

Package ‘ready4class’

September 13, 2020

Title Readyforwhatsnext Tools For Creating And Extending
Classes.

Version 0.0.0.9138

Description ready4class provides classes and methods that are designed to standardise and streamline the process for extending the readyforwhatsnext data synthesis and simulation framework with new classes. This development version of the ready4class package has been made available as part of the process of testing and documenting the package. That means this should be regarded as UNTESTED software, which is provided for free WITHOUT ANY WARRANTY. Importantly, the tools contained in this test release automate a number of tasks which MODIFY THE DIRECTORY STRUCTURE OF YOUR LOCAL MACHINE. While we welcome and appreciate anyone who takes the time to provide us with feedback on this test release, we caution you that you should only test this software if you feel confident you understand what it does and have created a sandpit area in which you can safely undertake testing. If you have any questions, please contact the authors (matthew.hamilton@orygen.org.au).

License GPL-3

URL <https://readyforwhatsnext.github.io/ready4class/>,
<https://github.com/readyforwhatsnext/ready4class>,
<https://readyforwhatsnext.github.io/readyforwhatsnext/>

Encoding UTF-8

LazyData true

Roxygen list(markdown = TRUE)

RoxygenNote 7.1.1

Imports lifecycle,
ready4fun (>= 0.0.0.9216),
devtools,
dplyr,
gtools,
magrittr,
methods,
purrr,
rlang,
stringi,

stringr,
testit,
tibble,
tidyr,
usethis

RdMacros lifecycle

Depends R (>= 2.10)

Remotes readyforwhatsnext/ready4fun

R topics documented:

ready4class-package	4
abbreviations_lup	5
add_class	5
add_class.ready4_class_pt_lup	6
fn_type_lup_tb	7
get_class_fl_nms	7
get_class_ns	8
get_nms_of_cls_to_inc	8
get_nms_of_curr_gnrcs	9
get_parent_cls_ns	10
get_parent_cls_pts	10
get_parent_cls_pt_fn	11
get_parent_cls_slot_nms	11
is_ready4_class_pt_lup	12
is_ready4_constructor_tbl	12
make_alg_to_gen_ref_to_cls	13
make_alg_to_get_pt_val	13
make_alg_to_set_gnrc	14
make_alg_to_set_mthd	15
make_alg_to_set_old_cls	15
make_alg_to_set_validity_of_r4_cls	16
make_alg_to_write_gtr_str_mthds	17
make_child_cls_fn_body	18
make_class_pts_tb	18
make_class_pt_tb_for_r3_and_r4_cls	19
make_dmt_inc_tag	19
make_fn_pt_to_check_r3_cls_inhrc	20
make_fn_pt_to_make_r3_cls_pt	20
make_fn_pt_to_make_unvld_r3_cls_inst	21
make_fn_pt_to_make_vld_r3_cls_inst	22
make_fn_pt_to_make_vld_r3_cls_inst	22
make_gnrc_fn	23
make_gnrc_mthd_pair_ls	24
make_helper_fn	24
make_lines_for_writing_dmt_d_fn	25
make_ls_of_pkgs_to_imp	26
make_ls_of_tfd_nms_of_curr_gnrcs	26
make_lup	27
make_lup.ready4_constructor_tbl	27
make_new_ready4_class_pt_lup	28

make_new_ready4_constructor_tbl	29
make_one_row_class_pt_tb	29
make_one_row_pt_tb_for_new_r3_cls	30
make_one_row_pt_tb_for_new_r4_cls	30
make_prototype_ready4_class_pt_lup	31
make_prototype_ready4_constructor_tbl	31
make_pt_ls	32
make_pt_ls_for_new_r3_cls	32
make_pt_tb_for_new_r3_cls	33
make_pt_tb_for_new_r4_cls	34
make_show_mthd_fn	34
order_tb	35
order_tb.ready4_constructor_tbl	35
prototype_lup	36
ready4_class_pt_lup	36
ready4_constructor_tbl	37
remake_ls_cols	37
remake_ls_cols.ready4_constructor_tbl	38
remove_class_from_cache	38
replace_NA_in_fn	39
transform_alg_to_ref_cls_nm	39
transform_class_ns	40
transform_fn_into_chr	40
transform_parent_ns_ls	41
transform_pt_ls_for_new_cls	41
update_lup_for_ns	42
update_lup_for_ns.ready4_class_pt_lup	42
validate_ready4_class_pt_lup	43
validate_ready4_constructor_tbl	43
write_classes	44
write_classes.ready4_constructor_tbl	44
write_classes_and_make_lup	45
write_classes_and_make_lup.ready4_constructor_tbl	45
write_gtr_str_mthds_for_r4	47
write_gtr_str_mthds_for_slots	47
write_mthds_for_r3_or_r4_cls	48
write_scripts_to_make_gnrc_and_mthd	49
write_scripts_to_mk_cls	50
write_scripts_to_mk_r3_cls	51
write_scripts_to_mk_r4_cls	52
write_script_to_make_gnrc	54
write_script_to_make_mthd	55
write_slot_gtr_str_mthds	56
write_std_mthd	57
write_to_delete_fls_with_ptrn	58
write_to_delete_gnrc_fn_fls	58
write_to_mk_r4_cls	59

ready4class-package	<i>ready4class: Readyforwhatsnext Tools For Creating And Extending Classes.</i>
---------------------	---

Description

ready4class provides classes and methods that are designed to standardise and streamline the process for extending the readyforwhatsnext data synthesis and simulation framework with new classes. This development version of the ready4class package has been made available as part of the process of testing and documenting the package. That means this should be regarded as UNTESTED software, which is provided for free WITHOUT ANY WARRANTY. Importantly, the tools contained in this test release automate a number of tasks which MODIFY THE DIRECTORY STRUCTURE OF YOUR LOCAL MACHINE. While we welcome and appreciate anyone who takes the time to provide us with feedback on this test release, we caution you that you should only test this software if you feel confident you understand what it does and have created a sandpit area in which you can safely undertake testing. If you have any questions, please contact the authors (matthew.hamilton@orygen.org.au).

Details

To learn more about ready4class, start with the vignettes: `browseVignettes(package = "ready4class")`

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]

See Also

Useful links:

- <https://readyforwhatsnext.github.io/ready4class/>
- <https://github.com/readyforwhatsnext/ready4class>
- <https://readyforwhatsnext.github.io/readyforwhatsnext/>

abbreviations_lup	<i>Common abbreviations lookup table</i>
-------------------	--

Description

A lookup table for abbreviations commonly used in object names in the ready4classpackage.

Usage

```
abbreviations_lup
```

Format

An object of class `tbl_df` (inherits from `tbl`, `data.frame`) with 248 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_class	<i>Add class</i>
-----------	------------------

Description

`add_class()` is an Add Class generic that adds information about a class.

Usage

```
add_class(x, ...)
```

Arguments

`x` An object

`...` Additional arguments

```
add_class.ready4_class_pt_lup
```

Add class method applied to readyforwhatsnext S3 class CLASS PROTOTYPE LOOKUP TABLE.

Description

add_class.ready4_class_pt_lup() is an Add Class method that adds information about a class. This method is implemented for the readyforwhatsnext S3 class CLASS PROTOTYPE LOOKUP TABLE. The function returns Instance (a readyforwhatsnext S3 class CLASS PROTOTYPE LOOKUP TABLE).

