

# Package ‘GlobalOptions’

April 8, 2026

**Type** Package

**Title** Generate Functions to Get or Set Global Options

**Version** 0.1.4

**Date** 2026-04-08

**Depends** R (>= 3.3.0), methods

**Imports** utils

**Suggests** testthat (>= 1.0.0), knitr, markdown, GetoptLong, rmarkdown

**VignetteBuilder** knitr

**Description** It provides more configurations on the option values such as validation and filtering on the values, making options invisible or private.

**URL** <https://github.com/jokergoo/GlobalOptions>

**License** MIT + file LICENSE

**RoxygenNote** 7.3.1

**Encoding** UTF-8

**NeedsCompilation** no

**Author** Zuguang Gu [aut, cre] (ORCID: <<https://orcid.org/0000-0002-7395-8709>>)

**Maintainer** Zuguang Gu <guzuguang@suat-sz.edu.cn>

**Repository** CRAN

**Date/Publication** 2026-04-08 09:50:09 UTC

## Contents

.v . . . . .	2
print.GlobalOptionsFun . . . . .	2
reset_opt . . . . .	3
setGlobalOptions . . . . .	4
[.GlobalOptionsFun . . . . .	5

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

---

```
.v
```

*Get other option values*

---

**Description**

Get other option values

**Usage**

```
.v(opt_name, name_is_character = NA)

## S3 method for class 'InternalOptionValue'
x$nm
```

**Arguments**

opt_name	The option name.
name_is_character	Please ignore, only used internally.
x	should always be written as <code>‘.v‘</code> .
nm	The option name.

**Details**

When setting one option, the value can be dependent on other option names. The current value of other option can be accessed by `‘.v(nm)‘` or `‘.v$nm‘`.

**Examples**

```
opt = set_opt(a = 1, b = function() .v$a*2)
opt$b
opt(a = 2); opt$b
opt(a = 4); opt$b
reset_opt(opt); opt$b
```

---

```
print.GlobalOptionsFun
```

*Print options*

---

**Description**

Print options

**Usage**

```
## S3 method for class 'GlobalOptionsFun'
print(x, ...)
```

**Arguments**

x                    The option object returned by [`set_opt()`] or [`setGlobalOptions()`].  
 ...                  Other arguments.

**Examples**

```
opt = set_opt(a = 1, b = "b")
opt
```

---

reset_opt	<i>Helper functions</i>
-----------	-------------------------

---

**Description**

Helper functions

**Usage**

```
reset_opt(opt)

add_opt(opt, ...)
```

**Arguments**

opt                    The option object returned by [`set_opt()`] or [`setGlobalOptions()`].  
 ...                  New options.

**Details**

`reset_opt()` is identical to `opt(RESET = TRUE)`.  
`add_opt()` is identical to `opt(..., ADD = TRUE)`.

**Examples**

```
opt = set_opt(a = 1, b = 2)
opt$a = 100; opt$b = 200
opt
reset_opt(opt)
opt
opt = set_opt(a = 1)
add_opt(opt, b = 2)
opt
```

---

setGlobalOptions      *Option Generator*

---

## Description

Option Generator

## Usage

```
setGlobalOptions(...)
```

```
set_opt(...)
```

## Arguments

...                    Specification of options, see the **\*\*Details\*\*** section.

## Details

The simplest way is to construct an option function (e.g. 'opt()') as:

```
““ opt = set_opt( "a" = 1, "b" = "text" ) ““
```

Then users can get or set the options by

```
““ opt() opt("a") opt$a opt[["a"]] opt(c("a", "b")) opt("a", "b") opt("a" = 2) opt$a = 2 opt[["a"]] = 2
opt("a" = 2, "b" = "new_text") ““
```

Options can be reset to their default values by:

```
““ opt(RESET = TRUE) # or reset_opt(opt) ““
```

The value for each option can be set as a list which contains more complex configurations:

```
““ opt = set_opt( "a" = list( .value = 1, .length = 1, .class = "numeric", .validate = function(x) x > 0
) ) ““
```

The different fields in the list can be used to filter or validate the option values.

