Package 'asciicast'

July 22, 2025

Title Create 'Ascii' Screen Casts from R Scripts

Version 2.3.1

Description Record 'asciicast' screen casts from R scripts. Convert them to animated SVG images, to be used in 'README' files, or blog posts. Includes 'asciinema-player' as an 'HTML' widget, and an 'asciicast' 'knitr' engine, to embed 'ascii' screen casts in 'Rmarkdown' documents.

License MIT + file LICENSE

URL https://asciicast.r-lib.org/, https://github.com/r-lib/asciicast

BugReports https://github.com/r-lib/asciicast/issues

Imports cli (>= 3.3.0.9000), curl, jsonlite, magick (>= 2.2.9002), processx (>= 3.7.0), tibble, utils, V8, withr

Suggests callr, covr, cpp11, decor, htmlwidgets, knitr, mockery, rmarkdown, rstudioapi, testthat (>= 3.2.0)

LinkingTo processx

Config/Needs/website r-lib/downlit, tidyverse/tidytemplate

Config/testthat/edition 3

Encoding UTF-8

RoxygenNote 7.2.3

NeedsCompilation yes

Author Gábor Csárdi [aut, cre],
Romain Francois [aut],
Mario Nebl [aut] (https://github.com/marionebl/svg-term author),
Marcin Kulik [aut] (https://github.com/asciinema/asciinema-player author),
Posit Software, PBC [cph, fnd]

Maintainer Gábor Csárdi <csardi.gabor@gmail.com>

Repository CRAN

Date/Publication 2024-01-18 06:30:02 UTC

Contents

`asciicast-package` *asciicast parameters*

Description

You can set asciicast parameters in the header of the recorded R script. The header is in DCF format (see `read.dcf()`), but all lines are prefixed with `#'` comments.

Details

The DCF header may specify arbitrary parameters. We list here the parameters that are interpreted by the asciicast functions.

Recording parameters:

- `allow_errors`: Whether to cast errors properly. If this is set to `TRUE`, then asciicast overwrites the `"error"` option. Only change this if you know what you are doing.
- `cols`: Width of the terminal, in number of characters.
- `empty_wait`: How long to wait for empty lines in the script file, in seconds.
- `end_wait`: Delay at the very end, in seconds.
- `env`: Environment variables to include in the case JSON file. Defaults to `list(TERM = "xterm-256color", SHELL = "/bin/zsh")`.
- `idle_time_limit`: Time limit for the cast not printing anything, in seconds. By default there is no limit.

- `record_env`: Environment variables to set for the R subprocess.
- `rows`: Height of the terminal, in number of characters.
- `start_wait`: Delay at the beginning, in seconds.
- `timeout`: Idle timeout, in seconds If the R subprocess running the recording does not answer within this limit, it is killed and the recording stops. Update this for slow running code, that produces no output as it runs.
- `timestamp`: Time stamp of the recording, defaults to `Sys.time()`, this is included in the cast JSON file.
- `title`: Title of the cast, this is included in the cast JSON file.
- `typing_speed`: Average typing speed, per keypress, in seconds.

Asciinema player parameters:

- `author`: Author, displayed in the titlebar in fullscreen mode.
- `author_img_url`: URL of the author's image, displayed in the titlebar in fullscreen mode.
- `author_url`: URL of the author's homepage/profile. Author name (author above) is linked to this URL.
- `autoplay`: Whether to start playing the cast automatically.
- `cols`: Width of the terminal, in number of characters.
- `font_size`: Size of terminal font. Possible values: small, medium, big, any css `font-size` value (e.g. 15px).
- `idle_time_limit`: Time limit for the cast not printing anything, in seconds. By default there is no limit.
- `loop`: Whether to loop the playback.
- `poster_frame`: Which frame to use (in seconds) as the preview picture.
- `poster_text`: Text to use as the preview picture. Defaults to the title.
- `rows`: Height of the terminal, in number of characters.
- `speed`: Whether to play slower or faster. 1 is normal speed.
- `start_at`: Where to start the playback from, in seconds.
- `theme`: Theme to use, currently it has to be a string, one of `"asciinema"`, `"tango"`, `"solarized-dark"`, `"solarized-light"`, `"monokai"`. The first one is the default.
- `title`: Title of the cast.

