Package 'captuR'

April 22, 2025

Type Package

Title Lightweight Object Capture and Metadata Tracking

Version 1.5

Description Provides a lightweight framework to automatically capture contextual metadata about R objects at creation time. Captured metadata includes the generating code, dependencies, digests, script and Git information, seeds, session info, and user-defined tags. Useful for reproducibility, validation, and provenance tracking. Supports capture from scripts, code blocks, and pipelines, and includes comparison and inspection utilities.

Depends R (>= 3.5.0)

Imports CodeDepends,
digest,
utils,
rstudioapi,
gert,
knitr

License MIT + file LICENSE

Encoding UTF-8

LazyData true

RoxygenNote 7.3.2

BugReports https://github.com/m-a-schultz/captuR/issues

URL https://github.com/m-a-schultz/captuR

Contents

`as.list.capture_diff` *Convert capture_diff to list*

Description

Convert capture_diff to list

Usage

```
## S3 method for class 'capture_diff'
as.list(x, ...)
```

Arguments

`x`	A ‘capture_diff‘ object.
`...`	Ignored.

Value

A named list of field comparisons.

`as.list.capture_env` *Convert capture_env to list*

Description

Convert capture_env to list

Usage

```
## S3 method for class 'capture_env'
as.list(x, ...)
```

Arguments

`x`	A ‘capture_env‘ object.
`...`	Ignored.

Value

Named list of contained objects.

`as.list.capture_info` *Convert a capture_info object to a list*

Description

Convert a capture_info object to a list

Usage

```
## S3 method for class 'capture_info'
as.list(x, ...)
```

Arguments

`x`	A ‘capture_info‘ object.
`...`	Ignored.

Value

Named list of metadata.

capture_block *#' Capture code block execution and context*

Description

Evaluates a block of R code and returns an object encapsulating both the resulting environment and the associated metadata for reproducibility. This includes inputs, outputs, code lines, digest values, and optional tags.

Usage

```
capture_block(
  code,
  save_env = TRUE,
  tags = NULL,
  digest_algo = "sha256",
  verbose = FALSE,
  include_session = FALSE
)

capture_code_lines(
  code_lines,
  save_env = TRUE,
  tags = NULL,
  digest_algo = "sha256",
  verbose = FALSE,
  include_session = FALSE,
  env = new.env()
)
```

Arguments

code	A block of code surrounded by ".
save_env	Logical; whether to return a named list of created objects.
tags	Optional character vector of user tags.
digest_algo	Digest algorithm to use (default: "sha256").
verbose	Logical; if TRUE, prints a summary of capture.
include_session	Logical; Should session info be included in metadata
code_lines	An R language object, like that returned by 'substitute'

Value

A named list with two components:

'capture' An object of class 'capture_info' with detailed metadata about the execution.

'objects' (Optional) A named list of created objects from the code block.

See Also

[capture_env()], [capture_info()], [get_capture()]

capture_checksum_valid *Validate a script checksum against the capture metadata.*

Description

Validate a script checksum against the capture metadata.

Usage

```
capture_checksum_valid(object)
```

Arguments

object	An object with a ‘capture‘ attribute.

Value

TRUE if the script is unchanged, FALSE otherwise.

capture_diff *Compare capture metadata for two objects*

Description

Compare capture metadata for two objects

Usage

```
capture_diff(object1, object2, ignore = NULL)
```

Arguments

object1	First captured object or capture_info
object2	Second captured object or capture_info
ignore	Optional character vector of field names to ignore

Value

An object of class ‘capture_diff‘

capture_digest *Get digest from capture metadata*

Description

Get digest from capture metadata

Usage

```
capture_digest(object)
```

Arguments

object	An R object with capture metadata.

Value

Digest string or NULL if not available.

capture_env *Create a captured environment object*

Description

Constructs a special list-like object that holds a group of captured R objects along with a shared ‘capture_info‘ metadata structure. This is typically returned by top-level capture functions that track multiple outputs.

Usage

```
capture_env(objects, capture)
```

Arguments

objects	A named list of R objects.
capture	A ‘capture_info‘ object with metadata.

