# Package ‘ready4fun’

September 8, 2020

**Title** Readyforwhatsnext Function Authoring And Documentation Tools

**Version** 0.0.0.9146

**Description** ready4fun is a collection of functions for authoring code libraries of functions and datasets for use in mental health simulations developed within the readyforwhatsnext ecosystem.

**License** FALSE

**Encoding** UTF-8

**LazyData** true

**Roxygen** list(markdown = TRUE)

**RoxygenNote** 7.1.1

**Depends** R (>= 2.10)

**Imports** methods,
Hmisc,
devtools,
dplyr,
magrittr,
purrr,
readxl,
rlang,
sinew,
stats,
stringi,
stringr,
testit,
tibble,
tools,
usethis,
xfun

## R topics documented:

## `ready4fun-package` *ready4fun: Readyforwhatsnext Function Authoring And Documentation Tools*

### Description

ready4fun is a collection of functions for authoring code libraries of functions and datasets for use in mental health simulations developed within the readyforwhatsnext ecosystem.

### Details

To learn more about ready4fun, start with the vignettes: `browseVignettes(package = "ready4fun")`

### Author(s)

**Maintainer**: Matthew Hamilton <`matthew.hamilton@orygen.org.au`> (ORCID)

Authors:

- Glen Wiesner <`Glen.Wiesner@vu.edu.au`> (ORCID)

Other contributors:

- Orygen [copyright holder, funder]
- VicHealth [funder]
- Victoria University [funder]

---

## `abbreviations_lup` *Common abbreviations lookup table*

---

### Description

A lookup table for abbreviations commonly used in object names in the ready4funpackage.

### Usage

```
abbreviations_lup
```

### Format

An object of class `tbl_df` (inherits from `tbl`, `data.frame`) with 198 rows and 3 columns.

### Details

A tibble

**short_name_chr** Short name (a character vector)

**long_name_chr** Long name (a character vector)

**plural_lgl** Plural (a logical vector)

### Source

`https://readyforwhatsnext.github.io/readyforwhatsnext/`

## add_indefartls_to_phrases

*Add indefinite articles to phrases*

### Description

add_indefartls_to_phrases() is an Add function that updates an object by adding data to that object. Specifically, this function implements an algorithm to add indefinite articles to phrases. Function argument abbreviated_phrase_1L_chr specifies the object to be updated. The function returns Phrases (a character vector).

### Usage

```
add_indefartls_to_phrases(
  abbreviated_phrase_1L_chr,
  abbreviations_lup = NULL,
  ignore_phrs_not_in_lup_1L_lgl = T
)
```

### Arguments

`abbreviated_phrase_1L_chr`
Abbreviated phrase (a character vector of length one)

`abbreviations_lup`
Abbreviations (a lookup table), Default: NULL

`ignore_phrs_not_in_lup_1L_lgl`
Ignore phrases not in lookup table (a logical vector of length one), Default: T

### Value

Phrases (a character vector)

## add_indef_artl_to_item

*Add indefinite article to item*

### Description

add_indef_artl_to_item() is an Add function that updates an object by adding data to that object. Specifically, this function implements an algorithm to add indefinite article to item. Function argument phrase_chr specifies the object to be updated. The function returns Indefinite item (a character vector).

### Usage

```
add_indef_artl_to_item(
  phrase_chr,
  abbreviations_lup = NULL,
  ignore_phrs_not_in_lup_1L_lgl = T
)
```

### Arguments

`phrase_chr` Phrase (a character vector)

`abbreviations_lup`
Abbreviations (a lookup table), Default: NULL

`ignore_phrs_not_in_lup_1L_lgl`
Ignore phrases not in lookup table (a logical vector of length one), Default: T

### Value

Indefinite item (a character vector)

---

## add_plurals_to_abbr_lup

*Add plurals to abbreviation*

---

### Description

add_plurals_to_abbr_lup() is an Add function that updates an object by adding data to that object. Specifically, this function implements an algorithm to add plurals to abbreviation lookup table. Function argument abbr_tb specifies the object to be updated. The function returns Abbreviation (a tibble).

### Usage

```
add_plurals_to_abbr_lup(
  abbr_tb,
  no_plural_chr = NA_character_,
  custom_plural_ls = NULL
)
```

### Arguments

`abbr_tb` Abbreviation (a tibble)

`no_plural_chr` No plural (a character vector), Default: 'NA'

`custom_plural_ls`
Custom plural (a list), Default: NULL

### Value

Abbreviation (a tibble)

## add_rows_to_fn_type_lup

*Add rows to function type*

### Description

add_rows_to_fn_type_lup() is an Add function that updates an object by adding data to that object. Specifically, this function implements an algorithm to add rows to function type lookup table. Function argument fn_type_lup_tb specifies the object to be updated. The function returns Updated function type lookup table (a tibble).

### Usage

```
add_rows_to_fn_type_lup(
  fn_type_lup_tb = make_fn_type_lup(),
  fn_type_nm_chr = NA_character_,
  fn_type_desc_chr = NA_character_,
  first_arg_desc_chr = NA_character_,
  second_arg_desc_chr = NA_character_,
  is_generic_lgl = F,
  is_method_lgl = F
)
```

### Arguments

| | |
|---|---|
| fn_type_lup_tb | Function type lookup table (a tibble), Default: make_fn_type_lup() |
| fn_type_nm_chr | Function type name (a character vector), Default: 'NA' |
| fn_type_desc_chr | Function type description (a character vector), Default: 'NA' |
| first_arg_desc_chr | First argument description (a character vector), Default: 'NA' |
| second_arg_desc_chr | Second argument description (a character vector), Default: 'NA' |
| is_generic_lgl | Is generic (a logical vector), Default: F |
| is_method_lgl | Is method (a logical vector), Default: F |

### Value

Updated function type lookup table (a tibble)

## `assert_inp_does_not_match_terms` *Assert input does not match terms*

### Description

assert_inp_does_not_match_terms() is an Assert function that validates that an object conforms to required condition(s). If the object does not meet all required conditions, program execution will be stopped and an error message provided. Specifically, this function implements an algorithm to assert input does not match terms. Function argument input_chr specifies the object on which assert validation checks are to be performed. Argument exclude_if_match_chr provides the object containing values used for validation tests. The function is called for its side effects and does not return a value.

### Usage

```
assert_inp_does_not_match_terms(input_chr, exclude_if_match_chr)
```

### Arguments

| | |
|---|---|
| `input_chr` | Input (a character vector) |
| `exclude_if_match_chr` | Exclude if match (a character vector) |

## `close_open_sinks` *Close open sinks*

### Description

close_open_sinks() is a Close function that closes specified connections. Specifically, this function implements an algorithm to close open sinks. The function is called for its side effects and does not return a value.

### Usage

```
close_open_sinks()
```

## `fn_type_lup_tb` *Function type lookup table*

### Description

A lookup table to find descriptions for different types of functions used within the ready4fun package suite.

### Usage

```
fn_type_lup_tb
```

### Format

An object of class `tbl_df` (inherits from `tbl`, `data.frame`) with 16 rows and 6 columns.

### Details

A tibble

**fn_type_nm_chr** Function type name (a character vector)

**fn_type_desc_chr** Function type description (a character vector)

**first_arg_desc_chr** First argument description (a character vector)

**second_arg_desc_chr** Second argument description (a character vector)

**is_generic_lgl** Is generic (a logical vector)

**is_method_lgl** Is method (a logical vector)

### Source

`https://readyforwhatsnext.github.io/readyforwhatsnext/`

## `force_instl_of_reqd_pkg` *Force install of required package*

### Description

force_instl_of_reqd_pkg() is a Force function that checks if a specified local or global environmental condition is met and if not, updates the specified environment to comply with the condition. Specifically, this function implements an algorithm to force install of required package. The function is called for its side effects and does not return a value.

### Usage

```
force_instl_of_reqd_pkg(package_nm_1L_chr)
```

### Arguments

`package_nm_1L_chr`
Package name (a character vector of length one)

## `get_arg_obj_type` *Get argument object type*

### Description

get_arg_obj_type() is a Get function that retrieves a pre-existing data object from memory, local file system or online repository. Specifically, this function implements an algorithm to get argument object type. Function argument argument_nm_1L_chr specifies the where to look for the required object. The function returns Argument object type (a character vector of length one).

### Usage

```
get_arg_obj_type(argument_nm_1L_chr, object_type_lup = NULL)
```

### Arguments

`argument_nm_1L_chr`
: Argument name (a character vector of length one)

`object_type_lup`
: Object type (a lookup table), Default: NULL

### Value

Argument object type (a character vector of length one)

## `get_dev_pkg_nm` *Get development package name*

### Description

get_dev_pkg_nm() is a Get function that retrieves a pre-existing data object from memory, local file system or online repository. Specifically, this function implements an algorithm to get development package name. Function argument path_to_pkg_rt_1L_chr specifies the where to look for the required object. The function returns Development package name (a character vector of length one).

