

Roxygen Vignette

Peter Danenberg <pcd@roxygen.org>

July 6, 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

1 Minimal Example: “Hello, Roxygen!” 1

2 Example: Pseudoprimality 3
   2.1 Package Description 3
   2.2 Fermat Test 4
   2.3 Pseudoprime 6
   2.4 Running Roxygen 6

1 Minimal Example: “Hello, Roxygen!”

`hello-roxygen.R`

```r
1  # A package to check Roxygen's sanity
2  # @name helloRoxygen-package
3  # @docType package
4  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:
> library(roxygen)
> package.skeleton('helloRoxygen',
>   code_files='hello-roxygen.R',
>   force=TRUE)
> # 'R CMD roxygen -d helloRoxygen' works, too.
> roxygenize('helloRoxygen',
>   roxygen.dir='helloRoxygen',
>   copy.package=FALSE,
>   unlink.target=FALSE)

Writing helloRoxygen-package to helloRoxygen/man/helloRoxygen-package.Rd
Writing namespace directives to helloRoxygen/NAMESPACE
Merging collate directive with helloRoxygen/DESCRIPTION to helloRoxygen/DESCRIPTION

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

```r
#' Tests pseudoprimality by Fermat's little theorem.
#'
#' @tabular{ll}{
#' Package: \tab pseudoprime\cr
#' Type: \tab Package\cr
#' Version: \tab 0.1\cr
#' Date: \tab 2008-08-24\cr
#' License: \tab GPL (>= 2)\cr
#' LazyLoad: \tab yes\cr
#' } 
#'
#'
#' Using the Fermat primality test, pseudoprime checks for primes
#' probabilistically; the test is fooled every time by Carmichael
#' numbers.
#'
#' \code{\link{is.pseudoprime}} checks a number \code{n} for
#' pseudoprimality, applying Fermat's test \code{times} times.
#'
#' @name pseudoprime-package
#' @aliases pseudoprime
#' @docType package
#' @title Tests pseudoprimality by Fermat's little theorem
#' @author Peter Danenberg \email{pcd@roxygen.org}
#' @references
#' \url{http://en.wikipedia.org/wiki/Fermat's_little_theorem}
#' @keywords package
#' @seealso \code{\link{is.pseudoprime}}
#' @examples
#' is.pseudoprime(13, 4)
#'
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.¹

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

¹`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.  
rroxygen() is provided as a NOOP (null statement) to stand in for such cases.

The first sentence of any R**oxygen** block briefly describes its object; and may be followed directly by a R**oxygen** 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 R**oxygen** tag (fig. 2, line 19).

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

Furthermore, although R**oxygen** 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 R**oxygen** tags. See “Writing R Extensions” for details. [R Development Core Team, 2008, §2.3 “Marking text”]

2.2 Fermat Test

```r
# Test an integer for primality with Fermat's little theorem.
#
# Fermat's little theorem states that if \( n \) is a prime
# number and \( a \) is any positive integer less than \( n \),
# then \( a \) raised to the \( n \)th power is congruent to
# \( a \mod n \).
#
# @param n the integer to test for primality
# @return Whether the integer passes the Fermat test
# @callGraphPrimitives
# @note \code{fermat.test} doesn't work for integers above
# approximately fifteen because modulus loses precision.
# @references
# \url{http://en.wikipedia.org/wiki/Fermat's_little_theorem}
# @author Peter Danenberg \email{pcd@@roxygen.org}
fermat.test <- function(n) {
  a <- floor(runif(1, min=1, max=n))
  a ^ n %% n == a
}
```

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.

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

```r
#' @include fermat.R
roxygen()

#' Check an integer for pseudo-primality to an arbitrary
#' precision.

#' A number is pseudo-prime if it is probably prime, the basis
#' of which is the probabilistic Fermat test; if it passes two
#' such tests, the chances are better than 3 out of 4 that
#' \( n \) is prime.

#' @param n the integer to test for pseudoprimality.
#' @param times the number of Fermat tests to perform
#' @return Whether the number is pseudoprime
#' @export

is.pseudoprime <- function(n, times) {
  if (times == 0) TRUE
  else if (fermat.test(n)) is.pseudoprime(n, times - 1)
  else FALSE
}
```

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`. The collate roclet 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:
Writing fermat.test to pseudoprime/man/fermat.test.Rd
Writing pseudoprime-package to pseudoprime/man/pseudoprime-package.Rd
Writing is.pseudoprime to pseudoprime/man/is.pseudoprime.Rd
Writing namespace directives to pseudoprime/NAMESPACE
Merging collate directive with pseudoprime/DESCRIPTION to pseudoprime/DESCRIPTION
Outputting call graph to 'pseudoprime/inst/doc/fermat.test-callgraph.pdf'

The roxygenize function is an alternative to ‘R CMD roxygen’; see the help
page for details.

References

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