Package ‘idbg’

February 20, 2015

Type Package

Title R debugger

Version 1.0

Date 2012-01-16

Author Ronen Kimchi

Maintainer Ronen Kimchi <mitzpaz@gmail.com>

Description An interactive R debugger

License GPL-2

LazyLoad yes

Repository CRAN

Repository/R-Forge/Project idbg

Repository/R-Forge/Revision 15

Date/Publication 2012-02-11 22:18:40

NeedsCompilation no

R topics documented:

`idbg-package` *idbg*

Description

An interactive debugger for R

Details

Package:	idbg
Type:	Package
Version:	1.0
Date:	2012-01-16
License:	GPL-2
LazyLoad:	yes

use idbg.bp to set a breakpoint in the debugged function.
Run your code and the debugger will stop at the debugged function

The following debugger options are available:
h - help. print this message
n - next. Empty line is the same as 'n'
s - step into
o - step out
c - continue
q - quit
b - print breakpoints
b FALSE - clear all breakpoints
b <func_name> [FALSE] - set/unset a breakpoint in first line of function
b <line_number> [FALSE] - set/unset a breakpoint in current function
b <func_name> <line_number> [FALSE] - set/unset a breakpoint in function at at line_number
w - print the stack
u - go up the stack
d - go down the stack
l [nlines] - print nlines of source before and after current position
l [nlines_back] [nlines_forward] - print source around current position
x expr - execute expression. Any expression that doesn't match the above options will also be executed

Author(s)

Ronen Kimchi

Maintainer: mitzpaz@gmail.com

See Also

`idbg.bp`

`idbg.bp` *idbg.bp*

Description

Prepare a function for debugging and set a breakpoint

Usage

```
idbg.bp(func_name, line_number = NA, condition = TRUE)
```

Arguments

`func_name`	name for function. Must be of type character
`line_number`	line number where to set the breakpoint
`condition`	a condition to evaluate for conditional breakpoint

Value

Logical. TRUE if the breakpoint was set, FALSE if it wasn't

Author(s)

Ronen Kimchi

See Also

debug undebug

Examples

```
bar <- function(a)
{
  if (a < 0)
    cat("a < 0\n")
  else
    cat("a >= 0\n")
  return(a)
}
foo <- function(x,y)
{
  tmp <- bar(x - y)
  return(tmp)
}

## Not run: idbg.bp("foo")

# call foo to enter the debugger
# in the debugger type h to see the help
# use n or enter to step over
# use s to step into bar
## Not run: foo(2,3)
```

`idbg.source` *idbg.source*

Description

Source R files and restore breakpoints

Usage

```
idbg.source(fname)
```

Arguments

`fname`	R file name to source

Value

None (invisible NULL).

Author(s)

Ronen Kimchi

See Also

`source`

Examples

```
## Not run: idbg.source("my_debugged_R_file.R")
```

Index