### Usage

```
get_dev_pkg_nm(path_to_pkg_rt_1L_chr = ".")
```

### Arguments

`path_to_pkg_rt_1L_chr`
: Path to package root (a character vector of length one), Default: '.'

### Value

Development package name (a character vector of length one)

## `get_fn_args` *Get function arguments*

### Description

get_fn_args() is a Get function that retrieves a pre-existing data object from memory, local file system or online repository. Specifically, this function implements an algorithm to get function arguments. Function argument fn specifies the where to look for the required object. The function returns Function arguments (a character vector).

### Usage

```
get_fn_args(fn)
```

### Arguments

| | |
|---|---|
| `fn` | Function (a function) |

### Value

Function arguments (a character vector)

## `get_fn_nms_in_file` *Get function names in file*

### Description

get_fn_nms_in_file() is a Get function that retrieves a pre-existing data object from memory, local file system or online repository. Specifically, this function implements an algorithm to get function names in file. Function argument path_1L_chr specifies the where to look for the required object. The function returns Local (a character vector).

### Usage

```
get_fn_nms_in_file(path_1L_chr)
```

### Arguments

| | |
|---|---|
| `path_1L_chr` | Path (a character vector of length one) |

### Value

Local (a character vector)

## get_from_lup_obj *Get from lookup table object*

### Description

get_from_lup_obj() is a Get function that retrieves a pre-existing data object from memory, local file system or online repository. Specifically, this function implements an algorithm to get from lookup table object. Function argument data_lookup_tb specifies the where to look for the required object. The function returns Return object (an output object of multiple potential types).

### Usage

```
get_from_lup_obj(
  data_lookup_tb,
  match_value_xx,
  match_var_nm_1L_chr,
  target_var_nm_1L_chr,
  evaluate_lgl = TRUE
)
```

### Arguments

| | |
|---|---|
| `data_lookup_tb` | Data lookup (a tibble) |
| `match_value_xx` | Match value (an output object of multiple potential types) |
| `match_var_nm_1L_chr` | Match var name (a character vector of length one) |
| `target_var_nm_1L_chr` | Target var name (a character vector of length one) |
| `evaluate_lgl` | Evaluate (a logical vector), Default: TRUE |

### Value

Return object (an output object of multiple potential types)

## get_outp_obj_type *Get output object type*

### Description

get_outp_obj_type() is a Get function that retrieves a pre-existing data object from memory, local file system or online repository. Specifically, this function implements an algorithm to get output object type. Function argument fns_chr specifies the where to look for the required object. The function returns Output object type (a character vector).

### Usage

```
get_outp_obj_type(fns_chr)
```

### Arguments

| | |
|---|---|
| `fns_chr` | Functions (a character vector) |

### Value

Output object type (a character vector)

## `get_r4_obj_slots` *Get readyforwhatsnext S4 object slots*

### Description

get_r4_obj_slots() is a Get function that retrieves a pre-existing data object from memory, local file system or online repository. Specifically, this function implements an algorithm to get readyforwhatsnext s4 object slots. Function argument fn_name_1L_chr specifies the where to look for the required object. The function returns Slots (a character vector).

### Usage

```
get_r4_obj_slots(fn_name_1L_chr, package_1L_chr = "")
```

### Arguments

| | |
|---|---|
| `fn_name_1L_chr` | Function name (a character vector of length one) |
| `package_1L_chr` | Package (a character vector of length one), Default: ” |

### Value

Slots (a character vector)

## `get_return_obj_nm` *Get return object name*

### Description

get_return_obj_nm() is a Get function that retrieves a pre-existing data object from memory, local file system or online repository. Specifically, this function implements an algorithm to get return object name. Function argument fn specifies the where to look for the required object. The function returns Return (a character vector of length one).

### Usage

```
get_return_obj_nm(fn)
```

### Arguments

| | |
|---|---|
| `fn` | Function (a function) |

### Value

Return (a character vector of length one)

## import_xls_sheets *Import Excel workbook sheets*

### Description

import_xls_sheets() is an Import function that reads a data object in its native format and converts it to an R object. Specifically, this function implements an algorithm to import excel workbook sheets. The function returns Tibble list (a list of tibbles).

### Usage

```
import_xls_sheets(range_1L_chr, sheet_names_chr, path_1L_chr)
```

### Arguments

`range_1L_chr` Range (a character vector of length one)

`sheet_names_chr`
Sheet names (a character vector)

`path_1L_chr` Path (a character vector of length one)

### Value

Tibble list (a list of tibbles)

## make_arg_desc *Make argument description*

### Description

make_arg_desc() is a Make function that creates a new R object. Specifically, this function implements an algorithm to make argument description. The function returns Argument description (a character vector).

### Usage

```
make_arg_desc(fn_args_chr, object_type_lup = NULL, abbreviations_lup = NULL)
```

### Arguments

`fn_args_chr` Function arguments (a character vector)

`object_type_lup`
Object type (a lookup table), Default: NULL

`abbreviations_lup`
Abbreviations (a lookup table), Default: NULL

### Value

Argument description (a character vector)

## make_arg_desc_ls *Make argument description*

### Description

make_arg_desc_ls() is a Make function that creates a new R object. Specifically, this function implements an algorithm to make argument description list. The function is called for its side effects and does not return a value.

### Usage

```
make_arg_desc_ls(fn_nms_chr, abbreviations_lup = NULL, object_type_lup = NULL)
```

### Arguments

`fn_nms_chr` Function names (a character vector)

`abbreviations_lup`
Abbreviations (a lookup table), Default: NULL

`object_type_lup`
Object type (a lookup table), Default: NULL

## make_arg_desc_spine *Make argument description spine*

### Description

make_arg_desc_spine() is a Make function that creates a new R object. Specifically, this function implements an algorithm to make argument description spine. The function is called for its side effects and does not return a value.

### Usage

```
make_arg_desc_spine(
  argument_nm_1L_chr,
  object_type_lup = NULL,
  abbreviations_lup = NULL
)
```

### Arguments

`argument_nm_1L_chr`
Argument name (a character vector of length one)

`object_type_lup`
Object type (a lookup table), Default: NULL

`abbreviations_lup`
Abbreviations (a lookup table), Default: NULL

### Value

NA ()

## make_arg_title *Make argument title*

### Description

make_arg_title() is a Make function that creates a new R object. Specifically, this function implements an algorithm to make argument title. The function returns Title (a character vector).

### Usage

```
make_arg_title(
  args_chr,
  match_chr,
  object_type_lup = NULL,
  abbreviations_lup = NULL
)
```

### Arguments

| | |
|---|---|
| `args_chr` | Arguments (a character vector) |
| `match_chr` | Match (a character vector) |
| `object_type_lup` | Object type (a lookup table), Default: NULL |
| `abbreviations_lup` | Abbreviations (a lookup table), Default: NULL |

### Value

Title (a character vector)

## make_arg_type *Make argument type*

### Description

make_arg_type() is a Make function that creates a new R object. Specifically, this function implements an algorithm to make argument type. The function returns Argument description (a character vector).

### Usage

```
make_arg_type(
  fn_args_chr,
  object_type_lup = NULL,
  abbreviations_lup = NULL,
  fn
)
```

### Arguments

| | |
|---|---|
| `fn_args_chr` | Function arguments (a character vector) |
| `object_type_lup` | Object type (a lookup table), Default: NULL |
| `abbreviations_lup` | Abbreviations (a lookup table), Default: NULL |
| `fn` | Function (a function) |

### Value

Argument description (a character vector)

---

## `make_arg_type_abbr` *Make argument type abbreviation*

---

### Description

make_arg_type_abbr() is a Make function that creates a new R object. Specifically, this function implements an algorithm to make argument type abbreviation. The function returns Argument type abbreviation (a character vector).

### Usage

```
make_arg_type_abbr(
  fn_args_chr,
  object_type_lup = NULL,
  abbreviations_lup = NULL
)
```

### Arguments

| | |
|---|---|
| `fn_args_chr` | Function arguments (a character vector) |
| `object_type_lup` | Object type (a lookup table), Default: NULL |
| `abbreviations_lup` | Abbreviations (a lookup table), Default: NULL |

### Value

Argument type abbreviation (a character vector)

## make_arg_type_abbr_spine *Make argument type abbreviation spine*

### Description

make_arg_type_abbr_spine() is a Make function that creates a new R object. Specifically, this function implements an algorithm to make argument type abbreviation spine. The function returns Argument type abbreviation spine (a character vector of length one).

