

# Package ‘GlobalOptions’

September 30, 2019

**Type** Package

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

**Version** 0.1.1

**Date** 2019-9-30

**Author** Zuguang Gu

**Maintainer** Zuguang Gu <z.gu@dkfz.de>

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

**Imports** utils

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

**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

**Repository** CRAN

**NeedsCompilation** no

**Date/Publication** 2019-09-30 19:40:02 UTC

## R topics documented:

<code>.DollarNames.GlobalOptionsFun</code> . . . . .	2
<code>.v</code> . . . . .	2
<code>dump_opt</code> . . . . .	3
<code>print.GlobalOptionsFun</code> . . . . .	4
<code>setGlobalOptions</code> . . . . .	4
<code>set_opt</code> . . . . .	6
<code>\$.GlobalOptionsFun</code> . . . . .	7
<code>[[.GlobalOptionsFun</code> . . . . .	7
<code>[[&lt;-.GlobalOptionsFun</code> . . . . .	8
<code>\$.GlobalOptionsFun</code> . . . . .	9
<code>\$.InternalOptionValue</code> . . . . .	9
<code>\$&lt;-.GlobalOptionsFun</code> . . . . .	10

---

`.DollarNames.GlobalOptionsFun`

*The `.DollarNames` method for the `GlobalOptionsFun` class*

---

### Description

The `.DollarNames` method for the `GlobalOptionsFun` class

### Usage

```
## S3 method for class 'GlobalOptionsFun'
.DollarNames(x, pattern = "")
```

### Arguments

`x`                    the object returned by `set_opt` or `setGlobalOptions`.  
`pattern`                pattern, please ignore it.

### Details

This makes the option object looks like a list that it allows option name completion after \$.

### Author(s)

z.gu@dkfz.de

### Examples

```
# There is no example
NULL
```

---

`.v`

*Get value of other options*

---

### Description

Get value of other options

### Usage

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

**Arguments**

opt\_name            name of the option, can be quoted or not quoted.  
name\_is\_character        whether opt\_name is a character, only used internally.

**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`.

**Author(s)**

Zuguang Gu <z.gu@dkfz.de>

**Examples**

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

---

dump_opt	<i>Print all fields of a single option</i>
----------	--

---

**Description**

Print all fields of a single option

**Usage**

```
dump_opt(opt, opt_name)
```

**Arguments**

opt                    the option object returned by [set\\_opt](#) or [setGlobalOptions](#).  
opt\_name                a single name of the option.

**Details**

Actually this function is identical to `opt[opt_name]`.

**Author(s)**

z.gu@dkfz.de

**Examples**

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

```
print.GlobalOptionsFun
```

*Print the GlobalOptionsFun object*

---

### Description

Print the GlobalOptionsFun object

### Usage

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

### Arguments

x                    the option object returned by [set\\_opt](#) or [setGlobalOptions](#).  
...                   other arguments

### Author(s)

z.gu@dkfz.de

### Examples

```
# There is no example  
NULL
```

---

```
setGlobalOptions
```

*Produce a function which can get or set global options*

---

### Description

Produce a function which can get or set global options

### Usage

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

### Arguments

...                   specification of options, see 'details' section

## Details

The function has a short name `set_opt`.

The most simple 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)
```

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

```
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.

### Author(s)

Zuguang Gu <z.gu@dkfz.de>

### Examples

```
# please go to the vignette
```

---

set_opt	<i>Produce a function which can get or set global options</i>
---------	---

---

### Description

Produce a function which can get or set global options

### Usage

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

### Arguments

```
... all go to setGlobalOptions
```

### Details

This is just a short name for [setGlobalOptions](#).

### Author(s)

z.gu@dkfz.de

### Examples

```
# There is no example
NULL
```

---

[.GlobalOptionsFun     *Get a single GlobalOption object*

---

### Description

Get a single GlobalOption object

### Usage

```
## S3 method for class 'GlobalOptionsFun'  
x[nm]
```

### Arguments

x                    the option object returned by [set\\_opt](#) or [setGlobalOptions](#).  
nm                   a single name of the option.

### Details

This function is only used internally.

