

# Package ‘GRANCore’

October 12, 2022

**Type** Package

**Title** Classes and Methods for 'GRANBase'

**Version** 0.2.7

**Author** Gabriel Becker[aut,cre], Dinakar Kulkarni [aut,ctb]

**Maintainer** Gabriel Becker <gabembecker@gmail.com>

**Copyright** Genentech Inc

**Description** Provides the classes and methods for GRANRepository objects that are used within the 'GRAN' build framework for R packages. This is primarily used by the 'GRANBase' package and repositories that are created by it.

**License** Artistic-2.0

**Depends** R (>= 3.1.0), switchr (>= 0.9.28), methods

**URL** <https://github.com/gmbecker/GRANCore>

**BugReports** <https://github.com/gmbecker/GRANCore/issues>

**RoxygenNote** 6.1.1

**NeedsCompilation** no

**Repository** CRAN

**Date/Publication** 2020-02-04 15:00:13 UTC

## R topics documented:

|                               |   |
|-------------------------------|---|
| archivedir . . . . .          | 2 |
| available.packages . . . . .  | 3 |
| backup_archive . . . . .      | 4 |
| checkout_dir . . . . .        | 4 |
| check_result_dir . . . . .    | 5 |
| contrib.url . . . . .         | 5 |
| coverage_report_dir . . . . . | 6 |
| destination . . . . .         | 6 |
| dest_base . . . . .           | 7 |

|  |    |
|--|----|
| email_notify . . . . .                                     | 7  |
| email_options . . . . .                                    | 8  |
| errlogfile . . . . .                                       | 8  |
| GRANRepository-class . . . . .                             | 9  |
| install_packages,character,GRANRepository-method . . . . . | 9  |
| install_result_dir . . . . .                               | 10 |
| loadRepo . . . . .   | 11 |
| logfile . . . . .  | 11 |
| make_windows_bins . . . . .                                | 16 |
| metadatadir . . . . .                                      | 17 |
| notrack,GRANRepository-method . . . . .                    | 17 |
| param . . . . .  | 18 |
| pkg_doc_dir . . . . .                                      | 18 |
| prepDirStructure . . . . .                                 | 19 |
| rebase . . . . .   | 19 |
| RepoBuildParam-class . . . . .                             | 20 |
| RepoToList . . . . .                                       | 23 |
| repo_results . . . . .                                     | 23 |
| repo_url . . . . .   | 24 |
| staging . . . . .  | 25 |
| temp_lib . . . . .   | 25 |
| updateGRANRepoObject . . . . .                             | 26 |
| windowsbindir . . . . .                                    | 26 |
| writeGRANLog . . . . .                                     | 27 |

## Index 28

---

|            |  |
|------------|--|
| archivedir | <i>archivedir</i> Return the full path to the archive directory for the final repository deployment. |
|------------|--|

---

### Description

archivedir Return the full path to the archive directory for the final repository deployment.

### Usage

```
archivedir(repo)

## S4 method for signature 'GRANRepository'
archivedir(repo)
```

### Arguments

repo            a GRANRepository object

### Value

The full path to the archive directory where the archived packages will be deployed to

---

|                    |  |
|--------------------|--|
| available.packages | <i>available.packages</i> A generic for available.packages and a method for GRANRepository objects |
|--------------------|--|

---

### Description

available.packages A generic for available.packages and a method for GRANRepository objects

### Usage

```
available.packages(contriburl, method, fields = NULL,
  type = getOption("pkgType"), filters = NULL, repos = NULL)

## S4 method for signature 'ANY'
available.packages(contriburl, method, fields = NULL,
  type = getOption("pkgType"), filters = NULL, repos = NULL)

## S4 method for signature 'GRANRepository'
available.packages(contriburl, method,
  fields = NULL, type = getOption("pkgType"), filters = NULL,
  repos = NULL)
```

### Arguments

|            |   |
|------------|---|
| contriburl | The repository or contrib url   |
| method     | See base documentation  |
| fields     | See base documentation  |
| type       | The type of packages to query   |
| filters    | See base documetnation  |
| repos      | Character string for the repository to query. GRANRepository objects should be passed to the contriburl argument. |

### Examples

```
repo = GRANRepository(GithubManifest("gmbecker/fastdigest"), basedir = tempdir())
## none because the repository hasn't been built...
available.packages(repo)
```

---

|                |  |
|----------------|--|
| backup_archive | <i>backup_archive</i> Return path where packages are backed up by default when clearing the repo |
|----------------|--|

---

**Description**

backup\_archive Return path where packages are backed up by default when clearing the repo

**Usage**

```
backup_archive(repo)
```

```
## S4 method for signature 'GRANRepository'
backup_archive(repo)
```

**Arguments**

repo                    a GRANRepository object

**Value**

Directory where packages are backed up by default when clearing repo

---

|              |   |
|--------------|---|
| checkout_dir | <i>checkout_dir</i> Return the directory that package sources will be checked-out into for use in the build process |
|--------------|---|

---

**Description**

checkout\_dir Return the directory that package sources will be checked-out into for use in the build process

**Usage**

```
checkout_dir(repo)
```

```
## S4 method for signature 'GRANRepository'
checkout_dir(repo)
```

```
## S4 method for signature '`NULL`'
checkout_dir(repo)
```

**Arguments**

repo                    a GRANRepository object

**Value**

For destination, the full path to the contrib directory the packages will be deployed to

---

|                  |  |
|------------------|--|
| check_result_dir | <i>check_result_dir Return the path where check results for packages will be deployed for use in the build report.</i> |
|------------------|--|

---

**Description**

check\_result\_dir Return the path where check results for packages will be deployed for use in the build report.