- `‘.value‘`: The default value. - `‘.length‘`: The valid length of the option value. It can be a vector, the check will be passed if one of the length fits. - `‘.class‘`: The valid class of the option value. It can be a vector, the check will be passed if one of the classes fits. - `‘.validate‘`: Validation function. The input parameter is the option value and should return a single logical value. - `‘.failed_msg‘`: Once validation failed, the error message that is printed. - `‘.filter‘`: Filtering function. The input parameter is the option value and it should return a filtered option value. - `‘.read.only‘`: Logical. The option value can not be modified if it is set to `‘TRUE‘`. - `‘.visible‘`: Logical. Whether the option is visible to users. - `‘.private‘`: Logical. The option value can only be modified in the same namespace where the option function is created. - `‘.synonymous‘`: a single option name which should have been already defined ahead of current option. The option specified will be shared by current option.
- `‘.description‘`: a short text for describing the option. The description is only used when printing the object.

For more detailed explanation, please go to the vignette.

**Examples**

```
opt = set_opt(  
  a = 1,  
  b = "text"  
)  
opt  
# for more examples, please go to the vignette
```

---

[.GlobalOptionsFun     *Getter and setter functions*

---

**Description**

Getter and setter functions

**Usage**

```
## S3 method for class 'GlobalOptionsFun'  
x[nm]  
  
dump_opt(x, nm)  
  
## S3 method for class 'GlobalOptionsFun'  
x[[nm]]  
  
## S3 replacement method for class 'GlobalOptionsFun'  
x[[nm]] <- value  
  
## S3 method for class 'GlobalOptionsFun'  
names(x)  
  
## S3 method for class 'GlobalOptionsFun'  
.DollarNames(x, pattern = "")  
  
## S3 method for class 'GlobalOptionsFun'  
x$nm  
  
## S3 replacement method for class 'GlobalOptionsFun'  
x$nm <- value
```

**Arguments**

x	The option object returned by [ <code>set_opt()</code> ] or [ <code>setGlobalOptions()</code> ].
nm	A single option name.
value	The value which is assigned to the option.
pattern	Ignore.

**Details**

'[' (single bracket) returns a single option object.

'dump\_opt()' is identical to '['.

'[[' (double brackets) returns the value of the option.

The '.DollarNames' method makes the option object look like a list that it allows option name completion after '\$' (by double clicking the "enter/return" key).

**Examples**

```
opt = set_opt(a = 1, b = "b")
opt["a"]
opt["b"]
dump_opt(opt, "a")
dump_opt(opt, "b")
opt[["a"]]
opt[["b"]]
opt[["a"]] = 200
opt[["a"]]
names(opt)
opt$a
opt$a = 100
opt$a
```

# Index

`.DollarNames.GlobalOptionsFun`  
    (`[.GlobalOptionsFun]`), 5  
`.v`, 2  
`[.GlobalOptionsFun]`, 5  
`[[.GlobalOptionsFun`  
    (`[.GlobalOptionsFun]`), 5  
`[[<-.GlobalOptionsFun`  
    (`[.GlobalOptionsFun]`), 5  
`$.GlobalOptionsFun`  
    (`[.GlobalOptionsFun]`), 5  
`$.InternalOptionValue (.v)`, 2  
`$<-.GlobalOptionsFun`  
    (`[.GlobalOptionsFun]`), 5  
  
`add_opt (reset_opt)`, 3  
  
`dump_opt ([.GlobalOptionsFun]`, 5  
  
`names.GlobalOptionsFun`  
    (`[.GlobalOptionsFun]`), 5  
  
`print.GlobalOptionsFun`, 2  
  
`reset_opt`, 3  
  
`set_opt (setGlobalOptions)`, 4  
`setGlobalOptions`, 4