

# MSDproject

February 4, 2019

**Type** Package

**Title** Project and Code Management

**Version** 0.1.5

**Maintainer** Oleg Stepanov <oleg.stepanov@msdecisions.ru>

**Description** Create and manage standardised data analysis directories with private package libraries and links to external code repositories.

**License** GPL-3 | file LICENSE

**LazyData** TRUE

**RoxygenNote** 6.1.1

**Suggests** covr,  
testthat,  
git2r,  
rmarkdown

**Imports** devtools,  
magrittr,  
packrat,  
tidyverse

## R topics documented:

attach_code_library . . . . .	2
check_if_MSDproject . . . . .	3
check_session . . . . .	3
code_library . . . . .	4
code_library_path . . . . .	4
code_snapshot_files . . . . .	4
commit_file . . . . .	5
copy_file . . . . .	5
copy_script . . . . .	6
dependency_tree . . . . .	6
do_test . . . . .	7
get_github_code_library . . . . .	7
get_script_field . . . . .	8
info_scripts . . . . .	8
is_full_path . . . . .	9
is_rstudio . . . . .	9
locate_file . . . . .	10

ls_code_library . . . . .	10
ls_scripts . . . . .	11
make_local_bare . . . . .	11
make_project . . . . .	12
models_dir . . . . .	12
MSDproject . . . . .	13
new_script . . . . .	13
preview . . . . .	14
pull_repo . . . . .	14
reexports . . . . .	14
Renvironment_info . . . . .	15
replace_code_library . . . . .	15
resetwd . . . . .	15
scripts_dir . . . . .	15
search_field . . . . .	16
search_keyword . . . . .	16
search_raw . . . . .	16
setup_file . . . . .	17
set_project_opts . . . . .	17
short_path . . . . .	17
snapshot2 . . . . .	18
system_cmd . . . . .	18
wait_for . . . . .	18

<b>Index</b>	<b>19</b>
--------------	-----------

---

attach_code_library	<i>Attach code library</i>
---------------------	----------------------------

---

## Description

Attaches a path(s) to to the code library search path

## Usage

```
attach_code_library(path)
```

## Arguments

path	character vector with paths to attach to
------	--

---

check_if_MSDproject	<i>Check if directory is a MSDproject</i>
---------------------	---

---

**Description**

Check if directory is a MSDproject

**Usage**

```
check_if_MSDproject(directory = getwd(), return_logical = FALSE)
```

**Arguments**

directory        character. working directory (default = current working directory)

return\_logical   logical. If true will return FALSE instead of error

---

check_session	<i>Check MSDproject for best practice compliance</i>
---------------	--

---

**Description**

Check MSDproject for best practice compliance

**Usage**

```
check_session(proj_name = getwd(), silent = FALSE,  
              check_rstudio = TRUE)
```

**Arguments**

proj\_name        character. default = current working directory. path to directory.

silent           logical. default = FALSE. suppress messages or not

check\_rstudio   logical (default = FALSE). Check rstudio studio project directory

---

code_library	<i>Show Code Library</i>
--------------	--------------------------

---

**Description**

Show Code Library

**Usage**

```
code_library(extn = NULL, fields = "Description", viewer = TRUE,
             silent = FALSE, return_info = FALSE)
```

**Arguments**

extn	vector string of extensions to include
fields	character vector of fields to extract
viewer	logical indicating if viewer should be used to display results (default=FALSE)
silent	logical indicating if messages should be silenced (default=FALSE)
return_info	logical (default = FALSE). Return data.frame of results (FALSE= returns file paths)

---

code_library_path	<i>Display code library search path</i>
-------------------	---

---

**Description**

Display code library search path

**Usage**

```
code_library_path()
```

---

code_snapshot_files	<i>Git commit of ctl files, SourceData and Scripts</i>
---------------------	--

---

**Description**

This function should not be used directly.

**Usage**

```
code_snapshot_files(message = "created automatic snapshot",
                    session = TRUE, files_to_stage, ...)
```

**Arguments**

message	character. Description to be added to commit
session	logical. Should sessionInfo be included in commit message
files_to_stage	character vector. file paths to add to commit
...	additional arguments for git2r::commit

---

commit_file	<i>commit individual file(s)</i>
-------------	----------------------------------

---

**Description**

Has side effect that staged changed will be updated to working tree

**Usage**

```
commit_file(file_name)
```

**Arguments**

file\_name          character vector. File(s) to be committed

**Examples**

```
## Not run:  
commit_file("Scripts/script1.R")  
  
## End(Not run)
```

---

copy_file	<i>Copy file to project directory</i>
-----------	---------------------------------------

