Package 'roprov'

October 14, 2022

Version 0.1.2

Title Low-Level Support for Provenance Capture Between in-Memory R Objects

Description A suite of classes and methods which provide low-level support for modeling provenance between in-memory R objects. This is an infrastructure package and is not intended to be used directly by end-users.

Author Gabriel Becker

Depends methods, fastdigest

Imports CodeDepends(>= 0.3.5), igraph

Copyright Genentech, Inc.

Maintainer Gabriel Becker <gabembecker@gmail.com>

License Artistic-2.0

RoxygenNote 6.0.1

NeedsCompilation no

Repository CRAN

Date/Publication 2018-11-06 09:10:03 UTC

R topics documented:

cback1 2
fullprovgraph 2
outputvars 3
provFromHistory 5
ProvStoreDF-class 6
ProvStores 7
rbind 8

Index **9**

cback1 *cback1*

Description

Determine the hash and class of a varaible based on the most recent ouput values for that variable captured as a previous output. Not intended for direct use by end users.

Usage

```
cback1(invar, prevouts, prevouthashes, prevoutclasses)
```

Arguments

invar	The variable to find
prevouts	Previous output variables
prevouthashes	Hashes for the previous output variables
prevoutclasses	Classes for the previous output variables

Details

If the variable is not found in the previous outputs, the .Global environment is inspected for a current value of the variable. This is necessary to capture things that can show up via lazyloading without ever being an output v, e.g., the mtcars dataset.

Value

a list containing the hash and class of the variable.

---

fullprovgraph *Create the full (multiple values per variable) provenance graph for a cache*

Description

This generates and returns the *full* provenance graph reflecting the information stored in the cache or data.frame specified. This can include the same variable multiple times if the corresponding expression is run with different inputs.

Usage

```
fullprovgraph(provstore)

## S4 method for signature 'ProvStoreDF'
fullprovgraph(provstore)

## S4 method for signature 'data.frame'
fullprovgraph(provstore)
```

Arguments

provstore	The provenance store data to use when generating the graph

Value

an igraph object representing the provenance graph

Examples

```
library(fastdigest)
code = c("x = 5", "y = x + 1")
outvars = c("x", "y")
outvarhashes = c(fastdigest(5), fastdigest(6))
outvarclasses = rep("numeric", 2)
invars = c("", "x")
invarhashes = c("", fastdigest(5))
invarclasses = c("", "numeric")

ps = ProvStoreDF(outvars = outvars,
                 outvarhashes = outvarhashes,
                 outvarclasses = outvarclasses,
                 invars = invars,
                 invarhashes = invarhashes,
                 invarclasses = invarclasses,
                 code = code,
                 agent = "coolguyorgirl")

plot(fullprovgraph(ps))
```

---

outputvars *Accessors*

Description

Accessors for information in roprov objects

Usage

```
outputvars(obj)

## S4 method for signature 'ProvStoreDF'
outputvars(obj)

outputvarhashes(obj)

## S4 method for signature 'ProvStoreDF'
outputvarhashes(obj)
```

```
outputvarclasses(obj)

## S4 method for signature 'ProvStoreDF'
outputvarclasses(obj)

inputvars(obj)

## S4 method for signature 'ProvStoreDF'
inputvars(obj)

inputvarhashes(obj)

## S4 method for signature 'ProvStoreDF'
inputvarhashes(obj)

inputvarclasses(obj)

## S4 method for signature 'ProvStoreDF'
inputvarclasses(obj)

provdata(obj)

## S4 method for signature 'ProvStoreDF'
provdata(obj)

hashprefix(obj)

## S4 method for signature 'ProvStoreDF'
hashprefix(obj)
```

Arguments

obj	The object.

