# Package: deps (via r-universe)

October 23, 2026

**Type** Package

**Title** Dependency Management with 'roxygen'-Style Comments

**Version** 0.4.1

**Description** Manage your source code dependencies by decorating your existing R code with special, 'roxygen'-style comments.

**License** MIT + file LICENSE

**LazyLoad** yes

**Imports** renv, jsonlite, remotes

**RoxygenNote** 7.3.3

**Encoding** UTF-8

**Roxygen** list(markdown = TRUE)

**BugReports** https://github.com/analythium/deps/issues

**URL** https://hub.analythium.io/deps/, https://github.com/analythium/deps

**Language** en-US

**Config/pak/sysreqs** git

**Repository** https://psolymos.r-universe.dev

**Date/Publication** 2026-01-04 22:36:13 UTC

**RemoteUrl** https://github.com/analythium/deps

**RemoteRef** HEAD

**RemoteSha** e4ab1ac9e74727e9458c04006b7b729c0ef44e77

## Contents


create . . . . . 2
install . . . . . 3

**Index** **4**

`create` *Create a Dependencies JSON File*

## Description

Discover dependencies and write a `dependencies.json` file.

## Usage

```
create(
  dir = getwd(),
  file = "dependencies.json",
  output = dir,
  installed = c("base", "recommended"),
  overwrite = TRUE,
  ask = TRUE
)
```

## Arguments

| | |
|---|---|
| `dir` | Path to the directory where the files to be scanned are located. |
| `file` | The name of the file to be save, default is `"dependencies.json"`. |
| `output` | Path to the directory where JSON file should be written to. |
| `installed` | The `priority` argument for `installed.packages()` for packages to be excluded. |
| `overwrite` | Logical, should the `file` in the `output` directory be overwritten if exists? |
| `ask` | Logical, asking confirmation before writing the `dependencies.json` file. |

## Value

Invisibly returns the list of file names that were created. The side effect is a JSON (and possibly a text for system requirements) file written to the hard drive. The function fails when there are no R related files in `dir`.

## Examples

```
dir <- system.file("examples/01-basic", package = "deps")
out <- tempdir()
create(dir, output = out, ask = interactive())
cat(readLines(file.path(out, "dependencies.json")), sep = "\n")
unlink(file.path(out, "dependencies.json"))
```

`install` *Install Dependencies*

## Description

Install dependencies from an existing `dependencies.json` file or after discovering the dependencies.

## Usage

```
install(
  dir = getwd(),
  file = "dependencies.json",
  upgrade = "never",
  cleanup = TRUE,
  timeout = 300L,
  ask = TRUE,
  ...
)
```

## Arguments

| | |
|---|---|
| `dir` | Path to the directory where the JSON file should be written to. |
| `file` | The name of the file to be save, default is `"dependencies.json"`. If the file is not found in `dir`, `create()` is called. |
| `upgrade` | Should package dependencies be upgraded? Argument passed to remotes functions. |
| `cleanup` | Logical, clean up files created by `create()` when `file` does not exist. |
| `timeout` | Integer, timeout for file downloads (default 60 seconds can be short). |
| `ask` | Logical, asking confirmation before writing the `dependencies.json` file. |
| `...` | Other argument passed to remotes functions. |

## Value

Returns `NULL` invisibly. The side effect is the dependencies installed.

## Examples

```
dir <- system.file("examples/01-basic", package = "deps")
out <- tempdir()
create(dir, output = out, ask = interactive())
cat(readLines(file.path(out, "dependencies.json")), sep = "\n")
## Not run:
install(out)

## End(Not run)
unlink(file.path(out, "dependencies.json"))
```

# Index

create, 2

install, 3