### Author(s)

Zuguang Gu <z.gu@dkfz.de>

### Examples

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

---

[ [.GlobalOptionsFun     *Get option value by subset operator*

---

### Description

Get option value by subset operator

### Usage

```
## S3 method for class 'GlobalOptionsFun'  
x[[nm]]
```

### Arguments

x                    the option object returned by [set\\_opt](#) or [setGlobalOptions](#).  
nm                   a single option name.

**Details**

opt[["a"]] is same as opt("a") or opt\$a.

**Author(s)**

Zuguang Gu <z.gu@dkfz.de>

**Examples**

```
opt = set_opt(a = 1)
opt[["a"]]
```

---

[[<-GlobalOptionsFun *Set option value by subset operator*

---

**Description**

Set option value by subset operator

**Usage**

```
## S3 replacement method for class 'GlobalOptionsFun'
x[[nm]] <- value
```

**Arguments**

x	the option object returned by <a href="#">set_opt</a> or <a href="#">setGlobalOptions</a> .
nm	a single option name.
value	the value which is assigned to the option.

**Details**

opt[["a"]] = 1 is same as opt("a" = 1) or opt\$a = 1.

**Author(s)**

Zuguang Gu <z.gu@dkfz.de>

**Examples**

```
opt = set_opt(a = 1)
opt[["a"]] = 2
opt$a
```



---

\$.GlobalOptionsFun *Get option value by dollar symbol*

---

### Description

Get option value by dollar symbol

### Usage

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

### Arguments

x                   the object returned by `set_opt` or `setGlobalOptions`.  
nm                   a single option name.

### Details

opt\$a is same as `opt("a")`.

### Author(s)

Zuguang Gu <z.gu@dkfz.de>

### Examples

```
opt = set_opt(a = 1)  
opt$a
```

---

\$.InternalOptionValue *Get value of other options*

---

### Description

Get value of other options

### Usage

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

### Arguments

x                   should always be `.v`  
nm                   name of the option

**Details**

`.v$nm` is basically a short version of `.v(nm)`.

**Author(s)**

Zuguang Gu <z.gu@dkfz.de>

**See Also**

[.v](#)

**Examples**

```
# There is no example
NULL
```

---

```
$<-.GlobalOptionsFun Set option value by dollar symbol
```

---

**Description**

Set option value by dollar symbol

**Usage**

```
## S3 replacement method for class 'GlobalOptionsFun'
x$nm <- value
```

**Arguments**

<code>x</code>	the object returned by <code>set_opt</code> or <code>setGlobalOptions</code> .
<code>nm</code>	a single option name.
<code>value</code>	the value which is assigned to the option.

**Details**

`opt$a = 1` is same as `opt("a" = 1)`.

Note you cannot reconfigure the option by assigning a configuration list.

**Author(s)**

Zuguang Gu <z.gu@dkfz.de>

**Examples**

```
opt = set_opt(a = 1)
opt$a = 2
opt$a
```

# Index

`.DollarNames.GlobalOptionsFun`, 2  
`.v`, 2, 10  
`[.GlobalOptionsFun`, 7  
`[[.GlobalOptionsFun`, 7  
`[[<-.GlobalOptionsFun`, 8  
`$.GlobalOptionsFun`, 9  
`$.InternalOptionValue`, 9  
`$<-.GlobalOptionsFun`, 10

`Assign.GlobalOptionsFun`  
    (`$<-.GlobalOptionsFun`), 10

`Dot.DollarNames.GlobalOptionsFun`  
    (`.DollarNames.GlobalOptionsFun`),  
    2  
`Dot.v (.v)`, 2  
`dump_opt`, 3

`Extract.GlobalOptionsFun`  
    (`[.GlobalOptionsFun`), 7  
`ExtractExtract.GlobalOptionsFun`  
    (`[[.GlobalOptionsFun`), 7  
`ExtractExtractAssign.GlobalOptionsFun`  
    (`[[<-.GlobalOptionsFun`), 8

`print.GlobalOptionsFun`, 4

`set_opt`, 2–5, 6, 7–10  
`setGlobalOptions`, 2–4, 4, 6–10  
`Subset.GlobalOptionsFun`  
    (`$.GlobalOptionsFun`), 9  
`Subset.InternalOptionValue`  
    (`$.InternalOptionValue`), 9