---

**Description**

Will copy any file from an external location (e.g. code library) into project No modification of that file will take place

**Usage**

```
copy_file(from, dest, overwrite = FALSE, alt_paths,  
          version_control = FALSE)
```

**Arguments**

from                character. file name or path of file to copy  
dest                character. file name file to create  
overwrite          logical. Overwrite 'to' file if exists?  
alt\_paths          character vector. paths to other candidate files to search  
version\_control    logical. Should file be added to version control (default = FALSE)

---

copy_script	<i>Copy script to project directory</i>
-------------	---

---

### Description

Will search code library and copy script and dependencies into scripts directory. Script will also be stamped with source location, time and user information

### Usage

```
copy_script(from, to, dependencies = TRUE, stamp_copy = TRUE,
            overwrite = FALSE, comment_char = "#", alt_paths, proj_path = ".")
```

### Arguments

from	character. file name or path of file to copy
to	character. file name file to create
dependencies	logical. Default = TRUE. will script copy dependencies
stamp_copy	logical. Create a commented timestamp at beginning of file
overwrite	logical. Overwrite 'to' file if exists?
comment_char	character. Comment character
alt_paths	character vector. paths to other candidate files to search
proj_path	character. Default = current working directory. path to MSDproject

---

dependency_tree	<i>Create a recursive dependency_tree of file names</i>
-----------------	---

---

### Description

Create a recursive dependency\_tree of file names

### Usage

```
dependency_tree(from, already_got = NULL)
```

### Arguments

from	character. file name or path of file to copy
already_got	logical. Default = NULL. If true will terminate loop Used by copy_script. Creates an an ordered vector of script dependencies.

---

do_test	<i>Testing interfact</i>
---------	--------------------------

---

**Description**

Testing interfact

**Usage**

```
do_test(..., silent = FALSE, append = data.frame())
```

**Arguments**

...	named expression to evaluate
silent	logical. Default = FALSE. Should messages be printed.
append	data.frame. Default = empty. Existing set of results to append to

---

get_github_code_library	<i>Download code repositor from github</i>
-------------------------	--

---

**Description**

Download code repositor from github

**Usage**

```
get_github_code_library(local_path, giturl, config_file)
```

**Arguments**

local_path	character. Path to install repository
giturl	character. URL to github repository
config_file	character. Name of config file (e.g. "~/.Rprofile")

---

get_script_field	<i>Get field from a code file</i>
------------------	-----------------------------------

---

**Description**

Get field from a code file

**Usage**

```
get_script_field(file_name, field_name, n = 10)
```

**Arguments**

file_name	character. path to file
field_name	character. name of field to find.
n	numeric. number of lines of each file to search

---

info_scripts	<i>List information about scripts</i>
--------------	---------------------------------------

---

**Description**

List information about scripts

**Usage**

```
info_scripts(files, fields = c("Description"), viewer = TRUE,
  silent = FALSE, base_dirs = NULL, shorten_paths = TRUE)
```

**Arguments**

files	vector string of file names/paths
fields	vector string of field tags to display
viewer	logical indicating if Rstudio viewer should be used (default = TRUE)
silent	run in quiet mode (default=FALSE)
base_dirs	character vector. group files together that belong to these directory paths
shorten_paths	logical. Default = TRUE. Long paths will be shortened if true in displayed output (not returned object)

**Examples**

```
## Not run:
ls_scripts('~ /AZD6094/PK_liver4/') %>%
  info_scripts('Description') %>%
  filter(grepl('mod',DESCRIPTION))

## End(Not run)
```



---

is_full_path	<i>Test if full path</i>
--------------	--------------------------

---

**Description**

Test if full path

**Usage**

```
is_full_path(x)
```

**Arguments**

x                      string giving file/path name

**Value**

TRUE only when path starts with ~, /, \ or X: (i.e. when x is a full path), FALSE otherwise

**Examples**

```
## Not run:  
is_full_path('file.text.ext')  
is_full_path('/path/to/file.text.ext')  
  
## End(Not run)
```

---

is_rstudio	<i>Logical flag for detecting if R session is on rstudio not</i>
------------	--

---

**Description**

Logical flag for detecting if R session is on rstudio not

**Usage**

```
is_rstudio()
```

---

locate_file	<i>Locate file from search path</i>
-------------	-------------------------------------

---

**Description**

Finds first file in search\_path that exists

**Usage**

```
locate_file(x, search_path = c("."), recursive = FALSE)
```

**Arguments**

x	string for file name
search_path	vector of strings giving search path
recursive	logical. Default TRUE. whether to do recursive search or not

**Value**

Path of located file. Returns error if file not found.