Details

The ‘capture_env‘ behaves like a named list, and individual objects may be accessed using ‘$‘ or ‘[[‘ operators. Each object accessed in this way will have the shared ‘"capture"‘ metadata attached automatically.

Value

A ‘capture_env‘ object that behaves like a list with attached metadata.

Structure

A ‘capture_env‘ is a list of named objects with an attached attribute:

[objects] Each named object created during a script or code block execution.

‘attr(., "capture")‘ A ‘capture_info‘ object describing shared metadata for all objects.

Access

- Use ‘x$name‘ or ‘x[[name]]‘ to access individual objects.
- The shared capture metadata is propagated to each extracted object.
- Use [get_capture()] to retrieve metadata from either the ‘capture_env‘ or its elements.

See Also

[capture_info()], [capture_block()], [capture_source()], [get_capture()]

---

capture_info *Create a capture_info object*

Description

Constructs a structured metadata object describing the full context in which an R object or environment was captured. This includes information about the object's name, digest, timestamp, script source, code history, dependencies, session info, Git metadata, seed state, and other provenance details.

Usage

```
capture_info(
  inputs,
  outputs,
  version,
  object_name,
  timestamp,
  script_path,
  code_lines,
  dependencies,
  script_checksum,
  seed,
  is_stochastic,
  session,
  object_digest,
  git_info = NULL,
  tags = NULL,
  package_versions = NULL,
  warnings = NULL,
  digest_algorithm = "sha256"
)
```

Arguments

version	captuR version string.
object_name	Name of the object being saved.
timestamp	Time when the object was saved.
script_path	Path to the script that generated the object.
code_lines	Character vector of code lines used to generate the object.
dependencies	A list of 'ScriptNodeInfo' objects (from CodeDepends).
script_checksum	Digest checksum of the script file.
seed	Any 'set.seed()' calls found in the generating code.
is_stochastic	Logical indicating whether stochastic code was detected
session	The output of 'sessionInfo()' at save time.
object_digest	Digest hash of the object itself.
git_info	Git metadata as a named list (optional).
tags	Character vector of user-supplied tags (optional).
package_versions	Named character vector of package versions used in dependencies.
warnings	List of warning messages generated during capture.
digest_algorithm	The digest algorithm used (e.g., "sha256").

Details

The 'capture_info' object is attached as an attribute to captured R objects or environments under the name '"capture"' and is used for auditing, reproducibility, validation, and code regeneration.

Value

An object of class 'capture_info', suitable for attaching as metadata.

An object of class 'capture_info'.

Structure

A 'capture_info' object is a named list with the following components:

inputs Character vector of objects that were inputs (read from) during generation.

outputs Character vector of objects created or updated.

capture_version Character string indicating captuR version.

object_name The name of the primary object (or comma-separated names for multiple).

timestamp POSIXct timestamp indicating when the capture was performed.

script_path Path to the R script that generated the object (if available).

code_lines Character vector of R code used to generate the object(s).

dependencies List of 'ScriptNodeInfo' objects from 'CodeDepends' that describe code structure.

script_checksum Digest hash of the script file at capture time.

seed The nearest 'set.seed()' call found before object creation, if present.

session Optional ‘sessionInfo()‘ result if session info was captured.

object_digest Named list of digest hashes for each object, or a single string for one object.

git_info Named list with Git commit, author, message, and date (if available).

tags Optional character vector of user-supplied tags.

package_versions Named character vector of package versions used in code.

warnings List of warning messages generated during capture (if any).

digest_algorithm Digest algorithm used (e.g., ‘"sha256"‘).

See Also

[get_capture()], [capture_env()], [capture_save()], [capture_block()]

`capture_object` *Attach metadata to a single object*

Description

Analyzes the provenance of an R object and attaches metadata to it, without saving it to disk. Metadata includes the code that generated the object, its digest, seed, dependencies, and script/Git information.