Usage

```
## S3 method for class 'ready4_class_pt_lup'
add_class(
  x,
  row_idx_1L_int,
  make_tb,
  dev_pkg_ns_1L_chr,
  name_pfx_1L_chr,
  output_dir_1L_chr,
  file_exists_cdn_1L_chr,
  nss_to_ignore_chr = NA_character_,
  req_pkgs_chr = NA_character_,
  class_in_cache_cdn_1L_chr = "stop",
  abbreviations_lup = NULL
)
```

Arguments

x	An instance of readyforwhatsnext S3 class CLASS PROTOTYPE LOOKUP TABLE
row_idx_1L_int	Row index (an integer vector of length one)
make_tb	Make (a tibble)
dev_pkg_ns_1L_chr	Development package namespace (a character vector of length one)
name_pfx_1L_chr	Name prefix (a character vector of length one)
output_dir_1L_chr	Output directory (a character vector of length one)
file_exists_cdn_1L_chr	File exists condition (a character vector of length one)
nss_to_ignore_chr	Namespaces to ignore (a character vector), Default: 'NA'
req_pkgs_chr	Req packages (a character vector), Default: 'NA'
class_in_cache_cdn_1L_chr	Class in cache condition (a character vector of length one), Default: 'stop'
abbreviations_lup	Abbreviations (a lookup table), Default: NULL

Value

Instance (a readyforwhatsnext S3 class CLASS PROTOTYPE LOOKUP TABLE)

fn_type_lup_tb	<i>Function type lookup table</i>
----------------	-----------------------------------

Description

A lookup table to find descriptions for different types of functions used within the ready4class package suite.

Usage

```
fn_type_lup_tb
```

Format

An object of class tbl_df (inherits from tbl, data.frame) with 23 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/>

get_class_fl_nms	<i>Get class file names</i>
------------------	-----------------------------

Description

get_class_fl_nms() 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 class file names. Function argument class_names_chr specifies the where to look for the required object. The function returns Class file names (a character vector).

Usage

```
get_class_fl_nms(class_names_chr, s3_1l_lgl = T, output_dir_1l_chr = NA)
```

Arguments

class_names_chr Class names (a character vector)

s3_1L_lgl S3 (a logical vector of length one), Default: T

output_dir_1L_chr Output directory (a character vector of length one), Default: NA

Value

Class file names (a character vector)

get_class_ns	<i>Get class namespace</i>
--------------	----------------------------

Description

get_class_ns() 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 class namespace. Function argument prototype_lup specifies the where to look for the required object. The function returns Class namespace (a character vector of length one).

Usage

```
get_class_ns(prototype_lup, class_nm_1L_chr)
```

Arguments

prototype_lup Prototype (a lookup table)

class_nm_1L_chr Class name (a character vector of length one)

Value

Class namespace (a character vector of length one)

get_nms_of_cls_to_inc	<i>Get names of classes to include</i>
-----------------------	--

Description

get_nms_of_cls_to_inc() 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 names of classes to include. Function argument parent_cls_nm_1L_chr specifies the where to look for the required object. The function returns Names of classes to include (a character vector).

Usage

```
get_nms_of_clss_to_inc(
  parent_cls_nm_1L_chr,
  parent_ns_ls,
  base_set_of_clss_to_inc_chr = NULL
)
```

Arguments

parent_cls_nm_1L_chr Parent class name (a character vector of length one)

parent_ns_ls Parent namespace (a list)

base_set_of_clss_to_inc_chr Base set of classes to include (a character vector), Default: NULL

Value

Names of classes to include (a character vector)

get_nms_of_curr_gnracs *Get names of current generics*

Description

get_nms_of_curr_gnracs() 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 names of current generics. Function argument req_pkgs_chr specifies the where to look for the required object. The function returns Names of current generics (a list).

Usage

```
get_nms_of_curr_gnracs(req_pkgs_chr, generic_1L_chr)
```

Arguments

req_pkgs_chr Req packages (a character vector)

generic_1L_chr Generic (a character vector of length one)

Value

Names of current generics (a list)

get_parent_cls_ns	<i>Get parent class namespace</i>
-------------------	-----------------------------------

Description

get_parent_cls_ns() 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 parent class namespace. Function argument prototype_lup specifies the where to look for the required object. The function is called for its side effects and does not return a value.

Usage

```
get_parent_cls_ns(prototype_lup, parent_cls_nm_1L_chr, dev_pkg_ns_1L_chr)
```

Arguments

prototype_lup	Prototype (a lookup table)
parent_cls_nm_1L_chr	Parent class name (a character vector of length one)
dev_pkg_ns_1L_chr	Development package namespace (a character vector of length one)

Value

Parent class (a namespace)

get_parent_cls_pts	<i>Get parent class prototypes</i>
--------------------	------------------------------------

Description

get_parent_cls_pts() 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 parent class prototypes. Function argument parent_cls_nm_1L_chr specifies the where to look for the required object. The function returns Parent class prototypes (a character vector).

Usage

```
get_parent_cls_pts(parent_cls_nm_1L_chr, parent_ns_ls, slot_names_chr)
```

Arguments

parent_cls_nm_1L_chr	Parent class name (a character vector of length one)
parent_ns_ls	Parent namespace (a list)
slot_names_chr	Slot names (a character vector)

Value

Parent class prototypes (a character vector)

get_parent_cls_pt_fn *Get parent class prototype*

Description

get_parent_cls_pt_fn() 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 parent class prototype function. Function argument parent_cls_nm_1L_chr specifies the where to look for the required object. The function returns Parent class prototype function (a character vector).

Usage

```
get_parent_cls_pt_fn(parent_cls_nm_1L_chr, prototype_lup)
```

Arguments

```
parent_cls_nm_1L_chr
                        Parent class name (a character vector of length one)
prototype_lup    Prototype (a lookup table)
```

Value

Parent class prototype function (a character vector)

get_parent_cls_slot_nms
 Get parent class slot names

Description

get_parent_cls_slot_nms() 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 parent class slot names. Function argument parent_cls_nm_1L_chr specifies the where to look for the required object. The function returns Parent class slot names (a character vector).

Usage

```
get_parent_cls_slot_nms(parent_cls_nm_1L_chr, parent_ns_ls)
```

Arguments

```
parent_cls_nm_1L_chr
                        Parent class name (a character vector of length one)
parent_ns_ls    Parent namespace (a list)
```

Value

Parent class slot names (a character vector)

is_ready4_class_pt_lup

Is readyforwhatsnext S3 class CLASS PROTOTYPE LOOKUP TABLE

Description

Check whether an object is a valid instance of the readyforwhatsnext S3 class CLASS PROTOTYPE LOOKUP TABLE

Usage

is_ready4_class_pt_lup(x)

Arguments

x An object of any type

Details

Readyforwhatsnext S3 class CLASS PROTOTYPE LOOKUP TABLE class of metadata of proto-type classes.

Value

A logical value, TRUE if a valid instance of the readyforwhatsnext S3 class CLASS PROTOTYPE LOOKUP TABLE

is_ready4_constructor_tbl

Is readyforwhatsnext S3 class CLASS CONSTRUCTOR TABLE

Description

Check whether an object is a valid instance of the readyforwhatsnext S3 class CLASS CONSTRUCTOR TABLE

Usage

is_ready4_constructor_tbl(x)

Arguments

x An object of any type

Details

Readyforwhatsnext S3 class CLASS CONSTRUCTOR TABLE of metadata required to make new classes.

Value

A logical value, TRUE if a valid instance of the readyforwhatsnext S3 class CLASS CONSTRUCTOR TABLE

make_alg_to_gen_ref_to_cls

Make algorithm to generate reference to class

Description

make_alg_to_gen_ref_to_cls() is a Make function that creates a new R object. Specifically, this function implements an algorithm to make algorithm to generate reference to class. The function returns Algorithm to generate reference to class (a character vector of length one).

Usage

```
make_alg_to_gen_ref_to_cls(class_nm_1L_chr, pkg_nm_1L_chr = ".GlobalEnv")
```

Arguments

class_nm_1L_chr

Class name (a character vector of length one)

pkg_nm_1L_chr

Package name (a character vector of length one), Default: '.GlobalEnv'

Value

Algorithm to generate reference to class (a character vector of length one)

make_alg_to_get_pt_val

Make algorithm to get prototype value

Description

make_alg_to_get_pt_val() is a Make function that creates a new R object. Specifically, this function implements an algorithm to make algorithm to get prototype value. The function returns Algorithm to get prototype value (a character vector of length one).

