

Package: woodendesc (via r-universe)

January 28, 2025

Type Package

Title Find Packages and Their Dependencies in Any 'R' Repository

Version 0.2.1

Date 2025-01-28

Description Simplifies getting the available packages, their version codes and dependencies from any 'R' repository. Uses extensive caching for repeated queries.

License GPL (>= 3)

Encoding UTF-8

Roxygen list(markdown = TRUE)

RoxygenNote 7.3.2

Depends R (>= 4.1.0)

Imports httr2 (>= 1.0.0), rlang (>= 1.0.0), utils, versionsort (>= 1.1.0)

Suggests covr, httptest2, knitr, rmarkdown, testthat (>= 3.0.0), usethis, withr, xml2

Config/testthat/edition 3

URL <https://github.com/turtletopia/woodendesc>,
<https://turtletopia.github.io/woodendesc/>

VignetteBuilder knitr

Config/pak/sysreqs libssl-dev

Repository <https://turtletopia.r-universe.dev>

RemoteUrl <https://github.com/turtletopia/woodendesc>

RemoteRef HEAD

RemoteSha e7bae8a57f4f95bb76a022f50c80a61d7a3d4112

Contents

filter_dependencies	3
squash	4
wood_bioc_dependencies	4
wood_bioc_packages	5
wood_bioc_releases	6
wood_bioc_version	7
wood_clear_cache	8
wood_core_dependencies	8
wood_core_packages	9
wood_core_version	9
wood_cran_dependencies	10
wood_cran_latest	11
wood_cran_packages	12
wood_cran_versions	12
wood_dependencies	13
wood_github_dependencies	14
wood_github_latest	15
wood_github_packages	16
wood_github_tags	17
wood_github_versions	18
wood_gitlab_dependencies	18
wood_gitlab_latest	19
wood_gitlab_packages	20
wood_gitlab_tags	21
wood_gitlab_versions	22
wood_local_dependencies	23
wood_local_packages	24
wood_local_versions	24
wood_packages	25
wood_runiverse_dependencies	26
wood_runiverse_packages	27
wood_runiverse_version	28
wood_url_dependencies	29
wood_url_packages	30
wood_url_version	30
wood_versions	31

Index

33

filter_dependencies *Filter dependencies by type*

Description

This function removes all dependencies that are not one of the specified types.

Usage

```
filter_dependencies(object, which = "strong")

## S3 method for class 'wood_deps'
filter_dependencies(object, which = "strong")

## S3 method for class 'wood_dep_squashed'
filter_dependencies(object, which = "strong")

## S3 method for class 'wood_dep_list'
filter_dependencies(object, which = "strong")

## S3 method for class '`NULL`'
filter_dependencies(object, which = "strong")
```

Arguments

object	wood_deps() wood_dep_list() Dependencies to filter.
which	character() A vector listing the types of dependencies, a subset of c("Depends", "Imports", "LinkingTo", "Suggests", "Enhances"). Character string "all" is shorthand for that vector, character string "most" for the same vector without "Enhances", character string "strong" (default) for the first three elements of that vector. The same convention as in tools::package_dependencies() .

Value

Object of the same class as object parameter, but with filtered dependencies.

Examples

```
# It can filter both single-package dependencies...
stats_deps <- wood_core_dependencies("stats")
filter_dependencies(stats_deps)

# ...and a list for multiple packages too.
core_pkgs <- wood_core_packages()
core_deps <- wood_dependencies(core_pkgs, "core")
filter_dependencies(core_deps, c("Imports", "Enhances"))
```

`squash`*Compress dependencies of multiple packages*

Description

This function combines multiple dependency lists into a single `data.frame` with additional info about "origin" package, i.e. which package has this dependency.

Usage

```
squash(object)

## S3 method for class 'wood_dep_list'
squash(object)
```

Arguments

<code>object</code>	<code>wood_dep_list()</code> A list of dependencies to compress.
---------------------	---

Value

A `data.frame` object similar to these returned by `_dependencies` functions, but with additional column named `origin`.

Examples

```
deps <- wood_dependencies(c("versionsort", "deepdep", "ggigbtq"))
squash(deps)
```

`wood_bioc_dependencies`*Get dependencies of a package in Bioconductor*

Description

This function queries Bioconductor for dependencies of the selected package for given Bioconductor release.

Usage

```
wood_bioc_dependencies(package, release = "release")
```

Arguments

package	character(1) Name of a package to query for.
release	character(1) Bioconductor release to query for. Can be either a numeric code (like 3.12) or one of two keywords: "release" and "devel".

Value

A data frame with three columns, all in string format:

- package (package name),
- version (minimum version requirement or NA if none),
- type (dependency type, e.g. "Imports").

See Also

Functions for Bioconductor: [wood_bioc_packages\(\)](#), [wood_bioc_releases\(\)](#), [wood_bioc_version\(\)](#)

Functions that query package dependencies: [wood_core_dependencies\(\)](#), [wood_cran_dependencies\(\)](#), [wood_dependencies\(\)](#), [wood_github_dependencies\(\)](#), [wood_gitlab_dependencies\(\)](#), [wood_local_dependencies\(\)](#), [wood_runiverse_dependencies\(\)](#), [wood_url_dependencies\(\)](#)

Examples

```
wood_bioc_dependencies("Biostrings")

# Will dependencies change?
wood_bioc_dependencies("Biostrings", "devel")
# And what about dependencies in the past?
wood_bioc_dependencies("Biostrings", "2.10")
wood_bioc_dependencies("Biostrings", "1.8")
```

`wood_bioc_packages` *List available packages in Bioconductor*

Description

This function queries Bioconductor for a list of available packages. Current and previous releases can be queried, up to Bioconductor 1.5.

Usage

```
wood_bioc_packages(release = "release")
```

Arguments

release character(1)
Bioconductor release to query for. Can be either a numeric code (like 3.12) or one of two keywords: "release" and "devel".

Value

A character vector of available packages.