**Usage**

```
check_result_dir(repo)

## S4 method for signature 'GRANRepository'
check_result_dir(repo)
```

**Arguments**

repo                    a GRANRepository object

**Value**

The directory where check results should be deployed for use in the build report

---

|             |   |
|-------------|---|
| contrib.url | <i>contrib.url A generic for contrib.url so that available.packages et al can interact with GRANRepository objects.</i> |
|-------------|---|

---

**Description**

contrib.url A generic for contrib.url so that available.packages et al can interact with GRANRepository objects.

**Usage**

```
contrib.url(repos, type = getOption("pkgType"))

## S4 method for signature 'GRANRepository'
contrib.url(repos,
  type = getOption("pkgType"))
```

**Arguments**

repo            A repository to extract the contrib url from  
 type            The type of package repository it is

**Examples**

```
repo = GRANRepository(GithubManifest("gmbecker/fastdigest"), basedir = tempdir())
contrib.url(repo)
```

---

coverage\_report\_dir    *coverage\_report\_dir* Return the path where test coverage reports for packages will be deployed for use in the build report.

---

**Description**

coverage\_report\_dir Return the path where test coverage reports for packages will be deployed for use in the build report.

**Usage**

```
coverage_report_dir(repo)

## S4 method for signature 'GRANRepository'
coverage_report_dir(repo)
```

**Arguments**

repo            a GRANRepository object

**Value**

The directory where test coverage results should be deployed for use in the build report

---

destination            *destination* Return the full path to the contrib directory for the final repository deployment.

---

**Description**

destination Return the full path to the contrib directory for the final repository deployment.

**Usage**

```
destination(repo)

## S4 method for signature 'GRANRepository'
destination(repo)
```

**Arguments**

repo                    a GRANRepository object

**Value**

For destination, the full path to the contrib directory the packages will be deployed to

---

|           |   |
|-----------|---|
| dest_base | <i>dest_base Return the full path to the contrib directory for the final repository deployment.</i> |
|-----------|---|

---

**Description**

dest\_base Return the full path to the contrib directory for the final repository deployment.

**Usage**

```
dest_base(repo)

## S4 method for signature 'GRANRepository'
dest_base(repo)
```

**Arguments**

repo                    a GRANRepository object

**Value**

For dest\_base, the base directory the repository will reside in

---

|              |  |
|--------------|--|
| email_notify | <i>email_notify Should emails be sent for build failure notifications?</i> |
|--------------|--|

---

**Description**

email\_notify Should emails be sent for build failure notifications?

**Usage**

```
email_notify(repo)

## S4 method for signature 'GRANRepository'
email_notify(repo)
```

**Arguments**

repo                    a GRANRepository object

**Value**

logical

---

|               |  |
|---------------|--|
| email_options | <i>email_options</i> Email options for sending build failure notifications |
|---------------|--|

---

**Description**

email\_options Email options for sending build failure notifications

**Usage**

```
email_options(repo)
```

```
## S4 method for signature 'GRANRepository'
email_options(repo)
```

**Arguments**

repo                    a GRANRepository object

**Value**

A list containing the email options

---

|            |   |
|------------|---|
| errlogfile | <i>Log file location of a GRAN (sub) repository</i> |
|------------|---|

---

**Description**

errlogfile Retrieve the path to the errors-only logfile for a GRAN repository

**Usage**

```
errlogfile(repo)
```

```
## S4 method for signature 'GRANRepository'
errlogfile(repo)
```

```
## S4 method for signature 'RepoBuildParam'
errlogfile(repo)
```



**Arguments**

repo                    a GRANRepository object

**Value**

file location of the errors-only logfile

---

GRANRepository-class    *GRANRepository*

---

**Description**

A constructor for the GRANRepository class of S4 objects representing individual repositories

**Usage**

```
GRANRepository(manifest, results, param = RepoBuildParam(...), ...)
```

**Arguments**

manifest                A PkgManifest object

results                 A data.frame containing previous build results

param                   A RepoBuildParam object controlling the location and behavior of the repository being built

...                      Passed through to the default value of param

**Examples**

```
man = GithubManifest("gmbecker/fastdigest")
repo = GRANRepository(man, basedir = tempdir())
```

---

install\_packages, character, GRANRepository-method  
*install\_packages method for GRANRepository objects*

---

**Description**

install\_packages method for GRANRepository objects

**Usage**

```
## S4 method for signature 'character,GRANRepository'
install_packages(pkgs, repos,
  versions = NULL, verbose = FALSE, ...)
```

**Arguments**

|          |   |
|----------|---|
| pkgs     | Character vector of package names to install            |
| repos    | A GRANRepository  |
| versions | Ignored   |
| verbose  | Whether verbose/debugging information should be printed |
| ...      | Passed (eventually) to install.packages                 |

**Details**

This method calls directly down to the character,character method of install\_packages in switchr

---

|                    |   |
|--------------------|---|
| install_result_dir | <i>install_result_dir</i> Return the path where instal results for packages will be deployed for use in the build report. |
|--------------------|---|

---

**Description**

install\_result\_dir Return the path where instal results for packages will be deployed for use in the build report.