Usage

```
make_alg_to_get_pt_val(
  pt_ns_1L_chr = "",
  fn_to_call_1L_chr = "",
  default_val_1L_chr = "",
  attached_nss_chr = c("base")
)
```

Arguments

pt_ns_1L_chr Prototype namespace (a character vector of length one), Default: ""

fn_to_call_1L_chr Function to call (a character vector of length one), Default: ""

default_val_1L_chr Default value (a character vector of length one), Default: ""

attached_nss_chr Attached namespaces (a character vector), Default: c("base")

Value

Algorithm to get prototype value (a character vector of length one)

make_alg_to_set_gnrc *Make algorithm to set generic*

Description

make_alg_to_set_gnrc() is a Make function that creates a new R object. Specifically, this function implements an algorithm to make algorithm to set generic. The function returns Algorithm to set generic (a character vector of length one).

Usage

```
make_alg_to_set_gnrc(
  name_1L_chr,
  args_chr = c("x"),
  signature_1L_chr = NA_character_,
  where_1L_chr = NA_character_
)
```

Arguments

name_1L_chr Name (a character vector of length one)

args_chr Arguments (a character vector), Default: c("x")

signature_1L_chr Signature (a character vector of length one), Default: 'NA'

where_1L_chr Where (a character vector of length one), Default: 'NA'

Value

Algorithm to set generic (a character vector of length one)

make_alg_to_set_mthd	<i>Make algorithm to set method</i>
----------------------	-------------------------------------

Description

make_alg_to_set_mthd() is a Make function that creates a new R object. Specifically, this function implements an algorithm to make algorithm to set method. The function returns Algorithm to set method (a character vector of length one).

Usage

```
make_alg_to_set_mthd(
  name_1L_chr,
  class_nm_1L_chr,
  fn,
  pkg_nm_1L_chr = NA_character_,
  where_1L_chr = NA_character_
)
```

Arguments

name_1L_chr	Name (a character vector of length one)
class_nm_1L_chr	Class name (a character vector of length one)
fn	Function (a function)
pkg_nm_1L_chr	Package name (a character vector of length one), Default: 'NA'
where_1L_chr	Where (a character vector of length one), Default: 'NA'

Value

Algorithm to set method (a character vector of length one)

make_alg_to_set_old_clss	<i>Make algorithm to set old classes</i>
--------------------------	--

Description

make_alg_to_set_old_clss() is a Make function that creates a new R object. Specifically, this function implements an algorithm to make algorithm to set old classes. The function returns Algorithm to set old classes (a character vector of length one).

Usage

```
make_alg_to_set_old_clss(type_chr, prototype_lup)
```

Arguments

type_chr Type (a character vector)
 prototype_lup Prototype (a lookup table)

Value

Algorithm to set old classes (a character vector of length one)

make_alg_to_set_validity_of_r4_cls

Make algorithm to set validity of readyforwhatsnext S4 class

Description

make_alg_to_set_validity_of_r4_cls() is a Make function that creates a new R object. Specifically, this function implements an algorithm to make algorithm to set validity of readyforwhatsnext s4 class. The function returns Algorithm to set validity of readyforwhatsnext S4 class (a character vector of length one).

Usage

```
make_alg_to_set_validity_of_r4_cls(  
  class_nm_1L_chr,  
  parent_cls_nm_1L_chr,  
  slots_of_dif_lnts_chr = NULL,  
  allowed_vals_ls = NULL,  
  names_must_match_ls = NULL,  
  print_validator_1L_lgl = FALSE  
)
```

Arguments

class_nm_1L_chr Class name (a character vector of length one)
 parent_cls_nm_1L_chr Parent class name (a character vector of length one)
 slots_of_dif_lnts_chr Slots of different lengths (a character vector), Default: NULL
 allowed_vals_ls Allowed values (a list), Default: NULL
 names_must_match_ls Names must match (a list), Default: NULL
 print_validator_1L_lgl Print validator (a logical vector of length one), Default: FALSE

Value

Algorithm to set validity of readyforwhatsnext S4 class (a character vector of length one)

`make_alg_to_write_gtr_str_mthds`*Make algorithm to write getter setter methods*

Description

`make_alg_to_write_gtr_str_mthds()` is a Make function that creates a new R object. Specifically, this function implements an algorithm to make algorithm to write getter setter methods. The function is called for its side effects and does not return a value.

Usage

```
make_alg_to_write_gtr_str_mthds(  
  class_nm_1L_chr,  
  parent_cls_nm_1L_chr,  
  print_gtrs_strs_1L_lgl,  
  output_dir_1L_chr,  
  nss_to_ignore_chr,  
  req_pkgs_chr,  
  parent_ns_ls  
)
```

Arguments

<code>class_nm_1L_chr</code>	Class name (a character vector of length one)
<code>parent_cls_nm_1L_chr</code>	Parent class name (a character vector of length one)
<code>print_gtrs_strs_1L_lgl</code>	Print getters setters (a logical vector of length one)
<code>output_dir_1L_chr</code>	Output directory (a character vector of length one)
<code>nss_to_ignore_chr</code>	Namespaces to ignore (a character vector)
<code>req_pkgs_chr</code>	Req packages (a character vector)
<code>parent_ns_ls</code>	Parent namespace (a list)

Value

Algorithm to write getter setter (methods)

make_child_cls_fn_body

Make child class function body

Description

make_child_cls_fn_body() is a Make function that creates a new R object. Specifically, this function implements an algorithm to make child class function body. The function returns Child class function body (a character vector of length one).

Usage

```
make_child_cls_fn_body(
  child_ext_fn_1L_chr,
  parent_cls_nm_1L_chr,
  prototype_lup,
  prepend_1L_lgl = T
)
```

Arguments

child_ext_fn_1L_chr Child ext function (a character vector of length one)

parent_cls_nm_1L_chr Parent class name (a character vector of length one)

prototype_lup Prototype (a lookup table)

prepend_1L_lgl Prepend (a logical vector of length one), Default: T

Value

Child class function body (a character vector of length one)

make_class_pts_tb

Make class prototypes

Description

make_class_pts_tb() is a Make function that creates a new R object. Specifically, this function implements an algorithm to make class prototypes tibble. The function returns Class prototypes (a tibble).

Usage

```
make_class_pts_tb(class_mk_ls)
```

Arguments

class_mk_ls Class make (a list)

Value

Class prototypes (a tibble)

```
make_class_pt_tb_for_r3_and_r4_cls
```

Make class prototype tibble for readyforwhatsnext S3 and readyforwhatsnext S4 classes

Description

make_class_pt_tb_for_r3_and_r4_cls() is a Make function that creates a new R object. Specifically, this function implements an algorithm to make class prototype tibble for readyforwhatsnext s3 and readyforwhatsnext s4 classes. The function returns Class prototype tibble for readyforwhatsnext S3 and readyforwhatsnext S4 classes (a tibble).

Usage

```
make_class_pt_tb_for_r3_and_r4_cls(class_mk_ls)
```

Arguments

class_mk_ls Class make (a list)

Value

Class prototype tibble for readyforwhatsnext S3 and readyforwhatsnext S4 classes (a tibble)

```
make_dmt_inc_tag
```

Make documentation include tag

Description

make_dmt_inc_tag() is a Make function that creates a new R object. Specifically, this function implements an algorithm to make documentation include tag. The function returns Documentation include tag (a character vector of length one).

Usage

```
make_dmt_inc_tag(class_names_chr, s3_1L_lgl = T)
```

Arguments

class_names_chr

Class names (a character vector)

s3_1L_lgl

S3 (a logical vector of length one), Default: T

Value

Documentation include tag (a character vector of length one)

```
make_fn_pt_to_check_r3_cls_inhtc
```

Make function prototype to check readyforwhatsnext S3 class inheritance

Description

`make_fn_pt_to_check_r3_cls_inhtc()` is a Make function that creates a new R object. Specifically, this function implements an algorithm to make function prototype to check readyforwhatsnext s3 class inheritance. The function is called for its side effects and does not return a value.

Usage

```
make_fn_pt_to_check_r3_cls_inhtc(class_nm_1L_chr, s3_validator_ls)
```

Arguments

```
class_nm_1L_chr
                Class name (a character vector of length one)
s3_validator_ls
                S3 validator (a list)
```

Value

Function prototype to check readyforwhatsnext S3 class (an inheritance)

```
make_fn_pt_to_make_r3_cls_pt
```

Make function prototype to make readyforwhatsnext S3 class prototype

Description

`make_fn_pt_to_make_r3_cls_pt()` is a Make function that creates a new R object. Specifically, this function implements an algorithm to make function prototype to make readyforwhatsnext s3 class prototype. The function is called for its side effects and does not return a value.