### Usage

```
make_arg_type_abbr_spine(argument_nm_1L_chr, lup_tb)
```

### Arguments

| | |
|---|---|
| `argument_nm_1L_chr` | Argument name (a character vector of length one) |
| `lup_tb` | Lookup table (a tibble) |

### Value

Argument type abbreviation spine (a character vector of length one)

## make_arg_type_lup_ls *Make argument type*

### Description

make_arg_type_lup_ls() is a Make function that creates a new R object. Specifically, this function implements an algorithm to make argument type lookup table list. The function returns Lookup table list (a list of lookup tables).

### Usage

```
make_arg_type_lup_ls(object_type_lup = NULL)
```

### Arguments

| | |
|---|---|
| `object_type_lup` | Object type (a lookup table), Default: NULL |

### Value

Lookup table list (a list of lookup tables)

## make_dmt_for_all_fns *Make documentation for all functions*

### Description

make_dmt_for_all_fns() is a Make function that creates a new R object. Specifically, this function implements an algorithm to make documentation for all functions. The function returns All functions documentation (a tibble).

### Usage

```
make_dmt_for_all_fns(
  paths_ls = make_fn_nms(),
  undocumented_fns_dir_chr = make_undmtd_fns_dir_chr(),
  custom_dmt_ls = list(details_ls = NULL, inc_for_main_user_lgl_ls =
   list(force_true_chr = NA_character_, force_false_chr = NA_character_), args_ls_ls =
    NULL),
  fn_type_lup_tb,
  abbreviations_lup = NULL
)
```

### Arguments

`paths_ls` Paths (a list), Default: make_fn_nms()

`undocumented_fns_dir_chr` Undocumented functions directory (a character vector), Default: make_undmtd_fns_dir_chr()

`custom_dmt_ls` Custom documentation (a list), Default: list(details_ls = NULL, inc_for_main_user_lgl_ls = list(force_true_chr = NA_character_, force_false_chr = NA_character_), args_ls_ls = NULL)

`fn_type_lup_tb` Function type lookup table (a tibble)

`abbreviations_lup` Abbreviations (a lookup table), Default: NULL

### Value

All functions documentation (a tibble)

## make_fn_desc *Make function description*

### Description

make_fn_desc() is a Make function that creates a new R object. Specifically, this function implements an algorithm to make function description. The function returns Function description (a character vector).

### Usage

```
make_fn_desc(
  fns_chr,
  title_chr,
  output_chr,
  fn_type_lup_tb = NULL,
  abbreviations_lup = NULL,
  test_for_write_R_warning_fn = NULL
)
```

### Arguments

`fns_chr` Functions (a character vector)

`title_chr` Title (a character vector)

`output_chr` Output (a character vector)

`fn_type_lup_tb` Function type lookup table (a tibble), Default: NULL

`abbreviations_lup` Abbreviations (a lookup table), Default: NULL

`test_for_write_R_warning_fn` Test for write warning (a function), Default: NULL

### Value

Function description (a character vector)

---

## make_fn_desc_spine *Make function description spine*

---

### Description

make_fn_desc_spine() is a Make function that creates a new R object. Specifically, this function implements an algorithm to make function description spine. The function returns Function description spine (a character vector of length one).

### Usage

```
make_fn_desc_spine(
  fn_name_1L_chr,
  fn_title_1L_chr,
  fn_type_lup_tb = NULL,
  abbreviations_lup = NULL
)
```

### Arguments

`fn_name_1L_chr` Function name (a character vector of length one)

`fn_title_1L_chr` Function title (a character vector of length one)

`fn_type_lup_tb` Function type lookup table (a tibble), Default: NULL

`abbreviations_lup` Abbreviations (a lookup table), Default: NULL

### Value

Function description spine (a character vector of length one)

## make_fn_dmt_spine *Make function documentation spine*

### Description

make_fn_dmt_spine() is a Make function that creates a new R object. Specifically, this function implements an algorithm to make function documentation spine. The function returns Function documentation spine (a list of character vectors).

### Usage

```
make_fn_dmt_spine(
  fn_name_1L_chr,
  fn_type_1L_chr,
  fn_title_1L_chr = NA_character_,
  fn,
  details_1L_chr = NA_character_,
  example_1L_lgl = F,
  export_1L_lgl = T,
  class_name_1L_chr,
  doc_in_class_1L_lgl
)
```

### Arguments

| | |
|---|---|
| `fn_name_1L_chr` | Function name (a character vector of length one) |
| `fn_type_1L_chr` | Function type (a character vector of length one) |
| `fn_title_1L_chr` | Function title (a character vector of length one), Default: 'NA' |
| `fn` | Function (a function) |
| `details_1L_chr` | Details (a character vector of length one), Default: 'NA' |
| `example_1L_lgl` | Example (a logical vector of length one), Default: F |
| `export_1L_lgl` | Export (a logical vector of length one), Default: T |
| `class_name_1L_chr` | Class name (a character vector of length one) |
| `doc_in_class_1L_lgl` | Document in class (a logical vector of length one) |

### Value

Function documentation spine (a list of character vectors)

## make_fn_dmt_tbl *Make function documentation table*

### Description

make_fn_dmt_tbl() is a Make function that creates a new R object. Specifically, this function implements an algorithm to make function documentation table. The function returns Function documentation table (a tibble).

### Usage

```
make_fn_dmt_tbl(
  fns_path_chr,
  fns_dir_chr = make_undmtd_fns_dir_chr(),
  pkg_nm_1L_chr,
  custom_dmt_ls = list(title_ls = NULL, desc_ls = NULL, details_ls = NULL,
   inc_for_main_user_lgl_ls = NULL, output_ls = NULL, example_ls = NULL, args_ls_ls =
    NULL),
  append_1L_lgl = T,
  fn_type_lup_tb = NULL,
  abbreviations_lup = NULL,
  object_type_lup = NULL,
  test_for_write_R_warning_fn = NULL
)
```

### Arguments

| | |
|---|---|
| `fns_path_chr` | Functions path (a character vector) |
| `fns_dir_chr` | Functions directory (a character vector), Default: make_undmtd_fns_dir_chr() |
| `pkg_nm_1L_chr` | Package name (a character vector of length one) |
| `custom_dmt_ls` | Custom documentation (a list), Default: list(title_ls = NULL, desc_ls = NULL, details_ls = NULL, inc_for_main_user_lgl_ls = NULL, output_ls = NULL, example_ls = NULL, args_ls_ls = NULL) |
| `append_1L_lgl` | Append (a logical vector of length one), Default: T |
| `fn_type_lup_tb` | Function type lookup table (a tibble), Default: NULL |
| `abbreviations_lup` | Abbreviations (a lookup table), Default: NULL |
| `object_type_lup` | Object type (a lookup table), Default: NULL |
| `test_for_write_R_warning_fn` | Test for write warning (a function), Default: NULL |

### Value

Function documentation table (a tibble)

## make_fn_dmt_tbl_tpl *Make function documentation table template*

### Description

make_fn_dmt_tbl_tpl() is a Make function that creates a new R object. Specifically, this function implements an algorithm to make function documentation table template. The function returns Function documentation table (a tibble).

### Usage

```
make_fn_dmt_tbl_tpl(
  fns_path_chr,
  fns_dir_chr = make_undmtd_fns_dir_chr(),
  fn_type_lup_tb = NULL,
  abbreviations_lup = NULL,
  object_type_lup = NULL,
  test_for_write_R_warning_fn = NULL
)
```

### Arguments

| | |
|---|---|
| fns_path_chr | Functions path (a character vector) |
| fns_dir_chr | Functions directory (a character vector), Default: make_undmtd_fns_dir_chr() |
| fn_type_lup_tb | Function type lookup table (a tibble), Default: NULL |
| abbreviations_lup | Abbreviations (a lookup table), Default: NULL |
| object_type_lup | Object type (a lookup table), Default: NULL |
| test_for_write_R_warning_fn | Test for write warning (a function), Default: NULL |

### Value

Function documentation table (a tibble)

## make_fn_nms *Make function names*

### Description

make_fn_nms() is a Make function that creates a new R object. Specifically, this function implements an algorithm to make function names. The function returns Functions (a list of character vectors of length one).

### Usage

```
make_fn_nms(path_1L_chr = "data-raw")
```

### Arguments

| | |
|---|---|
| `path_1L_chr` | Path (a character vector of length one), Default: 'data-raw' |

### Value

Functions (a list of character vectors of length one)

## `make_fn_title` *Make function title*

### Description

make_fn_title() is a Make function that creates a new R object. Specifically, this function implements an algorithm to make function title. The function returns Title (a character vector).