Parameters for SVG files:

- `at`: Timestamp of single frame to render, in seconds.
- `cols`: Width of the terminal, in number of characters.
- `cursor`: Enable cursor rendering.
- `end_at`: Upper range of timeline to render in seconds.
- `padding`: Distance between text and image bounds.
- `padding_x`: Distance between text and image bounds on x axis.
- `padding_y`: Distance between text and image bounds on y axis.

- rows: Height of the terminal, in number of characters.
- start_at: Where to start the playback from, in seconds.
- window: Render with window decorations.
- theme: Theme to use, currently it has to be a string referring to a build-in theme, or a named list of theme properties, see default_theme(). The built-in themes are "asciinema", "tango", "solarized-dark", "solarized-light", "seti", "monokai", "github-light", "github-dark", "pkgdown" ,"readme". "readme" is a special theme the switches between light and dark mode automatically in README.md files on GitHub.

See Also

Other asciicast functions: asciicast_start_process(), read_cast(), record(), write_json()

---

asciicast_options *Default options to set in the asciicast subprocess.*

Description

Default options to set in the asciicast subprocess.

Usage

```
asciicast_options()
```

Value

Named list.

Examples

```
asciicast_options()
```

---

asciicast_start_process *Start an asciicast background process*

Description

This is for expert use, if you want to run multiple recordings in the same process.

Usage

```
asciicast_start_process(
  startup = NULL,
  timeout = 10,
  record_env = NULL,
  interactive = TRUE,
  locales = get_locales(),
  options = NULL,
  show_output = FALSE
)
```

Arguments

startup	Quoted language object to run in the subprocess before starting the recording.
timeout	Idle timeout, in seconds If the R subprocess running the recording does not answer within this limit, it is killed and the recording stops.
record_env	Environment variables to set for the R subprocess.
interactive	Whether to run R in interactive mode. Note that in interactive mode R might ask for terminal input.
locales	Locales to set in the asciicast subprocess. Defaults to the current locales in the main R process. Specify a named character vector here to override some of the defaults. See also get_locales().
options	Options to set in the subprocess, a named list. They are deparsed to code, and then the code setting them is executed in the subprocess. See asciicast_options() for the defaults. Supply a named list here to override the defaults or set additionsl ones. Passing large and/or complicated options here might not work, or might be slow.
show_output	Whether to show the output of the subprocess in real time.

Value

The R process, a processx::process object.

See Also

Other asciicast functions: asciicast-package, read_cast(), record(), write_json()

Examples

```
# Use the same R process to record multiple casts
process <- asciicast_start_process()
script1 <- "a <- runif(10)\n"
script2 <- "a\n"
cast1 <- record(textConnection(script1), process = process)
cast2 <- record(textConnection(script2), process = process)
cast1
cast2
```

`asciinema_player` *asciinema player HTML widget*

Description

You can use this widget in Rmd files or Shiny applications, the same way as other HTML widgets.

Usage

```
asciinema_player(
  cast,
  start_at = 0,
  rows = NULL,
  cols = NULL,
  autoplay = NULL,
  loop = NULL,
  speed = NULL,
  title = NULL,
  author = NULL,
  author_url = NULL,
  author_img_url = NULL,
  poster_text = NULL,
  poster_frame = NULL,
  font_size = NULL,
  theme = NULL,
  idle_time_limit = NULL,
  html_height = NULL,
  html_width = NULL,
  element_id = NULL
)
```

Arguments

`cast`	`asciicast` object.
`start_at`	Where to start the playback from, in seconds.
`rows`	Number of rows, defaults to the number of rows in the recording, or 24 if not specified in the cast.
`cols`	Number of columns, defaults to the number columns in the recording, or 80 if not specified in the cast.
`autoplay`	Whether to start playing the cast automatically.
`loop`	Whether to loop the playback.
`speed`	Whether to play slower or faster. 1 is normal speed.
`title`	If specified, it overrides the title in the recording.
`author`	Author, displayed in the titlebar in fullscreen mode.