**Usage**

```
install_result_dir(repo)

## S4 method for signature 'GRANRepository'
install_result_dir(repo)
```

**Arguments**

|      |                         |
|------|-------------------------|
| repo | a GRANRepository object |
|------|-------------------------|

**Value**

The directory where install results should be deployed for use in the build report

---

|          |  |
|----------|--|
| loadRepo | <i>Backwards compatible load utility</i> |
|----------|--|

---

**Description**

Load a repository serialized to an R code file  
 serialize a repository to a file so that it does not require GRANBase to load

**Usage**

```
loadRepo(filename)

saveRepo(repo, filename)
```

**Arguments**

|          |                                   |
|----------|-----------------------------------|
| filename | The file to load                  |
| repo     | The GRANRepository object to save |

**Examples**

```
repo = GRANRepository(GithubManifest("gmbecker/rpath"), basedir = tempdir())
fil = file.path(tempdir(), "repo.R")
saveRepo(repo, fil)
repo2 = loadRepo(fil)
```

---

|         |   |
|---------|---|
| logfile | <i>Log file location of a GRAN (sub) repository</i> |
|---------|---|

---

**Description**

Accessors

**Usage**

```
logfile(repo)

## S4 method for signature 'GRANRepository'
logfile(repo)

## S4 method for signature 'RepoBuildParam'
logfile(repo)

## S4 method for signature 'GRANRepository'
dep_repos(x)
```

```
repo_name(x)

## S4 method for signature 'GRANRepository'
repo_name(x)

temp_repo(x)

## S4 method for signature 'GRANRepository'
temp_repo(x)

check_warn_ok(x)

## S4 method for signature 'GRANRepository'
check_warn_ok(x)

check_note_ok(x)

## S4 method for signature 'GRANRepository'
check_note_ok(x)

suspended_pkgs(x)

## S4 method for signature 'GRANRepository'
suspended_pkgs(x)

suspended_pkgs(x) <- value

## S4 replacement method for signature 'GRANRepository'
suspended_pkgs(x) <- value

## S4 method for signature 'GRANRepository'
sh_init_script(x)

## S4 replacement method for signature 'GRANRepository'
sh_init_script(x) <- value

extra_fun(x)

## S4 method for signature 'GRANRepository'
extra_fun(x)

check_test_on(x)

## S4 method for signature 'RepoBuildParam'
check_test_on(x)

## S4 method for signature 'GRANRepository'
```

```
check_test_on(x)

install_test_on(x)

## S4 method for signature 'RepoBuildParam'
install_test_on(x)

## S4 method for signature 'GRANRepository'
install_test_on(x)

## S4 method for signature 'GRANRepository'
logfun(x)

## S4 replacement method for signature 'GRANRepository'
logfun(x) <- value

use_cran_granbase(x)

use_cran_granbase(x) <- value

## S4 method for signature 'GRANRepository'
use_cran_granbase(x)

## S4 replacement method for signature 'GRANRepository'
use_cran_granbase(x) <- value

## S4 method for signature 'RepoBuildParam'
use_cran_granbase(x)

## S4 replacement method for signature 'RepoBuildParam'
use_cran_granbase(x) <- value

check_timeout(x)

check_timeout(x) <- value

## S4 method for signature 'GRANRepository'
check_timeout(x)

## S4 replacement method for signature 'GRANRepository'
check_timeout(x) <- value

## S4 method for signature 'RepoBuildParam'
check_timeout(x)

## S4 replacement method for signature 'RepoBuildParam'
check_timeout(x) <- value
```

```
build_timeout(x)

build_timeout(x) <- value

## S4 method for signature 'GRANRepository'
build_timeout(x)

## S4 replacement method for signature 'GRANRepository'
build_timeout(x) <- value

## S4 method for signature 'RepoBuildParam'
build_timeout(x)

## S4 replacement method for signature 'RepoBuildParam'
build_timeout(x) <- value

pkg_log_dir(x)

## S4 method for signature 'RepoBuildParam'
pkg_log_dir(x)

## S4 method for signature 'GRANRepository'
pkg_log_dir(x)

pkg_log_file(pkg, x)

## S4 method for signature 'ANY,RepoBuildParam'
pkg_log_file(pkg, x)

## S4 method for signature 'ANY,GRANRepository'
pkg_log_file(pkg, x)

platform(x)

platform(x) <- value

## S4 method for signature 'GRANRepository'
platform(x)

## S4 replacement method for signature 'GRANRepository'
platform(x) <- value

r_version(x)

r_version(x) <- value

## S4 method for signature 'GRANRepository'
r_version(x)
```

```
## S4 replacement method for signature 'GRANRepository'  
r_version(x) <- value  
  
bioc_version(x)  
  
bioc_version(x) <- value  
  
## S4 method for signature 'GRANRepository'  
bioc_version(x)  
  
## S4 replacement method for signature 'GRANRepository'  
bioc_version(x) <- value
```

### Arguments

|       |   |
|-------|---|
| repo  | a GRANRepository object                     |
| x     | A GRANRepository object                     |
| value | The new parameter value                     |
| pkg   | The package name, accepted by pkg_log_file. |

### Details

Set or retrieve the relevant values from a GRANRepository object

These functions get or set individual repository build parameters on a GRANRepository object.

