

# Package ‘bigbang’

August 19, 2026

**Type** Package

**Title** Build 'Tidyverse'-Style Meta-Packages from Local Package Files

**Version** 0.4.0

**Description** Turns a curated set of package archives (.tar.gz, .zip) into one meta-package in the style of the 'tidyverse', so that a group of interdependent packages can be distributed and installed as a single unit. The generated meta-package records the exact archive versions it was built from and installs its components in dependency order, so that whoever receives it does not have to work out which package to install first.

The component archives are copied into the generated meta-package, so it is the only artifact that has to be distributed and no directory has to be agreed on between machines. Resolves dependencies by building a graph with topological ordering and cycle detection, classifies them as local or external, and detects implicit dependencies by scanning source code. Installation needs no repository access unless a component depends on a package that only exists in one, which suits teams working behind institutional firewalls. Generates the complete meta-package scaffold, including installation helpers, vignettes and documentation.

**License** GPL (>= 3)

**URL** <https://sebollin.github.io/bigbang/>,  
<https://github.com/sebollin/bigbang>

**BugReports** <https://github.com/sebollin/bigbang/issues>

**Encoding** UTF-8

**Language** en

**VignetteBuilder** knitr

**Depends** R (>= 3.6.0)

**Imports** brio, glue, tools, utils, whisker

**Suggests** devtools, knitr, pkgbuild, rmarkdown, testthat (>= 3.0.0),  
withr

**Config/testthat/edition** 3

**Config/roxygen2/version** 8.0.0  
**NeedsCompilation** no  
**Author** Sebastián Lucas [aut, cre] (ORCID:  
    <<https://orcid.org/0009-0009-9068-0276>>),  
    Richard Detomasi [ctb] (ORCID: <<https://orcid.org/0000-0002-6725-0261>>,  
    Suggested the initial idea of a metapackage builder)  
**Maintainer** Sebastián Lucas <sebalucas@gmail.com>  
**Repository** CRAN  
**Date/Publication** 2026-08-19 18:00:02 UTC

Contents

|                                        |           |
|----------------------------------------|-----------|
| create_metapackage . . . . .           | 2         |
| diagnose_dependencies . . . . .        | 7         |
| install_local_pkg . . . . .            | 8         |
| print.bigbang_artifact_scan . . . . .  | 10        |
| print.bigbang_install_result . . . . . | 10        |
| print.bigbang_result . . . . .         | 11        |
| scan_bigbang_artifact . . . . .        | 11        |
| <b>Index</b>                           | <b>13</b> |

---

|                    |                                   |
|--------------------|-----------------------------------|
| create_metapackage | <i>Build a local meta-package</i> |
|--------------------|-----------------------------------|

---

Description

Creates the full structure and files of a meta-package that installs, manages and loads a set of locally stored R packages, resolving the dependencies between them with a graph-based (topologically ordered) approach.

Usage

```
create_metapackage(  
  name,  
  packages,  
  pkg_dir = NULL,  
  ext = ".tar.gz",  
  version = "0.1.0",  
  dest_dir,  
  reexport = FALSE,  
  document = TRUE,  
  verbose = getOption("bigbang.verbose", interactive()),  
  authors =  
    "person('First', 'Last', email = 'first.last@example.com', role = c('aut', 'cre'))",  
  description = "Local Package Metapackage",
```

```

license = "MIT + file LICENSE",
additional_deps = NULL,
ignore_deps = NULL,
import_deps = c("data.table", "dplyr", "ggplot2", "readr", "tibble", "tidyr", "xts",
  "zoo"),
force_deps = NULL,
debug = FALSE,
workflow = NULL,
include_archives = TRUE,
tolerate = character(),
dry_run = FALSE,
on_component_error = c("abort", "skip"),
update = FALSE,
install_upgrade = c("newer", "always", "never")
)