Usage

```
make_fn_pt_to_make_r3_cls_pt(
  type_1L_chr,
  pt_ns_1L_chr,
  vals_ls,
  ordered_1L_lgl,
  class_nm_1L_chr,
  parent_cls_nm_1L_chr,
  prototype_lup
)
```

Arguments

type_1L_chr	Type (a character vector of length one)
pt_ns_1L_chr	Prototype namespace (a character vector of length one)
vals_ls	Values (a list)
ordered_1L_lgl	Ordered (a logical vector of length one)
class_nm_1L_chr	Class name (a character vector of length one)
parent_cls_nm_1L_chr	Parent class name (a character vector of length one)
prototype_lup	Prototype (a lookup table)

Value

Function prototype to make readyforwhatsnext S3 class (a prototype)

```
make_fn_pt_to_make_unvld_r3_cls_inst
```

Make function prototype to make unvld readyforwhatsnext S3 class instance

Description

make_fn_pt_to_make_unvld_r3_cls_inst() is a Make function that creates a new R object. Specifically, this function implements an algorithm to make function prototype to make unvld readyforwhatsnext s3 class instance. The function is called for its side effects and does not return a value.

Usage

```
make_fn_pt_to_make_unvld_r3_cls_inst(
  type_1L_chr,
  pt_chkr_pfx_1L_chr,
  pt_ns_1L_chr,
  class_nm_1L_chr,
  s3_prototype_ls
)
```

Arguments

type_1L_chr	Type (a character vector of length one)
pt_chkr_pfx_1L_chr	Prototype checker prefix (a character vector of length one)
pt_ns_1L_chr	Prototype namespace (a character vector of length one)
class_nm_1L_chr	Class name (a character vector of length one)
s3_prototype_ls	S3 prototype (a list)

Value

Function prototype to make unvld readyforwhatsnext S3 class (an instance)

```
make_fn_pt_to_make_vld_r3_cls_inst
```

Make function prototype to make validated readyforwhatsnext S3 class instance

Description

`make_fn_pt_to_make_vld_r3_cls_inst()` is a Make function that creates a new R object. Specifically, this function implements an algorithm to make function prototype to make validated readyforwhatsnext s3 class instance. The function is called for its side effects and does not return a value.

Usage

```
make_fn_pt_to_make_vld_r3_cls_inst(
  class_nm_1L_chr,
  s3_prototype_ls,
  s3_constructor_ls,
  s3_validator_ls
)
```

Arguments

```
class_nm_1L_chr
  Class name (a character vector of length one)
s3_prototype_ls
  S3 prototype (a list)
s3_constructor_ls
  S3 constructor (a list)
s3_validator_ls
  S3 validator (a list)
```

Value

Function prototype to make validated readyforwhatsnext S3 class (an instance)

```
make_fn_pt_to_make_vld_r3_cls_inst
```

Make function prototype to make valid readyforwhatsnext S3 class instance

Description

`make_fn_pt_to_make_vld_r3_cls_inst()` is a Make function that creates a new R object. Specifically, this function implements an algorithm to make function prototype to make valid readyforwhatsnext s3 class instance. The function is called for its side effects and does not return a value.

Usage

```
make_fn_pt_to_make_vld_r3_cls_inst(
  type_1L_chr,
  class_nm_1L_chr,
  s3_prototype_ls,
  min_max_vals_dbl,
  start_end_vals_dbl,
  vals_ls
)
```

Arguments

type_1L_chr	Type (a character vector of length one)
class_nm_1L_chr	Class name (a character vector of length one)
s3_prototype_ls	S3 prototype (a list)
min_max_vals_dbl	Min max values (a double vector)
start_end_vals_dbl	Start end values (a double vector)
vals_ls	Values (a list)

Value

Function prototype to make valid readyforwhatsnext S3 class (an instance)

make_gnrc_fn	<i>Make generic</i>
--------------	---------------------

Description

make_gnrc_fn() is a Make function that creates a new R object. Specifically, this function implements an algorithm to make generic function. The function returns Generic function (a character vector of length one).

Usage

```
make_gnrc_fn(name_1L_chr, args_chr)
```

Arguments

name_1L_chr	Name (a character vector of length one)
args_chr	Arguments (a character vector)

Value

Generic function (a character vector of length one)

make_gnrc_mthd_pair_ls

Make generic method pair

Description

make_gnrc_mthd_pair_ls() is a Make function that creates a new R object. Specifically, this function implements an algorithm to make generic method pair list. The function returns Generic method pair (a list).

Usage

```
make_gnrc_mthd_pair_ls(
  name_1L_chr,
  args_chr = c("x"),
  signature_1L_chr = NA_character_,
  pkg_nm_1L_chr = NA_character_,
  where_1L_chr = NA_character_,
  class_nm_1L_chr,
  fn
)
```

Arguments

name_1L_chr	Name (a character vector of length one)
args_chr	Arguments (a character vector), Default: c("x")
signature_1L_chr	Signature (a character vector of length one), Default: 'NA'
pkg_nm_1L_chr	Package name (a character vector of length one), Default: 'NA'
where_1L_chr	Where (a character vector of length one), Default: 'NA'
class_nm_1L_chr	Class name (a character vector of length one)
fn	Function (a function)

Value

Generic method pair (a list)

make_helper_fn

Make helper

Description

make_helper_fn() is a Make function that creates a new R object. Specifically, this function implements an algorithm to make helper function. The function returns Helper function (a character vector of length one).

Usage

```
make_helper_fn(
  class_nm_1L_chr,
  parent_cls_nm_1L_chr,
  slots_chr,
  pt_ls,
  prototype_lup,
  parent_ns_ls
)
```

Arguments

class_nm_1L_chr	Class name (a character vector of length one)
parent_cls_nm_1L_chr	Parent class name (a character vector of length one)
slots_chr	Slots (a character vector)
pt_ls	Prototype (a list)
prototype_lup	Prototype (a lookup table)
parent_ns_ls	Parent namespace (a list)

Value

Helper function (a character vector of length one)

make_lines_for_writing_dmt_d_fn	<i>Make lines for writing documented</i>
---------------------------------	--

Description

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

Usage

```
make_lines_for_writing_dmt_d_fn(
  fn_name_1L_chr,
  fn_body_1L_chr,
  fn_type_1L_chr,
  class_nm_1L_chr,
  class_desc_1L_chr,
  abbreviations_lup = NULL
)
```

Arguments

fn_name_1L_chr Function name (a character vector of length one)
 fn_body_1L_chr Function body (a character vector of length one)
 fn_type_1L_chr Function type (a character vector of length one)
 class_nm_1L_chr
 Class name (a character vector of length one)
 class_desc_1L_chr
 Class description (a character vector of length one)
 abbreviations_lup
 Abbreviations (a lookup table), Default: NULL

make_ls_of_pkgs_to_imp

Make list of packages to import

Description

make_ls_of_pkgs_to_imp() is a Make function that creates a new R object. Specifically, this function implements an algorithm to make list of packages to import. The function returns Packages to import (a list).

Usage

```
make_ls_of_pkgs_to_imp(curr_gnracs_ls, fn_name_1L_chr, nss_to_ignore_chr)
```

Arguments

curr_gnracs_ls Current generics (a list)
 fn_name_1L_chr Function name (a character vector of length one)
 nss_to_ignore_chr
 Namespaces to ignore (a character vector)

Value

Packages to import (a list)

make_ls_of_tfd_nms_of_curr_gnracs

Make list of transformed names of current generics

Description

make_ls_of_tfd_nms_of_curr_gnracs() is a Make function that creates a new R object. Specifically, this function implements an algorithm to make list of transformed names of current generics. The function returns Current generics (a list).

