Package: vhsR (via r-universe)

June 4, 2026

Title Animated Terminal Demos for R via 'vhs'

Version 0.0.1

Description Wraps the 'vhs' command-line tool from Charm (<https://github.com/charmbracelet/vhs>) so that animated GIF demos of R code can be produced from a single R expression.

License MIT + file LICENSE

Encoding UTF-8

Roxygen list(markdown = TRUE)

Depends R (>= 4.4)

Imports cli, processx, withr

Suggests htmltools, knitr, rmarkdown, testthat (>= 3.0.0)

VignetteBuilder knitr

Config/testthat/edition 3

URL https://schochastics.github.io/vhsR/

Config/roxygen2/version 8.0.0

Repository https://schochastics.r-universe.dev

Date/Publication 2026-05-05 19:06:13 UTC

RemoteUrl https://github.com/schochastics/vhsR

RemoteRef HEAD

RemoteSha 57dbd456a8a562346a2e046de20b0d7c41f350ac

Contents

`record_demo` *Record a GIF demo of an R expression*

Description

Captures the unevaluated R expression `expr`, types it line-by-line into a freshly-spawned R session inside a `vhs` recording, and writes the result to `output`. The R front-end is chosen by `backend` (see below). Setup is a one-liner: if the required binaries are missing, the function points to `vhsr_install()`.

Usage

```
record_demo(
  expr,
  output = "demo.gif",
  ...,
  backend = c("auto", "arf", "radian", "R"),
  tape = NULL,
  quiet = FALSE
)
```

Arguments

`expr`	An R expression (typically a `{ ... }` block). Captured unevaluated; lines are typed verbatim into the REPL.
`output`	Path to the output file. The extension determines the format (`.gif`, `.mp4`, `.webm`).
`...`	Pacing and styling arguments forwarded to `tape_options()`. Pass any of `width`, `height`, `font_size`, `theme`, `typing_speed`, `playback_speed`, `line_pause`, `start_pause`, `end_pause`, `paragraph_pause`, `typing_speed_jitter` by name. Unknown names error.
`backend`	Which interactive R front-end to record. `"auto"` (the default) prefers `"arf"` (a Rust-based modern R console with tree-sitter syntax highlighting and zero runtime dependencies — see `https://github.com/eitsupi/arf`), falls back to `"radian"` if found, otherwise plain `"R"`.
`tape`	Optional. A raw `.tape` script as a string (or path to a `.tape` file). When supplied, `expr` and the styling args are ignored and the tape is run verbatim.
`quiet`	If `TRUE`, suppress vhs stdout/stderr.

Value

Invisibly returns the path to the output file.

See Also

tape_options() for the full pacing/styling argument list.

Examples

```
## Not run:
record_demo({
  x <- 1:5
  mean(x)
  head(cars, 3)
}, output = "demo.gif")

## End(Not run)
```

`record_demo_file` *Record a GIF demo from a script file*

Description

Reads `path` as R source, validates it parses, and records each line typed into the recorded REPL. Equivalent to inlining the file contents in `record_demo()` but accepts a path instead of an expression.

Usage

```
record_demo_file(
  path,
  output = "demo.gif",
  ...,
  backend = c("auto", "arf", "radian", "R"),
  quiet = FALSE
)
```

Arguments

`path`	Path to an R script file.
`output`	Path to the output file. The extension determines the format (`.gif`, `.mp4`, `.webm`).
`...`	Pacing and styling arguments forwarded to `tape_options()`. Pass any of `width`, `height`, `font_size`, `theme`, `typing_speed`, `playback_speed`, `line_pause`, `start_pause`, `end_pause`, `paragraph_pause`, `typing_speed_jitter` by name. Unknown names error.
`backend`	Which interactive R front-end to record. `"auto"` (the default) prefers `"arf"` (a Rust-based modern R console with tree-sitter syntax highlighting and zero runtime dependencies — see https://github.com/eitsupi/arf), falls back to `"radian"` if found, otherwise plain `"R"`.
`quiet`	If `TRUE`, suppress vhs stdout/stderr.

Value

Invisibly returns `output`.

Examples

```
## Not run:
record_demo_file("script.R", output = "demo.gif")

## End(Not run)
```

`record_demo_screenshot` *Capture a single PNG frame from a recorded R session*

Description

Records the same R expression that `record_demo()` would, but instead of (or in addition to) the animated output, writes a single PNG frame using vhs's `Screenshot` directive.

Usage

```
record_demo_screenshot(
  expr,
  output = "demo.png",
  at = NULL,
  ...,
  backend = c("auto", "arf", "radian", "R"),
  quiet = FALSE
)
```

Arguments

`expr`	An R expression. See `record_demo()`.
`output`	Path to the output PNG. Must end in `.png`.
`at`	Where to take the screenshot. `NULL` (default) → after the last typed line. A string `"after:N"` (1-indexed) → after the Nth line.
`...`	Pacing and styling arguments forwarded to `tape_options()`. Pass any of `width`, `height`, `font_size`, `theme`, `typing_speed`, `playback_speed`, `line_pause`, `start_pause`, `end_pause`, `paragraph_pause`, `typing_speed_jitter` by name. Unknown names error.
`backend`	Which interactive R front-end to record. `"auto"` (the default) prefers `"arf"` (a Rust-based modern R console with tree-sitter syntax highlighting and zero runtime dependencies — see `https://github.com/eitsupi/arf`), falls back to `"radian"` if found, otherwise plain `"R"`.
`quiet`	If `TRUE`, suppress vhs stdout/stderr.

