

Package ‘progressify’

May 9, 2026

Version 0.1.0

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 'futzurize' package for parallelization, e.g. `'lapply(x, fcn) |> progressify() |> futurize()'` and `'purrr::map(x, fcn) |> futurize() |> progressify()'`.

License GPL (>= 3)

Depends R (>= 4.1.0), progressr

Imports methods

Suggests base64enc, commonmark, crossmap, doFuture, foreach, furr, future.apply, futurize, partykit, plyr, purrr, tools

VignetteBuilder progressify

Encoding UTF-8

RoxygenNote 7.3.3

NeedsCompilation no

Author Henrik Bengtsson [aut, cre, cph]

Maintainer Henrik Bengtsson <henrikb@braju.com>

Repository CRAN

Date/Publication 2026-04-07 08:00:08 UTC

Contents

progressify	2
progressify_supported_packages	3

Index	4
--------------	----------

progressify

Evaluate a regular map-reduce call with progress updates

Description

Evaluate a regular map-reduce call with progress updates

Usage

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

Arguments

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

Value

Returns the value of the evaluated expression `expr`.

Examples

```
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](#)