

Package: imputesrcref (via r-universe)

July 15, 2026

Type Package

Title Impute Transparent Source References for Injected Braces

Version 0.0.0.9000

Author Filip Krikava [aut, cre, cph], Matej Kocourek [aut, cph],
Pierre Donat-Bouillud [aut, cph]

Maintainer Filip Krikava <krikava@gmail.com>

Description Rewrites R function ASTs by injecting transparent brace
calls in control-flow positions and imputing source references
(`'srcRef'`) from parse data. Supports individual functions,
sourced files, and installed package namespace functions,
including package-style `'srcRef'` line mappings.

Suggests ggplot2, mutator, testthat (>= 3.0.0)

Remotes PRL-PRG/mutator

Config/testthat/edition 3

License MIT + file LICENSE

Encoding UTF-8

Roxygen list(markdown = TRUE)

RoxygenNote 7.3.3

Repository <https://prl-prg.r-universe.dev>

Date/Publication 2026-07-15 11:06:14 UTC

RemoteUrl <https://github.com/PRL-PRG/imputesrcref>

RemoteRef HEAD

RemoteSha 6fb4435f3393c71da676956d126d2bc8f8e256c0

Contents

get_impute_blacklist	2
impute_package_srcrefs	3
impute_srcrefs	3
source_impute_srcrefs	4

Index	6
--------------	----------

get_impute_blacklist *Manage call blacklist for generic argument wrapping*

Description

By default, `impute_srcrefs()` skips argument wrapping for primitive SPECIALSXP calls discovered from `builtins()`. These functions inspect and modify the user-configured portion of that blacklist.

Usage

```
get_impute_blacklist(include_default = TRUE)

set_impute_blacklist(functions, append = TRUE)

reset_impute_blacklist()
```

Arguments

include_default	If TRUE, include built-in SPECIALSXP names.
functions	Character vector of call names. NULL clears user entries.
append	If TRUE append to existing user entries; otherwise replace.

Details

- `get_impute_blacklist()` returns the blacklist, optionally including the built-in defaults.
- `set_impute_blacklist()` adds or replaces user entries, stored in `options(imputesrcref.wrap_arg_blacklist = ...)`.
- `reset_impute_blacklist()` clears user entries.

Value

`get_impute_blacklist()` returns a sorted unique character vector of call names. `set_impute_blacklist()` invisibly returns the current user-configured entries. `reset_impute_blacklist()` invisibly returns an empty character vector.

Examples

```
head(get_impute_blacklist())
set_impute_blacklist(c("str_c", "paste"))
get_impute_blacklist(include_default = FALSE)
reset_impute_blacklist()
```

impute_package_srcrefs

Check parse data for a package and impute srcrefs on all its functions.

Description

Check parse data for a package and impute srcrefs on all its functions.

Usage

```
impute_package_srcrefs(package, include_internal = TRUE, verbose = TRUE)
```

Arguments

package	Package name.
include_internal	If TRUE, inspect all namespace functions.
verbose	If TRUE, print a patch summary.

Value

Invisibly returns a list with:

- package: package name
- fn_names: inspected function names
- failed: failure messages (NA for successfully patched functions)
- patched_count: number of patched functions

impute_srcrefs

Impute transparent srcrefs for injected braces in a function AST.

Description

Traverses a function's AST and wraps unbraced expressions in targeted positions with { . . . } while attaching transparent srcrefs to the injected brace calls. The srcref assigned to an injected brace matches the span of the wrapped expression so that source mapping stays aligned with original code.

Usage

```
impute_srcrefs(fn, wrap_call_args = TRUE, quiet = FALSE)
```

Arguments

<code>fn</code>	A function.
<code>wrap_call_args</code>	If TRUE (default), wrap generic function-call arguments that are call expressions.
<code>quiet</code>	If TRUE, suppress the informational message emitted when a function has no srcref metadata and deparse fallback is disabled. Useful for callers (such as coverage tools) that process many functions and handle the no-srcref case themselves. Defaults to FALSE.

Details

Covered constructs include:

- if / else
- for, while, repeat
- switch
- logical operators (&&, ||, &, |)
- function defaults and function bodies
- function call arguments (optional; call expressions only)

For functions without srcref metadata, deparse-based fallback is disabled by default. To allow fallback, set `options(impuresrcref.allow_deparse_fallback = TRUE)`.

Generic call argument wrapping skips blacklisted callee names. By default the blacklist includes primitive SPECIALSXP calls from `builtins()`. Use `set_impute_blacklist()` / `reset_impute_blacklist()` to customize.

Value

A function with transformed body/formals and preserved function-level attributes (including srcref/srcfile metadata when present).

Examples

```
options(keep.source = TRUE)
f <- eval(parse(text = "function(x, y) if (x && y) f() else g()", keep.source = TRUE)[[1]])
g <- impute_srcrefs(f)
g
```

`source_impute_srcrefs` *Source an R file and impute srcrefs on loaded functions.*

Description

Sources a file with `base::sys.source()` and then rewrites function bindings that were created or changed by that source call via `impute_srcrefs()`. This is the recommended entry point for source files because it can enforce both source retention and parse-data retention.

Usage

```
source_impute_srcrefs(  
  file,  
  envir = parent.frame(),  
  chdir = FALSE,  
  keep.source = TRUE,  
  keep.parse.data = TRUE,  
  toplevel.env = as.environment(envir),  
  all.names = TRUE  
)
```

Arguments

file	Path to an R source file.
envir	Environment where the file is sourced.
chdir	Passed to <code>base::sys.source()</code> .
keep.source	Passed to <code>base::sys.source()</code> . Defaults to TRUE.
keep.parse.data	Passed to <code>base::sys.source()</code> . Defaults to TRUE.
toplevel.env	Passed to <code>base::sys.source()</code> .
all.names	Include hidden bindings when scanning for functions.

Value

Invisibly returns a list with:

- file: normalized file path
- functions: names of patched functions
- count: number of patched functions
- keep.source / keep.parse.data: effective source flags

Examples

```
## Not run:  
tf <- tempfile(fileext = ".R")  
writeLines("f <- function(x, y) if (x && y) f() else g()", tf)  
env <- new.env(parent = baseenv())  
source_impute_srcrefs(tf, envir = env)  
env$f  
  
## End(Not run)
```

Index

`base::sys.source()`, [4](#), [5](#)

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

`get_impute_blacklist`, [2](#)

`impute_package_srcrefs`, [3](#)

`impute_srcrefs`, [3](#)

`impute_srcrefs()`, [2](#), [4](#)

`reset_impute_blacklist`

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

`reset_impute_blacklist()`, [4](#)

`set_impute_blacklist`

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

`set_impute_blacklist()`, [4](#)

`source_impute_srcrefs`, [4](#)