### Value

file location of the full logfile

### See Also

[RepoBuildParam](#)

### Examples

```
repo = GRANRepository(GithubManifest("gmbecker/fastdigest"), basedir = tempdir())  
# parameter object  
param(repo)  
##fundamental sub-objects  
manifest(repo)  
repo_results(repo)  
##important directories  
rebase(repo)  
staging(repo)  
temp_lib(repo)  
notrack(repo)  
destination(repo)  
dest_base(repo)  
windowsbindir(repo)
```

```

archivedir(repo)
metadatadir(repo)
check_result_dir(repo)
backup_archive(repo)
coverage_report_dir(repo)
pkg_doc_dir(repo)
install_result_dir(repo)
repo_url(repo)
checkout_dir(repo)
## logging
logfile(repo)
errlogfile(repo)
staging_logs(repo)
pkg_log_dir(repo)
pkg_log_file("switchr", repo)
## email and other behavior
email_options(repo)
email_notify(repo)
## miscellaneous
make_windows_bins(repo)
use_cran_granbase(repo)
check_timeout(repo)
build_timeout(repo)
platform(repo)
r_version(repo)
bioc_version(repo)

```

---

|                   |   |
|-------------------|---|
| make_windows_bins | <i>make_windows_bins</i> Return the logical that determines whether to build Windows binaries |
|-------------------|---|

---

## Description

make\_windows\_bins Return the logical that determines whether to build Windows binaries

## Usage

```

make_windows_bins(repo)

## S4 method for signature 'GRANRepository'
make_windows_bins(repo)

```

## Arguments

repo            a GRANRepository object

## Value

Logical indicating whether Windows binaries will be built



---

|             |  |
|-------------|--|
| metadatadir | <i>metadatadir</i> Return the full path to the pkg metadata directory for the final repository deployment. |
|-------------|--|

---

**Description**

metadatadir Return the full path to the pkg metadata directory for the final repository deployment.

**Usage**

```
metadatadir(repo)
```

```
## S4 method for signature 'GRANRepository'
metadatadir(repo)
```

**Arguments**

repo            a GRANRepository object

**Value**

The full path to the metadata directory

---

|                               |  |
|-------------------------------|--|
| notrack,GRANRepository-method | <i>notrack</i> Return the directory which stores retrieved versions of non-GRAN packages for use in virtual repositories |
|-------------------------------|--|

---

**Description**

notrack Return the directory which stores retrieved versions of non-GRAN packages for use in virtual repositories

**Usage**

```
## S4 method for signature 'GRANRepository'
notrack(repo)
```

**Arguments**

repo            a GRANRepository object

**Value**

The path to the notrack directory

---

|       |                                 |
|-------|---------------------------------|
| param | <i>Extract parameter object</i> |
|-------|---------------------------------|

---

**Description**

Extract parameter object

**Usage**

```
param(x)

## S4 method for signature 'GRANRepository'
param(x)

param(x) <- value

## S4 replacement method for signature 'GRANRepository'
param(x) <- value
```

**Arguments**

|       |  |
|-------|--|
| x     | An object with an associated paramater |
| value | A new parameter object                 |

---

|             |   |
|-------------|---|
| pkg_doc_dir | <i>pkg_doc_dir Return the path where test coverage reports for packages will be deployed for use in the build report.</i> |
|-------------|---|

---

**Description**

pkg\_doc\_dir Return the path where test coverage reports for packages will be deployed for use in the build report.

**Usage**

```
pkg_doc_dir(repo)

## S4 method for signature 'GRANRepository'
pkg_doc_dir(repo)
```

**Arguments**

|      |                         |
|------|-------------------------|
| repo | a GRANRepository object |
|------|-------------------------|

**Value**

The directory where test coverage results should be deployed for use in the build report

---

```
prepDirStructure      prepDirStructure
```

---

**Description**

Initialize the directory structure for the GRAN repo

**Usage**

```
prepDirStructure(basedir, repo_name, temp_repo, temp_checkout, tempLibLoc,
  destination, logfile, errlogfile)
```

**Arguments**

|               |  |
|---------------|--|
| basedir       | The base directory. By default the temporary repository, temporary install library, and package staging area will be located in <basedir>/<subrepoName>/, while the temporary source checkout will be in the basedir itself. |
| repo_name     | The name of the repository, e.g. stable or devel   |
| temp_repo     | Location to create the temporary repository  |
| temp_checkout | Location to create temporary checkouts/copies of package source code   |
| tempLibLoc    | Location to create the temporary installed package library for use during the testing process  |
| destination   | Base location (not including repository name) of the final repository to be built.   |
| logfile       | Comprehensive GRAN log file  |
| errlogfile    | Error log file   |

**Note**

This function is not intended for use by the end user.