author_url	URL of the author's homepage/profile. Author name (author above) is linked to this URL.
author_img_url	URL of the author's image, displayed in the titlebar in fullscreen mode.
poster_text	if not NULL, used as the text of the poster (preview).
poster_frame	Which frame to use for the preview. A number means seconds. Defaults to the last frame. This is only used if poster_text is NULL.
font_size	Size of terminal font. Possible values: small, medium, big, any css font-size value (e.g. 15px).
theme	Theme.
idle_time_limit	Time limit for the cast not printing anything, in seconds. By default there is no limit.
html_height	HTML height of the widget.
html_width	HTML width of the widget.
element_id	HTML id of the widget's element. If NULL, then the id is generated randomly.

Examples

```
cast <- read_cast(system.file("examples", "hello.cast", package = "asciicast"))
asciinema_player(cast)
```

---

clear_screen *Merge multiple ASCII casts into one*

Description

The new cast will inherit its options (screen size, etc.) from the first cast in the argument list. The options of the rest of the casts are ignored.

Usage

```
clear_screen()

pause(secs)

merge_casts(...)
```

Arguments

secs	Number of seconds to wait.
...	Ascii casts to merge or merge commands. Merge commands provide a way to insert pause, clear the screen, etc., between casts.

Details

`pause()` inserts a pause of the specified seconds between the casts.

`clear_screen()` clears the screen between two casts.

Value

An `asciicast` object.

Examples

```
# merge two casts, with a pause, and clear screen between them
cast1 <- read_cast(system.file("examples", "hello.cast", package = "asciicast"))
cast2 <- read_cast(system.file("examples", "dplyr.cast", package = "asciicast"))
cast <- merge_casts(cast1, pause(3), clear_screen(), cast2)
play(cast)
```

---

default_theme *The default asciicast theme*

Description

Currently only used for `write_svg()`

Usage

```
default_theme()
```

Value

A named list.

See Also

Other SVG functions: `play()`, `write_svg()`

Examples

```
cast <- read_cast(system.file("examples", "hello.cast", package = "asciicast"))
svg_file <- tempfile(fileext = ".svg")
mytheme <- modifyList(default_theme(), list(cursor = c(255, 0, 0)))
write_svg(cast, svg_file, theme = mytheme)
```

`expect_snapshot_r_process` *testthat snapshot test with asciicast*

Description

This function is very similar to `testthat::expect_snapshot_output()`, but it runs the code in an asciicast subprocess, using `record_output()`.

Usage

```
expect_snapshot_r_process(
  ...,
  interactive = TRUE,
  echo = TRUE,
  startup = NULL,
  transform = NULL,
  variant = NULL
)
```

Arguments

`...`	Code to run (unnamed arguments) and arguments to pass to `record_output()` (named arguments). The code is evaluated in a new asciicast subprocess. Their output is returned and used in a testthat snapshot test.
`interactive`	Whether to use an interactive R process to evaluate the code.
`echo`	Whether to echo the code in the subprocess before running it.
`startup`	Expression to evaluate in the subprocess before recording the snapshot. By default it loads and attaches the calling package, including its internal functions.
`transform`	Passed to `testthat::expect_snapshot()`.
`variant`	Passed to `testthat::expect_snapshot()`.

Details

THe `Code` part of the snapshot is always the same, but the `Output` part shows the code, assuming `echo = TRUE` (the default).

Examples

```
Sys.getpid()
testthat::local_edition(3)
expect_snapshot_r_process(Sys.getpid())
```

get_locales *Helper function to query locales as a named character vector.*

Description

Helper function to query locales as a named character vector.

Usage

```
get_locales()
```

Value

Named character vector with entries:

- LC_COLLATE, LC_CTYPE, LC_MONETARY, LC_NUMERIC and LC_TIME.

init_knitr_engine *Initialize the asciicast knitr engine*

Description

Call this function in your Rmd file to enable creating asciinema casts from code chunks.