Usage

```
make_ls_of_tfd_nms_of_curr_gnracs(
  req_pkgs_chr,
  generic_1L_chr,
  nss_to_ignore_chr
)
```

Arguments

req_pkgs_chr Req packages (a character vector)

generic_1L_chr Generic (a character vector of length one)

nss_to_ignore_chr
 Namespaces to ignore (a character vector)

Value

Current generics (a list)

make_lup	<i>Make lookup table</i>
----------	--------------------------

Description

make_lup() is a Make Lookup Table generic that applies a Make method and then updates the output of that method.

Usage

```
make_lup(x, ...)
```

Arguments

x An object

... Additional arguments

make_lup.ready4_constructor_tbl	<i>Make lookup table method applied to readyforwhatsnext S3 class CLASS CONSTRUCTOR TABLE.</i>
---------------------------------	--

Description

make_lup.ready4_constructor_tbl() is a Make Lookup Table method that applies a Make method and then updates the output of that method. This method is implemented for the readyforwhatsnext S3 class CLASS CONSTRUCTOR TABLE. The function returns Instance (a readyforwhatsnext S3 class CLASS PROTOTYPE LOOKUP TABLE).

Usage

```
## S3 method for class 'ready4_constructor_tbl'
make_lup(x, dev_pkg_ns_1L_chr, prefix_1L_chr)
```

Arguments

x An instance of readyforwhatsnext S3 class CLASS CONSTRUCTOR TABLE

dev_pkg_ns_1L_chr Development package namespace (a character vector of length one)

prefix_1L_chr Prefix (a character vector of length one)

Value

Instance (a readyforwhatsnext S3 class CLASS PROTOTYPE LOOKUP TABLE)

```
make_new_ready4_class_pt_lup
```

Make new readyforwhatsnext S3 class CLASS PROTOTYPE LOOKUP TABLE

Description

Create a new unvalidated instance of the readyforwhatsnext S3 class CLASS PROTOTYPE LOOKUP TABLE

Usage

```
make_new_ready4_class_pt_lup(x)
```

Arguments

x A prototype for the readyforwhatsnext S3 class CLASS PROTOTYPE LOOKUP TABLE

Details

Readyforwhatsnext S3 class CLASS PROTOTYPE LOOKUP TABLE class of metadata of prototype classes.

Value

An unvalidated instance of the readyforwhatsnext S3 class CLASS PROTOTYPE LOOKUP TABLE

make_new_ready4_constructor_tbl

Make new readyforwhatsnext S3 class CLASS CONSTRUCTOR TABLE

Description

Create a new unvalidated instance of the readyforwhatsnext S3 class CLASS CONSTRUCTOR TABLE

Usage

```
make_new_ready4_constructor_tbl(x)
```

Arguments

x A prototype for the readyforwhatsnext S3 class CLASS CONSTRUCTOR TABLE

Details

Readyforwhatsnext S3 class CLASS CONSTRUCTOR TABLE of metadata required to make new classes.

Value

An unvalidated instance of the readyforwhatsnext S3 class CLASS CONSTRUCTOR TABLE

make_one_row_class_pt_tb

Make one row class prototype

Description

make_one_row_class_pt_tb() is a Make function that creates a new R object. Specifically, this function implements an algorithm to make one row class prototype tibble. The function returns One row class prototype (a tibble).

Usage

```
make_one_row_class_pt_tb(class_type_mk_ls, make_s3_1L_lgl = T)
```

Arguments

class_type_mk_ls

Class type make (a list)

make_s3_1L_lgl Make S3 (a logical vector of length one), Default: T

Value

One row class prototype (a tibble)

`make_one_row_pt_tb_for_new_r3_cls`*Make one row prototype tibble for new readyforwhatsnext S3 class*

Description

`make_one_row_pt_tb_for_new_r3_cls()` is a Make function that creates a new R object. Specifically, this function implements an algorithm to make one row prototype tibble for new readyforwhatsnext s3 class. The function returns One row class prototype (a tibble).

Usage

```
make_one_row_pt_tb_for_new_r3_cls(x)
```

Arguments

x PARAM_DESCRIPTION

Value

One row class prototype (a tibble)

`make_one_row_pt_tb_for_new_r4_cls`*Make one row prototype tibble for new readyforwhatsnext S4 class*

Description

`make_one_row_pt_tb_for_new_r4_cls()` is a Make function that creates a new R object. Specifically, this function implements an algorithm to make one row prototype tibble for new readyforwhatsnext s4 class. The function returns One row class prototype (a tibble).

Usage

```
make_one_row_pt_tb_for_new_r4_cls(x)
```

Arguments

x PARAM_DESCRIPTION

Value

One row class prototype (a tibble)

make_prototype_ready4_class_pt_lup

*Make prototype readyforwhatsnext S3 class CLASS PROTOTYPE
LOOKUP TABLE*

Description

Create a new prototype for the readyforwhatsnext S3 class CLASS PROTOTYPE LOOKUP TABLE

Usage

make_prototype_ready4_class_pt_lup()

Details

Readyforwhatsnext S3 class CLASS PROTOTYPE LOOKUP TABLE class of metadata of prototype classes.

Value

A prototype for readyforwhatsnext S3 class CLASS PROTOTYPE LOOKUP TABLE

make_prototype_ready4_constructor_tbl

*Make prototype readyforwhatsnext S3 class CLASS CONSTRUCTOR
TABLE*

Description

Create a new prototype for the readyforwhatsnext S3 class CLASS CONSTRUCTOR TABLE

Usage

make_prototype_ready4_constructor_tbl()

Details

Readyforwhatsnext S3 class CLASS CONSTRUCTOR TABLE of metadata required to make new classes.

Value

A prototype for readyforwhatsnext S3 class CLASS CONSTRUCTOR TABLE

make_pt_ls	<i>Make prototype</i>
------------	-----------------------

Description

make_pt_ls() is a Make function that creates a new R object. Specifically, this function implements an algorithm to make prototype list. The function returns Prototype (a list).

Usage

```
make_pt_ls(
  slots_chr,
  type_chr = NULL,
  vals_ls = NULL,
  make_val_1L_lgl = TRUE,
  prototype_lup
)
```

Arguments

slots_chr	Slots (a character vector)
type_chr	Type (a character vector), Default: NULL
vals_ls	Values (a list), Default: NULL
make_val_1L_lgl	Make value (a logical vector of length one), Default: TRUE
prototype_lup	Prototype (a lookup table)

Value

Prototype (a list)

make_pt_ls_for_new_r3_cls	<i>Make prototype for new readyforwhatsnext S3 class</i>
---------------------------	--

Description

make_pt_ls_for_new_r3_cls() is a Make function that creates a new R object. Specifically, this function implements an algorithm to make prototype list for new readyforwhatsnext s3 class. The function returns Prototype list for new readyforwhatsnext S3 class (a list).

Usage

```
make_pt_ls_for_new_r3_cls(
  class_name_1L_chr,
  type_1L_chr,
  pt_ns_1L_chr,
  pt_chkr_pfx_1L_chr,
  vals_ls,
  ordered_1L_lgl,
  parent_cls_nm_1L_chr,
  prototype_lup,
  min_max_vals_dbl,
  start_end_vals_dbl,
  nss_to_ignore_chr
)
```

Arguments

class_name_1L_chr	Class name (a character vector of length one)
type_1L_chr	Type (a character vector of length one)
pt_ns_1L_chr	Prototype namespace (a character vector of length one)
pt_chkr_pfx_1L_chr	Prototype checker prefix (a character vector of length one)
vals_ls	Values (a list)
ordered_1L_lgl	Ordered (a logical vector of length one)
parent_cls_nm_1L_chr	Parent class name (a character vector of length one)
prototype_lup	Prototype (a lookup table)
min_max_vals_dbl	Min max values (a double vector)
start_end_vals_dbl	Start end values (a double vector)
nss_to_ignore_chr	Namespaces to ignore (a character vector)

Value

Prototype list for new readyforwhatsnext S3 class (a list)

```
make_pt_tb_for_new_r3_cls
```

Make prototype tibble for new readyforwhatsnext S3 class

Description

make_pt_tb_for_new_r3_cls() is a Make function that creates a new R object. Specifically, this function implements an algorithm to make prototype tibble for new readyforwhatsnext s3 class. The function returns Prototype tibble for new readyforwhatsnext S3 class (a tibble).