### Usage

```
make_fn_title(
  fns_chr,
  object_type_lup = NULL,
  abbreviations_lup = NULL,
  is_generic_lgl = F
)
```

### Arguments

| | |
|---|---|
| `fns_chr` | Functions (a character vector) |
| `object_type_lup` | Object type (a lookup table), Default: NULL |
| `abbreviations_lup` | Abbreviations (a lookup table), Default: NULL |
| `is_generic_lgl` | Is generic (a logical vector), Default: F |

### Value

Title (a character vector)

## `make_fn_types` *Make function types*

### Description

make_fn_types() is a Make function that creates a new R object. Specifically, this function implements an algorithm to make function types. The function returns Functions type (a character vector).

### Usage

```
make_fn_types()
```

## Value

Functions type (a character vector)

---

## make_fn_type_lup *Make function type*

---

### Description

make_fn_type_lup() is a Make function that creates a new R object. Specifically, this function implements an algorithm to make function type lookup table. The function returns Function type lookup table (a tibble).

### Usage

```
make_fn_type_lup(
  fn_type_nm_chr = character(0),
  fn_type_desc_chr = character(0),
  first_arg_desc_chr = character(0),
  second_arg_desc_chr = character(0),
  is_generic_lgl = logical(0),
  is_method_lgl = logical(0)
)
```

### Arguments

| | |
|---|---|
| fn_type_nm_chr | Function type name (a character vector), Default: character(0) |
| fn_type_desc_chr | Function type description (a character vector), Default: character(0) |
| first_arg_desc_chr | First argument description (a character vector), Default: character(0) |
| second_arg_desc_chr | Second argument description (a character vector), Default: character(0) |
| is_generic_lgl | Is generic (a logical vector), Default: logical(0) |
| is_method_lgl | Is method (a logical vector), Default: logical(0) |

### Value

Function type lookup table (a tibble)

## make_gtr_str_dmt_spine *Make getter setter documentation spine*

### Description

make_gtr_str_dmt_spine() is a Make function that creates a new R object. Specifically, this function implements an algorithm to make getter setter documentation spine. The function returns Getter setter documentation spine (a list of character vectors).

### Usage

```
make_gtr_str_dmt_spine(
  fn_type_1L_chr,
  fn_name_1L_chr,
  class_name_1L_chr,
  doc_in_class_1L_lgl,
  example_1L_lgl = F
)
```

### Arguments

| | |
|---|---|
| `fn_type_1L_chr` | Function type (a character vector of length one) |
| `fn_name_1L_chr` | Function name (a character vector of length one) |
| `class_name_1L_chr` | Class name (a character vector of length one) |
| `doc_in_class_1L_lgl` | Document in class (a logical vector of length one) |
| `example_1L_lgl` | Example (a logical vector of length one), Default: F |

### Value

Getter setter documentation spine (a list of character vectors)

## make_lines_for_fn_dmt *Make lines for function documentation*

### Description

make_lines_for_fn_dmt() is a Make function that creates a new R object. Specifically, this function implements an algorithm to make lines for function documentation. The function is called for its side effects and does not return a value.

### Usage

```
make_lines_for_fn_dmt(
  fn_name_1L_chr,
  fn_type_1L_chr,
  fn = NULL,
  fn_desc_1L_chr = NA_character_,
  fn_out_type_1L_chr = NA_character_,
  fn_title_1L_chr = NA_character_,
  example_1L_lgl = F,
  export_1L_lgl = T,
  class_name_1L_chr = "",
  details_1L_chr = "DETAILS",
  args_ls = NULL,
  import_chr = NA_character_,
  doc_in_class_1L_lgl = F,
  abbreviations_lup = NULL,
  object_type_lup = NULL
)
```

### Arguments

`fn_name_1L_chr` Function name (a character vector of length one)

`fn_type_1L_chr` Function type (a character vector of length one)

`fn` Function (a function), Default: NULL

`fn_desc_1L_chr` Function description (a character vector of length one), Default: 'NA'

`fn_out_type_1L_chr`
Function out type (a character vector of length one), Default: 'NA'

`fn_title_1L_chr`
Function title (a character vector of length one), Default: 'NA'

`example_1L_lgl` Example (a logical vector of length one), Default: F

`export_1L_lgl` Export (a logical vector of length one), Default: T

`class_name_1L_chr`
Class name (a character vector of length one), Default: ”

`details_1L_chr` Details (a character vector of length one), Default: 'DETAILS'

`args_ls` Arguments (a list), Default: NULL

`import_chr` Import (a character vector), Default: 'NA'

`doc_in_class_1L_lgl`
Document in class (a logical vector of length one), Default: F

`abbreviations_lup`
Abbreviations (a lookup table), Default: NULL

`object_type_lup`
Object type (a lookup table), Default: NULL

## make_new_fn_dmt *Make new function documentation*

### Description

make_new_fn_dmt() is a Make function that creates a new R object. Specifically, this function implements an algorithm to make new function documentation. The function returns New function documentation (a list of character vectors).

### Usage

```
make_new_fn_dmt(
  fn_type_1L_chr,
  fn_name_1L_chr,
  fn_desc_1L_chr = NA_character_,
  fn_det_1L_chr = NA_character_,
  fn_out_type_1L_chr = NA_character_,
  args_ls = NULL,
  fn = NULL,
  abbreviations_lup = NULL,
  object_type_lup = NULL
)
```

### Arguments

| | |
|---|---|
| `fn_type_1L_chr` | Function type (a character vector of length one) |
| `fn_name_1L_chr` | Function name (a character vector of length one) |
| `fn_desc_1L_chr` | Function description (a character vector of length one), Default: 'NA' |
| `fn_det_1L_chr` | Function det (a character vector of length one), Default: 'NA' |
| `fn_out_type_1L_chr` | Function out type (a character vector of length one), Default: 'NA' |
| `args_ls` | Arguments (a list), Default: NULL |
| `fn` | Function (a function), Default: NULL |
| `abbreviations_lup` | Abbreviations (a lookup table), Default: NULL |
| `object_type_lup` | Object type (a lookup table), Default: NULL |

### Value

New function documentation (a list of character vectors)

## `make_obj_lup` *Make object*

### Description

make_obj_lup() is a Make function that creates a new R object. Specifically, this function implements an algorithm to make object lookup table. The function returns Object (a tibble).

### Usage

```
make_obj_lup()
```

### Value

Object (a tibble)

## `make_ret_obj_desc` *Make ret object description*

### Description

make_ret_obj_desc() is a Make function that creates a new R object. Specifically, this function implements an algorithm to make ret object description. The function returns Ret object description (a character vector of length one).

### Usage

```
make_ret_obj_desc(fn, abbreviations_lup, starts_sentence_1L_lgl = T)
```

### Arguments

| | |
|---|---|
| `fn` | Function (a function) |
| `abbreviations_lup` | Abbreviations (a lookup table) |
| `starts_sentence_1L_lgl` | Starts sentence (a logical vector of length one), Default: T |

### Value

Ret object description (a character vector of length one)

## make_short_long_nms_vec *Make short long names vec*

### Description

make_short_long_nms_vec() is a Make function that creates a new R object. Specifically, this function implements an algorithm to make short long names vec. The function returns Short and long vec (a character vector).

### Usage

```
make_short_long_nms_vec(
  long_vecs_chr = character(0),
  short_vecs_chr = character(0)
)
```

### Arguments

| | |
|---|---|
| long_vecs_chr | Long vecs (a character vector), Default: character(0) |
| short_vecs_chr | Short vecs (a character vector), Default: character(0) |

### Value

Short and long vec (a character vector)

## make_std_fn_dmt_spine *Make standard function documentation spine*

### Description

make_std_fn_dmt_spine() is a Make function that creates a new R object. Specifically, this function implements an algorithm to make standard function documentation spine. The function returns Standard function documentation spine (a list of character vectors).

### Usage

```
make_std_fn_dmt_spine(
  fn_name_1L_chr,
  fn_type_1L_chr,
  fn_title_1L_chr,
  fn,
  details_1L_chr = NA_character_,
  example_1L_lgl = F,
  export_1L_lgl = T,
  class_name_1L_chr = "",
  exclude_if_match_chr
)
```

## Arguments

| | |
|---|---|
| `fn_name_1L_chr` | Function name (a character vector of length one) |
| `fn_type_1L_chr` | Function type (a character vector of length one) |
| `fn_title_1L_chr` | Function title (a character vector of length one) |
| `fn` | Function (a function) |
| `details_1L_chr` | Details (a character vector of length one), Default: 'NA' |
| `example_1L_lgl` | Example (a logical vector of length one), Default: F |
| `export_1L_lgl` | Export (a logical vector of length one), Default: T |
| `class_name_1L_chr` | Class name (a character vector of length one), Default: " |
| `exclude_if_match_chr` | Exclude if match (a character vector) |

## Value

Standard function documentation spine (a list of character vectors)

---

# `make_undmtd_fns_dir_chr` *Make undocumented functions directory*

---

## Description

make_undmtd_fns_dir_chr() is a Make function that creates a new R object. Specifically, this function implements an algorithm to make undocumented functions directory character vector. The function returns Undocumented functions directory (a character vector).