Usage

```
init_knitr_engine(
  echo = FALSE,
  same_process = TRUE,
  timeout = 10,
  startup = NULL,
  record_env = NULL,
  echo_input = TRUE,
  interactive = TRUE
)
```

Arguments

echo	Whether to print the code of asciicast chunks.
same_process	Whether to run all asciicast chunks *in the same* R process. To restart this R process, call `init_knitr_engine()` again.
timeout	Idle timeout, in seconds If the R subprocess running the recording does not answer within this limit, it is killed and the recording stops.
startup	Quoted language object to run in the subprocess before starting the recording.
record_env	Environment variables to set for the R subprocess.

echo_input	Whether to echo the input in the asciicast recording.
interactive	Whether to run R in interactive mode. Note that in interactive mode R might ask for terminal input.

Details

Limitations:

- `purl()` or setting the `purl = TRUE` chunk option, does not work properly, in that knitr thinks that asciicast chunks are not R code, so they will appear as comments. If you know how to fix this, please contact us.

Examples

Call this function from an Rmd chunk and then you can use the asciicast knitr engine:

````
```{r setup, include = FALSE}
asciicast::init_knitr_engine()
```

```{asciicast, cache = TRUE}`
#' Rows: 10
# This is an asciicast example
loadedNamespaces()
```
````

---

install_phantomjs *Install PhantomJS*

Description

Download the zip package, unzip it, and copy the executable to a system directory in which **asciicast** can look for the PhantomJS executable.

Usage

```
install_phantomjs(
  version = "2.1.1",
  baseURL = "https://github.com/wch/webshot/releases/download/v0.3.1/",
  quiet = FALSE
)
```

Arguments

version	The version number of PhantomJS.
baseURL	The base URL for the location of PhantomJS binaries for download. If the default download site is unavailable, you may specify an alternative mirror, such as `"https://bitbucket.org/ariya/phantomjs/downloads/"`.
quiet	If `TRUE` suppress status messages and progress bar.

Details

This function was designed primarily to help Windows users since it is cumbersome to modify the PATH variable. Mac OS X users may install PhantomJS via Homebrew. If you download the package from the PhantomJS website instead, please make sure the executable can be found via the PATH variable.

On Windows, the directory specified by the environment variable APPDATA is used to store 'phantomjs.exe'. On OS X, the directory '~/Library/Application Support' is used. On other platforms (such as Linux), the directory '~/bin' is used. If these directories are not writable, the directory 'PhantomJS' under the installation directory of the **asciicast** package will be tried. If this directory still fails, you will have to install PhantomJS by yourself.

Value

NULL (the executable is written to a system directory).

---

play *Play asciinema cast as an SVG image in the default browser*

Description

Uses write_svg() to create an SVG image for a cast, in a temporary file, and then previews a minimal HTML file with the SVG image, in the default browser.

Usage

```
play(cast, ...)
```

Arguments

cast	asciicast object
...	Additional arguments are passed to write_svg().

Value

The path of the temporary SVG file, invisibly.

See Also

Other SVG functions: default_theme(), write_svg()

Examples

```
cast <- read_cast(system.file("examples", "hello.cast", package = "asciicast"))
play(cast)
```

read_cast *Import an asciicast from an asciicast JSON file*

Description

Import an asciicast from an asciicast JSON file

Usage

```
read_cast(json)
```

Arguments

json	Path to JSON asciicast file, version 2: https://github.com/asciinema/asciinema/blob/master/doc/asciicast-v2.md. If a numeric id, then it is taken as a public https://asciinema.org recording id, that is downloaded. It can also be a URL of private https://asciinema.org link.

Value

asciicast object.

See Also

Other asciicast functions: asciicast-package, asciicast_start_process(), record(), write_json()

Examples

```
c1 <- read_cast("https://asciinema.org/a/uHQwIVpiZvu0Ioio8KYx6Uwlj.cast?dl=1")
play(c1)

c2 <- read_cast(258660)
play(c2)


c3 <- read_cast(system.file("examples", "hello.cast", package = "asciicast"))
play(c3)
```

record *Record an asciinema screencast*

Description

Record an asciinema screencast

Usage

```
record(
  script,
  typing_speed = NULL,
  empty_wait = NULL,
  cols = NULL,
  rows = NULL,
  title = NULL,
  timestamp = NULL,
  env = NULL,
  idle_time_limit = NULL,
  timeout = NULL,
  start_wait = NULL,
  end_wait = NULL,
  record_env = NULL,
  startup = NULL,
  echo = TRUE,
  speed = NULL,
  process = NULL,
  interactive = TRUE,
  locales = get_locales(),
  options = asciicast_options(),
  incomplete_error = NULL,
  show_output = FALSE
)
```