Usage

```
capture_object(
  object,
  script_path = NULL,
  tags = NULL,
  digest_algo = "sha256",
  verbose_guess = FALSE,
  contextual_code = TRUE,
  include_session = FALSE
)
```

Arguments

`object`	An R object to capture.
`script_path`	Optional path to source script.
`tags`	Optional tags.
`digest_algo`	Digest algorithm to use.
`verbose_guess`	Logical; whether to print guess path warnings.
`contextual_code`	Logical; if TRUE, return all code from set.seed() to object creation.
`include_session`	Logical; Should session info be included in metadata

Value

The input object with an attached ‘"capture"‘ attribute of class ‘capture_info‘, which encapsulates all associated metadata for reproducibility and tracking.

See Also

[capture_info()], [get_capture()]

`capture_save` *Save/Load an R object with capture metadata*

Description

Saves and loads an R object to disk as an RDS file, while attaching detailed metadata about its creation, including the originating code, seed, script path, and dependency information.

Usage

```
capture_save(
  object,
  file,
  script_path = NULL,
  save_env = FALSE,
  remove_seed = FALSE,
  tags = NULL,
  verbose_guess = FALSE,
  digest_algo = "sha256",
  contextual_code = FALSE
)

capture_load(file, check_script = TRUE)
```

Arguments

`file`	RDS file path.
`script_path`	Optional script path.
`save_env`	Save environment variables?
`remove_seed`	Drop set.seed lines?
`tags`	Optional tags.
`verbose_guess`	Verbose script path guessing?
`digest_algo`	Digest algorithm.
`contextual_code`	Logical; if TRUE, return all code from set.seed() to object creation.
`check_script`	Validate script?

Value

The same R object, with an attached ‘capture_info‘ object in the ‘"capture"‘ attribute. This metadata can be accessed using [get_capture()] and includes information such as object name, timestamp, script source, code lines, digest, Git commit (if applicable), and dependency structure.

The saved object with capture metadata.

R object with capture attribute.

See Also

[capture_info()], [get_capture()], [capture_load()]#' @param object The R object.

`capture_script_path` *Get the script path from capture metadata*

Description

Get the script path from capture metadata

Usage

```
capture_script_path(object)
```

Arguments

`object`	A ‘capture_info‘ object.

Value

Character string of the script path.

`capture_source` *Source an R script with capture metadata*

Description

Behaves like base::source() but returns an object summarizing all newly created objects along with detailed capture metadata for reproducibility.

Usage

```
capture_source(
  file,
  local = parent.frame(),
  tags = NULL,
  digest_algo = "sha256",
  verbose = FALSE,
  include_session = FALSE,
  ...
)
```

Arguments

file	Path to the script file.
local	Logical or environment. Passed to ‘base::source()‘. Defaults to parent.frame().
tags	Optional character vector of user-supplied tags.
digest_algo	Digest algorithm for checksums (default: "sha256").
verbose	Print metadata summary? Default FALSE.
include_session	Logical; include session info in metadata.
...	Additional arguments passed to base::source().

Value

A ‘capture_env‘ object containing all top-level outputs created by the script, with a shared ‘capture_info‘ metadata object accessible via [get_capture()]. The result is returned invisibly.

See Also

[capture_env()], [capture_info()], [get_capture()]

---

capture_summary *Capture Metadata Summary for Multiple Objects*

Description

Returns a data frame summarizing key metadata from a list of captured objects.

Usage

```
capture_summary(objects)
```

Arguments

objects	A named list of R objects (or captured environment).

Value

A data.frame with metadata summary.

capture_summary_list *Return summary list from capture metadata*

Description

Provides a simplified summary of key capture metadata.

Usage

```
capture_summary_list(object)
```

Arguments

object	An object with a ‘capture‘ attribute.

Value

Named list with name, timestamp, digest, script path, Git commit, and tags.

capture_tags *Get tags from capture metadata*

Description

Get tags from capture metadata

Usage

```
capture_tags(object)
```

Arguments

object	An R object with capture metadata.

Value

Character vector of tags or NULL if not available.

capture_timestamp *Retrieve the capture timestamp*

Description

Retrieve the capture timestamp

Usage

```
capture_timestamp(object)
```

Arguments

object	A captured object.