```

### Arguments

|          |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                |
|----------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| name     | Character. Name of the meta-package to create (must not contain underscores _).                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                |
| packages | Character vector. Archive paths or stems of the local packages to include. An existing file is always used as a path; otherwise the element is resolved as a stem in pkg_dir, e.g. "myPackage_1.0.0". A bare package name such as "myPackage" resolves when exactly one archive in those directories declares that Package identity. Zero matches use the usual unresolved-archive error; multiple matches are an ambiguity error. Supported archives that cannot be read during this identity search are excluded with a warning that names the archive. Existing paths may come from different directories. A single existing text file without a recognized archive extension is treated as a manifest, with one component per line; relative paths in that file are resolved relative to the manifest directory, absolute paths and ~ paths are used as written, and bare archive file-names may also be found in pkg_dir. |
| pkg_dir  | Character. Optional directory or directories containing local archives used to resolve stems and bare package names. It is not needed when every packages element is an existing archive path.                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                 |
| ext      | Character. Fallback archive extension for stems. Defaults to ".tar.gz"; each existing archive path keeps its own extension.                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                    |
| version  | Character. Version of the meta-package. Defaults to "0.1.0".                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                   |
| dest_dir | Character. Required destination directory. The function writes the generated meta-package exclusively inside this directory; there is no default path. Use tempdir() for disposable output.                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                    |
| reexport | Logical flag retained in its original position for positional-call compatibility. The default FALSE attaches installed components as usual. With TRUE, explicit exports read from each component's NAMESPACE are exposed through read-only active bindings. Components are never added to Imports or Depends, so the generated package still installs offline without them. Before installation, reading a binding returns a placeholder function whose clear missing-component                                                                                                                                                                                                                                                                                                                                                                                                                                                |

error appears only when that function is called. For non-function exports, access therefore returns the placeholder instead of the object until the component is installed. The same binding then works without reloading the metapackage. Only explicit `export()` directives become bindings. S4 classes and methods remain available by loading their component package. Non-syntactic explicit export names are quoted in the generated `NAMESPACE`. An object restored with `readRDS()` does not load a component by itself, so base R cannot dispatch that component's S3 method until it is loaded.

|                  |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                |
|------------------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| document         | Logical. If TRUE, runs <code>devtools::document()</code> automatically. Defaults to TRUE. The planned <code>man/&lt;name&gt;_*.Rd</code> and internal-helper Rd filenames are reserved for generated documentation; custom Rd files should use different names. A successful documentation run may adopt a reserved filename into the generation manifest, after which a later update with <code>document = FALSE</code> removes it as generated output.                                                                                                                                                                                                                                                                                                       |
| verbose          | Logical. If TRUE, shows verbose messages. The default follows <code>getOption("bigbang.verbose", interactive())</code> .                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                       |
| authors          | Character. Content for the <code>Authors@R</code> field of <code>DESCRIPTION</code> .                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                          |
| description      | Character. Description of the meta-package.                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                    |
| license          | Character. License of the meta-package.                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                        |
| additional_deps  | Character vector. Extra dependencies to add on top of the ones declared by components. Source-code guesses are diagnostic by default; use this argument when a guessed dependency should bind in the generated package.                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                        |
| ignore_deps      | Character vector. Dependencies to ignore even if detected.                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                     |
| import_deps      | Character vector. Packages that should go in the <code>Imports</code> field of <code>DESCRIPTION</code> rather than <code>Depends</code> . Imports are not attached when the user calls <code>library()</code> on the meta-package, but remain available via <code>::</code> (e.g. <code>dplyr::filter()</code> ), reducing name clashes in the user's workspace.                                                                                                                                                                                                                                                                                                                                                                                              |
| force_deps       | Character vector. Exact package names to use as dependencies, bypassing automatic detection. If supplied, only these are used as the meta-package's implicit dependencies.                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                     |
| debug            | Logical. If TRUE, emits detailed debugging messages. Defaults to FALSE.                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                        |
| workflow         | Optional named character vector mapping ordered stage labels to component package names. When supplied, every component must appear once and a pipeline vignette skeleton is generated.                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                        |
| include_archives | Logical. If TRUE, the default, the component archives are copied into <code>inst/archives/</code> of the generated meta-package, so that the meta-package is the only artifact that has to be distributed and <code>&lt;meta&gt;_install()</code> works with no arguments, without any path being agreed on beforehand. Components still install only where they can: a Windows binary archive is refused on other platforms. Shipping the archives also means redistributing them, so their licenses have to allow it, and it makes the generated tarball as large as its components: CRAN prefers source tarballs under 10 MB and does not accept binary executables in them, which matters only if a generated meta-package is ever submitted there. Set it |