---

```
repo_base              repo_base Generic accessor function to retrieve the repo specific sub-
  directory within the base directory
```

---

**Description**

repo\_base Generic accessor function to retrieve the repo specific subdirectory within the base directory

**Usage**

```
repo_base(repo)
```

```
## S4 method for signature 'GRANRepository'
repo_base(repo)
```

**Arguments**

repo                    a GRANRepository object

**Value**

The path to the repository specific directory

---

RepoBuildParam-class    *ResultsRow*

---

**Description**

A row from the 'results' slot in the GRANRepository object

Parameters for building a GRAN repository. Most behavior during the GRAN building process is specified via this object/constructor.

**Usage**

```
ResultsRow(name = NA_character_, building = TRUE, status = "ok",
  version = "0.0-0", lastAttempt = NA_character_,
  lastAttemptVersion = NA_character_,
  lastAttemptStatus = NA_character_, lastbuilt = NA_character_,
  lastbuiltversion = NA_character_, lastbuiltstatus = NA_character_,
  buildReason = NA_character_, maintainer = NA_character_,
  suspended = FALSE)
```

```
RepoBuildParam(basedir, repo_name = "current",
  temp_repo = file.path(basedir, repo_name, "tmprepo"),
  temp_checkout = file.path(basedir, "tmpcheckout"),
  errlog = file.path(basedir, repo_name, paste0("GRAN", repo_name,
    "-error.log")), logfile = file.path(basedir, repo_name, paste0("GRAN",
    repo_name, "-full.log")), check_note_ok = TRUE, check_warn_ok = TRUE,
  templibLoc = file.path(basedir, repo_name, "LibLoc"),
  extra_fun = function(...) NULL, destination = basedir, auth = "",
  dest_url = makeFileURL(normalizePath2(destination)),
  shell_init = character(), loginnerfun = writeGRANLog,
  install_test = TRUE, check_test = TRUE, use_cran_granbase = TRUE,
  archive_timing = 2, archive_retries = 2, build_timeout = 10 * 60,
  check_timeout = 15 * 60, email_notifications = FALSE,
  email_opts = list(smtp_server = "localhost", smtp_port = 25,
  sender_email = paste0("gran", repo_name, "@localhost"), unsubscribers =
  NULL), repo_archive = file.path(destination, repo_name, "src",
  "contrib", "Archive"), repo_metadata_dir = file.path(destination,
  repo_name, "src", "contrib", "Meta"), make_windows_bins = TRUE,
  platform = "", r_version = "", bioc_version = "")
```

**Arguments**

|                    |  |
|--------------------|--|
| name               | Name of the package  |
| building           | Logical indicating whether the package is building   |
| status             | Status of the package build  |
| version            | Package version  |
| lastAttempt        | Last attempt time of package build   |
| lastAttemptVersion | Package version when build was last attempted  |
| lastAttemptStatus  | Package build status of last attempt   |
| lastbuilt          | Last built time of package, if successful  |
| lastbuiltversion   | Last built version of package, if successful   |
| lastbuiltstatus    | Last built status of package, if successful  |
| buildReason        | The reason why the package build was attempted   |
| maintainer         | Package maintainer   |
| suspended          | Is the package suspended?  |
| basedir            | The base directory. By default the temporary repository, temporary install library, and package staging area will be located in <basedir>/<subrepoName>/, while the temporary source checkout will be in the basedir itself. |
| repo_name          | The name of the repository, e.g. stable or devel   |
| temp_repo          | Location to create the temporary repository  |
| temp_checkout      | Location to create temporary checkouts/copies of package source code   |
| errlog             | The file to append error output to during the building and testing processes   |
| logfile            | The file to append summary log information to during building and testing  |
| check_note_ok      | logical. Whether packages that raise notes during R CMD check should be considered to have passed  |
| check_warn_ok      | logical. Whether packages that raise warnings during R CMD check should be considered to have passed   |
| tempLibLoc         | Location to create the temporary installed package library for use during the testing process  |
| extra_fun          | currently ignored  |
| destination        | Base location (not including repository name) of the final repository to be built. NOTE: GRANBase:::buildReportURL() expects this argument to have a trailing slash.   |
| auth               | character. Authentication information required to add packages to the manifest.  |
| dest_url           | The base URL the destination directory corresponds to. The subrepository name will be appended to this to generate the URL used when installing from the repository.   |

|                     |   |
|---------------------|---|
| shell_init          | An optional shell script to source before invoking system commands, e.g. a bashrc file. Ignored if "" or not specified.   |
| loginnerfun         | The function to use to write log messages during the repository build process. It will be passed pkg, ..., errfile, logfile, and pkglog based on the other arguments to this function. Defaults to writeGRANLog specified as the full and error log locations, respectively.  |
| install_test        | logical. Should the install test be performed? Required to build packages with vignettes, and for the check test  |
| check_test          | logical. Should R CMD check be run on the packages as a cohort. Requires install test.  |
| use_cran_granbase   | logical. Currently ignored.   |
| archive_timing      | numeric. Number of seconds to wait between attempts to pull a package from the CRAN archive   |
| archive_retries     | numeric. Number of times to retry pulling a package from the CRAN archive.  |
| build_timeout       | numeric. Number of seconds before timeout during the build step for a single package. Defaults to 10 minutes.   |
| check_timeout       | numeric. Number of seconds before timeout during the check step for a single package. Defaults to 15 minutes.   |
| email_notifications | logical. Should email notifications be sent regarding packages that fail to build on GRAN? Defaults to FALSE  |
| email_opts          | List. Email options for sending emails regarding packages that fail to build on GRAN. The list contains 4 elements: smtp_server: the SMTP server - defaults to "localhost", smtp_port: SMTP port number - defaults to 25, sender_email: Whom should the emails be sent as? Defaults to "gran<repo_name>@localhost", unsubscribers: Vector of Perl-style regexes for unsubscribers - defaults to NULL. |
| repo_archive        | Archive directory where older package sources will be saved   |
| repo_metadata_dir   | Directory containing metadata files   |
| make_windows_bins   | Whether to make Windows binary packages   |
| platform            | Name of platform this GRANRepository is being build on  |
| r_version           | R version for this GRANRepository   |
| bioc_version        | Bioconductor version for this GRANRepository  |