Value

POSIXct timestamp or NULL.

capture_version *Get the current captuR version.*

Description

Get the current captuR version.

Usage

```
capture_version()
```

check_capture *Check the consistency of the generating script with the recorded checksum.*

Description

Check the consistency of the generating script with the recorded checksum.

Usage

```
check_capture(capture, verbose = TRUE)
```

Arguments

capture	The capture metadata object (of class 'capture_info').
verbose	Logical indicating whether to print messages (default: TRUE).

Value

TRUE if the script is consistent with the saved checksum.

check_captured_code *Re-run Captured Code and Validate Outputs*

Description

Re-evaluates the original code used to create a captured object and compares the resulting outputs to the original objects. Useful for verifying that captured code remains reproducible and consistent.

Usage

```
check_captured_code(obj, overall = FALSE)
```

Arguments

obj	A captured object or 'capture_env' with capture metadata.
overall	Logical; if 'TRUE', returns a single 'TRUE'/'FALSE' indicating whether all outputs matched. If 'FALSE' (default), returns a named logical vector indicating which outputs matched.

Value

A logical vector (named by object name) or a single logical value if 'overall = TRUE'.

Examples

```
## Not run:
result <- check_captured_code(my_model)
if (any(!result)) warning("Some outputs did not match.")

# Check if everything matches in a single logical
check_captured_code(my_model, overall = TRUE)

## End(Not run)
```

compare_capture *Compare capture fields (logical check)*

Description

Performs a field-by-field logical comparison of two capture metadata objects.

Usage

```
compare_capture(object1, object2, ignore = NULL)
```

Arguments

object1	First captured object or 'capture_info'.
object2	Second captured object or 'capture_info'.
ignore	Character vector of fields to ignore (optional).

Value

Named list indicating which fields are identical (‘TRUE‘), different (‘FALSE‘), or ignored (‘NA‘).

`export_capture_code` *Export generating code from captured object.*

Description

Export generating code from captured object.

Usage

```
export_capture_code(object, file, header = TRUE)
```

Arguments

`object`	A captured object.
`file`	File path for output.
`header`	Include header metadata? Default TRUE.

Value

File path (invisibly).

`get_capture` *Access capture metadata from an R object*

Description

Retrieves the metadata stored during object capture.

Usage

```
get_capture(object)
```

Arguments

`object`	An R object saved using ‘capture_save()‘.

Value

A ‘capture_info‘ object or NULL if unavailable.

get_generating_code *Extract Generating Code and Seed Call*

Description

Identifies the code lines that generated an object using 'CodeDepends', and locates the closest preceding 'set.seed()' call if available.

Usage

```
get_generating_code(
  expr,
  script_path = NULL,
  obj_name = NULL,
  contextual_code = TRUE
)
```

Arguments

expr	The object whose capture is being traced.
script_path	Optional path to the R script file. Guessed if NULL.
obj_name	Optional character name of the object. Defaults to 'deparse(substitute(expr))'.
contextual_code	Logical; if TRUE, return all code from set.seed() to object creation.
dependencies	Optional precomputed dependency list from 'get_object_dependencies()'.

Value

A list with two elements:

code List of code lines related to object creation.

seed Optional set.seed line found before the last code line.

get_git_info *Get Git Metadata for a Script*

Description

Returns Git commit information for the script's repository, if available.

Usage

```
get_git_info(script_path)
```

Arguments

script_path	Path to the script file.

Value

A named list with commit, author, message, and date; or NULL.

get_object_dependencies *Get Code Dependencies for an Object*

Description

Uses ‘CodeDepends‘ to find the statements in a script that generated a specific object.

Usage

```
get_object_dependencies(obj_name, script_path = NULL)
```

Arguments

obj_name	Character name of the object to trace.
script_path	Path to the script file (optional). Will guess if NULL.

Value

A list of ‘ScriptNodeInfo‘ objects or NULL.

get_package_versions *Get package versions from CodeDepends dependencies*

Description

Extracts used packages based on namespaced functions in CodeDepends analysis.

