

Package: progressify (via r-universe)

October 11, 2026

Version 0.2.0-9020

Title Progress Reporting of Common Functions via One Magic Function

Description The progressify() function rewrites (transpiles) calls to sequential and parallel map-reduce functions such as base::lapply(), purrr::map(), foreach::foreach(), and plyr::lply() to signal progress updates. By combining this function with R's native pipe operator, you have a straightforward way to report progress on iterative computations with minimal refactoring, e.g. 'lapply(x, fcn) |> progressify()' and 'purrr::map(x, fcn) |> progressify()'. It is compatible with the parallel-processing map-reduce packages 'future.apply', 'furry', 'crossmap', 'foreach', 'doFuture', and 'futzurize'. It also supports domain-specific packages including 'boot', 'fwb', 'lme4', 'partykit', 'sandwich', and 'SimDesign', e.g. 'boot::boot(data, stat, R) |> progressify()'.

License Apache License (>= 2)

URL <https://progressify.futureverse.org>,
<https://github.com/futureverse/progressify>

BugReports <https://github.com/futureverse/progressify/issues>

Depends R (>= 4.1.0), progressr

Imports methods

Suggests base64enc, boot, commonmark, crossmap, doFuture, foreach, furry, future.apply, fwb, futuzize, lme4, partykit, plyr, purrr, sandwich, SimDesign, tools

VignetteBuilder progressify

Encoding UTF-8

Roxygen list(markdown = TRUE)

Config/roxygen2/version 8.1.0

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

Date/Publication 2026-10-06 20:43:05 UTC

RemoteUrl <https://github.com/futureverse/progressify>
RemoteRef develop
RemoteSha 4ae67d78197092c31fe491f6c04313806faa90c6

Contents

progressify	2
progressify_supported_packages	3
Index	5

progressify	<i>Evaluate a regular map-reduce call with progress updates</i>
-------------	---

Description

Evaluate a regular map-reduce call with progress updates

Usage

```
progressify(  
  expr,  
  substitute = TRUE,  
  ...,  
  when = TRUE,  
  eval = TRUE,  
  envir = parent.frame()  
)
```

Arguments

- | | |
|------------|---|
| expr | An R expression. |
| substitute | If TRUE, expr is quoted. |
| when | If TRUE (default), the expression is progressified, otherwise not. |
| eval | If TRUE (default), the progressified expression is evaluated, otherwise it is returned. |
| envir | The environment in which expr is evaluated. |
| ... | Not used. |

Value

Returns the value of the evaluated expression expr.

Expression unwrapping

The transpilation mechanism includes logic to "unwrap" expressions enclosed in constructs such as `!, { }, ( ), local(), I(), identity(), invisible(), suppressMessages(), suppressWarnings(), suppressPackageStartupMessages(), withCallingHandlers(),` and `with()`. The transpiler descends through wrapping constructs until it finds a transpilable expression, avoiding the need to place `progressify()` inside such constructs. This allows for patterns like:

```
y <- {  
  lapply(xs, fcn)  
} |> suppressMessages() |> progressify()
```

avoiding having to write:

```
y <- {  
  lapply(xs, fcn) |> progressify()  
} |> suppressMessages()
```

Examples

```
handlers(global = TRUE) # listen to progress updates  
  
xs <- list(1, 1:2, 1:2, 1:5)  
y <- lapply(X = xs, FUN = sum) |> progressify()  
str(y)
```

progressify_supported_packages

List packages and functions supporting progressification

Description

List packages and functions supporting progressification

Usage

```
progressify_supported_packages()  
  
progressify_supported_functions(package)
```

Arguments

package A package name.

Value

A character vector of package or function names.

Examples

```
pkgs <- progressify_supported_packages()
pkgs

fcns <- progressify_supported_functions("base")
print(fcns)

if (requireNamespace("purrr")) {
  fcns <- progressify_supported_functions("purrr")
  print(fcns)
}
```

Index

progressify, [2](#)
progressify_supported_functions
 (progressify_supported_packages),
 [3](#)
progressify_supported_packages, [3](#)
pz (progressify), [2](#)