Usage

```
make_pt_tb_for_new_r3_cls(x)
```

Arguments

x PARAM_DESCRIPTION

Value

Prototype tibble for new readyforwhatsnext S3 class (a tibble)

```
make_pt_tb_for_new_r4_cls
```

Make prototype tibble for new readyforwhatsnext S4 class

Description

make_pt_tb_for_new_r4_cls() is a Make function that creates a new R object. Specifically, this function implements an algorithm to make prototype tibble for new readyforwhatsnext s4 class. The function returns Prototype tibble for new readyforwhatsnext S3 class (a tibble).

Usage

```
make_pt_tb_for_new_r4_cls(x)
```

Arguments

x PARAM_DESCRIPTION

Value

Prototype tibble for new readyforwhatsnext S3 class (a tibble)

```
make_show_mthd_fn
```

Make show method

Description

make_show_mthd_fn() is a Make function that creates a new R object. Specifically, this function implements an algorithm to make show method function. The function returns Show method function (a character vector of length one).

Usage

```
make_show_mthd_fn(class_nm_1L_chr, meaningful_nms_1s)
```

Arguments

class_nm_1L_chr	Class name (a character vector of length one)
meaningful_nms_ls	Meaningful names (a list)

Value

Show method function (a character vector of length one)

order_tb	<i>Order tibble</i>
----------	---------------------

Description

order_tb() is an Order Tibble generic that orders a tibble.

Usage

```
order_tb(x, ...)
```

Arguments

x	An object
...	Additional arguments

order_tb.ready4_constructor_tbl	<i>Order tibble method applied to readyforwhatsnext S3 class CLASS CONSTRUCTOR TABLE.</i>
---------------------------------	---

Description

order_tb.ready4_constructor_tbl() is an Order Tibble method that orders a tibble. This method is implemented for the readyforwhatsnext S3 class CLASS CONSTRUCTOR TABLE. The function is called for its side effects and does not return a value.

Usage

```
## S3 method for class 'ready4_constructor_tbl'
order_tb(x, name_pfx_1L_chr)
```

Arguments

x	An instance of readyforwhatsnext S3 class CLASS CONSTRUCTOR TABLE
name_pfx_1L_chr	Name prefix (a character vector of length one)

Value

Instance (a readyforwhatsnext S3 class CLASS CONSTRUCTOR TABLE)

prototype_lup	<i>Class prototype lookup table</i>
---------------	-------------------------------------

Description

Metadata on classes used in readyforwhatsnext suite

Usage

prototype_lup

Format

An object of class ready4_class_pt_lup (inherits from tbl_df, tbl, data.frame) with 7 rows and 6 columns.

Details

A tibble

- type_chr** Type (a character vector)
- val_chr** Val (a character vector)
- pt_ns_chr** Prototype namespace (a character vector)
- fn_to_call_chr** Function to call (a character vector)
- default_val_chr** Default val (a character vector)
- old_class_lgl** Old class (a logical vector)

ready4_class_pt_lup	<i>readyforwhatsnext S3 class CLASS PROTOTYPE LOOKUP TABLE</i>
---------------------	--

Description

Create a new valid instance of the readyforwhatsnext S3 class CLASS PROTOTYPE LOOKUP TABLE

Usage

ready4_class_pt_lup(x = make_prototype_ready4_class_pt_lup())

Arguments

- x A prototype for the readyforwhatsnext S3 class CLASS PROTOTYPE LOOKUP TABLE, Default: make_prototype_ready4_class_pt_lup()

Details

Readyforwhatsnext S3 class CLASS PROTOTYPE LOOKUP TABLE class of metadata of prototype classes.

Value

A validated instance of the readyforwhatsnext S3 class CLASS PROTOTYPE LOOKUP TABLE

ready4_constructor_tbl	<i>readyforwhatsnext S3 class CLASS CONSTRUCTOR TABLE</i>
------------------------	---

Description

Create a new valid instance of the readyforwhatsnext S3 class CLASS CONSTRUCTOR TABLE

Usage

ready4_constructor_tbl(x = make_prototype_ready4_constructor_tbl())

Arguments

x A prototype for the readyforwhatsnext S3 class CLASS CONSTRUCTOR TABLE, Default: make_prototype_ready4_constructor_tbl()

Details

Readyforwhatsnext S3 class CLASS CONSTRUCTOR TABLE of metadata required to make new classes.

Value

A validated instance of the readyforwhatsnext S3 class CLASS CONSTRUCTOR TABLE

remake_ls_cols	<i>Remake list columns</i>
----------------	----------------------------

Description

remake_ls_cols() is a Remake List Columns generic that remakes list columns.

Usage

remake_ls_cols(x, ...)

Arguments

x An object
... Additional arguments

```
remake_ls_cols.ready4_constructor_tbl
```

Remake list columns method applied to readyforwhatsnext S3 class CLASS CONSTRUCTOR TABLE.

Description

remake_ls_cols.ready4_constructor_tbl() is a Remake List Columns method that remakes list columns. This method is implemented for the readyforwhatsnext S3 class CLASS CONSTRUCTOR TABLE. The function is called for its side effects and does not return a value.

Usage

```
## S3 method for class 'ready4_constructor_tbl'
remake_ls_cols(x)
```

Arguments

x An instance of readyforwhatsnext S3 class CLASS CONSTRUCTOR TABLE

Value

Instance (a readyforwhatsnext S3 class CLASS CONSTRUCTOR TABLE)

```
remove_class_from_cache
```

Remove class from cache

Description

remove_class_from_cache() is a Remove function that edits an object, removing a specified element or elements. Specifically, this function implements an algorithm to remove class from cache. Function argument class_nm_1L_chr specifies the object to be updated. Argument class_in_cache_cdn_1L_chr provides the object to be updated. The function is called for its side effects and does not return a value.

Usage

```
remove_class_from_cache(class_nm_1L_chr, class_in_cache_cdn_1L_chr = "stop")
```

Arguments

class_nm_1L_chr

Class name (a character vector of length one)

class_in_cache_cdn_1L_chr

Class in cache condition (a character vector of length one), Default: 'stop'

replace_NA_in_fn	<i>Replace NA in</i>
------------------	----------------------

Description

replace_NA_in_fn() is a Replace function that edits an object, replacing a specified element with another specified element. Specifically, this function implements an algorithm to replace na in function. Function argument fn_body_1L_chr specifies the object to be updated. The function returns Function body (a character vector of length one).

Usage

```
replace_NA_in_fn(fn_body_1L_chr)
```

Arguments

fn_body_1L_chr Function body (a character vector of length one)

Value

Function body (a character vector of length one)

transform_alg_to_ref_cls_nm	<i>Transform algorithm to reference class name</i>
-----------------------------	--

Description

transform_alg_to_ref_cls_nm() is a Transform function that edits an object in such a way that core object attributes - e.g. shape, dimensions, elements, type - are altered. Specifically, this function implements an algorithm to transform algorithm to reference class name. Function argument alg_to_ref_cls_nm specifies the object to be updated. Argument pkg_nm_1L_chr provides the object to be updated. The function is called for its side effects and does not return a value.

Usage

```
transform_alg_to_ref_cls_nm(alg_to_ref_cls_nm, pkg_nm_1L_chr)
```

Arguments

alg_to_ref_cls_nm	PARAM_DESCRIPTION
pkg_nm_1L_chr	Package name (a character vector of length one)

Value

Transformed algorithm to reference class (a name)

transform_class_ns	<i>Transform class namespace</i>
--------------------	----------------------------------

Description

transform_class_ns() is a Transform function that edits an object in such a way that core object attributes - e.g. shape, dimensions, elements, type - are altered. Specifically, this function implements an algorithm to transform class namespace. Function argument class_ns_1L_chr specifies the object to be updated. Argument dev_pkg_ns_1L_chr provides the object to be updated. The function returns Transformed class namespace (a character vector of length one).

