

# Roxygen Vignette

Peter Danenberg <pcd@roxygen.org>

December 23, 2011

## Abstract

The purpose of the **Roxygen** Vignette is to show how to get up and running with **Roxygen**; for details, including a complete list of tags, consult the help pages or manual for:

- `make.callgraph.roclet`
- `make.collate.roclet`
- `make.namespace.roclet`
- `make.Rd.roclet`

## Contents

<b>1</b>	<b>Minimal Example: “Hello, Roxygen!”</b>	<b>1</b>
<b>2</b>	<b>Example: Pseudoprimality</b>	<b>3</b>
2.1	Package Description . . . . .	3
2.2	Fermat Test . . . . .	4
2.3	Pseudoprime . . . . .	6
2.4	Running Roxygen . . . . .	6

## 1 Minimal Example: “Hello, Roxygen!”

```
1 hello-roxygen.R
2 #' A package to check Roxygen's sanity
3 #' @name helloRoxygen-package
4 #' @docType package
NA
```

Figure 1: Roxygen sanity-check

`hello-roxygen.R` (fig. 1) is a minimal example to check the sanity of your **Roxygen** installation. It merely replaces the package description so that ‘**R CMD check**’ will run after **Roxygen** has processed the package skeleton.

First, create the package skeleton:

```
> library(roxygen)
> package.skeleton('helloRoxygen',
+                  code_files='hello-roxygen.R',
+                  force=TRUE)
```

then, run Roxygen on the new directory:

```
> roxygenize('helloRoxygen',
+            roxygen.dir='helloRoxygen',
+            copy.package=FALSE,
+            unlink.target=FALSE)
```

A new `helloRoxygen/man/helloRoxygen-package.Rd` should have been created with the contents of figure 1; and ‘R CMD check helloRoxygen’ should terminate successfully.

## 2 Example: Pseudoprimality

### 2.1 Package Description

```
----- pseudoprime-package.R -----
1  #' Tests pseudoprimality by Fermat's little theorem.
2  #'
3  #' \tabular{ll}{
4  #' Package: \tab pseudoprime\cr
5  #' Type: \tab Package\cr
6  #' Version: \tab 0.1\cr
7  #' Date: \tab 2008-08-24\cr
8  #' License: \tab GPL (>= 2)\cr
9  #' LazyLoad: \tab yes\cr
10 #' }
11 #'
12 #' Using the Fermat primality test, pseudoprime checks for primes
13 #' probabilistically; the test is fooled every time by Carmichael
14 #' numbers.
15 #'
16 #' \code{\link{is.pseudoprime}} checks a number \code{n} for
17 #' pseudoprimality, applying Fermat's test \code{times} times.
18 #'
19 #' @name pseudoprime-package
20 #' @aliases pseudoprime
21 #' @docType package
22 #' @title Tests pseudoprimality by Fermat's little theorem
23 #' @author Peter Danenberg \email{pcd@roxygen.org}
24 #' @references
25 #' \url{http://en.wikipedia.org/wiki/Fermat's_little_theorem}
26 #' @keywords package
27 #' @seealso \code{\link{is.pseudoprime}}
28 #' @examples
29 #' is.pseudoprime(13, 4)
30 roxygen()
```

Figure 2: Package description for `pseudoprime`

One could imagine, for instance, a less trivial package that actually does something; enter `pseudoprime`, a toy that tests for primes using Fermat's little theorem.<sup>1</sup>

A package description has been provided in figure 2; notice the `roxygen()` statement in line 30: each `Roxygen` description block must be followed by a

<sup>1</sup>[http://en.wikipedia.org/wiki/Fermat's\\_little\\_theorem](http://en.wikipedia.org/wiki/Fermat's_little_theorem)

statement, even header material that describes a file or package in lieu of a specific function. `roxygen()` is provided as a NOOP (null statement) to stand in for such cases.

The first sentence of any **Roxygen** block briefly describes its object; and may be followed directly by a **Roxygen** tag (fig. 1, line 2) or a more detailed description (fig. 2, line 3). The detailed description begins after the intervening blank line, and continues until the first **Roxygen** tag (fig. 2, line 19).

Each **Roxygen** tag begins with an asperand, like `@name`, `@author`, etc.; which means, incidentally, that all real asperands have to be escaped with a double-asperand, as in `\email{pcd@@roxygen.org}` (fig. 2, line 23).