**Value**

data.frame

**Note**

This function is not intended for use by the end user.

**Examples**

```
row = ResultsRow(name = "mypkg")
rbp = RepoBuildParam(basedir = tempdir(), repo_name = "myrepo")
```

---

|            |  |
|------------|--|
| RepoToList | <i>Transform a GRANRepository object into a list</i> |
|------------|--|

---

**Description**

Utility to transform a GRANRepository object into a list so that repos saved using GRANBase can be loaded by GRAN without requiring GRANBase

**Usage**

```
RepoToList(repo)
```

```
RepoFromList(rlist)
```

**Arguments**

|       |  |
|-------|--|
| repo  | repository   |
| rlist | A list with entries that are slot name-value pairs for a GRANRepository object |

**Value**

a list suitable for use with RepoFromList

a GRANRepository object

**Examples**

```
repo = GRANRepository(GithubManifest("gmbecker/switchr"), basedir = tempdir())
lst = RepoToList(repo)
repo2 = RepoFromList(lst)
```

---

|              |                                 |
|--------------|---------------------------------|
| repo_results | <i>Repository build results</i> |
|--------------|---------------------------------|

---

**Description**

Repository build results

**Usage**

```

repo_results(x)

## S4 method for signature 'GRANRepository'
repo_results(x)

repo_results(x) <- value

## S4 replacement method for signature 'GRANRepository'
repo_results(x) <- value

```

**Arguments**

```

x          A GRANRepository object
value     The new results data.frame

```

**Value**

A data.frame of build results

---

|          |  |
|----------|--|
| repo_url | <i>repo_url</i> Return the url that the repository will be served at. This is the web address corresponding to repository, suitable for calling contrib.url. e.g <a href="http://www.website.com/gran/current/">http://www.website.com/gran/current/</a> |
|----------|--|

---

**Description**

repo\_url Return the url that the repository will be served at. This is the web address corresponding to repository, suitable for calling contrib.url. e.g <http://www.website.com/gran/current/>

**Usage**

```

repo_url(repo)

## S4 method for signature 'GRANRepository'
repo_url(repo)

## S4 method for signature ''NULL''
repo_url(repo)

```

**Arguments**

```

repo          a GRANRepository object

```

**Value**

For destination, the full path to the contrib directory the packages will be deployed to



---

|         |  |
|---------|--|
| staging | <i>staging</i> Return the staging directory or the staging_logs to be used when building the repository. If the directory does not exist it will be created. |
|---------|--|

---

**Description**

staging Return the staging directory or the staging\_logs to be used when building the repository. If the directory does not exist it will be created.

**Usage**

```
staging(repo)

## S4 method for signature 'GRANRepository'
staging(repo)

staging_logs(repo)

## S4 method for signature 'GRANRepository'
staging_logs(repo)
```

**Arguments**

repo                    a GRANRepository object

**Value**

The path to the repository specific directory  
The path to the repository specific directory

---

|          |                          |
|----------|--------------------------|
| temp_lib | <i>temporary library</i> |
|----------|--------------------------|

---

**Description**

temporary library

**Usage**

```
temp_lib(repo)

## S4 method for signature 'GRANRepository'
temp_lib(repo)
```

**Arguments**

repo                    A GRANRepository object

---

updateGRANRepoObject    *updateGRANRepoObject*

---

**Description**

Update the GRAN repo object with newer information. This is called internally by loadRepo to ensure that the resulting repo doesn't have stale logging closures and the like.

**Usage**

updateGRANRepoObject(object, ...)

**Arguments**

object                    The GRAN repo object  
 ...                        Other parameters from RepoBuildParam

**Note**

This function is cal not intended for use by the end user. This is called internally by loadRepo.

---

windowsbindir            *windowsbindir Return the full path to the location of the windows binary builds*

---

**Description**

windowsbindir Return the full path to the location of the windows binary builds

**Usage**

```
windowsbindir(repo)

## S4 method for signature 'GRANRepository'
windowsbindir(repo)
```

**Arguments**

repo                    a GRANRepository object

**Value**

Full path to the location of the windows binary builds

---

|              |                     |
|--------------|---------------------|
| writeGRANLog | <i>writeGRANLog</i> |
|--------------|---------------------|

---

**Description**

Utility function which writes gran logs

**Usage**

```
writeGRANLog(pkg, msg, type = "full", logfile, errfile, pkglog = NULL)
```

**Arguments**

|         |   |
|---------|---|
| pkg     | The name of the package the log is about  |
| msg     | The log message, collapsed if length>1  |
| type    | "full", "error", "warn", or "both" indicating which log(s) the message should be written to |
| logfile | The location of the full log file to write/append to  |
| errfile | the location of the error log file to write/append to                                       |
| pkglog  | character. The package-specific log file to write to if applicable.                         |

**Note**

This function is not intended for direct use by the end user.

