

# Package ‘packS4’

May 28, 2015

**Type** Package

**Title** Toy Example of S4 Package

**Version** 0.9.3

**Date** 2015-05-27

**Author** Christophe Genolini

**Maintainer** Christophe Genolini <genolini@u-paris10.fr>

**Description** Illustration of the book ``Petit Manuel de Programmation Orientee Objet sous R''. The english version ``A (Not so) Short Introduction to S4'' is on CRAN, 'Contributed documentation'.

**License** GPL (>= 2)

**LazyLoad** yes

**Depends** methods, graphics, codetools

**Collate** global.R ClassU.R ClassV.R ClassW.R ClassV-ClassW.R  
classCreator.R codeVerif.R

**URL** [www.r-project.org](http://www.r-project.org),[christophe.genolini.free.fr/webTutorial](http://christophe.genolini.free.fr/webTutorial)

**NeedsCompilation** no

**Repository** CRAN

**Date/Publication** 2015-05-28 01:54:05

## R topics documented:

packS4-package . . . . .	2
classCreator . . . . .	3
ClassV-class . . . . .	4
ClassW-class . . . . .	4
dataAges . . . . .	5
detectGlobal . . . . .	6
functionClassicA . . . . .	7
plot-methods . . . . .	8
publicA . . . . .	9
publicA-ClassV-methods . . . . .	10
publicA-methods . . . . .	10
publicB . . . . .	11
tryBug . . . . .	12

---

packS4-package	<i>Toy example of a S4 package</i>
----------------	------------------------------------

---

## Description

This package is a toy example build to illustrate the construction of an S4 package as explain in the tutorial *S4 : From An Idea To Its Package*.

## Details

```

Package:    packS4
Type:      Package
Version:   0.9.1
Date:      2012-05-01
License:   GPL (>=2.0)
LazyLoad:  yes
Depends:   methods,graphics

```

This package is a toy example build to illustrate the construction of a package as explain in the book *Book: "Petit Manuel de Programmation Orientee Objet sous R"* There is mainly two classes. May be there is another one, but it is a secret...

## Author(s)

Christophe Genolini <genolini@u-paris10.fr>

## References

Book: "Petit Manuel de Programmation Orientee Objet sous R"

## See Also

[packClassic](#) is another toy example build to illustrate the construction of a classic (non S4) package.

## Examples

```

### classW is the constructor of ClassW object
www <- classW(u1=3,u2=4,w1="Title")

### plot is a method for ClassW
plot(www)

### There is no constructor for ClassV (which is bad !)
new("ClassV",v1=2,v2=3)

```

---

classCreator	<i>classCreator</i>
--------------	---------------------

---

**Description**

Write the 'R code' for creating a new class

**Usage**

```
classCreator(name, field, type)
```

**Arguments**

name	[character] Name of the class (with its first letter in upper case.
field	[vector(character)] Name(s) of the field of the class.
type	[vector(character)] Type of the field of the class.

**Details**

Given its three argument, this function display on screen the R code for creating a new class. More precisely, it write code for 'validiy', 'constructor', 'show', 'get' & 'set'.

**Value**

None

**Author(s)**

Christophe Genolini <genolini@u-paris10.fr>

**References**

Book: "Petit Manuel de Programmation Orientee Objet sous R"

**Examples**

```
classCreator("Complex", c("xx", "yy"), c("numerix", "numeric"))
```

---

ClassV-class

*Class "ClassV"*

---

### Description

This is a nice class.

### Objects from the Class

Objects can be created by calls of the form `new("ClassV", v1=val1, v2=val2)`.

### Slots

v1: ["numeric"] first slot

v2: ["numeric"] second slot

### Methods

**plot** signature(x = "ClassV"): for more detail, see [plot for ClassV](#)

**publicA** signature(object = "ClassV"): for more detail, see [plot for ClassV](#)

### Author(s)

Christophe Genolini <genolini@u-paris10.fr>

### References

Book: "Petit Manuel de Programmation Orientee Objet sous R"

### Examples

```
showClass("ClassV")
new("ClassV", v1=1, v2=2)
```

---

ClassW-class

*Class "ClassW"*

---

### Description

Another nice class

### Objects from the Class

Objects can be created by calls of the form `classW(x1=val1, x2=val2, w1=val3)`.

**Slots**

w1: ["numeric"] first slot  
u1: ["numeric"] second slot  
u2: ["character"] third slot

**Extends**

Class ClassU, directly. (ClassX is for internal use only).