See Also

Functions for Bioconductor: [wood_bioc_dependencies\(\)](#), [wood_bioc_releases\(\)](#), [wood_bioc_version\(\)](#)
Functions that query available packages: [wood_core_packages\(\)](#), [wood_cran_packages\(\)](#), [wood_github_packages\(\)](#), [wood_gitlab_packages\(\)](#), [wood_local_packages\(\)](#), [wood_packages\(\)](#), [wood_runiverse_packages\(\)](#), [wood_url_packages\(\)](#)

Examples

```
print(wood_bioc_packages(), max = 15)
# is the same as
print(wood_bioc_packages("release"), max = 15)

# A glimpse in the future:
print(wood_bioc_packages("devel"), max = 15)
# Previous versions can be queried as well:
print(wood_bioc_packages("3.9"), max = 15)
# The oldest available version is 1.8:
wood_bioc_packages("1.8")
```

wood_bioc_releases *List Bioconductor releases*

Description

This function scrapes the Bioconductor site for version codes of all its releases.

Usage

```
wood_bioc_releases()
```

Value

A character vector, where each element is a Bioconductor release.

See Also

Functions for Bioconductor: [wood_bioc_dependencies\(\)](#), [wood_bioc_packages\(\)](#), [wood_bioc_version\(\)](#)

Examples

```
wood_bioc_releases()
```

wood_bioc_version	<i>Get available package version in Bioconductor</i>
-------------------	--

Description

This function queries Bioconductor for the code of the available version of the selected package for given Bioconductor release.

Usage

```
wood_bioc_version(package, release = "release")
```

Arguments

package	character(1) Name of a package to query for.
release	character(1) Bioconductor release to query for. Can be either a numeric code (like 3.12) or one of two keywords: "release" and "devel".

Value

A single string with a version code.

See Also

Functions for Bioconductor: [wood_bioc_dependencies\(\)](#), [wood_bioc_packages\(\)](#), [wood_bioc_releases\(\)](#)

Functions that query package versions: [wood_core_version\(\)](#), [wood_cran_latest\(\)](#), [wood_cran_versions\(\)](#), [wood_github_latest\(\)](#), [wood_github_versions\(\)](#), [wood_gitlab_latest\(\)](#), [wood_gitlab_versions\(\)](#), [wood_local_versions\(\)](#), [wood_runiverse_version\(\)](#), [wood_url_version\(\)](#), [wood_versions\(\)](#)

Examples

```
wood_bioc_version("Biostrings")

# What's coming next?
wood_bioc_version("Biostrings", release = "devel")
# Can query releases as old as 1.8:
wood_bioc_version("Biostrings", release = "1.8")
```

wood_clear_cache	<i>Clear cache</i>
------------------	--------------------

Description

Remove all **woodendesc**-related cached files.

Usage

```
wood_clear_cache()
```

Value

An integer code, invisibly, the same as in `base::unlink()`.

wood_core_dependencies	<i>Get dependencies of a core R package</i>
------------------------	---

Description

This function collects the dependencies of the selected core R package.

Usage

```
wood_core_dependencies(package)
```

Arguments

package	character(1) Name of a package to query for.
---------	---

Value

A data frame with three columns, all in string format:

- package (package name),
- version (minimum version requirement or NA if none),
- type (dependency type, e.g. "Imports").

See Also

Functions for core R packages: [wood_core_packages\(\)](#), [wood_core_version\(\)](#)

Functions that query package dependencies: [wood_bioc_dependencies\(\)](#), [wood_cran_dependencies\(\)](#), [wood_dependencies\(\)](#), [wood_github_dependencies\(\)](#), [wood_gitlab_dependencies\(\)](#), [wood_local_dependencies\(\)](#), [wood_runiverse_dependencies\(\)](#), [wood_url_dependencies\(\)](#)

Examples

```
wood_core_dependencies("tcltk")
```

wood_core_packages *List core R packages*

Description

This function returns the list of base R packages.

Usage

```
wood_core_packages()
```

Value

A character vector of available packages.

See Also

Functions for core R packages: [wood_core_dependencies\(\)](#), [wood_core_version\(\)](#)

Functions that query available packages: [wood_bioc_packages\(\)](#), [wood_cran_packages\(\)](#), [wood_github_packages\(\)](#), [wood_gitlab_packages\(\)](#), [wood_local_packages\(\)](#), [wood_packages\(\)](#), [wood_runiverse_packages\(\)](#), [wood_url_packages\(\)](#)

Examples

```
wood_core_packages()
```

wood_core_version *Get current version of a core R package*

Description

This function checks the version of the selected core R package.

Usage

```
wood_core_version(package)
```

Arguments

package character(1)
Name of a package to query for.

Value

A single string with a version code. Should be equal to the R version (`base::getRversion()`).

See Also

Functions for core R packages: `wood_core_dependencies()`, `wood_core_packages()`

Functions that query package versions: `wood_bioc_version()`, `wood_cran_latest()`, `wood_cran_versions()`, `wood_github_latest()`, `wood_github_versions()`, `wood_gitlab_latest()`, `wood_gitlab_versions()`, `wood_local_versions()`, `wood_runiverse_version()`, `wood_url_version()`, `wood_versions()`

Examples

```
wood_core_version("graphics")
```

```
wood_cran_dependencies
```

Get dependencies of a package on CRAN

Description

This function queries CRAN for dependencies of the selected version of a selected package. By default, it queries the latest version.

Usage

```
wood_cran_dependencies(package, version = "latest")
```

Arguments

package	character(1) Name of a package to query for.
version	character(1) A version code without leading "v", e.g. "1.6.0".

Value

A data frame with three columns, all in string format:

- package (package name),
- version (minimum version requirement or NA if none),
- type (dependency type, e.g. "Imports").

