Package 'echo'

July 22, 2025

Title Echo Code Evaluations

Version 0.1.0

Maintainer Jordan Mark Barbone <jmbarbone@gmail.com>

Description Capture code evaluations and script executions by expressions, outputs, and condition calls for logging.

License MIT + file LICENSE

Depends R (>= 3.6)

Imports utils

Suggests covr, spelling, testthat (>= 3.0.0)

Encoding UTF-8

RoxygenNote 7.2.3

URL https://github.com/jmbarbone/echo, https://jmbarbone.github.io/echo/

BugReports https://github.com/jmbarbone/echo/issues

Language en-US

Config/testthat/edition 3

NeedsCompilation no

Author Jordan Mark Barbone [aut, cph, cre] (ORCID: <https://orcid.org/0000-0001-9788-3628>)

Repository CRAN

Date/Publication 2023-05-25 20:00:12 UTC

Contents

echo *Echo*

Description

Echo expression or a file

Usage

```
echo(
  expr,
  log = echo_get_log(),
  msg = echo_get_msg(),
  level = echo_get_level(),
  file = NULL
)
```

Arguments

`expr`	Expression to evaluate; should be written with curly braces (see examples)
`log`	A connection or file name for outputs; defaults to `stdout()`
`msg`	Logical, if `FALSE` does not output a message; defaults to `TRUE`
`level`	Sets the echo level (see details); defaults to `0L`
`file`	File path to evaluate (like `base::source()`). If `file` is not `NULL`, then `expr` must be missing‘

Details

Levels of output can be controlled with `level`:

`0 EXP`: logs expressions that were evaluated

`1 OUT`: logs outputs from expressions

`2 MSG`: logs messages

`3 WRN`: logs warnings

`4 ERR`: logs errors

When set, all outputs at the `level` or below are run. Errors are always logged as they will interrupt and stop the program.

Timestamps are printed in UTC by default. To control this, set the option value, such as `options(echo.timezone = "EST")`.

Value

Nothing, called for side-effects

Examples

```
# make sure to use braces for expr
echo(letters, level = 0)   # bad
echo({letters}, level = 0) # good

try(echo(
  expr = {
    print(1 + 1)
    Sys.sleep(2)
    head(mtcars)
    message(1)
    warning(2)
    stop(3)
  },
  level = 0
))


# Parse lines in a file instead
try(echo(file = system.file("example-script.R", package = "echo")))

# Note that
x <- c("example for", "writing lines")
echo({
  x
  print(x)
  writeLines(x)
}, level = 0)
```

Index