Usage

```
transform_class_ns(class_ns_1L_chr, dev_pkg_ns_1L_chr)
```

Arguments

class_ns_1L_chr	Class namespace (a character vector of length one)
dev_pkg_ns_1L_chr	Development package namespace (a character vector of length one)

Value

Transformed class namespace (a character vector of length one)

transform_fn_into_chr	<i>Transform function into</i>
-----------------------	--------------------------------

Description

transform_fn_into_chr() is a Transform function that edits an object in such a way that core object attributes - e.g. shape, dimensions, elements, type - are altered. Specifically, this function implements an algorithm to transform function into character vector. Function argument fn specifies the object to be updated. The function returns Function (a character vector of length one).

Usage

```
transform_fn_into_chr(fn)
```

Arguments

fn	Function (a function)
----	-----------------------

Value

Function (a character vector of length one)

`transform_parent_ns_ls`*Transform parent namespace*

Description

`transform_parent_ns_ls()` is a Transform function that edits an object in such a way that core object attributes - e.g. shape, dimensions, elements, type - are altered. Specifically, this function implements an algorithm to transform parent namespace list. Function argument `parent_ns_ls` specifies the object to be updated. The function returns Transformed parent namespace (a list).

Usage

```
transform_parent_ns_ls(parent_ns_ls)
```

Arguments

`parent_ns_ls` Parent namespace (a list)

Value

Transformed parent namespace (a list)

`transform_pt_ls_for_new_clss`*Transform prototype list for new classes*

Description

`transform_pt_ls_for_new_clss()` is a Transform function that edits an object in such a way that core object attributes - e.g. shape, dimensions, elements, type - are altered. Specifically, this function implements an algorithm to transform prototype list for new classes. Function argument `pts_for_new_clss_ls` specifies the object to be updated. The function returns Transformed prototypes for new classes (a list).

Usage

```
transform_pt_ls_for_new_clss(pts_for_new_clss_ls)
```

Arguments

`pts_for_new_clss_ls`
Prototypes for new classes (a list)

Value

Transformed prototypes for new classes (a list)

update_lup_for_ns	<i>Update lookup table for namespace</i>
-------------------	--

Description

update_lup_for_ns() is an Update Lookup Table for Namespace generic that updates a lookup table with namespace data.

Usage

```
update_lup_for_ns(x, attached_nss_chr, ...)
```

Arguments

x	An object
attached_nss_chr	Attached namespaces (a character vector)
...	Additional arguments

update_lup_for_ns.ready4_class_pt_lup	<i>Update lookup table for namespace method applied to readyforwhat-snext S3 class CLASS PROTOTYPE LOOKUP TABLE.</i>
---------------------------------------	--

Description

update_lup_for_ns.ready4_class_pt_lup() is an Update Lookup Table for Namespace method that updates a lookup table with namespace data. This method is implemented for the readyforwhatsnext S3 class CLASS PROTOTYPE LOOKUP TABLE. The function returns Instance (a readyforwhat-snext S3 class CLASS PROTOTYPE LOOKUP TABLE).

Usage

```
## S3 method for class 'ready4_class_pt_lup'
update_lup_for_ns(x, attached_nss_chr)
```

Arguments

x	An instance of readyforwhatsnext S3 class CLASS PROTOTYPE LOOKUP TABLE
attached_nss_chr	Attached namespaces (a character vector)

Value

Instance (a readyforwhatsnext S3 class CLASS PROTOTYPE LOOKUP TABLE)

validate_ready4_class_pt_lup
<i>Validate readyforwhatsnext S3 class CLASS PROTOTYPE LOOKUP TABLE</i>

Description

Validate an instance of the readyforwhatsnext S3 class CLASS PROTOTYPE LOOKUP TABLE

Usage

validate_ready4_class_pt_lup(x)

Arguments

x	An unvalidated instance of the readyforwhatsnext S3 class CLASS PROTOTYPE LOOKUP TABLE
---	--

Details

Readyforwhatsnext S3 class CLASS PROTOTYPE LOOKUP TABLE class of metadata of prototype classes.

Value

A prototpe for readyforwhatsnext S3 class CLASS PROTOTYPE LOOKUP TABLE

validate_ready4_constructor_tbl
<i>Validate readyforwhatsnext S3 class CLASS CONSTRUCTOR TABLE</i>

Description

Validate an instance of the readyforwhatsnext S3 class CLASS CONSTRUCTOR TABLE

Usage

validate_ready4_constructor_tbl(x)

Arguments

x	An unvalidated instance of the readyforwhatsnext S3 class CLASS CONSTRUCTOR TABLE
---	---

Details

Readyforwhatsnext S3 class CLASS CONSTRUCTOR TABLE of metadata required to make new classes.

Value

A prototpe for readyforwhatsnext S3 class CLASS CONSTRUCTOR TABLE

write_classes	<i>Write classes</i>
---------------	----------------------

Description

write_classes() is a Write Classes generic that writes new classes.

Usage

```
write_classes(x, ...)
```

Arguments

x	An object
...	Additional arguments

write_classes.ready4_constructor_tbl	<i>Write classes method applied to readyforwhatsnext S3 class CLASS CONSTRUCTOR TABLE.</i>
--------------------------------------	--

Description

write_classes.ready4_constructor_tbl() is a Write Classes method that writes new classes. This method is implemented for the readyforwhatsnext S3 class CLASS CONSTRUCTOR 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

```
## S3 method for class 'ready4_constructor_tbl'
write_classes(
  x,
  name_pfx_1L_chr,
  output_dir_1L_chr,
  file_exists_cdn_1L_chr = NULL,
  prototype_lup = NULL,
  nss_to_ignore_chr,
  req_pkgs_chr = NA_character_,
  class_in_cache_cdn_1L_chr = "stop",
  abbreviations_lup = NULL
)
```

Arguments

x	An instance of readyforwhatsnext S3 class CLASS CONSTRUCTOR TABLE
name_pfx_1L_chr	Name prefix (a character vector of length one)
output_dir_1L_chr	Output directory (a character vector of length one)
file_exists_cdn_1L_chr	File exists condition (a character vector of length one), Default: NULL
prototype_lup	Prototype (a lookup table), Default: NULL
nss_to_ignore_chr	Namespaces to ignore (a character vector)
req_pkgs_chr	Req packages (a character vector), Default: 'NA'
class_in_cache_cdn_1L_chr	Class in cache condition (a character vector of length one), Default: 'stop'
abbreviations_lup	Abbreviations (a lookup table), Default: NULL

write_classes_and_make_lup

Write classes and make lookup table

Description

write_classes_and_make_lup() is a Write Classes and Make Lookup Table generic that makes new classes and creates or updates a class prototype lookup table.

Usage

```
write_classes_and_make_lup(x, ...)
```

Arguments

x	An object
...	Additional arguments

write_classes_and_make_lup.ready4_constructor_tbl

*Write classes and make lookup table method applied to readyforwhat-
snext S3 class CLASS CONSTRUCTOR TABLE.*

Description

write_classes_and_make_lup.ready4_constructor_tbl() is a Write Classes and Make Lookup Table method that makes new classes and creates or updates a class prototype lookup table. This method is implemented for the readyforwhatsnext S3 class CLASS CONSTRUCTOR TABLE. The function returns Instance (a readyforwhatsnext S3 class CLASS PROTOTYPE LOOKUP TABLE).