**Methods**

[ signature(object = "ClassW"): accessor  
[<- signature(object = "ClassW"): accessor  
**plot** signature(x = "ClassW"): for more detail, see [plot for ClassW](#)  
**publicA** signature(object = "ClassW"): for more detail, see [publicA for ClassW](#)

**Author(s)**

Christophe Genolini <genolini@u-paris10.fr>

**References**

Book: "Petit Manuel de Programmation Orientee Objet sous R"

**Examples**

```
showClass("ClassW")  
www <- classW(w1="az", u1=2, u2=2)  
www["w1"]  
show(www)  
plot(www)
```

---

dataAges

*Toy data frame for packClassic and packS4*

---

**Description**

This data.frame is a fake toy example made up to illustrate the inclusion of data in a package.

**Usage**

```
data(dataAges)
```

**Format**

A data frame with 5 observations on the following 2 variables.

sex a factor with levels F H, which denote the gender of the subject

age a numeric vector for teh age.

**Details**

So simple, no detail are needed.

**Author(s)**

Christophe Genolini <genolini@u-paris10.fr>

**Source**

Fake data.

**References**

Book: "Petit Manuel de Programmation Orientee Objet sous R"

**Examples**

```
data(dataAges)
str(dataAges)
```

---

detectGlobal                    ~ Function: detectGlobal ~

---

**Description**

Detect if there is some global variable in a function.

**Usage**

```
detectGlobal(realResult, tolerance = 0, theoResult = "", result = TRUE)
```

**Arguments**

tolerance	Some keyword are detected as global variable whereas they are not. Tolerance is the number of false detection that should be ignored.
realResult	Either the name of a function (see example 1), or a function with its argument (see example 2).
theoResult	The theoretical result of the computation of the function with its argument.
result	TRUE or FALSE: shall realResult should be the same than the theoResult, or not ? This is usefull to test conter example.

**Details**

Detect if there is some global variable in a function.

**Value**

None

**Examples**

```
### example 1
f <- function(x)x^2
detectGlobal(f)

g <- function(x)x^2*y
detectGlobal(g)

#####
### example 2
f <- function(x)x^2

### the next line should ring a bell, because 2^2 is not 8
detectGlobal(f(2),8)

### the next line should not ring a bell, because 2^2 is not 8, and we ask for counter-example
detectGlobal(f(2),8,FALSE)

#####
### example 3
h <- function(x){
  apply(matrix(1:x),1,length)
}

### 'length' is detected as a global variable whereas it is a function
detectGlobal(h)

### So we use tolerance=1
detectGlobal(h,,1)
```

---

functionClassicA      *functionClassicA*

---

**Description**

A very nice function

**Usage**

```
functionClassicA(age)
```

**Arguments**

age [numeric] The age of the patient

**Details**

Double the age

**Value**

An age [numeric]

**Author(s)**

Christophe Genolini <genolini@u-paris10.fr>

**References**

Book: "Petit Manuel de Programmation Orientee Objet sous R"

**Examples**

```
myAge <- 35
functionClassicA(myAge)
```

---

plot-methods

*plot-ClassV ; plot-ClassW*

---

**Description**

plot for two objects.

**Methods**

**x = "ClassV"** just plot a point  
**x = "ClassW"** plot a point and add a title

**Author(s)**

Christophe Genolini <genolini@u-paris10.fr>

**References**

Book: "Petit Manuel de Programmation Orientee Objet sous R"

**Examples**

```
www <- classW(w1="A",u1=2,u2=-4)
plot(www)
```



---

publicA	<i>publicA-generic</i>
---------	------------------------

---

### Description

A nice method

### Usage

```
publicA(object)
```

### Arguments

object            can be either ClassV or ClassW

### Details

For detail, see [publicA for ClassV](#) or [publicA for ClassW](#)

### Value

One of the slot [numeric]

### Author(s)

Christophe Genolini <genolini@u-paris10.fr>

### References

Book: "Petit Manuel de Programmation Orientee Objet sous R"

### Examples

```
vvv <- new("ClassV", v1=2, v2=3)
publicA(vvv)
www <- classW(u1=4, u2=5, w1="E")
publicA(www)
```

---

publicA-ClassV-methods

*publicA for ClassV*

---

**Description**

publicA is a great function.