See Also

Functions for CRAN: [wood_cran_latest\(\)](#), [wood_cran_packages\(\)](#), [wood_cran_versions\(\)](#)

Functions that query package dependencies: [wood_bioc_dependencies\(\)](#), [wood_core_dependencies\(\)](#), [wood_dependencies\(\)](#), [wood_github_dependencies\(\)](#), [wood_gitlab_dependencies\(\)](#), [wood_local_dependencies\(\)](#), [wood_runiverse_dependencies\(\)](#), [wood_url_dependencies\(\)](#)

Examples

```
wood_cran_dependencies("deepdep")
wood_cran_dependencies("ggplot2", version = "3.4.0")
```

wood_cran_latest	<i>Get latest package version on CRAN</i>
------------------	---

Description

This function queries CRAN for the code of the latest version of the selected package.

Usage

```
wood_cran_latest(package)
```

Arguments

package	character(1) Name of a package to query for.
---------	---

Value

A single string with a version code.

See Also

Functions for CRAN: [wood_cran_dependencies\(\)](#), [wood_cran_packages\(\)](#), [wood_cran_versions\(\)](#)

Functions that query package versions: [wood_bioc_version\(\)](#), [wood_core_version\(\)](#), [wood_cran_versions\(\)](#), [wood_github_latest\(\)](#), [wood_github_versions\(\)](#), [wood_gitlab_latest\(\)](#), [wood_gitlab_versions\(\)](#), [wood_local_versions\(\)](#), [wood_runiverse_version\(\)](#), [wood_url_version\(\)](#), [wood_versions\(\)](#)

Examples

```
wood_cran_latest("versionsort")
```

wood_cran_packages *List available packages on CRAN*

Description

This function queries CRAN for a list of available packages. They are returned as a vector of strings, each element being a package name.

Usage

```
wood_cran_packages()
```

Value

A character vector of available packages.

See Also

Functions for CRAN: [wood_cran_dependencies\(\)](#), [wood_cran_latest\(\)](#), [wood_cran_versions\(\)](#)

Functions that query available packages: [wood_bioc_packages\(\)](#), [wood_core_packages\(\)](#), [wood_github_packages\(\)](#), [wood_gitlab_packages\(\)](#), [wood_local_packages\(\)](#), [wood_packages\(\)](#), [wood_runiverse_packages\(\)](#), [wood_url_packages\(\)](#)

Examples

```
print(wood_cran_packages(), max = 15)
```

wood_cran_versions *Get all package versions on CRAN*

Description

This function queries CRAN for the codes of all available package versions, both current and archived, for a given package.

Usage

```
wood_cran_versions(package)
```

Arguments

package character(1)
Name of a package to query for.

Value

A character vector of version codes.

See Also

Functions for CRAN: [wood_cran_dependencies\(\)](#), [wood_cran_latest\(\)](#), [wood_cran_packages\(\)](#)

Functions that query package versions: [wood_bioc_version\(\)](#), [wood_core_version\(\)](#), [wood_cran_latest\(\)](#), [wood_github_latest\(\)](#), [wood_github_versions\(\)](#), [wood_gitlab_latest\(\)](#), [wood_gitlab_versions\(\)](#), [wood_local_versions\(\)](#), [wood_runiverse_version\(\)](#), [wood_url_version\(\)](#), [wood_versions\(\)](#)

Examples

```
wood_cran_versions("versionsort")
```

wood_dependencies *Find dependencies from multiple repositories*

Description

This function queries the selected repositories for dependencies of the selected packages and returns the first working occurrence.

Usage

```
wood_dependencies(packages, repos = "cran")
```

Arguments

- | | |
|----------|--|
| packages | character()
Names of the packages to query for. |
| repos | character()
A vector of repositories to query. The following values are available: <ul style="list-style-type: none"> • "cran", meaning CRAN; • "bioc@release", meaning Bioconductor, where release should be replaced with a valid Bioconductor release code or one of "release", "devel"; • "runiverse@universe", meaning R-universe, where universe should be replaced with a valid universe name (e.g. <code>turtletopia</code>); • "github/user", meaning GitHub, where user should be replaced with a valid user or organization name (e.g. <code>turtletopia</code>); • "gitlab/user", meaning GitLab, where user should be replaced with a valid user or group name (e.g. <code>r-packages</code>); • "local#index", meaning a local library, where index should be replaced with an integer index of a library in <code>base::libPaths()</code> or an all keyword; • "core", meaning base R packages; • a URL if none of the above is matched. |

Value

A list named after queried packages, each element being a data frame with three columns, all in string format:

- package (package name),
- version (minimum version requirement or NA if none),
- type (dependency type, e.g. "Imports").

See Also

Functions for all repositories: [wood_packages\(\)](#), [wood_versions\(\)](#)

Functions that query package dependencies: [wood_bioc_dependencies\(\)](#), [wood_core_dependencies\(\)](#), [wood_cran_dependencies\(\)](#), [wood_github_dependencies\(\)](#), [wood_gitlab_dependencies\(\)](#), [wood_local_dependencies\(\)](#), [wood_runiverse_dependencies\(\)](#), [wood_url_dependencies\(\)](#)

Examples

```
wood_dependencies("stats", "core")
# Multiple packages are also possible:
wood_dependencies(
  c("tibble", "Biostrings", "woodendesc"),
  repos = c("runiverse@turtletopia", "cran", "bioc@1.5")
)
# By default, only CRAN is queried:
wood_dependencies("versionsort")
```

```
wood_github_dependencies
```

Get dependencies of a package on GitHub

Description

This function queries GitHub for dependencies of the selected tagged commit of a repo. By default, it queries the latest commit instead.

Usage

```
wood_github_dependencies(package, user, tag = "latest")
```

Arguments

package	character(1) Name of a package to query for.
user	character(1) Name of a user or organization.
tag	character(1) Tag of a commit on GitHub or "latest" for the latest (possibly untagged) commit.

Value

A data frame with three columns, all in string format:

- package (package name),
- version (minimum version requirement or NA if none),
- type (dependency type, e.g. "Imports").

