Package: dcmodifydt (via r-universe)

June 25, 2024

Title Modification Rules on 'data.table'

Version 0.1.0.9000

Description Apply 'dcmodify' rules on a 'data.table', code generated is optimized for changing a data.table in place / by reference.

License MIT + file LICENSE

Encoding UTF-8

LazyData true

Roxygen list(markdown = TRUE)

RoxygenNote 7.1.1

Remotes data-cleaning/dcmodify/pkg

Depends dcmodify (>= 0.1.9), R (>= 3.5.0)

Imports validate, data.table, methods

Suggests covr, tinytest

URL https://github.com/data-cleaning/dcmodifydt

BugReports https://github.com/data-cleaning/dcmodifydt/issues

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

RemoteUrl https://github.com/data-cleaning/dcmodifydt

RemoteRef HEAD

RemoteSha 258f5f5ea07a91293adfdfd752f57770c1d495f6

Contents

dump_dt *Dump data.table modification script*

Description

Dump data.table modification script

Usage

```
dump_dt(x, name = "dat", file = stdout())
```

Arguments

x	dcmodify::modifier() object with the modification rules
name	character name of data.table to be used.
file	where should the generated file be written?

Examples

```
library(dcmodify)

m <- modifier( if (age > 130) age = 130
               , income[age < 12] <- 0
)

dump_dt(m, "my_data")
```

modify,data.table,modifier-method *Modify records in a data.table*

Description

Modify records in a data.table using modification rules specified in a modifier object.

Usage

```
## S4 method for signature 'data.table,modifier'
modify(dat, x, copy = NULL, sequential = TRUE, ...)
```

Arguments

dat	data.table() object
x	dcmodify::modifier object.
copy	if TRUE modify copy of table.
sequential	if TRUE (default), steps will be executed in sequence, so order matters.
...	unused

Details

This is a more efficient implementation then coercing the data.table to a data.frame and use that implementation.

See Also

Other modify: `setmodify()`

Examples

```
library(data.table)

m <- modifier( if (age > 130) age = 130
             , income[age < 12] <- 0
             )

dat <- fread(text =
"age, income
 140,  300
  11, 2000
  25, 3000"
)

# modify a copy of the data
dat_m <- modify(dat, m, copy = TRUE)
print(dat_m)

# the data it self
setmodify(dat, m)
print(dat)
```

`setmodify` *modifies data.table in place*

Description

modifies data.table in place, alias for `modify` with `copy=TRUE` and `sequential=TRUE`. It follows the naming convention in `data.table` to prefix methods that change objects (byreference) with `set`.

Usage

```
setmodify(dat, x, ...)
```

Arguments

`dat`	`data.table()` object
`x`	`dcmodify::modifier` object.
`...`	not used

See Also

Other modify: modify,data.table,modifier-method

Examples

```
library(data.table)

m <- modifier( if (age > 130) age = 130
             , income[age < 12] <- 0
             )

dat <- fread(text =
"age, income
 140,  300
  11, 2000
  25, 3000"
)

# modify a copy of the data
dat_m <- modify(dat, m, copy = TRUE)
print(dat_m)

# the data it self
setmodify(dat, m)
print(dat)
```

Index