## Usage

```
make_undmtd_fns_dir_chr(path_1L_chr = "data-raw")
```

## Arguments

| | |
|---|---|
| `path_1L_chr` | Path (a character vector of length one), Default: 'data-raw' |

## Value

Undocumented functions directory (a character vector)

## `object_type_lup` *Object abbreviations lookup table*

### Description

A lookup table to identify R object types from an abbreviation that can be used as object name suffices.

### Usage

```
object_type_lup
```

### Format

An object of class `tbl_df` (inherits from `tbl`, `data.frame`) with 109 rows and 3 columns.

### Details

A tibble

- **short_name_chr** Short name (a character vector)
- **long_name_chr** Long name (a character vector)
- **plural_lgl** Plural (a logical vector)

### Source

https://readyforwhatsnext.github.io/readyforwhatsnext/

## `read_fns` *Read functions*

### Description

read_fns() is a Read function that reads an R script into memory. Specifically, this function implements an algorithm to read functions. Function argument fns_dir_1L_chr specifies the path to object. The function returns Functions path (a character vector).

### Usage

```
read_fns(fns_dir_1L_chr = "data-raw/fns/")
```

### Arguments

`fns_dir_1L_chr` Functions directory (a character vector of length one), Default: 'data-raw/fns/'

### Value

Functions path (a character vector)

## remove_obj_type_from_nm

*Remove object type from name*

### Description

remove_obj_type_from_nm() is a Remove function that edits an object, removing a specified element or elements. Specifically, this function implements an algorithm to remove object type from name. Function argument nms_chr specifies the object to be updated. Argument object_type_lup provides the object to be updated. The function returns Names (a character vector).

### Usage

```
remove_obj_type_from_nm(
  nms_chr,
  object_type_lup = NULL,
  abbreviations_lup = NULL,
  is_generic_lgl = F
)
```

### Arguments

| | |
|---|---|
| nms_chr | Names (a character vector) |
| object_type_lup | Object type (a lookup table), Default: NULL |
| abbreviations_lup | Abbreviations (a lookup table), Default: NULL |
| is_generic_lgl | Is generic (a logical vector), Default: F |

### Value

Names (a character vector)

## replace_abbr

*Replace abbreviation*

### Description

replace_abbr() is a Replace function that edits an object, replacing a specified element with another specified element. Specifically, this function implements an algorithm to replace abbreviation. Function argument title_chr specifies the object to be updated. Argument abbreviations_lup provides the object to be updated. The function returns Title (a character vector).

### Usage

```
replace_abbr(title_chr, abbreviations_lup = NULL, collapse_lgl = T)
```

### Arguments

| | |
|---|---|
| `title_chr` | Title (a character vector) |
| `abbreviations_lup` | Abbreviations (a lookup table), Default: NULL |
| `collapse_lgl` | Collapse (a logical vector), Default: T |

### Value

Title (a character vector)

## rowbind_all_tbs_in_r4_obj *Rowbind all tibbles in readyforwhatsnext S4 object*

### Description

rowbind_all_tbs_in_r4_obj() is a Rowbind function that performs custom rowbind operations on table objects. Specifically, this function implements an algorithm to rowbind all tibbles in readyforwhatsnext s4 object. The function returns Tibbles (a readyforwhatsnext S4).

### Usage

```
rowbind_all_tbs_in_r4_obj(tbs_r4, second_tbs_r4, r4_name_1L_chr)
```

### Arguments

| | |
|---|---|
| `tbs_r4` | Tibbles (a readyforwhatsnext S4) |
| `second_tbs_r4` | Second tibbles (a readyforwhatsnext S4) |
| `r4_name_1L_chr` | Readyforwhatsnext S4 name (a character vector of length one) |

### Value

Tibbles (a readyforwhatsnext S4)

## rowbind_tbs_in_r4_obj *Rowbind tibbles in readyforwhatsnext S4 object*

### Description

rowbind_tbs_in_r4_obj() is a Rowbind function that performs custom rowbind operations on table objects. Specifically, this function implements an algorithm to rowbind tibbles in readyforwhatsnext s4 object. The function returns Tibbles (a readyforwhatsnext S4).

### Usage

```
rowbind_tbs_in_r4_obj(tbs_r4, slot_nm_1L_chr, second_tbs_r4, r4_name_1L_chr)
```

### Arguments

| | |
|---|---|
| `tbs_r4` | Tibbles (a readyforwhatsnext S4) |
| `slot_nm_1L_chr` | Slot name (a character vector of length one) |
| `second_tbs_r4` | Second tibbles (a readyforwhatsnext S4) |
| `r4_name_1L_chr` | Readyforwhatsnext S4 name (a character vector of length one) |

### Value

Tibbles (a readyforwhatsnext S4)

---

## `unload_packages` *Unload packages*

---

### Description

unload_packages() is an Unload function that performs a custom detaching of a package from the search path. Specifically, this function implements an algorithm to unload packages. Function argument package_chr specifies the package(s) to be detached from the search path. The function is called for its side effects and does not return a value.

### Usage

```
unload_packages(package_chr)
```

### Arguments

| | |
|---|---|
| `package_chr` | Package (a character vector) |

---

## `update_abbr_lup` *Update abbreviation*

---

### Description

update_abbr_lup() is an Update function that edits an object, while preserving core object attributes. Specifically, this function implements an algorithm to update abbreviation lookup table. Function argument abbr_tb specifies the object to be updated. Argument short_name_chr provides the object to be updated. The function returns Abbreviation (a tibble).

### Usage

```
update_abbr_lup(
  abbr_tb,
  short_name_chr,
  long_name_chr,
  no_plural_chr = NA_character_,
  custom_plural_ls = NULL,
  pfx_rgx = NA_character_
)
```

### Arguments

| | |
|---|---|
| abbr_tb | Abbreviation (a tibble) |
| short_name_chr | Short name (a character vector) |
| long_name_chr | Long name (a character vector) |
| no_plural_chr | No plural (a character vector), Default: 'NA' |
| custom_plural_ls | Custom plural (a list), Default: NULL |
| pfx_rgx | Prefix (a regular expression vector), Default: 'NA' |

### Value

Abbreviation (a tibble)

---

## update_first_word_case

*Update first word case*

---

### Description

update_first_word_case() is an Update function that edits an object, while preserving core object attributes. Specifically, this function implements an algorithm to update first word case. Function argument phrase_1L_chr specifies the object to be updated. Argument fn provides the object to be updated. The function returns Phrase (a character vector of length one).

### Usage

```
update_first_word_case(phrase_1L_chr, fn = tolower)
```

### Arguments

| | |
|---|---|
| phrase_1L_chr | Phrase (a character vector of length one) |
| fn | Function (a function), Default: tolower |

### Value

Phrase (a character vector of length one)

## `update_fns_dmt_tb` *Update functions documentation*

### Description

update_fns_dmt_tb() is an Update function that edits an object, while preserving core object attributes. Specifically, this function implements an algorithm to update functions documentation tibble. Function argument fns_dmt_tb specifies the object to be updated. Argument title_ls provides the object to be updated. The function returns Functions documentation (a tibble).

### Usage

```
update_fns_dmt_tb(
  fns_dmt_tb,
  title_ls = NULL,
  desc_ls = NULL,
  details_ls = NULL,
  inc_for_main_user_lgl_ls = NULL,
  output_ls = NULL,
  example_ls = NULL,
  args_ls_ls = NULL,
  append_1L_lgl = T
)
```

### Arguments

| | |
|---|---|
| `fns_dmt_tb` | Functions documentation (a tibble) |
| `title_ls` | Title (a list), Default: NULL |
| `desc_ls` | Description (a list), Default: NULL |
| `details_ls` | Details (a list), Default: NULL |
| `inc_for_main_user_lgl_ls` | Inc for main user (a list of logical vectors), Default: NULL |
| `output_ls` | Output (a list), Default: NULL |
| `example_ls` | Example (a list), Default: NULL |
| `args_ls_ls` | Arguments (a list of lists), Default: NULL |
| `append_1L_lgl` | Append (a logical vector of length one), Default: T |

### Value

Functions documentation (a tibble)

## update_fns_dmt_tb_chr_vars

*Update functions documentation tibble character vector vars*

### Description

update_fns_dmt_tb_chr_vars() is an Update function that edits an object, while preserving core object attributes. Specifically, this function implements an algorithm to update functions documentation tibble character vector vars. Function argument fns_dmt_tb specifies the object to be updated. Argument data_1L_chr provides the object to be updated. The function returns Functions documentation (a tibble).