See Also

Functions for GitHub: [wood_github_latest\(\)](#), [wood_github_packages\(\)](#), [wood_github_tags\(\)](#), [wood_github_versions\(\)](#)

Functions that query package dependencies: [wood_bioc_dependencies\(\)](#), [wood_core_dependencies\(\)](#), [wood_cran_dependencies\(\)](#), [wood_dependencies\(\)](#), [wood_gitlab_dependencies\(\)](#), [wood_local_dependencies\(\)](#), [wood_runiverse_dependencies\(\)](#), [wood_url_dependencies\(\)](#)

Examples

```
wood_github_dependencies("gglgltq", "turtletopia")
wood_github_dependencies("versionsort", "turtletopia", tag = "v1.0.0")
```

wood_github_latest	<i>Get current package version on GitHub</i>
--------------------	--

Description

This function queries GitHub for the code of the current package version. This may reference a non-tagged commit; for the analysis of tagged commits only, see [wood_github_versions\(\)](#).

Usage

```
wood_github_latest(package, user)
```

Arguments

package	character(1) Name of a package to query for.
user	character(1) Name of a user or organization.

Value

A character vector of version codes.

See Also

Functions for GitHub: [wood_github_dependencies\(\)](#), [wood_github_packages\(\)](#), [wood_github_tags\(\)](#), [wood_github_versions\(\)](#)

Functions that query package versions: [wood_bioc_version\(\)](#), [wood_core_version\(\)](#), [wood_cran_latest\(\)](#), [wood_cran_versions\(\)](#), [wood_github_versions\(\)](#), [wood_gitlab_latest\(\)](#), [wood_gitlab_versions\(\)](#), [wood_local_versions\(\)](#), [wood_runiverse_version\(\)](#), [wood_url_version\(\)](#), [wood_versions\(\)](#)

Examples

```
# Latest version code is returned
wood_github_latest("gglgbtq", "turtletopia")

# To get the latest *tagged* version code instead, use:
codes <- wood_github_versions("gglgbtq", "turtletopia")
versionsort::ver_latest(codes)
```

`wood_github_packages` *List available package on a Github account*

Description

This function finds packages among repositories belonging to a selected account. They are returned as a vector of strings, each element being a repository (and in most cases, package) name.

Usage

```
wood_github_packages(user, include_forks = FALSE)
```

Arguments

<code>user</code>	<code>character(1)</code> Name of a user or organization.
<code>include_forks</code>	<code>logical(1)</code> Whether to include packages forked from other accounts.

Value

A character vector of available packages.

See Also

Functions for GitHub: [wood_github_dependencies\(\)](#), [wood_github_latest\(\)](#), [wood_github_tags\(\)](#), [wood_github_versions\(\)](#)

Functions that query available packages: [wood_bioc_packages\(\)](#), [wood_core_packages\(\)](#), [wood_cran_packages\(\)](#), [wood_gitlab_packages\(\)](#), [wood_local_packages\(\)](#), [wood_packages\(\)](#), [wood_runiverse_packages\(\)](#), [wood_url_packages\(\)](#)

Examples

```
wood_github_packages("turtletopia")
# I moved most my packages to turtletopia
wood_github_packages("ErdaradunGaztea", include_forks = TRUE)
```

wood_github_tags	<i>List GitHub tags for a repository</i>
------------------	--

Description

This function queries GitHub for a list of tags for a given repository (indicated by a combination of package name and username). Each element is a tag, which need not be identical or even related to version code of the package at the given time.

Usage

```
wood_github_tags(package, user)
```

Arguments

package	character(1) Name of a package to query for.
user	character(1) Name of a user or organization.

Value

A character vector of repository tags.

See Also

Functions for GitHub: [wood_github_dependencies\(\)](#), [wood_github_latest\(\)](#), [wood_github_packages\(\)](#), [wood_github_versions\(\)](#)

Examples

```
wood_github_tags("gglgbtq", "turtletopia")
# Sometimes there are no tags (yet?)
wood_github_tags("ggpizza", "turtletopia")
```

wood_github_versions *Get all tagged package versions on GitHub*

Description

This function queries GitHub for the codes of all tagged package versions. To get the latest available version, see [wood_github_latest\(\)](#).

Usage

```
wood_github_versions(package, user)
```

Arguments

package	character(1) Name of a package to query for.
user	character(1) Name of a user or organization.

Value

A character vector of version codes.

See Also

Functions for GitHub: [wood_github_dependencies\(\)](#), [wood_github_latest\(\)](#), [wood_github_packages\(\)](#), [wood_github_tags\(\)](#)

Functions that query package versions: [wood_bioc_version\(\)](#), [wood_core_version\(\)](#), [wood_cran_latest\(\)](#), [wood_cran_versions\(\)](#), [wood_github_latest\(\)](#), [wood_gitlab_latest\(\)](#), [wood_gitlab_versions\(\)](#), [wood_local_versions\(\)](#), [wood_runiverse_version\(\)](#), [wood_url_version\(\)](#), [wood_versions\(\)](#)

Examples

```
wood_github_versions("versionsort", "turtletopia")
```

wood_gitlab_dependencies

Get dependencies of a package on GitLab

Description

This function queries GitLab for dependencies of the selected tagged commit of a repo. By default, it queries the latest commit instead.

Usage

```
wood_gitlab_dependencies(package, user, tag = "latest")
```

Arguments

package	character(1) Name of a package to query for.
user	character(1) Name of a user or organization.
tag	character(1) Tag of a commit on GitLab or "latest" for the latest (possibly untagged) commit.

Value

A data frame with three columns, all in string format:

- package (package name),
- version (minimum version requirement or NA if none),
- type (dependency type, e.g. "Imports").