Usage

```
get_package_versions(dependencies)
```

Arguments

dependencies	List of ScriptNodeInfo objects

Value

Named character vector of package versions

has_capture *Check if an object has capture metadata*

Description

Determines whether the object has a valid capture metadata attribute.

Usage

```
has_capture(object)
```

Arguments

object	An R object.

Value

TRUE if the object has a capture attribute of class ‘capture_info‘, FALSE otherwise.

is_script_checksum_valid *Check if capture script checksum is valid*

Description

Validates that the current script content matches the recorded checksum in the capture metadata.

Usage

```
is_script_checksum_valid(capture)
```

Arguments

capture	A ‘capture_info‘ object.

Value

Logical, TRUE if checksum matches current script.

is_stochastic_code *Detect Stochastic Behavior in Code Lines*

Description

Determines whether any known stochastic functions appear in the code.

Usage

```
is_stochastic_code(code_lines, funcs = stochastic_functions())
```

Arguments

code_lines	Character vector of R code lines.
funcs	Optional character vector of stochastic function names to match.

Value

Logical indicating whether code is stochastic.

list_capture_dependencies *List input dependencies from capture metadata*

Description

List input dependencies from capture metadata

Usage

```
list_capture_dependencies(object)
```

Arguments

object	A captured object.

Value

A list of dependencies or NULL.

print.capture_diff *Print method for capture_diff*

Description

Print method for capture_diff

Usage

```
## S3 method for class 'capture_diff'
print(x, ...)
```

Arguments

x	A ‘capture_diff‘ object.
...	Ignored.

print.capture_env *Print method for capture_env*

Description

Displays a summary of the captured environment.

Usage

```
## S3 method for class 'capture_env'
print(x, ...)
```

Arguments

x	A ‘capture_env‘ object.
...	Ignored.

Value

The object (invisible).

print.capture_info *Print summary of capture metadata*

Description

Print summary of capture metadata

Usage

```
## S3 method for class 'capture_info'
print(x, ...)
```

Arguments

x	A ‘capture_info‘ object.
...	Ignored.

stochastic_functions *List Known Stochastic Function Names*

Description

Returns a list of commonly used R functions that generate randomness.

Usage

```
stochastic_functions()
```

Value

A character vector of function names (e.g., "rnorm", "sample").

summary.capture_diff *Summary method for capture_diff*

Description

Summary method for capture_diff

Usage

```
## S3 method for class 'capture_diff'
summary(object, ...)
```

Arguments

object	A ‘capture_diff‘ object.
...	Ignored.

summary.capture_info *Summary method for capture metadata*

Description

Summary method for capture metadata

Usage

```
## S3 method for class 'capture_info'
summary(object, ...)
```

Arguments

object	A ‘capture_info‘ object.
...	Ignored.

Value

A named list summarizing key metadata.

view_capture_code *View the generating code from capture metadata*

Description

View the generating code from capture metadata

Usage

```
view_capture_code(object)
```

Arguments

object	A captured object.

view_capture_script *View Full Script with Metadata from Capture*

Description

Prints the full reconstructed code used to create an object, with metadata headers and line numbers.

Usage

```
view_capture_script(object, numbered = TRUE, header = TRUE)
```

Arguments

object	An R object with capture metadata.
numbered	Logical. Whether to add line numbers. Default TRUE.
header	Logical. Whether to include metadata header. Default TRUE.

[[.capture_env *Extract an object from capture_env*

Description

Allows access to objects via '$' operator.

Usage

```
## S3 method for class 'capture_env'
x[[i]]
```

Arguments

x	A 'capture_env' object.
i	Index of object to extract.

Value

The extracted object.

$.capture_env *Extract an object from capture_env*

Description

Allows access to objects via '$' operator.

Usage

```
## S3 method for class 'capture_env'
x$name
```

Arguments

x	A 'capture_env' object.
name	Name of object to extract.

Value

The extracted object.

%capture% *Pipe-friendly capture-save operator*

Description

A forward-pipeable operator to save an object with capture metadata.

Usage

```
object %capture% file
```

Arguments

object	The object to be saved.
file	File path to save the object.

Value

The saved object (invisible).

Index