**Examples**

```
## Not run:  
locate_file('script.R',c('.', 'Scripts')) ## looks in current working directory, then Scripts folder  
  
## End(Not run)
```

---

ls_code_library	<i>List files in code library</i>
-----------------	-----------------------------------

---

**Description**

List files in code library

**Usage**

```
ls_code_library()
```

---

ls_scripts	<i>List scripts</i>
------------	---------------------

---

**Description**

List scripts

**Usage**

```
ls_scripts(folder = ".", extn = "r|R", recursive = TRUE)
```

**Arguments**

folder	string describing folder to search recursively in
extn	character (can be regex) giving extension to limit search to
recursive	by default TRUE

**Examples**

```
## Not run:
ls_scripts('~ /AZD6094/PK_liver4/') %>%
  info_scripts('Description') %>%
  filter(grepl('mod', DESCRIPTION))

## End(Not run)
```

---

make_local_bare	<i>create local bare repository</i>
-----------------	-------------------------------------

---

**Description**

create local bare repository

**Usage**

```
make_local_bare(proj_name = getwd())
```

**Arguments**

proj_name	character vector indicating path to MSDproject
-----------	--

---

make_project	<i>Create new_project</i>
--------------	---------------------------

---

### Description

Creates directory structure. User install MSDproject again in

### Usage

```
make_project(proj_name, keep_readme_file = TRUE,
             remove_user_lib = TRUE, overwrite_rprofile = TRUE)
```

### Arguments

proj_name	character string of full path to new_project
keep_readme_file	logical (default=TRUE) if you do not want to see readme files in project use FALSE
remove_user_lib	logical (default=TRUE) if TRUE will attempt to remove user R package library from .libPaths()
overwrite_rprofile	logical. should project .Rprofile be overwritten (default=FALSE)

---

models_dir	<i>Shortcut to Models directory</i>
------------	-------------------------------------

---

### Description

Shortcut to Models directory

### Usage

```
models_dir(proj_name = getwd())
```

### Arguments

proj_name	character. Working directory (default = getwd())
-----------	--

**Description**

MSDproject is designed to manage tidy project directories and facilitate reproducibility and reuse of code

**Strategy**

- Code management
  - Project structure: Defines standardised directory structure for statistical projects
  - Long term reproducibility: Local (project level) installation of packages
  - Code sharing/reuse: code library + search functionality

**Options**

`scripts.dir` = names of the 'scripts' directory in the project

`models.dir` = names of the 'models' directory in the project

`git.exists` = TRUE if git is installed (only works on unix currently)

`code_library_path` = character vector. paths of directories containing code files

To modify these options use the `options()` interface. To modify these values permanently set `options()` in `~/.Rprofile` or `R_HOME/etc/Rprofile.site`

**Examples**

```
## Not run:
make_project('~/.MSDXXX/PKAE1')      ## creates new_project
code_library()                        ## display summary of code_library
preview('template1.R')               ## preview file in code_library
copy_script('template1.R')            ## copy_script from code_library to project

new_script('test.R')                  ## Creates empty test.R
copy_script('output.data.R')          ## copies from code_library
copy_script('../pathto/script.R')     ## copies from other location

## End(Not run)
```

**Description**

Create new R script

**Usage**

```
new_script(name, overwrite = FALSE, open_file = TRUE,
  libs = c("MSDproject"))
```

**Arguments**

name	character indicating name of script to create
overwrite	logical. Whether to overwrite existing file (default = FALSE)
open_file	logical. Whether function should open script (default = TRUE)
libs	character. What libraries to add.

---

preview	<i>Preview code_library file</i>
---------	----------------------------------

---

**Description**

Preview code\_library file

**Usage**

```
preview(name)
```

**Arguments**

name	character indicating script in code_library to preview
------	--

---

pull_repo	<i>pull repository</i>
-----------	------------------------

---

**Description**

pull repository

**Usage**

```
pull_repo(local_path)
```

**Arguments**

local_path	character. Path to repository
------------	-------------------------------

---

reexports	<i>Objects exported from other packages</i>
-----------	---

---

**Description**

These objects are imported from other packages. Follow the links below to see their documentation.