See Also

Functions for GitLab: [wood_gitlab_latest\(\)](#), [wood_gitlab_packages\(\)](#), [wood_gitlab_tags\(\)](#), [wood_gitlab_versions\(\)](#)

Functions that query package dependencies: [wood_bioc_dependencies\(\)](#), [wood_core_dependencies\(\)](#), [wood_cran_dependencies\(\)](#), [wood_dependencies\(\)](#), [wood_github_dependencies\(\)](#), [wood_local_dependencies\(\)](#), [wood_runiverse_dependencies\(\)](#), [wood_url_dependencies\(\)](#)

Examples

```
wood_gitlab_dependencies("limonaid", "r-packages")  
wood_gitlab_dependencies("rock", "r-packages", tag = "0.6.3")
```

wood_gitlab_latest *Get current package version on GitLab*

Description

This function queries GitLab for the code of the current package version. This may reference a non-tagged commit; for the analysis of tagged commits only, see [wood_gitlab_versions\(\)](#).

Usage

```
wood_gitlab_latest(package, user)
```

Arguments

package	character(1) Name of a package to query for.
user	character(1) Name of a user or organization.

Value

A character vector of version codes.

See Also

Functions for GitLab: [wood_gitlab_dependencies\(\)](#), [wood_gitlab_packages\(\)](#), [wood_gitlab_tags\(\)](#), [wood_gitlab_versions\(\)](#)

Functions that query package versions: [wood_bioc_version\(\)](#), [wood_core_version\(\)](#), [wood_cran_latest\(\)](#), [wood_cran_versions\(\)](#), [wood_github_latest\(\)](#), [wood_github_versions\(\)](#), [wood_gitlab_versions\(\)](#), [wood_local_versions\(\)](#), [wood_runiverse_version\(\)](#), [wood_url_version\(\)](#), [wood_versions\(\)](#)

Examples

```
if (interactive()) {
  # Latest version code is returned
  wood_gitlab_latest("rock", "r-packages")

  # To get the latest *tagged* version code instead, use:
  wood_gitlab_versions("rock", "r-packages") |>
    versionsort::ver_latest()
}
```

`wood_gitlab_packages` *List available package on a Gitlab account*

Description

This function finds packages among repositories belonging to a selected account. They are returned as a vector of strings, each element being a repository (and in most cases, package) name.

Usage

```
wood_gitlab_packages(user, include_forks = FALSE)
```

Arguments

user	character(1) Name of a user or organization.
include_forks	logical(1) Whether to include packages forked from other accounts.

Value

A character vector of available packages.

See Also

Functions for GitLab: [wood_gitlab_dependencies\(\)](#), [wood_gitlab_latest\(\)](#), [wood_gitlab_tags\(\)](#), [wood_gitlab_versions\(\)](#)

Functions that query available packages: [wood_bioc_packages\(\)](#), [wood_core_packages\(\)](#), [wood_cran_packages\(\)](#), [wood_github_packages\(\)](#), [wood_local_packages\(\)](#), [wood_packages\(\)](#), [wood_runiverse_packages\(\)](#), [wood_url_packages\(\)](#)

Examples

```
if (interactive()) {  
  wood_gitlab_packages("r-packages")  
  # The function takes care of differentiating  
  # between users and groups internally  
  wood_gitlab_packages("matherion")  
}
```

wood_gitlab_tags	<i>List GitLab tags for a repository</i>
------------------	--

Description

This function queries GitLab for a list of tags for a given repository (indicated by a combination of package name and username). Each element is a tag, which need not be identical or even related to version code of the package at the given time.

Usage

```
wood_gitlab_tags(package, user)
```

Arguments

package	character(1) Name of a package to query for.
user	character(1) Name of a user or organization.

Value

A character vector of repository tags.

See Also

Functions for GitLab: [wood_gitlab_dependencies\(\)](#), [wood_gitlab_latest\(\)](#), [wood_gitlab_packages\(\)](#), [wood_gitlab_versions\(\)](#)

Examples

```
wood_gitlab_tags("rock", "r-packages")
```

wood_gitlab_versions *Get all tagged package versions on GitLab*

Description

This function queries GitLab for the codes of all tagged package versions. To get the latest available version, see [wood_gitlab_latest\(\)](#).

Usage

```
wood_gitlab_versions(package, user)
```

Arguments

package	character(1) Name of a package to query for.
user	character(1) Name of a user or organization.

Value

A character vector of version codes.

See Also

Functions for GitLab: [wood_gitlab_dependencies\(\)](#), [wood_gitlab_latest\(\)](#), [wood_gitlab_packages\(\)](#), [wood_gitlab_tags\(\)](#)

Functions that query package versions: [wood_bioc_version\(\)](#), [wood_core_version\(\)](#), [wood_cran_latest\(\)](#), [wood_cran_versions\(\)](#), [wood_github_latest\(\)](#), [wood_github_versions\(\)](#), [wood_gitlab_latest\(\)](#), [wood_local_versions\(\)](#), [wood_runiverse_version\(\)](#), [wood_url_version\(\)](#), [wood_versions\(\)](#)

Examples

```
if (interactive()) {  
  wood_gitlab_versions("rock", "r-packages")  
}
```

`wood_local_dependencies`*Get dependencies of locally available package*

Description

This function extracts dependencies of the selected package, searching through specified library paths.

Usage

```
wood_local_dependencies(package, paths = .libPaths()[1])
```

Arguments

<code>package</code>	<code>character(1)</code> Name of a package to query for.
<code>paths</code>	<code>character()</code> Paths to local libraries, by default the first element of <code>.libPaths()</code> . If equal to "all", uses all elements of <code>.libPaths()</code> .

Value

A data frame with three columns, all in string format:

- `package` (package name),
- `version` (minimum version requirement or NA if none),
- `type` (dependency type, e.g. "Imports").

See Also

Functions for local libraries: [wood_local_packages\(\)](#), [wood_local_versions\(\)](#)

Functions that query package dependencies: [wood_bioc_dependencies\(\)](#), [wood_core_dependencies\(\)](#), [wood_cran_dependencies\(\)](#), [wood_dependencies\(\)](#), [wood_github_dependencies\(\)](#), [wood_gitlab_dependencies\(\)](#), [wood_runiverse_dependencies\(\)](#), [wood_url_dependencies\(\)](#)

Examples

```
wood_local_dependencies("woodendesc")  
wood_local_dependencies("httr2", paths = "all")
```

wood_local_packages *List locally available packages*

Description

This function searches for installed packages inside specified library paths.