### Usage

```
update_fns_dmt_tb_chr_vars(fns_dmt_tb, data_1L_chr, new_ls, append_1L_lgl)
```

### Arguments

| | |
|---|---|
| `fns_dmt_tb` | Functions documentation (a tibble) |
| `data_1L_chr` | Data (a character vector of length one) |
| `new_ls` | New (a list) |
| `append_1L_lgl` | Append (a logical vector of length one) |

### Value

Functions documentation (a tibble)

## update_fns_dmt_tb_lgl_vars

*Update functions documentation tibble logical vector vars*

### Description

update_fns_dmt_tb_lgl_vars() is an Update function that edits an object, while preserving core object attributes. Specifically, this function implements an algorithm to update functions documentation tibble logical vector vars. Function argument fns_dmt_tb specifies the object to be updated. Argument data_1L_chr provides the object to be updated. The function returns Functions documentation (a tibble).

### Usage

```
update_fns_dmt_tb_lgl_vars(fns_dmt_tb, data_1L_chr, new_ls)
```

### Arguments

| | |
|---|---|
| `fns_dmt_tb` | Functions documentation (a tibble) |
| `data_1L_chr` | Data (a character vector of length one) |
| `new_ls` | New (a list) |

### Value

Functions documentation (a tibble)

---

## `update_fns_dmt_tb_ls_vars` *Update functions documentation tibble list vars*

---

### Description

update_fns_dmt_tb_ls_vars() is an Update function that edits an object, while preserving core object attributes. Specifically, this function implements an algorithm to update functions documentation tibble list vars. Function argument fns_dmt_tb specifies the object to be updated. Argument data_1L_chr provides the object to be updated. The function returns Functions documentation (a tibble).

### Usage

```
update_fns_dmt_tb_ls_vars(fns_dmt_tb, data_1L_chr, new_ls, append_1L_lgl)
```

### Arguments

| | |
|---|---|
| `fns_dmt_tb` | Functions documentation (a tibble) |
| `data_1L_chr` | Data (a character vector of length one) |
| `new_ls` | New (a list) |
| `append_1L_lgl` | Append (a logical vector of length one) |

### Value

Functions documentation (a tibble)

---

## `update_fn_dmt` *Update function documentation*

---

### Description

update_fn_dmt() is an Update function that edits an object, while preserving core object attributes. Specifically, this function implements an algorithm to update function documentation. Function argument fn_tags_spine_ls specifies the object to be updated. Argument new_tag_chr_ls provides the object to be updated. The function returns Function documentation (a character vector of length one).

### Usage

```
update_fn_dmt(
  fn_tags_spine_ls,
  new_tag_chr_ls,
  fn_name_1L_chr,
  fn_type_1L_chr,
  import_chr,
  abbreviations_lup = NULL
)
```

## Arguments

| | |
|---|---|
| `fn_tags_spine_ls` | Function tags spine (a list) |
| `new_tag_chr_ls` | New tag (a list of character vectors) |
| `fn_name_1L_chr` | Function name (a character vector of length one) |
| `fn_type_1L_chr` | Function type (a character vector of length one) |
| `import_chr` | Import (a character vector) |
| `abbreviations_lup` | Abbreviations (a lookup table), Default: NULL |

## Value

Function documentation (a character vector of length one)

---

# `update_fn_dmt_with_slots` *Update function documentation with slots*

---

## Description

update_fn_dmt_with_slots() is an Update function that edits an object, while preserving core object attributes. Specifically, this function implements an algorithm to update function documentation with slots. Function argument fn_name_1L_chr specifies the object to be updated. Argument fn_dmt_1L_chr provides the object to be updated. The function returns Function documentation (a character vector of length one).

## Usage

```
update_fn_dmt_with_slots(fn_name_1L_chr, fn_dmt_1L_chr)
```

## Arguments

| | |
|---|---|
| `fn_name_1L_chr` | Function name (a character vector of length one) |
| `fn_dmt_1L_chr` | Function documentation (a character vector of length one) |

## Value

Function documentation (a character vector of length one)

## update_ns *Update namespace*

### Description

update_ns() is an Update function that edits an object, while preserving core object attributes. Specifically, this function implements an algorithm to update namespace. Function argument package_1L_chr specifies the object to be updated. The function returns Package name (a character vector).

### Usage

```
update_ns(package_1L_chr)
```

### Arguments

| | |
|---|---|
| package_1L_chr | Package (a character vector of length one) |

### Value

Package name (a character vector)

## write_abbr_lup *Write abbreviation*

### Description

write_abbr_lup() is a Write function that writes a file to a specified local directory. Specifically, this function implements an algorithm to write abbreviation lookup table. The function is called for its side effects and does not return a value. WARNING: This function writes R scripts to your local environment. Make sure to only use if you want this behaviour

### Usage

```
write_abbr_lup(
  short_name_chr = NA_character_,
  long_name_chr = NA_character_,
  no_plural_chr = NA_character_,
  custom_plural_ls = NULL,
  overwrite_1L_lgl = T,
  seed_lup = NULL,
  url_1L_chr,
  pkg_nm_1L_chr = get_dev_pkg_nm()
)
```

### Arguments

| | |
|---|---|
| `short_name_chr` | Short name (a character vector), Default: 'NA' |
| `long_name_chr` | Long name (a character vector), Default: 'NA' |
| `no_plural_chr` | No plural (a character vector), Default: 'NA' |
| `custom_plural_ls` | Custom plural (a list), Default: NULL |
| `overwrite_1L_lgl` | Overwrite (a logical vector of length one), Default: T |
| `seed_lup` | Seed (a lookup table), Default: NULL |
| `url_1L_chr` | Url (a character vector of length one) |
| `pkg_nm_1L_chr` | Package name (a character vector of length one), Default: get_dev_pkg_nm() |

## write_all_tbs_in_tbs_r4_to_csvs

*Write all tibbles in tibbles readyforwhatsnext S4 to comma separated variables files*

### Description

write_all_tbs_in_tbs_r4_to_csvs() is a Write function that writes a file to a specified local directory. Specifically, this function implements an algorithm to write all tibbles in tibbles readyforwhatsnext s4 to comma separated variables files. The function is called for its side effects and does not return a value. WARNING: This function writes R scripts to your local environment. Make sure to only use if you want this behaviour

### Usage

```
write_all_tbs_in_tbs_r4_to_csvs(
  tbs_r4,
  r4_name_1L_chr,
  lup_dir_1L_chr,
  pfx_1L_chr
)
```

### Arguments

| | |
|---|---|
| `tbs_r4` | Tibbles (a readyforwhatsnext S4) |
| `r4_name_1L_chr` | Readyforwhatsnext S4 name (a character vector of length one) |
| `lup_dir_1L_chr` | Lookup table directory (a character vector of length one) |
| `pfx_1L_chr` | Prefix (a character vector of length one) |

## write_and_doc_ds *Write and document dataset*

### Description

write_and_doc_ds() is a Write function that writes a file to a specified local directory. Specifically, this function implements an algorithm to write and document dataset. The function is called for its side effects and does not return a value. WARNING: This function writes R scripts to your local environment. Make sure to only use if you want this behaviour

### Usage

```
write_and_doc_ds(
  db,
  overwrite_1L_lgl = T,
  db_1L_chr,
  title_1L_chr,
  desc_1L_chr,
  format_1L_chr = "A tibble",
  url_1L_chr = NA_character_,
  vars_ls = NULL,
  R_dir_1L_chr = "R",
  abbreviations_lup = NULL,
  object_type_lup = NULL
)
```

### Arguments

| | |
|---|---|
| db | PARAM_DESCRIPTION |
| overwrite_1L_lgl | Overwrite (a logical vector of length one), Default: T |
| db_1L_chr | Database (a character vector of length one) |
| title_1L_chr | Title (a character vector of length one) |
| desc_1L_chr | Description (a character vector of length one) |
| format_1L_chr | Format (a character vector of length one), Default: 'A tibble' |
| url_1L_chr | Url (a character vector of length one), Default: 'NA' |
| vars_ls | Vars (a list), Default: NULL |
| R_dir_1L_chr | R directory (a character vector of length one), Default: 'R' |
| abbreviations_lup | Abbreviations (a lookup table), Default: NULL |
| object_type_lup | Object type (a lookup table), Default: NULL |