|                                 |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                 |
|---------------------------------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
|                                 | to FALSE when the archives stay in a shared location that recipients can reach; then <code>&lt;meta&gt;_install()</code> requires an explicit <code>pkg_dir</code> .                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                            |
| <code>tolerate</code>           | Character vector of explicitly named validation relaxations. Use <code>"filename_mismatch"</code> to silence filename-versus-DESCRIPTION mismatch warnings, or <code>"unincluded_local_dep"</code> to turn an available-but-unincluded local dependency error into a warning. With the latter relaxation, the generated metapackage does not ship that dependency: the recipient must provide it through <code>pkg_dir</code> or a repository with <code>cran_deps = "install"</code> . Unknown names are errors. Each applied relaxation is recorded in the returned <code>tolerated</code> table.                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                             |
| <code>dry_run</code>            | Logical. If TRUE, resolves and validates components and returns the planned generation without creating <code>dest_dir</code> or writing a project.                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                             |
| <code>on_component_error</code> | Character policy for component-level failures: <code>"abort"</code> (default) stops generation, while <code>"skip"</code> omits the failed component and transitively omits components that depend on it. When a failed archive still exposes its DESCRIPTION, propagation uses its declared Package; otherwise the filename-derived name is used and the limitation is reported. If that fallback name differs from Package, a dependent may fail on the recipient. During an update, omitted inputs never authorize deletion of a previously shipped archive. When the old component cannot be identified unambiguously, archive reconciliation is deferred until a clean update rather than risking the only surviving copy.                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                 |
| <code>update</code>             | Logical. If TRUE, update a previously generated project only when its bigbang manifest is present and all generated files are unchanged. Files outside that manifest are never touched. Updates are refused when the generated project root, a manifest file, or any path component inside the project is a symbolic link, so writes cannot escape the project tree. Generated files no longer in the plan are reported in <code>removed_files</code> . Removing a component also removes its shipped archive, which may be the last available copy. The same field includes partial documentation outputs created and cleaned up after a failed documentation run. A dry run cannot predict those failure-dependent cleanups and reports only planned removals. Before changing the project, an update backs up every generated file and its manifest. A failed update restores that state so the same update can be retried. Documentation files requested by <code>document = TRUE</code> can always be regenerated: they can be restored after documentation was disabled, and an update that cannot regenerate them retains the previously tracked Rd files. See <code>document</code> for the reserved generated-documentation filenames. |
| <code>install_upgrade</code>    | Character default upgrade policy emitted in the generated installer function: <code>"newer"</code> , <code>"always"</code> , or <code>"never"</code> . This controls whether a generated installer keeps newer installed versions, reinstalls every component, or skips archive inspection.                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                     |

## Details

The function performs the following steps:

1. Creates the basic R package structure (R, man, vignettes, etc.).

2. Detects dependencies between packages, both explicit (from DESCRIPTION) and possible implicit uses (found by scanning executable source tokens). The latter are reported for diagnosis and are not hard dependencies unless explicitly supplied through `additional_deps` or `force_deps`.
3. Generates DESCRIPTION and NAMESPACE with the appropriate dependencies.
4. Creates a basic vignette documenting the meta-package.
5. Generates R files with functions to install and load the component packages:
  - `<name>_install()`: installs the component packages from the local archives.
  - `<name>_attach()`: attaches the components that are already installed.
  - `<name>_detach()`: detaches all the meta-package's components.
  - `<name>_packages()`: lists the included packages.