**Methods**

**object = "ClassV"** publicA on object ClassV

**Author(s)**

Christophe Genolini <genolini@u-paris10.fr>

**References**

Book: "Petit Manuel de Programmation Orientee Objet sous R"

**Examples**

```
vvv <- new("ClassV",v1=2,v2=-4)
publicA(vvv)
```

---

publicA-methods

*publicA for ClassW*

---

**Description**

publicA is a great function.

**Methods**

**object = "ClassW"** publicA on object ClassW

**Author(s)**

Christophe Genolini <genolini@u-paris10.fr>

**References**

Book: "Petit Manuel de Programmation Orientee Objet sous R"

**Examples**

```
www <- classW(w1="A",u1=2,u2=-4)
publicA(www)
```

---

`publicB`                      *publicB*

---

### **Description**

`publicB` is a great function too

### **Usage**

```
publicB(objectV, objectW)
```

### **Arguments**

<code>objectV</code>	Object of classV
<code>objectW</code>	Object of classV

### **Details**

`publicB` is a great function

### **Value**

A numeric

### **Author(s)**

Christophe Genolini <genolini@u-paris10.fr>

### **References**

Book: "Petit Manuel de Programmation Orientee Objet sous R"

### **Examples**

```
vvv <- new("ClassV", v1=1, v2=2)
www <- classW(w1="Z", u1=2, u2=3)
publicB(vvv, www)
```

---

`tryBug`*~ Function: tryBug ~*

---

### Description

This function "try" to run its argement (like the function `try`). If the evaluated argument is not correct, then everything is fine. If the argument is correct, then `tryBug` stop the execution.

### Usage

```
tryBug(...)
```

### Arguments

```
...           A command
```

### Details

This function "try" to run its argement (like the function `try`). If the evaluated argument is not correct, then everything is fine. If the argument is correct, then `tryBug` stop the execution. This is usefull in the programme tests file: in some case, on some specific argument, a function should not work, and if it does work, then there is a bug. The `tryBug` function will detect this kind of bug.

### Value

None

### Examples

```
### A function...
f <- function(oldYoung){
  if(oldYoung=="old"){
    cat("You are not that old!")
  }else{
    cat("You are young, great for you!")
  }
}

### ... that we test
# f("old") # ok
# f("young") # ok
# tryBug(f("dead")) #not ok

### The corrected function...
f <- function(oldYoung){
  if(oldYoung=="old"){
    cat("You are not that old!")
  }else{
    if(oldYoung=="young"){
```

```
        cat("You are young, great for you!")
      }else{
        stop("We deal only with young and old peoples!")
      }
    }
  }

### ... with its new tests.
f("old")
f("young")
tryBug(f("dead"))
```

# Index

## \*Topic **datasets**

dataAges, [5](#)

## \*Topic **documentation**

classCreator, [3](#)

ClassV-class, [4](#)

ClassW-class, [4](#)

dataAges, [5](#)

functionClassicA, [7](#)

packS4-package, [2](#)

plot-methods, [8](#)

publicA, [9](#)

publicA-ClassV-methods, [10](#)

publicA-methods, [10](#)

publicB, [11](#)

## \*Topic **methods**

classCreator, [3](#)

## \*Topic **package**

packS4-package, [2](#)

[,ClassW-method (ClassW-class), [4](#)

[<-,ClassW-method (ClassW-class), [4](#)

classCreator, [3](#)

ClassV (ClassV-class), [4](#)

ClassV-class, [4](#)

ClassW (ClassW-class), [4](#)

classW (ClassW-class), [4](#)

ClassW-class, [4](#)

dataAges, [5](#)

detectGlobal, [6](#)

functionClassicA, [7](#)

packClassic, [2](#)

packS4 (packS4-package), [2](#)

packS4-package, [2](#)

plot,ClassU-method (plot-methods), [8](#)

plot,ClassV-method (plot-methods), [8](#)

plot,ClassW-method (plot-methods), [8](#)

plot-methods, [8](#)

publicA, [9](#)

publicA,ClassV-method  
(publicA-ClassV-methods), [10](#)

publicA,ClassW-method  
(publicA-methods), [10](#)

publicA-ClassV-methods, [10](#)

publicA-methods, [10](#)

publicA-methods (publicA), [9](#)

publicB, [11](#)

publicB,ClassV,ClassW-method (publicB),  
[11](#)

publicB-methods (publicB), [11](#)

tryBug, [12](#)