

# Package: sourcemodify (via r-universe)

August 30, 2024

**Type** Package

**Title** Helper functions to modify source code of R scripts

**Version** 0.1.0

**Author** Sebastian Kranz

**Maintainer** Sebastian Kranz <sebastian.kranz@uni-ulm.de>

**Description** Helper functions to modify source code of R scripts

**License** GPL >= 2.0

**Encoding** UTF-8

**LazyData** true

**Remotes** skranz/stringtools

**Depends** stringi, restorepoint, stringtools, dplyr

**RoxygenNote** 7.1.1

**Repository** <https://repboxr.r-universe.dev>

**RemoteUrl** <https://github.com/repboxr/sourcemodify>

**RemoteRef** main

**RemoteSha** da2bd07e1422819de8126539ac55a6d2250263b0

## Contents

getAugmentedParseData . . . . .	2
somo_calls_info . . . . .	2
somo_change_calls_fun_name . . . . .	3
somo_find_fun_arg . . . . .	3
somo_find_fun_call . . . . .	4
somo_found . . . . .	4
somo_init . . . . .	4
somo_make_code . . . . .	5
somo_surround_calls . . . . .	5
somo_surround_calls_arg . . . . .	6

<b>Index</b>	<b>7</b>
--------------	----------

`getAugmentedParseData` *Key internal function fo sourcemodify Returns parsed code as a tibble with detailed information where each subexpression is found in the code*

---

### **Description**

Key internal function fo sourcemodify Returns parsed code as a tibble with detailed information where each subexpression is found in the code

### **Usage**

```
getAugmentedParseData(  
  code,  
  calls = parse(text = code, keep.source = TRUE),  
  line.starts = find.line.starts(code)  
)
```

---

`somo_calls_info` *A simplified interface to `somo_find_fun_arg`*

---

### **Description**

Only works with named arguments.

### **Usage**

```
somo_calls_info(  
  somo,  
  fun,  
  arg_names,  
  arg_pos = NULL,  
  arg_def = NULL,  
  call_pd_rows = NULL,  
  arg_prefix = ""  
)
```

---

somo\_change\_calls\_fun\_name

*Change the name of a called function without adapting any arguments*


---

**Description**

If the function call directly

**Usage**

```
somo_change_calls_fun_name(somo, fun, new, ignore_if_pkg = TRUE)
```

**Arguments**

somo	The somo object generated with somo_init.
fun	Function name that shall be changed in calls
new	New function name to be called
ignore_if_pkg	If TRUE don't change calls of the form pkg::fun(). If FALSE a call like pkg::fun() will be replaced by just new().

**Value**

The modified somo object

---

somo\_find\_fun\_arg

*Find the text of the function argument in all calls to fun*


---

**Description**

Find the text of the function argument in all calls to fun

**Usage**

```
somo_find_fun_arg(
  somo,
  fun,
  arg_name = NULL,
  arg_pos = NULL,
  arg_def = NULL,
  call_pd_rows = NULL
)
```

---

somo_find_fun_call	<i>Find the text of the function argument in all calls to fun</i>
--------------------	---

---

**Description**

Find the text of the function argument in all calls to fun

**Usage**

```
somo_find_fun_call(somo, fun)
```

---

somo_found	<i>Returns the results from the last somo_find_* call as a tibble</i>
------------	---

---

**Description**

Returns the results from the last somo\_find\_\* call as a tibble

**Usage**

```
somo_found(somo)
```

---

somo_init	<i>Init a new source modifier object</i>
-----------	--

---

**Description**

Init a new source modifier object

**Usage**

```
somo_init(code, start.size = 200)
```

**Arguments**

code	The R source code as text file
------	--------------------------------

---

somo_make_code	<i>Returns the modified code of a somo object</i>
----------------	---

---

**Description**

Returns the modified code of a somo object

**Usage**

```
somo_make_code(somo)
```

---

somo_surround_calls	<i>Surround calls to the specified functions</i>
---------------------	--

---

**Description**

Will by typically used to wrap a function call into another function call that can analyse or modify the result.

**Usage**

```
somo_surround_calls(somo, funs, pre, post)
```

**Arguments**

somo	The somo object generated with somo_init.
funs	One or several function names.
pre	Text to be added before the function call.
post	Text to be added the function call.

**Value**

The modified somo object

---

`somo_surround_calls_arg`*Surround an argument of specific function*

---

**Description**

Will by typically used to wrap the argument (like a filename) of function call into another function call that can analyse or modify the argument.

**Usage**

```
somo_surround_calls_arg(  
    somo,  
    fun,  
    arg_name = NULL,  
    arg_pos = NULL,  
    pre,  
    post,  
    arg_def = NULL  
)
```

**Arguments**

<code>somo</code>	The somo object generated with <code>somo_init</code> .
<code>fun</code>	Name of the function whose argument shall be wrapped
<code>arg_name</code>	The name of the argument (must be given if <code>arg_pos</code> is NULL)
<code>arg_pos</code>	Position of the argument (must be given if <code>arg_name</code> is NULL)
<code>pre</code>	Text to be added before the function call.
<code>post</code>	Text to be added the function call.
<code>arg_def</code>	Names of all function arguments in the correct order. The result of calling <code>names(as.list(args(fun)))</code> . This argument is optional, but without it matching may fail.

**Value**

The modified somo object

# Index

[getAugmentedParseData](#), [2](#)

[somo\\_calls\\_info](#), [2](#)

[somo\\_change\\_calls\\_fun\\_name](#), [3](#)

[somo\\_find\\_fun\\_arg](#), [3](#)

[somo\\_find\\_fun\\_call](#), [4](#)

[somo\\_found](#), [4](#)

[somo\\_init](#), [4](#)

[somo\\_make\\_code](#), [5](#)

[somo\\_surround\\_calls](#), [5](#)

[somo\\_surround\\_calls\\_arg](#), [6](#)