```
## Not run:
insertPackage("foo_0.2.3.tar.gz") # inserts into (default) repo
insertPackage("foo_0.2.3.tar.gz", "/nas/R/") # ... into local dir

## End(Not run)
## Not run:
insertPackage("foo_0.2.3.tar.gz", action = "prune") # prunes any older copies
insertPackage("foo_0.2.3.tar.gz", action = "archive") # archives any older copies

## End(Not run)
```

pruneRepo

*Prune repository from older copies of packages***Description**

The function determines which packages in a repositories can be removed as they are being ‘shadowed’ by a newer version of the same packages.

**Usage**

```
getRepoInfo(repopath = getOption("dratRepo", "~/git/drat"),
  type = c("source", "binary", "mac.binary", "mac.binary.big-sur-x86_64",
    "mac.binary.big-sur-arm64", "mac.binary.el-capitan", "mac.binary.mavericks",
    "win.binary", "both"), pkg, version = getRversion(),
  location = getOption("dratBranch", "gh-pages"))

pruneRepo(repopath = getOption("dratRepo", "~/git/drat"), type = c("source",
  "binary", "mac.binary", "mac.binary.big-sur-x86_64",
  "mac.binary.big-sur-arm64", "mac.binary.el-capitan", "mac.binary.mavericks",
  "win.binary", "both"), pkg, version = getRversion(), remove = FALSE,
  location = getOption("dratBranch", "gh-pages"))

pruneRepoForAllRversions(repopath = getOption("dratRepo", "~/git/drat"),
  type = c("source", "mac.binary", "mac.binary.big-sur-x86_64",
    "mac.binary.big-sur-arm64", "mac.binary.el-capitan", "mac.binary.mavericks",
    "win.binary", "both"), pkg, remove = FALSE)

updateRepo(repopath = getOption("dratRepo", "~/git/drat"),
  type = c("source", "mac.binary", "mac.binary.big-sur-x86_64",
    "mac.binary.big-sur-arm64", "mac.binary.el-capitan", "mac.binary.mavericks",
    "win.binary", "both"), version = NA, ...)
```

**Arguments**

|          |   |
|----------|---|
| repopath | Character variable with the path to the repo; defaults to the value of the “dratRepo” option with “~/git/drat” as fallback  |
| type     | Character variable for the type of repository, so far “source”, “binary”, “win.binary”, “mac.binary”, “mac.binary.mavericks”, “mac.binary.el-capitan”, “mac.binary.big-sur-x86_64”, “mac.binary.big-sur-arm64”, or “both”   |
| pkg      | Optional character variable specifying a package name, whose older versions should be pruned. If missing (the default), pruning is performed on all packages.   |
| version  | R version information in the format X.Y or X.Y.Z. Only used, if pruning binary packages. (default: <code>version = getRversion()</code> ). If <code>version = NA</code> , all available R versions will be used. If <code>version = NULL</code> , this defaults to <code>getRversion()</code> .   |
| location | An optional character variable with the GitHub Pages location: either “gh-pages” indicating a branch of that name, or “docs/” directory in the main branch. The default value can be overridden via the “dratBranch” option.  |
| remove   | Character or logical variable indicating whether files should be removed. Nothing happens if ‘FALSE’. If different from (logical) ‘FALSE’ and equal to character “git” files are removed via <code>git rm</code> else via a straight file deletion.   |
| ...      | For <code>updateRepo</code> a catch-all collection of parameters. Arguments passed to <code>update_PACKAGES</code> currently include <code>latestOnly</code> , for which the default value is set here to FALSE. See <a href="#">update_PACKAGES</a> . Please note that this has an effect for <code>update_PACKAGES</code> only, if new packages are found, e.g. manually added. |

**Details**

Given a package name, R will always find the newest version of that package. Older versions are therefore effectively shadowed and can be removed without functionally changing a repository.

However, if a current package file is removed without `pruneRepo`, the `PACKAGES`, `PACKAGES.gz` and `PACKAGES.rds` file might be not up to date. To ensure the correct information is available in these indices, run `updateRepo`.

These functions are still undergoing development and polish and may change in subsequent versions.

**Value**

A data frame describing the repository is returned containing columns with columns “file”, “package” (just the name), “version” and a logical variable “newest” indicating if the package can be removed.

**Author(s)**

Dirk Eddelbuettel

# Index

- \* **package**
  - drat-package, 2
  
- add (addRepo), 3
- addRepo, 3
- archivePackages, 4
- archivePackagesForAllRversions (archivePackages), 4
- available.packages, 3
  
- drat (drat-package), 2
- drat-package, 2
- drat:::add (addRepo), 3
- drat:::insert (insertPackage), 7
  
- getPackageInfo, 5
- getRepoInfo (pruneRepo), 9
- git2r, 8
  
- identifyPackageType, 6
- initRepo, 6
- insert (insertPackage), 7
- insertPackage, 7
- insertPackages, 5, 8
- insertPackages (insertPackage), 7
- install.packages, 3
  
- options, 8
  
- pruneRepo, 8, 9
- pruneRepoForAllRversions (pruneRepo), 9
  
- update.packages, 3
- update\_PACKAGES, 10
- updateRepo (pruneRepo), 9
  
- write\_PACKAGES, 8