**magrittr** [%>%](#)

---

Renvironment_info	<i>Make R session record</i>
-------------------	------------------------------

---

**Description**

Create a record of the R version and package versions used in a particular MSDproject

**Usage**

```
Renvironment_info()
```

---

replace_code_library	<i>Replaces code library</i>
----------------------	------------------------------

---

**Description**

Replace code library search path with path(s)

**Usage**

```
replace_code_library(path)
```

**Arguments**

path	character vector with paths to attach to
------	--

---

resetwd	<i>Reset working directory to rstudio working directory</i>
---------	---

---

**Description**

Reset working directory to rstudio working directory

**Usage**

```
resetwd()
```

---

scripts_dir	<i>Shortcut to Scripts directory</i>
-------------	--------------------------------------

---

**Description**

Shortcut to Scripts directory

**Usage**

```
scripts_dir(proj_name = getwd())
```

**Arguments**

proj_name	character. Working directory (default = getwd())
-----------	--

---

search_field	<i>Search for files matching field values</i>
--------------	---

---

**Description**

Search for files matching field values

**Usage**

```
search_field(files, field, text)
```

**Arguments**

files	vector string of file names/paths
field	character. field name (can be regex) to search for
text	string (can be regex) to search for

---

search_keyword	<i>Search for files matching key words</i>
----------------	--

---

**Description**

Search for files matching key words

**Usage**

```
search_keyword(files, text)
```

**Arguments**

files	vector string of file names/paths
text	string (can be regex) to search for

---

search_raw	<i>Search for files matching raw text search</i>
------------	--

---

**Description**

Search for files matching raw text search

**Usage**

```
search_raw(files, text, search_title = TRUE, search_contents = TRUE)
```



**Arguments**

files	vector string of file names/paths
text	string (can be regex) to search for
search_title	logical (default=TRUE). should matching occur in title
search_contents	logical (default=TRUE). should matching occur in file contents

---

 setup\_file

*Setup files*


---

**Description**

This function should not be used directly. Only exported for use by dependent packages

**Usage**

```
setup_file(file_name, version_control = getOption("git.exists"))
```

**Arguments**

file_name	character. Indicating name of file to set up
version_control	logical. Should file be added to version control

---

 set\_project\_opts

*Set project options*


---

**Description**

Set project options

**Usage**

```
set_project_opts()
```

---

 short\_path

*shorten path name*


---

**Description**

shorten path name

**Usage**

```
short_path(x)
```

**Arguments**

x	character vector. Path to shorten.
---	------------------------------------

---

snapshot2	<i>Git commit of ctl files, SourceData and Scripts</i>
-----------	--

---

**Description**

Git commit of ctl files, SourceData and Scripts

**Usage**

```
snapshot2(message = "created automatic snapshot", session = TRUE, ...)
```

**Arguments**

message	character. Description to be added to commit
session	logical. Should sessionInfo be included in commit message
...	additional arguments for git2r::commit

---

system_cmd	<i>system/shell command wrapper</i>
------------	-------------------------------------

---

**Description**

system/shell command wrapper

**Usage**

```
system_cmd(cmd, dir = ".", ...)
```

**Arguments**

cmd	character. command to send to shell
dir	character. directory to run command in
...	other arguments passed to system command

---

wait_for	<i>Wait for statement to be true</i>
----------	--------------------------------------

---

**Description**

Wait for statement to be true

**Usage**

```
wait_for(x, timeout = NULL, interval = 1)
```

**Arguments**

x	expression to evaluate
timeout	numeric. Maximum time (seconds) to wait
interval	numeric. Number of seconds (default=1s) to wait till rechecking

# Index

## \*Topic **internal**

reexports, [14](#)

%>% (reexports), [14](#)

%>%, [14](#)

attach\_code\_library, [2](#)

check\_if\_MSDproject, [3](#)

check\_session, [3](#)

code\_library, [4](#)

code\_library\_path, [4](#)

code\_snapshot\_files, [4](#)

commit\_file, [5](#)

copy\_file, [5](#)

copy\_script, [6](#)

dependency\_tree, [6](#)

do\_test, [7](#)

get\_github\_code\_library, [7](#)

get\_script\_field, [8](#)

info\_scripts, [8](#)

is\_full\_path, [9](#)

is\_rstudio, [9](#)

locate\_file, [10](#)

ls\_code\_library, [10](#)

ls\_scripts, [11](#)

make\_local\_bare, [11](#)

make\_project, [12](#)

models\_dir, [12](#)

MSDproject, [13](#)

MSDproject-package (MSDproject), [13](#)

new\_script, [13](#)

preview, [14](#)

pull\_repo, [14](#)

reexports, [14](#)

Renvironment\_info, [15](#)

replace\_code\_library, [15](#)

resetwd, [15](#)

scripts\_dir, [15](#)

search\_field, [16](#)

search\_keyword, [16](#)

search\_raw, [16](#)

set\_project\_opts, [17](#)

setup\_file, [17](#)

short\_path, [17](#)

snapshot2, [18](#)

system\_cmd, [18](#)

wait\_for, [18](#)