Furthermore, although **Roxygen** tags replace many of the structural Rd elements such as `\title`, `\keyword`, etc.; stylistic Rd elements such as `\emph` and `\email` can be used freely within **Roxygen** tags. See “Writing R Extensions” for details. [?, §2.3 “Marking text”]

## 2.2 Fermat Test

```

1  #' Test an integer for primality with Fermat's little theorem.
2  #'
3  #' Fermat's little theorem states that if \eqn{n} is a prime
4  #' number and \eqn{a} is any positive integer less than \eqn{n},
5  #' then \eqn{a} raised to the \eqn{n}th power is congruent to
6  #' \eqn{a \ modulo \ n}{a modulo n}.
7  #'
8  #' @param n the integer to test for primality
9  #' @return Whether the integer passes the Fermat test
10 #'   for a randomized \eqn{0 < a < n}
11 #' @callGraphPrimitives
12 #' @note \code{fermat.test} doesn't work for integers above
13 #'   approximately fifteen because modulus loses precision.
14 #' @references
15 #' \url{http://en.wikipedia.org/wiki/Fermat's_little_theorem}
16 #' @author Peter Danenberg \email{pcd@@roxygen.org}
17 fermat.test <- function(n) {
18   a <- floor(runif(1, min=1, max=n))
19   a ^ n %% n == a
20 }

```

Figure 3: Roxygen example `fermat.R`

When documenting functions (fig. 3), every parameter must be documented with a `@param` tag (line 8); which consists of `@param <variable> <description>`. Similarly, the return value must be documented with `@return <description>` (lines 9-10).

Notice `@callGraphPrimitives` (line 11): it creates a call graph at the default depth similar to figure 4, including primitive functions; `@callGraph`, on the other hand, would exclude primitive functions.<sup>2</sup>

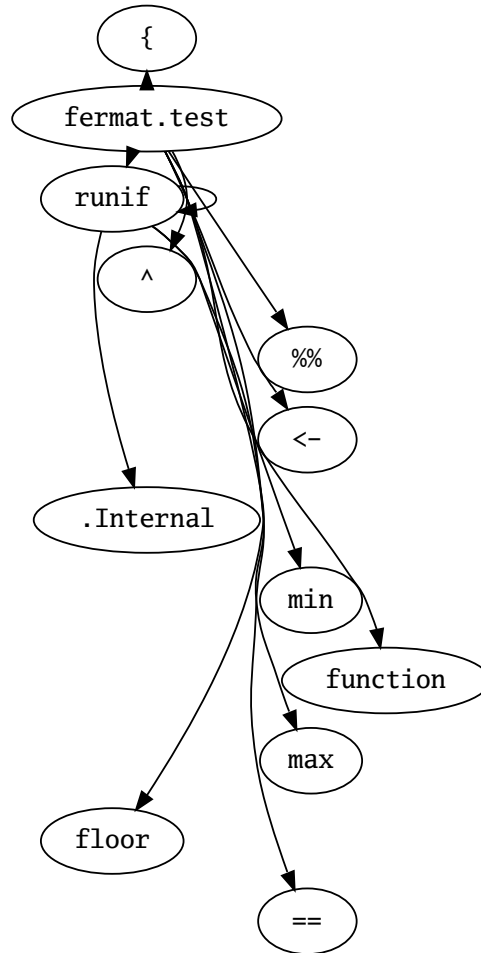


Figure 4: `fermat.test` call graph with primitives

---

<sup>2</sup>Note that this feature requires package `Rgraphviz`  $\geq 1.19.2$ , and as of `Roxygen` 0.1 – 1 the `callgraph` feature is not automatically executed; see help for `make.callgraph.roclet`.

## 2.3 Pseudoprime

```

1  #' @include fermat.R
2  roxygen()
3
4  #' Check an integer for pseudo-primality to an arbitrary
5  #' precision.
6  #'
7  #' A number is pseudo-prime if it is probably prime, the basis
8  #' of which is the probabilistic Fermat test; if it passes two
9  #' such tests, the chances are better than 3 out of 4 that
10 #' \eqn{n} is prime.
11 #'
12 #' @param n the integer to test for pseudoprimality.
13 #' @param times the number of Fermat tests to perform
14 #' @return Whether the number is pseudoprime
15 #' @export
16 #' @seealso \code{\link{fermat.test}}
17 #' @references Abelson, Hal; Jerry Sussman, and Julie Sussman.
18 #'   Structure and Interpretation of Computer Programs.
19 #'   Cambridge: MIT Press, 1984.
20 #' @author Peter Danenberg \email{pcd@@roxygen.org}
21 #' @examples
22 #' is.pseudoprime(13, 4) # TRUE most of the time
23 is.pseudoprime <- function(n, times) {
24   if (times == 0) TRUE
25   else if (fermat.test(n)) is.pseudoprime(n, times - 1)
26   else FALSE
27 }
```

Figure 5: Roxygen example `pseudoprime.R`

Notice the header in `pseudoprime.R` (fig. 5) terminated by `roxygen()`; `@include fermat.R` (line 1) signals that `fermat.R` should be loaded before `pseudoprime.R`. Roxygen will update `DESCRIPTION` accordingly.

`@export` (line 15) signifies that `is.pseudoprime` will be added to an export directive in `NAMESPACE`.

## 2.4 Running Roxygen

Running ‘R CMD roxygen -d pseudoprime’ populates `man` with Rd files, edits `DESCRIPTION` and `NAMESPACE`, and puts call graphs in `inst/doc`:

The `roxygenize` function is an alternative to ‘R CMD roxygen’; see the help page for details. Note that version 0.1-3 ‘R CMD roxygen’ is not available

anymore.

## References

R Development Core Team. *Writing R Extensions*. R Foundation for Statistical Computing, Vienna, Austria, 2008.