Usage

```
wood_local_packages(paths = .libPaths()[1])
```

Arguments

paths character()
Paths to local libraries, by default the first element of `.libPaths()`. If equal to "all", uses all elements of `.libPaths()`.

Value

A character vector of available packages.

See Also

Functions for local libraries: [wood_local_dependencies\(\)](#), [wood_local_versions\(\)](#)

Functions that query available packages: [wood_bioc_packages\(\)](#), [wood_core_packages\(\)](#), [wood_cran_packages\(\)](#), [wood_github_packages\(\)](#), [wood_gitlab_packages\(\)](#), [wood_packages\(\)](#), [wood_runiverse_packages\(\)](#), [wood_url_packages\(\)](#)

Examples

```
wood_local_packages()  
wood_local_packages("all")
```

wood_local_versions *Get locally available package versions*

Description

This function searches for installed package versions inside specified library paths.

Usage

```
wood_local_versions(package, paths = .libPaths()[1])
```


Arguments

package	character(1) Name of a package to query for.
paths	character() Paths to local libraries, by default the first element of <code>.libPaths()</code> . If equal to "all", uses all elements of <code>.libPaths()</code> .

Value

A character vector of version codes.

See Also

Functions for local libraries: `wood_local_dependencies()`, `wood_local_packages()`

Functions that query package versions: `wood_bioc_version()`, `wood_core_version()`, `wood_cran_latest()`, `wood_cran_versions()`, `wood_github_latest()`, `wood_github_versions()`, `wood_gitlab_latest()`, `wood_gitlab_versions()`, `wood_runiverse_version()`, `wood_url_version()`, `wood_versions()`

Examples

```
wood_local_versions("httr2")
```

wood_packages

Collect available packages from multiple repositories

Description

This function queries the selected repositories for the available packages and collects the unique occurrences.

Usage

```
wood_packages(repos = "cran")
```

Arguments

repos	character() A vector of repositories to query. The following values are available: <ul style="list-style-type: none"> • "cran", meaning CRAN; • "bioc@release", meaning Bioconductor, where release should be replaced with a valid Bioconductor release code or one of "release", "devel"; • "runiverse@universe", meaning R-universe, where universe should be replaced with a valid universe name (e.g. turtletopia); • "github/user", meaning GitHub, where user should be replaced with a valid user or organization name (e.g. turtletopia);
-------	---

- "gitlab/user", meaning GitLab, where user should be replaced with a valid user or group name (e.g. r-packages);
- "local#index", meaning a local library, where index should be replaced with an integer index of a library in `base::.libPaths()` or an all keyword;
- "core", meaning base R packages;
- a URL if none of the above is matched.

Value

A character vector of available packages.

See Also

Functions for all repositories: `wood_dependencies()`, `wood_versions()`

Functions that query available packages: `wood_bioc_packages()`, `wood_core_packages()`, `wood_cran_packages()`, `wood_github_packages()`, `wood_gitlab_packages()`, `wood_local_packages()`, `wood_runiverse_packages()`, `wood_url_packages()`

Examples

```
print(wood_packages(c("local#all", "bioc@1.7", "cran", "core")), max = 15)
wood_packages(c("https://colinfay.me", "runiverse@turtletopia"))
# By default, only CRAN is queried:
print(wood_packages(), max = 15)
```

wood_runiverse_dependencies

Get dependencies of a package in one of R universes

Description

This function queries the selected universe for dependencies of the selected package.

Usage

```
wood_runiverse_dependencies(package, universe = "ropensci")
```

Arguments

package	character(1) Name of a package to query for.
universe	character(1) Name of a universe within R-universe, e.g. "ropensci".

Value

A data frame with three columns, all in string format:

- package (package name),
- version (minimum version requirement or NA if none),
- type (dependency type, e.g. "Imports").

See Also

Functions for R-universe: [wood_runiverse_packages\(\)](#), [wood_runiverse_version\(\)](#)

Functions that query package dependencies: [wood_bioc_dependencies\(\)](#), [wood_core_dependencies\(\)](#), [wood_cran_dependencies\(\)](#), [wood_dependencies\(\)](#), [wood_github_dependencies\(\)](#), [wood_gitlab_dependencies\(\)](#), [wood_local_dependencies\(\)](#), [wood_url_dependencies\(\)](#)

Examples

```
wood_runiverse_dependencies("targets")
wood_runiverse_dependencies("ggplot2", universe = "tidyverse")
```

wood_runiverse_packages

List available packages in one of R universes

Description

This function queries the selected universe for a list of available packages. They are returned as a vector of strings, each element being a package name.

Usage

```
wood_runiverse_packages(universe = "ropensci")
```

Arguments

universe	character(1)
----------	--------------

Name of a universe within R-universe, e.g. "ropensci".

Value

A character vector of available packages.

See Also

Functions for R-universe: [wood_runiverse_dependencies\(\)](#), [wood_runiverse_version\(\)](#)

Functions that query available packages: [wood_bioc_packages\(\)](#), [wood_core_packages\(\)](#), [wood_cran_packages\(\)](#), [wood_github_packages\(\)](#), [wood_gitlab_packages\(\)](#), [wood_local_packages\(\)](#), [wood_packages\(\)](#), [wood_url_packages\(\)](#)

Examples

```
wood_runiverse_packages()
wood_runiverse_packages("tidyverse")
```

```
wood_runiverse_version
```

Get current package version in one of R universes

Description

This function queries the selected universe for the code of the current version of the selected package.

Usage

```
wood_runiverse_version(package, universe = "ropensci")
```

Arguments

package	character(1) Name of a package to query for.
universe	character(1) Name of a universe within R-universe, e.g. "ropensci".

Value

A single string with a version code.