Examples

```
library(fastdigest)
code = c("x = 5", "y = x + 1")
outvars = c("x", "y")
outvarhashes = c(fastdigest(5), fastdigest(6))
outvarclasses = rep("numeric", 2)
invars = c("", "x")
invarhashes = c("", fastdigest(5))
invarclasses = c("", "numeric")

ps = ProvStoreDF(outvars = outvars,
                 outvarhashes = outvarhashes,
                 outvarclasses = outvarclasses,
                 invars = invars,
                 invarhashes = invarhashes,
```

```
                invarclasses = invarclasses,
                code = code,
                agent = "coolguyorgirl")
outputvars(ps)
outputvarhashes(ps)
outputvarclasses(ps)
inputvars(ps)
inputvarhashes(ps)
inputvarclasses(ps)
provdata(ps)
hashprefix(ps)
```

`provFromHistory` *Generate a provenance store from hashed output history*

Description

Generate provenance of a single variable from a history of hashed outputs.

Usage

```
provFromHistory(outvar, outvarhash, outvarclass, invars, prevouts,
  prevouthashes, prevoutclasses, code)
```

Arguments

`outvar`	character(1). A single output variable
`outvarhash`	character(1). The hash of the value of `outvar`.
`outvarclass`	character(1). The (top level) class of the output variable
`invars`	A vector of input variables for `outvar`.
`prevouts`	character. A vector of previous output variables
`prevouthashes`	character. A vector of previous output hashes corresponding to `prevouts`
`prevoutclasses`	character. A vector of (top level) classes corresponding to `prevouts`
`code`	The code corresponding with the generation of `outvar`

Value

A ProvStoreDF object

`ProvStoreDF-class` *ProvStoreDF constructors and class*

Description

Functions to create a valid ProvStore object from raw data about evaluation history (including hashes of input and output variable values).

Usage

```
makeProvStore(invarhashes = NULL, outvarhashes = NULL, code = NULL,
  codehash = fastdigest(code), invars = names(invarhashes),
  outvars = names(outvarhashes), invarclasses = character(),
  outvarclasses = character(), agent = getUser(), hashprefix = "SpookyV2")

ProvStoreDF(outvars = character(), outvarhashes = character(),
  outvarclasses = character(), invars = character(),
  invarhashes = character(), invarclasses = character(),
  agent = character(), code = character(), codehash = vapply(code,
  fastdigest, character(1)), hashprefix = "SpookyV2", df = NULL)
```

Arguments

`invarhashes`	character. Hashes of values for input variables (or NULL).
`outvarhashes`	character. Hashes of values for output variables (or NULL).
`code`	character. The code associated with the input and output variables in question.
`codehash`	character. The hash of the (parsed and deparsed) code.
`invars`	character. The set of input variables (symbols) to `code`
`outvars`	character. The output variable(s) (symbols) to `code`.
`invarclasses`	character. The (top level) classes of the input variable values.
`outvarclasses`	character. The (top level) classes of the output variable values.
`agent`	character. a string identifying the user. Defaults to the output of a `whoami` system call.
`hashprefix`	character. A prefix to append to the hashes if not present already to make them more self-describing.
`df`	data.frame. Optional. An already constructed data.frame do use as the provdata of the constructed `ProvStoreDF` object.

Details

makeProvStore expects information about a single code unit. i.e., all input values will be counted as inputs for all outputs. For makeProvStore the number of inputs and outputs need not match, and the appropriate replication on inputs will happen automatically.

ProvStoreDF is a direct constructor and thus expects the replciation of inputs for multiple outputs to have already occurred.