## write_and_doc_fn_fls *Write and document function files*

### Description

write_and_doc_fn_fls() is a Write function that writes a file to a specified local directory. Specifically, this function implements an algorithm to write and document function files. The function is called for its side effects and does not return a value. WARNING: This function writes R scripts to your local environment. Make sure to only use if you want this behaviour

### Usage

```
write_and_doc_fn_fls(
  fns_dmt_tb,
  r_dir_1L_chr = "R",
  path_to_user_dmt_dir_1L_chr = "../../../../Documentation/Code/User",
  path_to_dvpr_dmt_dir_1L_chr = "../../../../Documentation/Code/Developer",
  make_pdfs_1L_lgl = T,
  dev_pkgs_chr = NA_character_
)
```

### Arguments

| | |
|---|---|
| `fns_dmt_tb` | Functions documentation (a tibble) |
| `r_dir_1L_chr` | R directory (a character vector of length one), Default: 'R' |
| `path_to_user_dmt_dir_1L_chr` | Path to user documentation directory (a character vector of length one), Default: '../../../../Documentation/Code/User' |
| `path_to_dvpr_dmt_dir_1L_chr` | Path to developer documentation directory (a character vector of length one), Default: '../../../../Documentation/Code/Developer' |
| `make_pdfs_1L_lgl` | Make pdfs (a logical vector of length one), Default: T |
| `dev_pkgs_chr` | Development packages (a character vector), Default: 'NA' |

## write_dmtd_fn_type_lup *Write documented function type*

### Description

write_dmtd_fn_type_lup() is a Write function that writes a file to a specified local directory. Specifically, this function implements an algorithm to write documented function type lookup table. The function is called for its side effects and does not return a value. WARNING: This function writes R scripts to your local environment. Make sure to only use if you want this behaviour

### Usage

```
write_dmtd_fn_type_lup(
  fn_type_lup_tb = make_fn_type_lup(),
  overwrite_1L_lgl = T,
  pkg_nm_1L_chr = get_dev_pkg_nm(),
  url_1L_chr = url_1L_chr,
  abbreviations_lup = NULL
)
```

### Arguments

`fn_type_lup_tb` Function type lookup table (a tibble), Default: make_fn_type_lup()

`overwrite_1L_lgl` Overwrite (a logical vector of length one), Default: T

`pkg_nm_1L_chr` Package name (a character vector of length one), Default: get_dev_pkg_nm()

`url_1L_chr` Url (a character vector of length one), Default: url_1L_chr

`abbreviations_lup` Abbreviations (a lookup table), Default: NULL

## write_documented_fns *Write documented functions*

### Description

write_documented_fns() is a Write function that writes a file to a specified local directory. Specifically, this function implements an algorithm to write documented functions. The function is called for its side effects and does not return a value. WARNING: This function writes R scripts to your local environment. Make sure to only use if you want this behaviour

### Usage

```
write_documented_fns(tmp_fn_dir_1L_chr, R_dir_1L_chr)
```

### Arguments

`tmp_fn_dir_1L_chr` Temporary function directory (a character vector of length one)

`R_dir_1L_chr` R directory (a character vector of length one)

## write_ds_dmt *Write dataset documentation*

### Description

write_ds_dmt() is a Write function that writes a file to a specified local directory. Specifically, this function implements an algorithm to write dataset documentation. The function is called for its side effects and does not return a value. WARNING: This function writes R scripts to your local environment. Make sure to only use if you want this behaviour

### Usage

```
write_ds_dmt(
  db,
  db_1L_chr,
  title_1L_chr,
  desc_1L_chr,
  format_1L_chr = "A tibble",
  url_1L_chr = NA_character_,
  vars_ls = NULL,
  R_dir_1L_chr = "R",
  abbreviations_lup = NULL,
  object_type_lup = NULL
)
```

### Arguments

| | |
|---|---|
| db | PARAM_DESCRIPTION |
| db_1L_chr | Database (a character vector of length one) |
| title_1L_chr | Title (a character vector of length one) |
| desc_1L_chr | Description (a character vector of length one) |
| format_1L_chr | Format (a character vector of length one), Default: 'A tibble' |
| url_1L_chr | Url (a character vector of length one), Default: 'NA' |
| vars_ls | Vars (a list), Default: NULL |
| R_dir_1L_chr | R directory (a character vector of length one), Default: 'R' |
| abbreviations_lup | Abbreviations (a lookup table), Default: NULL |
| object_type_lup | Object type (a lookup table), Default: NULL |

## `write_fn_fl` *Write function file*

### Description

write_fn_fl() is a Write function that writes a file to a specified local directory. Specifically, this function implements an algorithm to write function file. The function is called for its side effects and does not return a value. WARNING: This function writes R scripts to your local environment. Make sure to only use if you want this behaviour

### Usage

```
write_fn_fl(fns_dmt_tb, r_dir_1L_chr = "R", document_unexp_lgl = T)
```

### Arguments

| | |
|---|---|
| `fns_dmt_tb` | Functions documentation (a tibble) |
| `r_dir_1L_chr` | R directory (a character vector of length one), Default: 'R' |
| `document_unexp_lgl` | Document unexported (a logical vector), Default: T |

## `write_fn_type_dirs` *Write function type directories*

### Description

write_fn_type_dirs() is a Write function that writes a file to a specified local directory. Specifically, this function implements an algorithm to write function type directories. The function is called for its side effects and does not return a value. WARNING: This function writes R scripts to your local environment. Make sure to only use if you want this behaviour

### Usage

```
write_fn_type_dirs(path_1L_chr = "data-raw")
```

### Arguments

| | |
|---|---|
| `path_1L_chr` | Path (a character vector of length one), Default: 'data-raw' |

## write_from_tmp *Write from temporary*

### Description

write_from_tmp() is a Write function that writes a file to a specified local directory. Specifically, this function implements an algorithm to write from temporary. The function is called for its side effects and does not return a value. WARNING: This function writes R scripts to your local environment. Make sure to only use if you want this behaviour

### Usage

```
write_from_tmp(
  temp_path_1L_chr,
  dest_path_1L_chr,
  edit_fn = function(x) {     x },
  args_ls = NULL
)
```

### Arguments

| | |
|---|---|
| temp_path_1L_chr | Temp path (a character vector of length one) |
| dest_path_1L_chr | Dest path (a character vector of length one) |
| edit_fn | Edit (a function), Default: function(x) x |
| args_ls | Arguments (a list), Default: NULL |

## write_new_arg_sfxs *Write new argument sfxs*

### Description

write_new_arg_sfxs() is a Write function that writes a file to a specified local directory. Specifically, this function implements an algorithm to write new argument sfxs. The function returns Function arguments to rnm (a list).

### Usage

```
write_new_arg_sfxs(
  arg_nms_chr,
  fn_type_1L_chr,
  dir_path_chr,
  rt_dev_dir_path_1L_chr = normalizePath("../../../"),
  pkg_nm_1L_chr,
  inc_fns_idx_dbl = NA_real_
)
```

### Arguments

| | |
|---|---|
| `arg_nms_chr` | Argument names (a character vector) |
| `fn_type_1L_chr` | Function type (a character vector of length one) |
| `dir_path_chr` | Directory path (a character vector) |
| `rt_dev_dir_path_1L_chr` | Root development directory path (a character vector of length one), Default: normalizePath("../../../") |
| `pkg_nm_1L_chr` | Package name (a character vector of length one) |
| `inc_fns_idx_dbl` | Inc functions idx (a double vector), Default: NA |

### Value

Function arguments to rnm (a list)

## `write_ns_imps_to_desc` *Write namespace imports to description*

### Description

write_ns_imps_to_desc() is a Write function that writes a file to a specified local directory. Specifically, this function implements an algorithm to write namespace imports to description. The function is called for its side effects and does not return a value. WARNING: This function writes R scripts to your local environment. Make sure to only use if you want this behaviour

### Usage

```
write_ns_imps_to_desc(dev_pkgs_chr = NA_character_, incr_ver_1L_lgl = T)
```

### Arguments

| | |
|---|---|
| `dev_pkgs_chr` | Development packages (a character vector), Default: 'NA' |
| `incr_ver_1L_lgl` | Incr ver (a logical vector of length one), Default: T |

## `write_pkg` *Write package*

### Description

write_pkg() is a Write function that writes a file to a specified local directory. Specifically, this function implements an algorithm to write package. The function is called for its side effects and does not return a value. WARNING: This function writes R scripts to your local environment. Make sure to only use if you want this behaviour

### Usage

```
write_pkg(package_1L_chr, R_dir_1L_chr = "R")
```

### Arguments

package_1L_chr Package (a character vector of length one)

R_dir_1L_chr R directory (a character vector of length one), Default: 'R'

