Package ‘evaluate’

November 1, 2023

Type Package
Title Parsing and Evaluation Tools that Provide More Details than the Default
Version 0.23
Description Parsing and evaluation tools that make it easy to recreate the command line behaviour of R.
License MIT + file LICENSE
URL https://github.com/r-lib/evaluate
BugReports https://github.com/r-lib/evaluate/issues
Depends R (>= 3.0.2)
Imports methods
Suggests covr, ggplot2, lattice, rlang, testthat (>= 3.0.0), withr
RoxygenNote 7.2.3
Encoding UTF-8
Config/testthat/edition 3
NeedsCompilation no
Author Hadley Wickham [aut], Yihui Xie [aut, cre] (<https://orcid.org/0000-0003-0645-5666>), Michael Lawrence [ctb], Thomas Kluyver [ctb], Jeroen Ooms [ctb], Barret Schloerke [ctb], Adam Ryczkowski [ctb], Hiroaki Yutani [ctb], Michel Lang [ctb], Karolis Koncevičius [ctb], Posit Software, PBC [cph, fnd]
Maintainer Yihui Xie <xie@yihui.name>
Repository CRAN
Date/Publication 2023-11-01 14:10:02 UTC
**R topics documented:**

- evaluate .............................................................. 2
- flush_console ......................................................... 3
- new_output_handler ................................................. 4
- parse_all ............................................................. 5
- replay ................................................................. 5

---

**Index**

| evaluate | Evaluate input and return all details of evaluation. |

---

**Description**

Compare to `eval()`, `evaluate` captures all of the information necessary to recreate the output as if you had copied and pasted the code into a R terminal. It captures messages, warnings, errors and output, all correctly interleaved in the order in which they occurred. It stores the final result, whether or not it should be visible, and the contents of the current graphics device.

**Usage**

```r
evaluate(
  input,
  envir = parent.frame(),
  enclos = NULL,
  debug = FALSE,
  stop_on_error = 0L,
  keep_warning = TRUE,
  keep_message = TRUE,
  log_echo = FALSE,
  log_warning = FALSE,
  new_device = TRUE,
  output_handler = default_output_handler,
  filename = NULL,
  include_timing = FALSE
)
```

**Arguments**

- `input` input object to be parsed and evaluated. May be a string, file connection or function. Passed on to `parse_all()`.
- `envir` environment in which to evaluate expressions.
- `enclos` when `envir` is a list or data frame, this is treated as the parent environment to `envir`.
- `debug` if TRUE, displays information useful for debugging, including all output that `evaluate` captures.
flush_console

An emulation of flush.console() in evaluate()

Description

When evaluate() is evaluating code, the text output is diverted into an internal connection, and there is no way to flush that connection. This function provides a way to "flush" the connection so that any text output can be immediately written out, and more importantly, the text handler (specified in the output_handler argument of evaluate()) will be called, which makes it possible for users to know it when the code produces text output using the handler.

Usage

flush_console()

Note

This function is supposed to be called inside evaluate() (e.g. either a direct evaluate() call or in knitr code chunks).
**new_output_handler**  
Custom output handlers.

**Description**

An output_handler handles the results of `evaluate()`, including the values, graphics, conditions. Each type of output is handled by a particular function in the handler object.

**Usage**

```r
custom_output_handler(
  source = identity,
  text = identity,
  graphics = identity,
  message = identity,
  warning = identity,
  error = identity,
  value = render,
  calling_handlers = list()
)
```

**Arguments**

- **source**: Function to handle the echoed source code under evaluation.
- **text**: Function to handle any textual console output.
- **graphics**: Function to handle graphics, as returned by `recordPlot()`.
- **message**: Function to handle `message()` output.
- **warning**: Function to handle `warning()` output.
- **error**: Function to handle `stop()` output.
- **value**: Function to handle the values returned from evaluation. If it only has one argument, only visible values are handled; if it has more arguments, the second argument indicates whether the value is visible.
- **calling_handlers**: List of calling handlers. These handlers have precedence over the exiting handler installed by `evaluate()` when `stop_on_error` is set to 0.

**Details**

The handler functions should accept an output object as their first argument. The return value of the handlers is ignored, except in the case of the value handler, where a visible return value is saved in the output list.

Calling the constructor with no arguments results in the default handler, which mimics the behavior of the console by printing visible values.

Note that recursion is common: for example, if value does any printing, then the text or graphics handlers may be called.
parse_all

Value

A new output_handler object

Description

Works very similarly to parse, but also keeps original formatting and comments.

Usage

parse_all(x, filename = NULL, allow_error = FALSE)

Arguments

x object to parse. Can be a string, a file connection, or a function. If a connection, will be opened and closed only if it was closed initially.
filename string overriding the file name
allow_error whether to allow syntax errors in x

Value

A data.frame with columns src, the source code, and expr. If there are syntax errors in x and allow_error = TRUE, the data frame has an attribute PARSE_ERROR that stores the error object.

replay

Replay a list of evaluated results.

Description

Replay a list of evaluated results, as if you’d run them in an R terminal.

Usage

replay(x)

Arguments

x result from evaluate()
Examples

samples <- system.file("tests", "testthat", package = "evaluate")
if (file_test("-d", samples)) {
  replay(evaluate(file(file.path(samples, "order.R"))))
  replay(evaluate(file(file.path(samples, "plot.R"))))
  replay(evaluate(file(file.path(samples, "data.R"))))
}
Index

base::srcfile(), 3

calling handlers, 4

eval(), 2
evaluate, 2
evaluate(), 3–5

flush_console, 3

message(), 4

new_output_handler, 4

output_handler (new_output_handler), 4
output_handler(), 3

parse_all, 5
parse_all(), 2

recordPlot(), 4
replay, 5

stop(), 4

warning(), 4