Package ‘debugr’

July 30, 2018

Title Debug Tool to Watch Objects/Expressions While Running an R Script
Version 0.0.1
Maintainer Joachim Zuckarelli <joachim@zuckarelli.de>
Description Tool to print out the value of R objects/expressions while running an R script. Outputs can be made dependent on user-defined conditions/criteria. Debug messages only appear when a global option for debugging is set. This way, ‘debugr’ code can even remain in the debugged code for later use without any negative effects during normal runtime.

BugReports https://github.com/jsugarelli/debugr/issues

URL https://github.com/jsugarelli/debugr/
Depends R (>= 3.5.0)
License GPL-3
Encoding UTF-8
LazyData true
RoxygenNote 6.0.1
Imports utils, rprojroot, rstudioapi
Suggests knitr, rmarkdown
VignetteBuilder knitr
NeedsCompilation no
Author Joachim Zuckarelli [aut, cre]
Repository CRAN
Date/Publication 2018-07-30 11:50:03 UTC

R topics documented:

debugmode . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 2
dwatch . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 2

Index 6
**debugmode**  

*Switching debug mode on and off*

**Description**

The behavior of debugr’s main function, `dwatch`, depends on whether or not the `debugr` debug mode is activated or not. The debug mode is turned on and off by setting the global option `debugr.active` to `TRUE` and `FALSE`, respectively. This can be accomplished with the `debugr_switchOn` and `debugr_switchOff` functions, or manually by running `options(debugr.active = TRUE)`.

**Usage**

- `debugr_switchOn()`  
- `debugr_switchOff()`  
- `debugr_isActive()`

**Details**

When `debugr.active = TRUE` the debug mode is enabled and `dwatch` produces debugging outputs to the console (or to a file). In contrast, when the debug mode is disabled, `dwatch` remains "silent" and no output whatsoever will be shown.

**Functions**

- `debugr_switchOn`: Switches on the global option for debugging  
- `debugr_switchOff`: Switches off the global option for debugging  
- `debugr_isActive`: Check if debug mode is currently active or not

---

**dwatch**  

*Printing debug outputs during runtime*

**Description**

Prints a debug output to the console or to a file. A debug output can consist of a static text message, the values of one or more objects (potentially transformed by applying some functions) or the value of one or multiple (more complex) R expressions. Whether or not a debug message is displayed can be made dependent on the evaluation of a criterion phrased as an R expression. Generally, debug messages are only shown if the debug mode is activated. The debug mode is activated and deactivated with `debugr_switchOn` and `debugr_switchOff`, respectively, which change the logical `debugr.active` value in the global options. Since debug messages are only displayed in debug mode, the `dwatch` function calls can even remain in the original code as they remain silent and won’t have any effect until the debug mode is switched on again.
**dwatch**

**Usage**

```r
dwatch(crit = "", objs = NULL, funs = NULL, args = NULL,
show.all = FALSE, expr = NULL, msg = "", halt = FALSE,
unique.id = "", suppress.source = FALSE, show.frame = TRUE,
filename = "")
```

**Arguments**

- **crit**: A string containing an expression that determines if any debug outputs shall be displayed at all. Only, if `crit` evaluates to `TRUE`, a debug output will be shown.
- **objs**: A vector of object names (as strings). The values of these objects will be displayed in the debug output.
- **funs**: A vector of function names (as strings) that shall be applied to the objects in `objs`, one function per object. `funs` must have the same length as `objs`. If no function shall be applied to an object, the respective element in the `funs` vector must be `NULL`. The functions in `funs` must undertake the task of printing the object.
- **args**: A list of vectors containing additional arguments for the functions in `funs`. It is assumed that the first argument of each function in `funs` is the respective object from `objs`. Additional arguments can then be supplied with `args`. The `args` list must have the same number of elements as `funs`. If a function does not receive any additional arguments, the respective element in the `args` list must be `NULL`. Each element of `args` is a vector of named elements. The element name is the name of the additions argument to the respective `funs` function, the elements value is the argument's value.
- **show.all**: Prints all objects from the (calling) environment. If set to `TRUE`, `objs` is ignored and all objects in the environment (with the exception of functions) are included in the debug output.
- **expr**: A vector of strings containing expressions to be evaluated and displayed in the debug output. This output comes on top of any `msg` or `objs` output.
- **msg**: A string containing a general message to be displayed.
- **halt**: If `TRUE`, the execution of the debugged R script is stopped after printing the output.
- **unique.id**: A unique string ID that can be chosen by the user. This ID is displayed in the debug output and is used to identify the code section that contains the `dwatch` call. By default, when a debug output is displayed, `dwatch` tries to show an extract from the code that surrounds the `dwatch` call (this feature can be turned off by setting `suppress.source` to `TRUE`).
- **suppress.source**: If `TRUE` (default), `dwatch` tries to find the code section that includes the `dwatch` call and displays it as part of the debug output. Requires `unique.id` to be set.
- **show.frame**: If `TRUE` (default), a frame is displayed at the top and the bottom of the debug output.
- **filename**: If a filename is provided, all debug message are only printed to the file and not shown on the R console.
See Also
debugr_switchOn, debugr_switchOff, debugr_isActive

Examples

library(debugr)

# --- A simple example to print the value of an object
myfunction <- function(x) {
  justastr = "Not much information here"
  z <- 1
  for(i in 1:x) {
    # This call can remain in your code; it is only activated when
    # the debug mode is switched on
    dwatch(crit = "z > 40000",objs = c("z"))
    z <- z * i
  }  
  invisible(z)
}

# Turn debug mode on
debugr_switchon()

# Call function for debugging
myfunction(10)

# --- Applying a function to the object that is printed
myfunction <- function(x) {
  justastr = "Not much information here"
  z <- 1
  for(i in 1:x) {
    dwatch(crit = "z > 40000",objs = c("z"),funs = c("format"),
      arge = as.list(c(big.mark = "","\"")))
    z <- z * i
  }  
  invisible(z)
}

myfunction(10)

# --- Same thing, this time with a expression
myfunction <- function(x) {
  justastr = "Not much information here"
  z <- 1
for(i in 1:x) {
    dwatch(crit = "z > 40000", expr=c("format(z, big.mark = ",\")"))
    z <- z * i
}
invisible(z)

myfunction(10)
Index

ddebugmode, 2
ddebugr_isActive (debugmode), 2
ddebugr_switchOff, 2
ddebugr_switchOff (debugmode), 2
ddebugr_switchOn, 2
ddebugr_switchOn (debugmode), 2
ddwatch, 2, 2