## write_pkg_setup_fls *Write package setup files*

### Description

write_pkg_setup_fls() is a Write function that writes a file to a specified local directory. Specifically, this function implements an algorithm to write package setup files. The function is called for its side effects and does not return a value. WARNING: This function writes R scripts to your local environment. Make sure to only use if you want this behaviour

### Usage

```
write_pkg_setup_fls(
  path_to_pkg_rt_1L_chr = ".",
  dev_pkg_nm_1L_chr = get_dev_pkg_nm(),
  make_tmpl_vignette_1L_lgl = F,
  incr_ver_1L_lgl = T,
  delete_contents_of_R_dir = F
)
```

### Arguments

path_to_pkg_rt_1L_chr
Path to package root (a character vector of length one), Default: '.'

dev_pkg_nm_1L_chr
Development package name (a character vector of length one), Default: get_dev_pkg_nm()

make_tmpl_vignette_1L_lgl
Make tmpl vignette (a logical vector of length one), Default: F

incr_ver_1L_lgl
Incr ver (a logical vector of length one), Default: T

delete_contents_of_R_dir
PARAM_DESCRIPTION, Default: F

## write_pt_lup_db *Write prototype lookup table database*

### Description

write_pt_lup_db() is a Write function that writes a file to a specified local directory. Specifically, this function implements an algorithm to write prototype lookup table database. The function is called for its side effects and does not return a value. WARNING: This function writes R scripts to your local environment. Make sure to only use if you want this behaviour

### Usage

```
write_pt_lup_db(R_dir_1L_chr = "R")
```

### Arguments

| | |
|---|---|
| `R_dir_1L_chr` | R directory (a character vector of length one), Default: 'R' |

## `write_std_imp` *Write standard import*

### Description

write_std_imp() is a Write function that writes a file to a specified local directory. Specifically, this function implements an algorithm to write standard import. The function is called for its side effects and does not return a value. WARNING: This function writes R scripts to your local environment. Make sure to only use if you want this behaviour

### Usage

```
write_std_imp(R_dir_1L_chr = "R")
```

### Arguments

| | |
|---|---|
| `R_dir_1L_chr` | R directory (a character vector of length one), Default: 'R' |

## `write_tb_to_csv` *Write tibble to comma separated variables file*

### Description

write_tb_to_csv() is a Write function that writes a file to a specified local directory. Specifically, this function implements an algorithm to write tibble to comma separated variables file. The function is called for its side effects and does not return a value. WARNING: This function writes R scripts to your local environment. Make sure to only use if you want this behaviour

### Usage

```
write_tb_to_csv(
  tbs_r4,
  slot_nm_1L_chr,
  r4_name_1L_chr,
  lup_dir_1L_chr,
  pfx_1L_chr
)
```

### Arguments

| | |
|---|---|
| `tbs_r4` | Tibbles (a readyforwhatsnext S4) |
| `slot_nm_1L_chr` | Slot name (a character vector of length one) |
| `r4_name_1L_chr` | Readyforwhatsnext S4 name (a character vector of length one) |
| `lup_dir_1L_chr` | Lookup table directory (a character vector of length one) |
| `pfx_1L_chr` | Prefix (a character vector of length one) |

## write_to_remove_collate

*Write to remove collate*

### Description

write_to_remove_collate() is a Write function that writes a file to a specified local directory. Specifically, this function implements an algorithm to write to remove collate. The function returns Description (a character vector).

### Usage

```
write_to_remove_collate(description_chr)
```

### Arguments

| | |
|---|---|
| `description_chr` | Description (a character vector) |

### Value

Description (a character vector)

## write_to_replace_fn_nms

*Write to replace function names*

### Description

write_to_replace_fn_nms() is a Write function that writes a file to a specified local directory. Specifically, this function implements an algorithm to write to replace function names. The function is called for its side effects and does not return a value. WARNING: This function writes R scripts to your local environment. Make sure to only use if you want this behaviour

### Usage

```
write_to_replace_fn_nms(
  rename_tb,
  undocumented_fns_dir_chr = make_undmtd_fns_dir_chr(),
  rt_dev_dir_path_1L_chr = normalizePath("../../../"),
  dev_pkg_nm_1L_chr = get_dev_pkg_nm()
)
```

### Arguments

`rename_tb` Rename (a tibble)

`undocumented_fns_dir_chr`
Undocumented functions directory (a character vector), Default: make_undmtd_fns_dir_chr()

`rt_dev_dir_path_1L_chr`
Root development directory path (a character vector of length one), Default: normalizePath("../../../")

`dev_pkg_nm_1L_chr`
Development package name (a character vector of length one), Default: get_dev_pkg_nm()

---

## write_to_replace_sfx_pair

*Write to replace suffix pair*

---

### Description

write_to_replace_sfx_pair() is a Write function that writes a file to a specified local directory. Specifically, this function implements an algorithm to write to replace suffix pair. The function is called for its side effects and does not return a value. WARNING: This function writes R scripts to your local environment. Make sure to only use if you want this behaviour

### Usage

```
write_to_replace_sfx_pair(
  args_nm_chr,
  sfxs_chr,
  replacements_chr,
  file_path_1L_chr = NA_character_,
  dir_path_1L_chr = NA_character_
)
```

### Arguments

`args_nm_chr` Arguments name (a character vector)

`sfxs_chr` Sfxs (a character vector)

`replacements_chr`
Replacements (a character vector)

`file_path_1L_chr`
File path (a character vector of length one), Default: 'NA'

`dir_path_1L_chr`
Directory path (a character vector of length one), Default: 'NA'

## write_to_reset_pkg_files

*Write to reset package files*

### Description

write_to_reset_pkg_files() is a Write function that writes a file to a specified local directory. Specifically, this function implements an algorithm to write to reset package files. The function is called for its side effects and does not return a value. WARNING: This function writes R scripts to your local environment. Make sure to only use if you want this behaviour

### Usage

```
write_to_reset_pkg_files(
  delete_contents_of_1L_chr,
  package_1L_chr = get_dev_pkg_nm(),
  package_dir_1L_chr = getwd(),
  description_ls = NULL,
  keep_version_lgl = T
)
```

### Arguments

`delete_contents_of_1L_chr`
Delete contents of (a character vector of length one)

`package_1L_chr` Package (a character vector of length one), Default: get_dev_pkg_nm()

`package_dir_1L_chr`
Package directory (a character vector of length one), Default: getwd()

`description_ls` Description (a list), Default: NULL

`keep_version_lgl`
Keep version (a logical vector), Default: T

## write_to_rpl_1L_and_indefL_sfcs

*Write to replace length one and indefinite length suffices*

### Description

write_to_rpl_1L_and_indefL_sfcs() is a Write function that writes a file to a specified local directory. Specifically, this function implements an algorithm to write to replace length one and indefinite length suffices. The function is called for its side effects and does not return a value. WARNING: This function writes R scripts to your local environment. Make sure to only use if you want this behaviour

### Usage

```
write_to_rpl_1L_and_indefL_sfcs(
  indefL_arg_nm_1L_chr,
  file_path_1L_chr = NA_character_,
  dir_path_1L_chr = NA_character_
)
```

### Arguments

`indefL_arg_nm_1L_chr`
: Indefinite length argument name (a character vector of length one)

`file_path_1L_chr`
: File path (a character vector of length one), Default: 'NA'

`dir_path_1L_chr`
: Directory path (a character vector of length one), Default: 'NA'

## write_vignette *Write vignette*

### Description

write_vignette() is a Write function that writes a file to a specified local directory. Specifically, this function implements an algorithm to write vignette. The function is called for its side effects and does not return a value. WARNING: This function writes R scripts to your local environment. Make sure to only use if you want this behaviour

### Usage

```
write_vignette(package_1L_chr, pkg_rt_dir_chr = ".")
```

### Arguments

| | |
|---|---|
| `package_1L_chr` | Package (a character vector of length one) |
| `pkg_rt_dir_chr` | Package root directory (a character vector), Default: '.' |

## write_ws *Write workspace*

### Description

write_ws() is a Write function that writes a file to a specified local directory. Specifically, this function implements an algorithm to write workspace. The function is called for its side effects and does not return a value. WARNING: This function writes R scripts to your local environment. Make sure to only use if you want this behaviour

### Usage

```
write_ws(path_1L_chr)
```

### Arguments

| | |
|---|---|
| `path_1L_chr` | Path (a character vector of length one) |

## `%>%` *Pipe operator*

### Description

Implements: https://github.com/sckott/analogsea/issues/32 and https://github.com/rstudio/ggvis/blob/master/R/pipe.R.

### Arguments

| | |
|---|---|
| `lhs, rhs` | An object and a function to apply to it |

# Index