Installation is **explicit**: calling `library(<meta>)` attaches the components that are already installed and reports which ones are missing, but does not install anything or delete any files. To install the components from the local archives, the user calls `<meta>_install()`. Installation resolves dependencies with a graph-based topological ordering that also detects circular dependencies.

Generation validates every supplied component and its dependency graph eagerly before writing the metapackage. This hard validation protects an artifact that will be distributed to another machine. The installer is more tolerant: when an already installed component does not need to be changed, it can retain that installation without reading an archive that will not be used.

## Value

Invisibly, a `bigbang_result` containing the generated path, component archives, dependency classification, applied tolerations, files removed by the call, and documentation status.

## Validation strictness

During generation, validations that protect the recipient cannot be disabled: malformed or unsafe archives, invalid component metadata, duplicate components, cycles, and unsatisfied local version constraints remain hard errors. Checks about project tidiness can be relaxed individually through `tolerate`; there is no switch that disables validation as a whole. `bigbang` does not run R CMD check on component packages, so component warnings and notes do not prevent generation. Component source directories are built in a temporary directory with the optional `pkgbuild` package; passing an already built archive avoids that optional dependency.

## Component installation

The generated meta-package installs component packages only when the user explicitly calls `<meta>_install()`. Loading it with `library()` never installs packages. By default, the generated installer does not access a repository.

With `include_archives = TRUE`, the default, the component archives travel inside the generated meta-package and `pkg_dir` defaults to `system.file("archives", package = "<meta>")`. That default is resolved when the installer is called, so it points at the library of whoever installed the meta-package: recipients need nothing beyond the meta-package itself, and no path has to be agreed on between machines. Network access is needed only when a component depends on a package that must come from a repository, which happens exclusively under `cran_deps = "install"`.

Loading the generated meta-package attaches installed components, so their exported functions can be called directly or through `component::function()`. With `reexport = TRUE`, explicit component exports are instead exposed through read-only active bindings in the meta-package namespace. This does not add components to Imports or Depends: loading remains possible without them, and a binding resolves the component on every access. Only explicit `export()` directives are rebound; S4 classes and methods are used through the loaded component namespace. An object restored with `readRDS()` cannot load a component by itself, so base R cannot dispatch that component's S3 method until the component has been loaded.

## Requirements

- Each component must be an existing archive path or a stem resolvable in one of the optional `pkg_dir` directories; `ext` is only a fallback for stems.
- Files in the supplied archive directories that cannot be read are excluded from the inventory with a warning. A requested component still fails validation, while an unreadable file matching a declared dependency is reported as an unavailable local archive.
- Automatic documentation (`document = TRUE`) requires the `devtools` package.

## Examples

```
archives <- system.file("extdata", package = "bigbang")
destination <- tempfile("bigbang-example-")
dir.create(destination)

result <- create_metapackage(
  name = "toyverse",
  packages = "toycomponent_0.1.0",
  pkg_dir = archives,
  dest_dir = destination,
  document = FALSE,
  verbose = FALSE,
  import_deps = character(),
  force_deps = character()
)
list.files(result$path)

unlink(destination, recursive = TRUE)
```

---

`diagnose_dependencies` *Diagnose implicit dependencies of local packages*

---

## Description

Scans local packages for references to the recommended packages 'Matrix' and 'class', which can cause R CMD check failures when they are used implicitly but not declared as dependencies.

## Usage

```
diagnose_dependencies(packages, pkg_dir = NULL, ext = ".tar.gz")
```

**Arguments**

|                       |                                                                                      |
|-----------------------|--------------------------------------------------------------------------------------|
| <code>packages</code> | Character vector. Archive paths or stems to examine, e.g. "conexiones_0.8.3".        |
| <code>pkg_dir</code>  | Character. Directory or directories containing local archives (.tar.gz, .zip, etc.). |
| <code>ext</code>      | Character. Archive extension. Defaults to ".tar.gz".                                 |

**Details**

Extracts and scans the R source of each package for patterns that suggest implicit use of 'Matrix' or 'class'. Useful for debugging R CMD check errors such as "there is no package called 'Matrix'" even when the package does not appear to use it directly.