# Index

archivedir, 2  
archivedir, GRANRepository-method  
    (archivedir), 2  
available.packages, 3  
available.packages, ANY  
    (available.packages), 3  
available.packages, ANY-method  
    (available.packages), 3  
available.packages, GRANRepository  
    (available.packages), 3  
available.packages, GRANRepository-method  
    (available.packages), 3

backup\_archive, 4  
backup\_archive, (backup\_archive), 4  
backup\_archive, GRANRepository-method  
    (backup\_archive), 4  
bioc\_version (logfile), 11  
bioc\_version, GRANRepository (logfile),  
    11  
bioc\_version, GRANRepository-method  
    (logfile), 11  
bioc\_version<- (logfile), 11  
bioc\_version<-, GRANRepository  
    (logfile), 11  
bioc\_version<-, GRANRepository-method  
    (logfile), 11  
build\_timeout (logfile), 11  
build\_timeout, GRANRepository (logfile),  
    11  
build\_timeout, GRANRepository-method  
    (logfile), 11  
build\_timeout, RepoBuildParam (logfile),  
    11  
build\_timeout, RepoBuildParam-method  
    (logfile), 11  
build\_timeout<- (logfile), 11  
build\_timeout<-, GRANRepository  
    (logfile), 11  
build\_timeout<-, GRANRepository-method  
    (logfile), 11  
build\_timeout<-, RepoBuildParam  
    (logfile), 11  
build\_timeout<-, RepoBuildParam-method  
    (logfile), 11  
check\_note\_ok (logfile), 11  
check\_note\_ok, GRANRepository (logfile),  
    11  
check\_note\_ok, GRANRepository-method  
    (logfile), 11  
check\_result\_dir, 5  
check\_result\_dir, GRANRepository-method  
    (check\_result\_dir), 5  
check\_test\_on (logfile), 11  
check\_test\_on, GRANRepository (logfile),  
    11  
check\_test\_on, GRANRepository-method  
    (logfile), 11  
check\_test\_on, RepoBuildParam (logfile),  
    11  
check\_test\_on, RepoBuildParam-method  
    (logfile), 11  
check\_timeout (logfile), 11  
check\_timeout, GRANRepository (logfile),  
    11  
check\_timeout, GRANRepository-method  
    (logfile), 11  
check\_timeout, RepoBuildParam (logfile),  
    11  
check\_timeout, RepoBuildParam-method  
    (logfile), 11  
check\_timeout<- (logfile), 11  
check\_timeout<-, GRANRepository  
    (logfile), 11  
check\_timeout<-, GRANRepository-method  
    (logfile), 11  
check\_timeout<-, RepoBuildParam  
    (logfile), 11

- check\_timeout<- ,RepoBuildParam-method (logfile), 11
- check\_warn\_ok (logfile), 11
- check\_warn\_ok,GRANRepository (logfile), 11
- check\_warn\_ok,GRANRepository-method (logfile), 11
- checkout\_dir, 4
- checkout\_dir,GRANRepository-method (checkout\_dir), 4
- checkout\_dir,NULL (checkout\_dir), 4
- checkout\_dir,NULL-method (checkout\_dir), 4
- contrib.url, 5
- contrib.url,GRANRepository (contrib.url), 5
- contrib.url,GRANRepository-method (contrib.url), 5
- coverage\_report\_dir, 6
- coverage\_report\_dir,GRANRepository-method (coverage\_report\_dir), 6
- dep\_repos,GRANRepository-method (logfile), 11
- dep\_repos,RepoBuildParam (logfile), 11
- dest\_base, 7
- dest\_base,GRANRepository-method (dest\_base), 7
- destination, 6
- destination,GRANRepository-method (destination), 6
- email\_notify, 7
- email\_notify,GRANRepository-method (email\_notify), 7
- email\_options, 8
- email\_options,GRANRepository-method (email\_options), 8
- errlogfile, 8
- errlogfile,GRANRepository-method (errlogfile), 8
- errlogfile,RepoBuildParam-method (errlogfile), 8
- extra\_fun (logfile), 11
- extra\_fun,GRANRepository-method (logfile), 11
- GRANRepository (GRANRepository-class), 9
- GRANRepository-class, 9
- GRANRepository-method (backup\_archive), 4
- install\_packages, character, GRANRepository-method, 9
- install\_result\_dir, 10
- install\_result\_dir,GRANRepository-method (install\_result\_dir), 10
- install\_test\_on (logfile), 11
- install\_test\_on,GRANRepository (logfile), 11
- install\_test\_on,GRANRepository-method (logfile), 11
- install\_test\_on,RepoBuildParam (logfile), 11
- install\_test\_on,RepoBuildParam-method (logfile), 11
- loadRepo, 11
- logfile, 11
- logfile,GRANRepository-method (logfile), 11
- logfile,RepoBuildParam-method (logfile), 11
- logfile-method,GRANRepository (logfile), 11
- logfun,GRANRepository (logfile), 11
- logfun,GRANRepository-method (logfile), 11
- logfun<- ,GRANRepository (logfile), 11
- logfun<- ,GRANRepository-method (logfile), 11
- make\_windows\_bins, 16
- make\_windows\_bins,GRANRepository-method (make\_windows\_bins), 16
- metadatadir, 17
- metadatadir,GRANRepository-method (metadatadir), 17
- notrack,GRANRepository-method, 17
- param, 18
- param,GRANRepository (param), 18
- param,GRANRepository-method (param), 18
- param<- (param), 18
- param<- ,GRANRepository (param), 18
- param<- ,GRANRepository-method (param), 18