Arguments

script	Path of an R script to record. It can also be a readable R connection or URL, as it is passed to base::readLines(). It can also be a language object, which is deparsed, or a character vector with the source code itself.
typing_speed	Average typing speed, per keypress, in seconds.
empty_wait	How long to wait for empty lines in the script file, in seconds.
cols	Width of the terminal, in number of characters.
rows	Height of the terminal, in number of characters. If it the string "auto", then it will be determined automatically, by including all output on the screen.
title	Title of the cast, this is included in the cast JSON file.

timestamp	Time stamp of the recording, defaults to `Sys.time()`, this is included in the cast JSON file.
env	Environment variables to include in the case JSON file. Defaults to `list(TERM = "xterm-256color", SHELL = "/bin/zsh")`.
idle_time_limit	Time limit for the cast not printing anything, in seconds. By default there is no limit.
timeout	Idle timeout, in seconds If the R subprocess running the recording does not answer within this limit, it is killed and the recording stops. Update this for slow running code, that produces no output as it runs.
start_wait	Delay at the beginning, in seconds.
end_wait	Delay at the very end, in seconds.
record_env	Environment variables to set for the R subprocess.
startup	Quoted language object to run in the subprocess before starting the recording.
echo	Whether to echo the input to the terminal. If `FALSE`, then only the output is shown.
speed	Rescale the speed of the recorded cast with this factor. The delay of the first frame is kept constant.
process	A processx subprocess to run the cast in. By default a new subprocess is started. You can reuse a process by calling `asciicast_start_process()` first, and supplying the returned process here.
interactive	Whether to run R in interactive mode. This argument is ignored if `process` is specified. If `process` is `NULL` then it is passed to `asciicast_start_process()`.
locales	Locales to set in the asciicast subprocess. Defaults to the current locales in the main R process. Specify a named character vector here to override some of the defaults. See also `get_locales()`.
options	Options to set in the subprocess, a named list. They are deparsed to code, and then the code setting them is executed in the subprocess. See `asciicast_options()` for the defaults. Supply a named list here to override the defaults or set additionsl ones. Passing large and/or complicated options here might not work, or might be slow.
incomplete_error	Whether to error on incomplete expressions. You might need to set this to `FALSE` for R code that does keyboard input, e.g. in `browser()`. The default is `TRUE`.
show_output	Whether to show the output of the subprocess in real time.

Value

An `asciicast` object, write this to file with `write_json()`.

See Also

Other asciicast functions: `asciicast-package`, `asciicast_start_process()`, `read_cast()`, `write_json()`

Examples

```
script <- system.file("examples", "hello.R", package = "asciicast")
cast <- record(script)
play(cast)
```

`record_output` *Record output of an R script and return it as a character vector*

Description

This function uses `record()` internally, but instead of creating an ascii cast, it just returns the output of the code in a character vector.

Usage

```
record_output(
  script,
  echo = FALSE,
  prompt = echo,
  stdout = TRUE,
  stderr = TRUE,
  ...
)
```

Arguments

`script`	The code to record, passed to `record()`.
`echo`	Whether to include the input in the return value.
`prompt`	Whether to include the R prompt in the return value.
`stdout`	Whether to include the standard output in the return value.
`stderr`	Whether to include the standard error in the return value.
`...`	Addiitonal arguments are passed to `record()`. (You cannot use `typing_speed` and `echo`, though, because these are used internally by `record_output()`.

Value

Character vector of output (plus input if `echo`, plus prompt if `prompt`), as it would appear on a terminal.