**Value**

A named list with one entry per local package, each a list with two elements:

**matrix\_refs** Character vector of references to 'Matrix', with file and line.

**class\_refs** Character vector of references to 'class', with file and line.

**Examples**

```
archives <- system.file("extdata", package = "bigbang")
res <- diagnose_dependencies(
  packages = "toycomponent_0.1.0",
  pkg_dir = archives
)
res[["toycomponent_0.1.0"]]
lapply(res, function(x) x$matrix_refs)
```

---

|                                |                                                               |
|--------------------------------|---------------------------------------------------------------|
| <code>install_local_pkg</code> | <i>Install a local package together with its dependencies</i> |
|--------------------------------|---------------------------------------------------------------|

---

**Description**

Installs a package from a local archive. Dependencies available as local archives are installed recursively; missing non-local dependencies follow the explicit `cran_deps` policy. ZIP archives containing `Meta/package.rds` are treated as Windows binaries, while other ZIP archives are unpacked and installed as source packages.

**Usage**

```
install_local_pkg(
  package,
  pkg_dir = NULL,
  ext = ".tar.gz",
  repos = getOption("repos"),
  cran_deps = c("skip", "error", "install"),
```



```

    verbose = getOption("bigbang.verbose", interactive()),
    force = FALSE,
    upgrade = c("newer", "always", "never"),
    lib = .libPaths()[[1L]]
  )

```

## Arguments

|           |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                               |
|-----------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| package   | Character. An existing archive path, or a package stem such as "uspr_0.8.5" to resolve in pkg_dir. An existing file is always treated as a path; only a non-existing element is resolved as a stem. A bare package name such as "uspr" resolves when exactly one archive in those directories declares that Package identity; multiple matches are an ambiguity error. Supported archives that cannot be read during this identity search are excluded with a warning that names the archive. |
| pkg_dir   | Character. Optional directory or directories containing local archives. It is not needed when package is an existing path.                                                                                                                                                                                                                                                                                                                                                                    |
| ext       | Character. Fallback archive extension for stems; existing paths keep their own extension.                                                                                                                                                                                                                                                                                                                                                                                                     |
| repos     | Character. Repositories used only when cran_deps = "install".                                                                                                                                                                                                                                                                                                                                                                                                                                 |
| cran_deps | Character. Policy for missing non-local dependencies: "skip" (the default) never accesses the network, "error" fails without accessing it, and "install" attempts installation from repos.                                                                                                                                                                                                                                                                                                    |
| verbose   | Logical. Whether to emit progress and summary messages. The default follows getOption("bigbang.verbose", interactive()).                                                                                                                                                                                                                                                                                                                                                                      |
| force     | Logical. Reinstall every local archive. This is a convenience alias for upgrade = "always".                                                                                                                                                                                                                                                                                                                                                                                                   |
| upgrade   | Character. Installed-version policy: "newer" installs only when the archive is newer, "always" reinstalls, and "never" keeps any installed version. Explicitly combining force = TRUE with a value other than "always" is an error.                                                                                                                                                                                                                                                           |
| lib       | Character. Library in which the local component must be installed and verified when supplied explicitly. When omitted, the legacy lookup considers all of .libPaths() and installs into its first entry. Non-local dependencies may be available either in lib or anywhere on .libPaths().                                                                                                                                                                                                    |

## Value

Invisibly, a list describing installed, unchanged, failed, and skipped packages. Components that an upgrade policy left in place are reported in unchanged, not in installed.