Examples

```
## spoof the information needed to create a provstore
library(fastdigest)
code = c("x = 5", "y = x + 1")
outvars = c("x", "y")
outvarhashes = c(fastdigest(5), fastdigest(6))
outvarclasses = rep("numeric", 2)
invars = c("", "x")
invarhashes = c("", fastdigest(5))
invarclasses = c("", "numeric")

ps = ProvStoreDF(outvars = outvars,
                 outvarhashes = outvarhashes,
                 outvarclasses = outvarclasses,
                 invars = invars,
                 invarhashes = invarhashes,
                 invarclasses = invarclasses,
                 code = code,
                 agent = "coolguyorgirl")

 df = data.frame(outputvar = outvars,
                 outputvarhash = outvarhashes,
                 outputvarclass = outvarclasses,
                 inputvar= invars,
                 inputvarhash = invarhashes,
                 inputvarclass = invarclasses,
                 agent = "coolgirloruy",
                 code = code,
                 codehash = sapply(code, fastdigest),
                 stringsAsFactors = FALSE)

ps2 = ProvStoreDF(df = df)
```

`ProvStores` *Combine provenance stores*

Description

This operation is conceptually

Usage

```
ProvStores(...)
```

Arguments

`...`	Two or more ProvStoreDF objects

Value

A ProvStoreDF object

`rbind` *rbind method*

Description

an rbind method for ProvStoreDF objects

Usage

```
rbind(..., deparse.level = 1)

## S4 method for signature 'ProvStoreDF'
rbind(..., deparse.level = 1)
```

Arguments

`...`	Two or more ProvStoreDF objects. Must all have identical hashprefix values
`deparse.level`	ignored.

Value

A ProvStoreDF object.

Examples

```
library(fastdigest)
library(roprov)
code = c("x = 5", "y = x + 1")
outvars = c("x", "y")
outvarhashes = c(fastdigest(5), fastdigest(6))
outvarclasses = rep("numeric", 2)
invars = c("", "x")
invarhashes = c("", fastdigest(5))
invarclasses = c("", "numeric")
 df = data.frame(outputvar = outvars,
                 outputvarhash = outvarhashes,
                 outputvarclass = outvarclasses,
                 inputvar= invars,
                 inputvarhash = invarhashes,
                 inputvarclass = invarclasses,
                 agent = "coolgirloruy",
                 code = code,
                 codehash = sapply(code, fastdigest),
                 stringsAsFactors = FALSE)

ps2 = ProvStoreDF(df = df)
rbind(ps2, ps2)
```

Index

cback1, 2

fullprovgraph, 2
fullprovgraph,data.frame
(fullprovgraph), 2
fullprovgraph,data.frame-method
(fullprovgraph), 2
fullprovgraph,ProvStoreDF
(fullprovgraph), 2
fullprovgraph,ProvStoreDF-method
(fullprovgraph), 2

hashprefix (outputvars), 3
hashprefix,ProvStoreDF (outputvars), 3
hashprefix,ProvStoreDF-method
(outputvars), 3

inputvarclasses (outputvars), 3
inputvarclasses,ProvStoreDF
(outputvars), 3
inputvarclasses,ProvStoreDF-method
(outputvars), 3
inputvarhashes (outputvars), 3
inputvarhashes,ProvStoreDF-method
(outputvars), 3
inputvars (outputvars), 3
inputvars,ProvStoreDF (outputvars), 3
inputvars,ProvStoreDF-method
(outputvars), 3

makeProvStore (ProvStoreDF-class), 6

outputvarclasses (outputvars), 3
outputvarclasses,ProvStoreDF
(outputvars), 3
outputvarclasses,ProvStoreDF-method
(outputvars), 3
outputvarhashes (outputvars), 3
outputvarhashes,ProvStoreDF
(outputvars), 3
outputvarhashes,ProvStoreDF-method
(outputvars), 3
outputvars, 3
outputvars,ProvStoreDF (outputvars), 3
outputvars,ProvStoreDF-method
(outputvars), 3

provdata (outputvars), 3
provdata,ProvStoreDF (outputvars), 3
provdata,ProvStoreDF-method
(outputvars), 3
provFromHistory, 5
ProvStoreDF (ProvStoreDF-class), 6
ProvStoreDF-class, 6
ProvStores, 7

rbind, 8
rbind,ProvStoreDF-method (rbind), 8