

# Package ‘updateme’

May 8, 2026

**Type** Package

**Title** Informative Messages About Outdated Packages

**Version** 0.1.0

**Description** When a package is loaded, the source repository is checked for new versions and a message is shown in the console indicating whether the package is out of date.

**License** MIT + file LICENSE

**Encoding** UTF-8

**URL** <https://github.com/wurli/updateme>,  
<https://wurli.github.io/updateme/>

**BugReports** <https://github.com/wurli/updateme/issues>

**Imports** cachem, cli (>= 3.6.0), curl, memoise, rlang, utils

**RoxygenNote** 7.2.3

**Suggests** BiocManager, gitcreds, knitr, rmarkdown, testthat (>= 3.0.0),  
withr

**Config/testthat/edition** 3

**NeedsCompilation** no

**Author** Jacob Scott [aut, cre]

**Maintainer** Jacob Scott <jscott2718@gmail.com>

**Repository** CRAN

**Date/Publication** 2024-03-14 20:10:02 UTC

## Contents

updateme_on . . . . .	2
updateme_sources_set . . . . .	2

<b>Index</b>	<b>5</b>
--------------	----------

---

update_me_on	<i>Temporarily turn update_me off for all packages</i>
--------------	--

---

### Description

These functions may be useful if you find the slight delay caused by update\_me looking up new package versions to be slowing down your workflow.

### Usage

```
update_me_on()
```

```
update_me_off()
```

### Value

The old update\_me.on option, as returned by options()

### See Also

[update\\_me\\_sources\\_set\(\)](#) to turn update\_me off for individual packages

---

update_me_sources_set	<i>Configure update_me lookup of new package versions</i>
-----------------------	---

---

### Description

This function is a helper for setting the "update\_me.sources" global option. It provides a user-friendly interface and validation of the options you set.

### Usage

```
update_me_sources_set(...)
```

### Arguments

- ...
- Named or unnamed arguments. Values should be either:
- One of names(getOption("repo")): latest versions will be taken from this source, if available
  - "bioc": new versions will be looked for on Bioconductor
  - "github"/"gitlab": new versions will be looked for on GitHub/GitLab, if a repo can be identified using the package DESCRIPTION
  - A URL pointing to a GitHub/GitLab repo, e.g. "https://github.com/wurli/update\_me": the latest version *for this particular package* will be taken from this project

- NA: update\_me will not attempt to query new versions. Note that NA inputs must always be named (i.e. you must specify this 'per package')
- NULL: return to the default behaviour

If arguments are named, names should indicate package which the option should apply to. If unnamed, the option will apply to all packages. See examples for more information.

## Value

The result of setting options(update\_me.sources = <new\_options>)

## Private Repositories

update\_me supports packages installed from private repositories on GitHub and GitLab. To get upstream package version from either, you should only have to configure a personal access token (PAT):

- For GitHub packages, update\_me checks, in order:
  - The GITHUB\_PAT environmental variable
  - The GITHUB\_TOKEN environmental variable
  - Any personal access tokens configured using `gitcreds::gitcreds_set()`
- For GitLab packages, update\_me checks, in order:
  - The GITLAB\_PAT environmental variable
  - The GITLAB\_TOKEN environmental variable
  - Any personal access tokens configured using `gitcreds::gitcreds_set()`

## See Also

`update_me_on()` and `update_me_off()` to disable update\_me for all packages

## Examples

```
# If you want to check non-standard repos for new versions of packages,
# you'll first have to set the repos global option. Note that each
# option must be named for compatibility with update_me
old_repos <- options(repos = c(

  # Your default CRAN mirror will likely be something like this
  CRAN = "https://cloud.r-project.org",

  # The r-lib r-universe, including dev versions of infrastructure packages
  # like cli, rlang, etc
  `r-lib` = "https://r-lib.r-universe.dev"
))

# 1. New versions will first be looked up from the r-lib R universe by default
# 2. If not found, they will be looked up from the usual CRAN mirror
# 3. dplyr will always be first looked up from GitHub
# 4. ggplot2 won't be looked up or notified about
```

```
old_updateme_sources <- updateme_sources_set(  
  "r-lib",  
  "CRAN",  
  dplyr = "https://github.com/tidyverse/dplyr", # Name is optional here  
  ggplot2 = NA  
)  
  
# memoise should now be looked up from the r-lib r-universe  
if (interactive()) {  
  library(memoise)  
}  
  
# Restore old options  
options(old_repos)  
options(old_updateme_sources)
```

# Index

`gitcreds::gitcreds_set()`, [3](#)

`updateme_off (updateme_on)`, [2](#)

`updateme_off()`, [3](#)

`updateme_on`, [2](#)

`updateme_on()`, [3](#)

`updateme_sources_set`, [2](#)

`updateme_sources_set()`, [2](#)