## Installation

This function installs packages into lib, which defaults to the user's active R library. When lib is supplied explicitly, a component found only in another library is still installed into lib; non-local dependencies are resolved from lib plus .libPaths(). When lib is omitted, an installed component found anywhere on .libPaths() retains the pre-0.3.0 behavior. Installation occurs only when the user calls the function; loading bigbang never installs packages. With the default

`cran_deps = "skip"`, it does not access the network. An installed package can be kept without reading its archive when `upgrade = "never"`. Under the default policy, `bigbang` reads only the archive DESCRIPTION first; if that metadata cannot be verified for an already installed package, the installed package is kept and the reason is reported. With `upgrade = "never"`, the shortcut takes the component identity from the archive filename because the archive is not read. Use `upgrade = "newer"` when the declared Package field must be checked against the installed package.

### See Also

[`create\_metapackage\(\)`](#) for generating a meta-package with an explicit component installer.

---

```
print.bigbang_artifact_scan
```

*Print an artifact scan*

---

### Description

Print an artifact scan

### Usage

```
## S3 method for class 'bigbang_artifact_scan'
print(x, ...)
```

### Arguments

|                  |                                              |
|------------------|----------------------------------------------|
| <code>x</code>   | A <code>bigbang_artifact_scan</code> object. |
| <code>...</code> | Unused.                                      |

### Value

`x`, invisibly.

---

```
print.bigbang_install_result
```

*Print a local package installation result*

---

### Description

Print a local package installation result

### Usage

```
## S3 method for class 'bigbang_install_result'
print(x, ...)
```

### Arguments

|     |                                  |
|-----|----------------------------------|
| x   | A bigbang_install_result object. |
| ... | Unused.                          |

### Value

x, invisibly.

---

`print.bigbang_result`    *Print a metapackage generation result*

---

### Description

Print a metapackage generation result

### Usage

```
## S3 method for class 'bigbang_result'
print(x, ...)
```

### Arguments

|     |                                                                            |
|-----|----------------------------------------------------------------------------|
| x   | A bigbang_result object returned by <a href="#">create_metapackage()</a> . |
| ... | Unused.                                                                    |

### Value

x, invisibly.

---

`scan_bigbang_artifact`    *Scan a generated metapackage for historical deletion signatures*

---

### Description

Inspects a generated metapackage source directory, source archive, or installed package without loading it. The scanner looks for the historical V1, V2, V3, and V7 deletion signatures from the pre-release security investigation. Provenance is read from both the current generator fields and the legacy pre-rename fields so development artifacts remain classifiable.

### Usage

```
scan_bigbang_artifact(path, dry_run = TRUE)
```

**Arguments**

|         |                                                                                                        |
|---------|--------------------------------------------------------------------------------------------------------|
| path    | Character scalar. Source directory, .tar.gz/.tar/.zip source archive, or installed package directory.  |
| dry_run | Logical. Must be TRUE, the default. Automatic mutation or remediation is deliberately not implemented. |

**Details**

Installed packages are inspected through R's internal lazy-load database API. That code is isolated in `.scan_installed_lazydb()` and has been exercised with R 4.6.1. Because this is an internal R format, callers should re-run the scanner tests when adopting a new R minor release.

**Value**

A list with the artifact type, vulnerability flag, detected signatures, evidence locations, provenance fields, and R version used for the scan.

**Examples**

```
archives <- system.file("extdata", package = "bigbang")
destination <- tempfile("bigbang-scan-example-")
dir.create(destination)

result <- create_metapackage(
  name = "toyverse",
  packages = "toycomponent_0.1.0",
  pkg_dir = archives,
  dest_dir = destination,
  document = FALSE,
  verbose = FALSE,
  import_deps = character(),
  force_deps = character()
)
scan_bigbang_artifact(result$path)

unlink(destination, recursive = TRUE)
```

# Index

`create_metapackage`, [2](#)  
`create_metapackage()`, [10](#), [11](#)  
`diagnose_dependencies`, [7](#)  
`install_local_pkg`, [8](#)  
`print.bigbang_artifact_scan`, [10](#)  
`print.bigbang_install_result`, [10](#)  
`print.bigbang_result`, [11](#)  
`scan_bigbang_artifact`, [11](#)