See Also

Functions for R-universe: [wood_runiverse_dependencies\(\)](#), [wood_runiverse_packages\(\)](#)

Functions that query package versions: [wood_bioc_version\(\)](#), [wood_core_version\(\)](#), [wood_cran_latest\(\)](#), [wood_cran_versions\(\)](#), [wood_github_latest\(\)](#), [wood_github_versions\(\)](#), [wood_gitlab_latest\(\)](#), [wood_gitlab_versions\(\)](#), [wood_local_versions\(\)](#), [wood_url_version\(\)](#), [wood_versions\(\)](#)

Examples

```
wood_runiverse_version("targets")
wood_runiverse_version("ggplot2", universe = "tidyverse")
```

wood_url_dependencies *Get dependencies of a package in any repository*

Description

This function queries any online repository for dependencies of the selected package.

Usage

```
wood_url_dependencies(package, repository)
```

Arguments

package	character(1) Name of a package to query for.
repository	character(1) URL to repository, e.g. "http://www.omegahat.net/R".

Value

A data frame with three columns, all in string format:

- package (package name),
- version (minimum version requirement or NA if none),
- type (dependency type, e.g. "Imports").

See Also

Functions for any URL: [wood_url_packages\(\)](#), [wood_url_version\(\)](#)

Functions that query package dependencies: [wood_bioc_dependencies\(\)](#), [wood_core_dependencies\(\)](#), [wood_cran_dependencies\(\)](#), [wood_dependencies\(\)](#), [wood_github_dependencies\(\)](#), [wood_gitlab_dependencies\(\)](#), [wood_local_dependencies\(\)](#), [wood_runiverse_dependencies\(\)](#)

Examples

```
wood_url_dependencies("tidystringdist", repository = "https://colinfay.me")
```

wood_url_packages *List available packages in any repository*

Description

This function queries any online repository for a list of available packages. They are returned as a vector of strings, each element being a package name. The data is retrieved from `/src/contrib/PACKAGES` or `/src/contrib/PACKAGES.gz` file.

Usage

```
wood_url_packages(repository)
```

Arguments

repository character(1)
URL to repository, e.g. "http://www.omegahat.net/R".

Value

A character vector of available packages.

See Also

Functions for any URL: [wood_url_dependencies\(\)](#), [wood_url_version\(\)](#)

Functions that query available packages: [wood_bioc_packages\(\)](#), [wood_core_packages\(\)](#), [wood_cran_packages\(\)](#), [wood_github_packages\(\)](#), [wood_gitlab_packages\(\)](#), [wood_local_packages\(\)](#), [wood_packages\(\)](#), [wood_runiverse_packages\(\)](#)

Examples

```
wood_url_packages("https://colinfay.me")  
# Trailing slashes are removed  
wood_url_packages("https://colinfay.me/")
```

wood_url_version *Get current package version in any repository*

Description

This function queries any online repository for the code of the current version of the selected package.

Usage

```
wood_url_version(package, repository)
```

Arguments

package	character(1) Name of a package to query for.
repository	character(1) URL to repository, e.g. "http://www.omegahat.net/R".

Value

A single string with a version code.

See Also

Functions for any URL: [wood_url_dependencies\(\)](#), [wood_url_packages\(\)](#)

Functions that query package versions: [wood_bioc_version\(\)](#), [wood_core_version\(\)](#), [wood_cran_latest\(\)](#), [wood_cran_versions\(\)](#), [wood_github_latest\(\)](#), [wood_github_versions\(\)](#), [wood_gitlab_latest\(\)](#), [wood_gitlab_versions\(\)](#), [wood_local_versions\(\)](#), [wood_runiverse_version\(\)](#), [wood_versions\(\)](#)

Examples

```
wood_url_version("dockerfiler", repository = "https://colinfay.me")
```

wood_versions	<i>Collect all package versions from multiple repositories</i>
---------------	--

Description

This function queries the selected repositories for version codes of the selected packages and collects the unique occurrences.

Usage

```
wood_versions(packages, repos = "cran")
```

Arguments

packages	character() Names of the packages to query for.
repos	character() A vector of repositories to query. The following values are available: <ul style="list-style-type: none"> • "cran", meaning CRAN; • "bioc@release", meaning Bioconductor, where release should be replaced with a valid Bioconductor release code or one of "release", "devel"; • "runiverse@universe", meaning R-universe, where universe should be replaced with a valid universe name (e.g. turtletopia);

- "github/user", meaning GitHub, where user should be replaced with a valid user or organization name (e.g. turtletopia);
- "gitlab/user", meaning GitLab, where user should be replaced with a valid user or group name (e.g. r-packages);
- "local#index", meaning a local library, where index should be replaced with an integer index of a library in `base::.libPaths()` or an all keyword;
- "core", meaning base R packages;
- a URL if none of the above is matched.

Value

A list named after queried packages, each element being a character vector of version codes.

See Also

Functions for all repositories: `wood_dependencies()`, `wood_packages()`

Functions that query package versions: `wood_bioc_version()`, `wood_core_version()`, `wood_cran_latest()`, `wood_cran_versions()`, `wood_github_latest()`, `wood_github_versions()`, `wood_gitlab_latest()`, `wood_gitlab_versions()`, `wood_local_versions()`, `wood_runiverse_version()`, `wood_url_version()`

Examples

```
wood_versions("woodendesc", c("local#all", "runiverse@turtletopia"))
# Multiple packages are also possible:
wood_versions(
  c("ggplot2", "Biostrings", "woodendesc"),
  repos = c("runiverse@turtletopia", "cran", "bioc@1.5")
)
# By default, only CRAN is queried:
wood_versions("versionsort")
```