Value

Invisibly returns `output`.

Examples

```
## Not run:
record_demo_screenshot(
  { x <- 1:5; mean(x) },
  output = "demo.png"
)

## End(Not run)
```

---

vhsr_cache_dir *Locate the per-user vhsR binary cache*

Description

The directory under `tools::R_user_dir()` where `vhsr_install()` places downloaded `vhs`, `ttyd`, and `ffmpeg` binaries. The path is platform-and-architecture-specific so a single `$HOME` can serve multiple machines (e.g. via NFS).

Usage

```
vhsr_cache_dir()
```

Value

A length-1 character path. The directory is not created on read.

---

vhsr_check *Check whether all required binaries are available*

Description

Silent variant: returns `TRUE` when `vhs`, `ttyd`, and `ffmpeg` can all be located via `vhsr_vhs_path()` / `vhsr_ttyd_path()` / `vhsr_ffmpeg_path()`, `FALSE` otherwise. Used at the top of `record_demo()` to bail with an actionable error.

Usage

```
vhsr_check()
```

Value

`TRUE` or `FALSE`.

vhsr_doctor *Report on the state of vhs and its dependencies*

Description

Prints two sections. The required binaries (`vhs`, `ttyd`, `ffmpeg`) are reported one per line with name, version, where they were found (cache or system), and full path. The optional highlighting tools (`arf`, `radian`) are reported below with install hints when missing. If any required binary is missing, a one-line `vhsr_install()` call is suggested at the end.

Usage

```
vhsr_doctor()
```

Value

Invisibly returns the result of `vhsr_check()`.

vhsr_install *Install vhs and its runtime dependencies*

Description

Downloads `vhs`, `ttyd`, and `ffmpeg` into the per-user cache returned by `vhsr_cache_dir()`. Where binaries cannot be auto-downloaded for a given platform (notably `ttyd` and `ffmpeg` on macOS), the function aborts with a one-line `brew install` instruction; system-installed binaries on `PATH` are picked up automatically by `vhsr_doctor()` afterwards.

Usage

```
vhsr_install(
  tools = c("vhs", "ttyd", "ffmpeg"),
  force = FALSE,
  versions = list()
)
```

Arguments

`tools`	Character vector of tools to install. Defaults to all three.
`force`	If `TRUE`, re-download even if the binary already exists.
`versions`	A named list overriding the default version of any tool, for example `list(vhs = "0.10.0")`.

Value

Invisibly returns the cache directory path.

vhsr_paths *Resolve paths to vhs and its dependencies*

Description

Each function returns the first match in this order: option (`vhsR.<tool>_path`) -> env var (`VHSR_<TOOL>`) -> per-user cache (see `vhsr_cache_dir()`) -> system `PATH`. Returns `""` when nothing matches.

Usage

```
vhsr_vhs_path()

vhsr_ttyd_path()

vhsr_ffmpeg_path()

vhsr_radian_path()

vhsr_arf_path()
```

Value

A length-1 character path, or `""` if not found.

vhsr_run_tape *Run a hand-written tape script*

Description

Escape hatch for users who want full control over the vhs script. Pass either a path to a `.tape` file or the tape contents as a string.

Usage

```
vhsr_run_tape(tape, quiet = FALSE)
```

Arguments

`tape`	Path to a `.tape` file, or a string containing tape syntax.
`quiet`	If `TRUE`, suppress vhs stdout/stderr.

Value

Invisibly returns the `processx::run()` result.

vhsr_widget *Embed a vhsR recording as an HTML widget*

Description

Returns an `htmltools::tagList()` containing the recorded GIF / video, suitable for embedding in an Rmarkdown chunk (`results = "asis"`) or a Shiny UI. Auto-detects the format from `file`'s extension: `.gif` becomes a styled `<img>`, `.mp4` / `.webm` become a `<video controls loop muted playsinline>`.

Usage

```
vhsr_widget(file, width = NULL, height = NULL)
```

Arguments

`file`	Path to a `.gif`, `.mp4`, or `.webm` file produced by `record_demo()` / `vhsr_run_tape()`.
`width, height`	Optional CSS dimensions. Numeric values are treated as pixels (e.g. `800` → `"800px"`); strings pass through verbatim (e.g. `"100%"`).

Value

An `htmltools::tagList`.

Examples

```
## Not run:
record_demo({ x <- 1:5; mean(x) }, output = "demo.gif")
vhsr_widget("demo.gif")

## End(Not run)
```

Index