- pkg\_doc\_dir, 18
- pkg\_doc\_dir, GRANRepository-method (pkg\_doc\_dir), 18
- pkg\_log\_dir (logfile), 11
- pkg\_log\_dir, GRANRepository (logfile), 11
- pkg\_log\_dir, GRANRepository-method (logfile), 11
- pkg\_log\_dir, RepoBuildParam (logfile), 11
- pkg\_log\_dir, RepoBuildParam-method (logfile), 11
- pkg\_log\_file (logfile), 11
- pkg\_log\_file, ANY, GRANRepository-method (logfile), 11
- pkg\_log\_file, ANY, RepoBuildParam-method (logfile), 11
- pkg\_log\_file, GRANRepository (logfile), 11
- pkg\_log\_file, RepoBuildParam (logfile), 11
- platform (logfile), 11
- platform, GRANRepository (logfile), 11
- platform, GRANRepository-method (logfile), 11
- platform<- (logfile), 11
- platform<-, GRANRepository (logfile), 11
- platform<-, GRANRepository-method (logfile), 11
- prepDirStructure, 19
- r\_version (logfile), 11
- r\_version, GRANRepository (logfile), 11
- r\_version, GRANRepository-method (logfile), 11
- r\_version<- (logfile), 11
- r\_version<-, GRANRepository (logfile), 11
- r\_version<-, GRANRepository-method (logfile), 11
- repo\_name (logfile), 11
- repo\_name, GRANRepository (logfile), 11
- repo\_name, GRANRepository-method (logfile), 11
- repo\_results, 23
- repo\_results, GRANRepository (repo\_results), 23
- repo\_results, GRANRepository-method (repo\_results), 23
- repo\_results<- (repo\_results), 23
- repo\_results<-, GRANRepository (repo\_results), 23
- repo\_results<-, GRANRepository-method (repo\_results), 23
- repo\_results<-, GRANRepository-method (repo\_results), 23
- repo\_url, 24
- repo\_url, GRANRepository-method (repo\_url), 24
- repo\_url, NULL (repo\_url), 24
- repo\_url, NULL-method (repo\_url), 24
- repobase, 19
- repobase, GRANRepository-method (repobase), 19
- RepoBuildParam, 15
- RepoBuildParam (RepoBuildParam-class), 20
- RepoBuildParam-class, 20
- RepoFromList (RepoToList), 23
- RepoToList, 23
- ResultsRow (RepoBuildParam-class), 20
- saveRepo (loadRepo), 11
- sh\_init\_script, GRANRepository (logfile), 11
- sh\_init\_script, GRANRepository-method (logfile), 11
- sh\_init\_script<-, GRANRepository (logfile), 11
- sh\_init\_script<-, GRANRepository-method (logfile), 11
- staging, 25
- staging, GRANRepository-method (staging), 25
- staging\_logs (staging), 25
- staging\_logs, GRANRepository-method (staging), 25
- suspended\_pkgs (logfile), 11
- suspended\_pkgs, GRANRepository (logfile), 11
- suspended\_pkgs, GRANRepository-method (logfile), 11
- suspended\_pkgs<- (logfile), 11
- suspended\_pkgs<-, GRANRepository (logfile), 11
- suspended\_pkgs<-, GRANRepository-method (logfile), 11
- temp\_lib, 25
- temp\_lib, GRANRepository (temp\_lib), 25
- temp\_lib, GRANRepository-method (temp\_lib), 25
- temp\_repo (logfile), 11

temp\_repo, GRANRepository (logfile), [11](#)  
temp\_repo, GRANRepository-method  
    (logfile), [11](#)

updateGRANRepoObject, [26](#)

use\_cran\_granbase (logfile), [11](#)  
use\_cran\_granbase, GRANRepository  
    (logfile), [11](#)  
use\_cran\_granbase, GRANRepository-method  
    (logfile), [11](#)  
use\_cran\_granbase, RepoBuildParam  
    (logfile), [11](#)  
use\_cran\_granbase, RepoBuildParam-method  
    (logfile), [11](#)  
use\_cran\_granbase<- (logfile), [11](#)  
use\_cran\_granbase<-, GRANRepository  
    (logfile), [11](#)  
use\_cran\_granbase<-, GRANRepository-method  
    (logfile), [11](#)  
use\_cran\_granbase<-, RepoBuildParam  
    (logfile), [11](#)  
use\_cran\_granbase<-, RepoBuildParam-method  
    (logfile), [11](#)

windowsbindir, [26](#)  
windowsbindir, GRANRepository-method  
    (windowsbindir), [26](#)

writeGRANLog, [27](#)