Package: abitools (via r-universe)

November 11, 2026

Type Package

Title Detect When Installed R Packages Need to be Rebuilt

Version 0.1.0

Description An installed R package may contain state copied from its dependencies at build time: compiled calls into the C API of a 'LinkingTo' package, S4 class definitions that extend classes from other packages, or copies of functions. When such a dependency changes, the binary package must be rebuilt. This package finds these critical dependencies and computes fingerprints of the parts that dependents copy, such that package repositories can rebuild exactly the packages that are affected by a change, rather than relying on version numbers.

License MIT + file LICENSE

URL https://github.com/r-universe-org/abitools

BugReports https://github.com/r-universe-org/abitools/issues

Encoding UTF-8

SystemRequirements libclang (deb: libclang-dev, rpm: clang-devel, brew: llvm)

Depends R (>= 4.5.0)

Imports methods, tools

Suggests jsonlite, knitr, rmarkdown, testthat (>= 3.0.0)

VignetteBuilder knitr

Config/testthat/edition 3

Roxygen list(markdown = TRUE)

Config/roxygen2/version 8.0.0

Config/pak/sysreqs libclang-dev

Repository https://ropensci.r-universe.dev

Date/Publication 2026-11-10 23:36:34 UTC

RemoteUrl https://github.com/r-universe-org/abitools

RemoteRef HEAD

RemoteSha ce28cb2a84e90477d0790634c7a350b8cb810127

Contents

abi_check 2
abi_deps 3
abi_list 4

Index **6**

`abi_check` *Check if a package needs to be rebuilt*

Description

`abi_check()` compares the ABI imports of a package, as stored when it was built (see `abi_list()`), with the current ABI exports of its dependencies. The package needs to be rebuilt if the result has any rows.

Usage

```
abi_check(x, exports = NULL, lib.loc = NULL, flags = NULL)

stale_classes(pkg, lib.loc = NULL)
```

Arguments

`x`	list as returned by `abi_list()`, possibly read from JSON. Only the imports are used.
`exports`	data frame with the current exports of the dependencies, e.g. the combined `exports` of `abi_list()` for each of them. If `NULL`, they are computed from the installed packages.
`lib.loc`	library to search, defaults to `.libPaths()`
`flags`	extra arguments for the compiler when parsing headers, such as the include directories of system libraries that the headers need. Use `install_flags()` to take these from the log of `R CMD INSTALL`.
`pkg`	name of an installed package

Details

`stale_classes()` needs no stored imports: it compares the S4 class definitions in an installed package with the current definitions of their superclasses in other packages. This finds packages in a local library that must be reinstalled after a dependency was updated.

Value

a data frame with the dependency and reason for each change

abi_deps *Critical dependencies of an installed package*

Description

Finds the dependencies of which the installed package holds a copy, or whose binary interface it was compiled against. When one of these changes, the package may need to be rebuilt.

Usage

```
abi_deps(pkg, lib.loc = NULL, ignore = "R6")
```

Arguments

pkg	name of an installed package
lib.loc	library to search, defaults to `.libPaths()`
ignore	namespaces whose code is copied by design, such as the generator functions of R6 classes.

Details

The `type` column in the result is one of:

- `C API`: a `LinkingTo` dependency that registers C functions with `R_RegisterCCallable()`, which the shared library of the package calls at runtime via `R_GetCCallable()`. Changes in its headers (struct layouts, function signatures) break the binary.
- `linked library`: a `LinkingTo` dependency that ships a shared library in its `lib` directory (such as TBB in RcppParallel), which the shared library of the package links to.
- `headers`: a `LinkingTo` dependency that is not called at runtime, for example a header-only or static library. The binary keeps working when it changes, so this is informational only.
- `S4 class`: an S4 or reference class in the package extends this class from the dependency. The definition of the subclass, including the inherited slots, was copied at install time.
- `S4 generic`: the package defines methods for this generic of the dependency. The methods are stored in a table for the generic of that package, and are lost if the generic moves to another package.
- `copied code`: objects in the package (such as `f <- dep:::g`) contain functions or environments of the dependency.

The package is inspected without loading it, but namespaces referenced by its lazyload database are loaded. Call this in the library that was used to build the package.

Value

a data frame with columns `package`, `type` and `detail`

Examples

```
abi_deps("stats4")
```

abi_list *ABI imports and exports of a package*

Description

Lists the parts of the binary interface of an installed package that matter for deciding when packages need to be rebuilt. Run this right after a package was built, in the library that it was built with.

Usage

```
abi_list(pkg, lib.loc = NULL, flags = NULL)

install_flags(log)
```

Arguments

pkg	name of an installed package
lib.loc	library to search, defaults to `.libPaths()`
flags	extra arguments for the compiler when parsing headers, such as the include directories of system libraries that the headers need. Use `install_flags()` to take these from the log of `R CMD INSTALL`.
log	path to the output of `R CMD INSTALL` for the package (e.g. `00install.out`). The `-I` flags of its compiler commands that point to existing directories are returned.

Details

The **exports** are the parts of the package that dependents may copy at build time: a hash of each S4 class definition and S4 generic, a hash of the C headers in its `include` directory, and a hash of each C function that it registers with `R_RegisterCCallable()`. The hash of a C function covers its declaration in the headers, and the definitions of the types that it uses. Unlike the version, these hashes only change when something changes that affects dependents.

The **imports** are the exports of its dependencies that the package copied when it was built: the classes that it extends, the generics that it defines methods for, the C functions of `C API` dependencies that it calls, and the headers of `linked library` dependencies (see `abi_deps()`). Other classes and functions of the dependency are not included, so a change in those does not trigger a rebuild.

A package needs to be rebuilt when one of its imports no longer matches the current export of the dependency, see `abi_check()`. Code that was copied from a dependency (such as `f <- dep:::g`) has no hash, and is left to periodic rebuilds.

The data frames are flat, so that `jsonlite::toJSON()` gives lists of entries that are easy to store and query in a database.

Value

a list with data frames `exports` and `imports`, both with columns `key` and `hash`. The key is `<package>::<class>` for an S4 class, `<package>::generic:<name>` for an S4 generic, `<package>::<function>()` for a C function, or `<package>/include` for the C headers. Package names cannot contain `:` or `/`, so the package is everything before the first `::` or `/`.

Examples

```
abi_list("stats4")
```

Index

.libPaths(), *2–4*

abi_check, 2
abi_check(), *4*
abi_deps, 3
abi_deps(), *4*
abi_list, 4
abi_list(), *2*

install_flags *(*abi_list*)*, 4
install_flags(), *2*, *4*

jsonlite::toJSON(), *4*

stale_classes *(*abi_check*)*, 2