Index

- * **bioc**
 - wood_bioc_dependencies, 4
 - wood_bioc_packages, 5
 - wood_bioc_releases, 6
 - wood_bioc_version, 7
 - * **core**
 - wood_core_dependencies, 8
 - wood_core_packages, 9
 - wood_core_version, 9
 - * **cran**
 - wood_cran_dependencies, 10
 - wood_cran_latest, 11
 - wood_cran_packages, 12
 - wood_cran_versions, 12
 - * **dependencies**
 - wood_bioc_dependencies, 4
 - wood_core_dependencies, 8
 - wood_cran_dependencies, 10
 - wood_dependencies, 13
 - wood_github_dependencies, 14
 - wood_gitlab_dependencies, 18
 - wood_local_dependencies, 23
 - wood_runiverse_dependencies, 26
 - wood_url_dependencies, 29
 - * **github**
 - wood_github_dependencies, 14
 - wood_github_latest, 15
 - wood_github_packages, 16
 - wood_github_tags, 17
 - wood_github_versions, 18
 - * **gitlab**
 - wood_gitlab_dependencies, 18
 - wood_gitlab_latest, 19
 - wood_gitlab_packages, 20
 - wood_gitlab_tags, 21
 - wood_gitlab_versions, 22
 - * **local**
 - wood_local_dependencies, 23
 - wood_local_packages, 24
 - wood_local_versions, 24
 - * **packages**
 - wood_bioc_packages, 5
 - wood_core_packages, 9
 - wood_cran_packages, 12
 - wood_github_packages, 16
 - wood_gitlab_packages, 20
 - wood_local_packages, 24
 - wood_packages, 25
 - wood_runiverse_packages, 27
 - wood_url_packages, 30
 - * **runiverse**
 - wood_runiverse_dependencies, 26
 - wood_runiverse_packages, 27
 - wood_runiverse_version, 28
 - * **url**
 - wood_url_dependencies, 29
 - wood_url_packages, 30
 - wood_url_version, 30
 - * **versions**
 - wood_bioc_version, 7
 - wood_core_version, 9
 - wood_cran_latest, 11
 - wood_cran_versions, 12
 - wood_github_latest, 15
 - wood_github_versions, 18
 - wood_gitlab_latest, 19
 - wood_gitlab_versions, 22
 - wood_local_versions, 24
 - wood_runiverse_version, 28
 - wood_url_version, 30
 - wood_versions, 31
 - * **wood**
 - wood_dependencies, 13
 - wood_packages, 25
 - wood_versions, 31
- base::.libPaths(), 13, 26, 32
base::getRversion(), 10
base::unlink(), 8

- filter_dependencies, 3
- squash, 4
- tools::package_dependencies(), 3
- wood_bioc_dependencies, 4, 6–8, 11, 14, 15, 19, 23, 27, 29
- wood_bioc_packages, 5, 5, 6, 7, 9, 12, 16, 21, 24, 26, 27, 30
- wood_bioc_releases, 5, 6, 6, 7
- wood_bioc_version, 5, 6, 7, 10, 11, 13, 16, 18, 20, 22, 25, 28, 31, 32
- wood_clear_cache, 8
- wood_core_dependencies, 5, 8, 9–11, 14, 15, 19, 23, 27, 29
- wood_core_packages, 6, 8, 9, 10, 12, 16, 21, 24, 26, 27, 30
- wood_core_version, 7–9, 9, 11, 13, 16, 18, 20, 22, 25, 28, 31, 32
- wood_cran_dependencies, 5, 8, 10, 11–15, 19, 23, 27, 29
- wood_cran_latest, 7, 10, 11, 11, 12, 13, 16, 18, 20, 22, 25, 28, 31, 32
- wood_cran_packages, 6, 9, 11, 12, 13, 16, 21, 24, 26, 27, 30
- wood_cran_versions, 7, 10–12, 12, 16, 18, 20, 22, 25, 28, 31, 32
- wood_dependencies, 5, 8, 11, 13, 15, 19, 23, 26, 27, 29, 32
- wood_github_dependencies, 5, 8, 11, 14, 14, 16–19, 23, 27, 29
- wood_github_latest, 7, 10, 11, 13, 15, 15, 16–18, 20, 22, 25, 28, 31, 32
- wood_github_latest(), 18
- wood_github_packages, 6, 9, 12, 15, 16, 16, 17, 18, 21, 24, 26, 27, 30
- wood_github_tags, 15, 16, 17, 18
- wood_github_versions, 7, 10, 11, 13, 15–17, 18, 20, 22, 25, 28, 31, 32
- wood_github_versions(), 15
- wood_gitlab_dependencies, 5, 8, 11, 14, 15, 18, 20–23, 27, 29
- wood_gitlab_latest, 7, 10, 11, 13, 16, 18, 19, 19, 21, 22, 25, 28, 31, 32
- wood_gitlab_latest(), 22
- wood_gitlab_packages, 6, 9, 12, 16, 19, 20, 20, 22, 24, 26, 27, 30
- wood_gitlab_tags, 19–21, 21, 22
- wood_gitlab_versions, 7, 10, 11, 13, 16, 18–22, 22, 25, 28, 31, 32
- wood_gitlab_versions(), 19
- wood_local_dependencies, 5, 8, 11, 14, 15, 19, 23, 24, 25, 27, 29
- wood_local_packages, 6, 9, 12, 16, 21, 23, 24, 25–27, 30
- wood_local_versions, 7, 10, 11, 13, 16, 18, 20, 22–24, 24, 28, 31, 32
- wood_packages, 6, 9, 12, 14, 16, 21, 24, 25, 27, 30, 32
- wood_runiverse_dependencies, 5, 8, 11, 14, 15, 19, 23, 26, 27–29
- wood_runiverse_packages, 6, 9, 12, 16, 21, 24, 26, 27, 27, 28, 30
- wood_runiverse_version, 7, 10, 11, 13, 16, 18, 20, 22, 25, 27, 28, 31, 32
- wood_url_dependencies, 5, 8, 11, 14, 15, 19, 23, 27, 29, 30, 31
- wood_url_packages, 6, 9, 12, 16, 21, 24, 26, 27, 29, 30, 31
- wood_url_version, 7, 10, 11, 13, 16, 18, 20, 22, 25, 28–30, 30, 32
- wood_versions, 7, 10, 11, 13, 14, 16, 18, 20, 22, 25, 26, 28, 31, 31