See `record()` for additional options.

write_gif *Export ascii screencast to animated GIF file*

Description

Export ascii screencast to animated GIF file

Usage

```
write_gif(
  cast,
  path,
  show = NULL,
  cols = NULL,
  rows = NULL,
  theme = NULL,
  scale = 2,
  speed = 1,
  max_colors = 256,
  loop = 0,
  end_wait = 10,
  optimize = TRUE
)
```

Arguments

cast	asciicast object.
path	Path to GIF file to create.
show	Whether to show the GIF on the screen, in the viewer pane in RStudio, or using the image viewer in the magick package. By default it only show the image in RStudio.
cols	If not NULL, *clip* terminal width to this number of columns.
rows	If not NULL, *clip* terminal height to this number of rows.
theme	Theme. Currently supported themes: asciinema, tango, solarized-dark, solarized-light, monokai. Defaults to the theme specified in the cast, or asciiname if not specified.
scale	Image scale / pixel density.
speed	Playback speed. Higher number means faster.
max_colors	Maximum number of colors in the GIF. This is currently per frame.
loop	How many times to loop the animation. Zero means infinite loop.
end_wait	Number of seconds to wait at the end, before looping.
optimize	Whether to try to create smaller GIF files. This might be slow for casts with many frames.

Value

path, invisibly.

`write_html` *Create a HTML snapshot of an asciicast*

Description

Create a HTML snapshot of an asciicast

Usage

```
write_html(
  cast,
  path,
  at = "end",
  omit_last_line = NULL,
  prefix = "",
  theme = NULL,
  details = FALSE,
  summary = "See output"
)
```

Arguments

cast	asciicast object.
path	Path to the HTML file to create.
at	When to take the snapshot, defaults to the end of the cast ("end"). Can also be a number, in seconds.
omit_last_line	Whether to omit the last line of the cast. This often just the prompt, and sometimes it is not worth showing.
prefix	Prefix to add to the beginning to every line. E.g. #> is usually added to knitr output.
theme	A theme name to use, or a a named list to override the default theme (see default_theme()).
details	Whether to put the output in a <details> tag.
summary	Summary of the <details> tag, ignored if details is FALSE.

write_json *Write an ascii cast to file*

Description

The file uses the asciinema file format, version 2: https://github.com/asciinema/asciinema/blob/master/doc/asciicast-v2.md.

Usage

```
write_json(cast, path)
```

Arguments

cast	asciicast object.
path	Path to write to.

See Also

Other asciicast functions: asciicast-package, asciicast_start_process(), read_cast(), record()

Examples

```
script <- system.file("examples", "hello.R", package = "asciicast")
cast <- record(script)
json <- tempfile(fileext = ".json")
write_json(cast, json)
```

write_svg *Create animated SVG from an asciicast*

Description

Create animated SVG from an asciicast

Usage

```
write_svg(
  cast,
  path,
  window = NULL,
  start_at = NULL,
  end_at = NULL,
  at = NULL,
```

```
  cursor = NULL,
  rows = NULL,
  cols = NULL,
  padding = NULL,
  padding_x = NULL,
  padding_y = NULL,
  omit_last_line = NULL,
  theme = NULL,
  show = NULL
)
```

Arguments

cast	asciicast object.
path	Path to the SVG file to create.
window	Render with window decorations.
start_at	Lower range of timeline to render in seconds.
end_at	Upper range of timeline to render in seconds.
at	Timestamp of single frame to render, in seconds. Alternatively it can be "end", to take a snapshot at the end of the cast, after all output is done.
cursor	Enable cursor rendering.
rows	Height in lines.
cols	Width in columns.
padding	Distance between text and image bounds.
padding_x	Distance between text and image bounds on x axis.
padding_y	Distance between text and image bounds on y axis.
omit_last_line	Whether to omit the last line of the cast. This often just the prompt, and sometimes it is not worth showing.
theme	A named list to override the default theme (see default_theme()).
show	Whether to show the SVG file on the screen, in the viewer pane in RStudio, or in the web browser.

See Also

Other SVG functions: default_theme(), play()

Examples

```
cast <- read_cast(system.file("examples", "hello.cast", package = "asciicast"))
svg_file <- tempfile(fileext = ".svg")
write_svg(cast, svg_file)
```

Index