drake CHEAT SHEET

Logo

A computational engine for data analysis that skips computing up to date results with support for distributed computing

build and plans

make(plan, ...**)**
Run your project (build the outdated targets)

drake_config(plan, ...**)**
does all the preprocessing for make()

plan
Workflow plan data frame.

drake_build(target, ...**)**
Build/process a single target or import.

drake_plan(...**)**
A data frame with columns target and command

...
A collection of symbols/targets with commands assigned to them

drake_slice(data, slices, index, margin, drop**)**
Take a strategic subset of a dataset.

drake plan keywords

target(command, transform**)**
Must be called inside drake_plan(). Invalid otherwise.

transform
A call to map(), etc. to apply a static transformation

trigger(command, depend, file, seed**)**
Customize the decision rules for rebuilding targets

file_in(...**)**
Declare input files and directories

...
Character vector, paths to files and directories.

file_out(...**)**
Declare output files and directories

knitr_in(...**)**
Declare knitr/rmarkdown source files as dependencies

ignore(x**)**
Ignore sections of commands and imported functions

visualization

vis_drake_graph(config, ...**)**
Show an interactive visual network representation of your drake project

config
A drake_config() configuration list

drake_ggraph(config, ...**)**
Visualize the workflow with ggraph/ggplot2

target status

drake_history(cache, history, analyze, ...**)**
See the history and provenance of your targets: what you ran, when you ran it, the function arguments you used, and how to get old data back.

history
Logical, whether to record the build history of your targets. Must be TRUE for drake_history() to work later.

https://github.com/ropensci/drake