Usage

```
## S3 method for class 'ready4_constructor_tbl'
write_classes_and_make_lup(
  x,
  dev_pkg_ns_1L_chr = ready4fun::get_dev_pkg_nm(),
  name_pfx_1L_chr = paste0(ready4fun::get_dev_pkg_nm(), "_"),
  output_dir_1L_chr = "R",
  delete_cdn_ptrn_chr = NA_character_,
  file_exists_cdn_1L_chr = "overwrite",
  init_class_pt_lup = NULL,
  nss_to_ignore_chr = NA_character_,
  req_pkgs_chr = NA_character_,
  class_in_cache_cdn_1L_chr = "stop",
  abbreviations_lup = NULL
)
```

Arguments

x	An instance of readyforwhatsnext S3 class CLASS CONSTRUCTOR TABLE
dev_pkg_ns_1L_chr	Development package namespace (a character vector of length one), Default: ready4fun::get_dev_pkg_nm()
name_pfx_1L_chr	Name prefix (a character vector of length one), Default: paste0(ready4fun::get_dev_pkg_nm(), "_")
output_dir_1L_chr	Output directory (a character vector of length one), Default: 'R'
delete_cdn_ptrn_chr	Delete condition pattern (a character vector), Default: 'NA'
file_exists_cdn_1L_chr	File exists condition (a character vector of length one), Default: 'overwrite'
init_class_pt_lup	Init class prototype (a lookup table), Default: NULL
nss_to_ignore_chr	Namespaces to ignore (a character vector), Default: 'NA'
req_pkgs_chr	Req packages (a character vector), Default: 'NA'
class_in_cache_cdn_1L_chr	Class in cache condition (a character vector of length one), Default: 'stop'
abbreviations_lup	Abbreviations (a lookup table), Default: NULL

Value

Instance (a readyforwhatsnext S3 class CLASS PROTOTYPE LOOKUP TABLE)

```
write_gtr_str_mthds_for_r4
```

Write getter setter methods for

Description

`write_gtr_str_mthds_for_r4()` is a Write function that writes a file to a specified local directory. Specifically, this function implements an algorithm to write getter setter methods for readyforwhat-snext s4. 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_gtr_str_mthds_for_r4(
  slot_nm_1L_chr,
  set_only_1L_lgl,
  pkgs_to_imp_ls,
  class_nm_1L_chr,
  print_gtrs_strs_1L_lgl,
  output_dir_1L_chr
)
```

Arguments

`slot_nm_1L_chr` Slot name (a character vector of length one)
`set_only_1L_lgl` Set only (a logical vector of length one)
`pkgs_to_imp_ls` Packages to import (a list)
`class_nm_1L_chr` Class name (a character vector of length one)
`print_gtrs_strs_1L_lgl` Print getters setters (a logical vector of length one)
`output_dir_1L_chr` Output directory (a character vector of length one)

```
write_gtr_str_mthds_for_slots
```

Write getter setter methods for slots

Description

`write_gtr_str_mthds_for_slots()` is a Write function that writes a file to a specified local directory. Specifically, this function implements an algorithm to write getter setter methods for slots. 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_gtr_str_mthds_for_slots(
  slot_names_chr,
  set_only_chr,
  parent_cls_nm_1L_chr,
  class_nm_1L_chr,
  print_gtrs_strs_1L_lgl,
  output_dir_1L_chr,
  nss_to_ignore_chr,
  req_pkgs_chr
)
```

Arguments

slot_names_chr Slot names (a character vector)

set_only_chr Set only (a character vector)

parent_cls_nm_1L_chr
Parent class name (a character vector of length one)

class_nm_1L_chr
Class name (a character vector of length one)

print_gtrs_strs_1L_lgl
Print getters setters (a logical vector of length one)

output_dir_1L_chr
Output directory (a character vector of length one)

nss_to_ignore_chr
Namespaces to ignore (a character vector)

req_pkgs_chr Req packages (a character vector)

```
write_mthds_for_r3_or_r4_clss
```

Write methods for readyforwhatsnext S3 or readyforwhatsnext S4 classes

Description

write_mthds_for_r3_or_r4_clss() is a Write function that writes a file to a specified local directory. Specifically, this function implements an algorithm to write methods for readyforwhatsnext s3 or readyforwhatsnext s4 classes. 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_mthds_for_r3_or_r4_clss(
  methods_tb,
  fn_ls,
  pkg_nm_1L_chr,
  output_dir_1L_chr
)
```


Arguments

methods_tb	Methods (a tibble)
fn_ls	Function list (a list of functions)
pkg_nm_1L_chr	Package name (a character vector of length one)
output_dir_1L_chr	Output directory (a character vector of length one)

write_scripts_to_make_gnrc_and_mthd

Write scripts to make generic and method

Description

write_scripts_to_make_gnrc_and_mthd() is a Write function that writes a file to a specified local directory. Specifically, this function implements an algorithm to write scripts to make generic and method. 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_scripts_to_make_gnrc_and_mthd(
  fn_name_1L_chr,
  args_chr = c("x"),
  signature_1L_chr = NA_character_,
  pkg_nm_1L_chr = NA_character_,
  where_chr = NA_character_,
  class_nm_1L_chr,
  fn,
  fn_type_chr,
  fn_desc_chr = rep(NA_character_, 2),
  fn_title_1L_chr = NA_character_,
  fn_outp_type_1L_chr = NA_character_,
  imports_chr,
  write_file_ls,
  output_dir_1L_chr,
  append_1L_lgl = T,
  doc_in_class_1L_lgl = F,
  gnrc_exists_1L_lgl,
  overwrite_1L_lgl = F,
  s3_1L_lgl,
  write_1L_lgl
)
```

Arguments

fn_name_1L_chr	Function name (a character vector of length one)
args_chr	Arguments (a character vector), Default: c("x")
signature_1L_chr	Signature (a character vector of length one), Default: 'NA'

pkg_nm_1L_chr	Package name (a character vector of length one), Default: 'NA'
where_chr	Where (a character vector), Default: 'NA'
class_nm_1L_chr	Class name (a character vector of length one)
fn	Function (a function)
fn_type_chr	Function type (a character vector)
fn_desc_chr	Function description (a character vector), Default: rep(NA_character_, 2)
fn_title_1L_chr	Function title (a character vector of length one), Default: 'NA'
fn_outp_type_1L_chr	Function output type (a character vector of length one), Default: 'NA'
imports_chr	Imports (a character vector)
write_file_ls	Write file (a list)
output_dir_1L_chr	Output directory (a character vector of length one)
append_1L_lgl	Append (a logical vector of length one), Default: T
doc_in_class_1L_lgl	Document in class (a logical vector of length one), Default: F
gnrc_exists_1L_lgl	Generic exists (a logical vector of length one)
overwrite_1L_lgl	Overwrite (a logical vector of length one), Default: F
s3_1L_lgl	S3 (a logical vector of length one)
write_1L_lgl	Write (a logical vector of length one)

```
write_scripts_to_mk_clss
```

Write scripts to make classes

Description

write_scripts_to_mk_clss() is a Write function that writes a file to a specified local directory. Specifically, this function implements an algorithm to write scripts to make classes. 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_scripts_to_mk_clss(
  pts_for_new_clss_ls,
  pkg_nm_1L_chr,
  class_pfx_1L_chr,
  R_dir_1L_chr = "R",
  pt_lup,
  description_ls = NULL,
  nss_to_ignore_chr = NA_character_,
  req_pkgs_chr = NA_character_
)
```

Arguments

pts_for_new_cls_ls
 Prototypes for new classes (a list)

pkg_nm_1L_chr Package name (a character vector of length one)

class_pfx_1L_chr
 Class prefix (a character vector of length one)

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

pt_lup Prototype (a lookup table)

description_ls Description (a list), Default: NULL

nss_to_ignore_chr
 Namespaces to ignore (a character vector), Default: 'NA'

req_pkgs_chr Req packages (a character vector), Default: 'NA'

write_scripts_to_mk_r3_cls

Write scripts to make readyforwhatsnext S3 class

Description

write_scripts_to_mk_r3_cls() is a Write function that writes a file to a specified local directory. Specifically, this function implements an algorithm to write scripts to make readyforwhatsnext s3 class. 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_scripts_to_mk_r3_cls(
  name_stub_1L_chr,
  name_pfx_1L_chr = "ready4_",
  output_dir_1L_chr = "data-raw",
  class_desc_1L_chr = "",
  parent_cls_nm_1L_chr = NULL,
  type_1L_chr,
  pt_ns_1L_chr = "",
  pt_chkr_pfx_1L_chr = "is.",
  vals_ls = NULL,
  ordered_1L_lgl = FALSE,
  allowed_vals_ls = NULL,
  min_max_vals_dbl = NULL,
  start_end_vals_dbl = NULL,
  prototype_lup,
  nss_to_ignore_chr = NA_character_,
  file_exists_cdn_1L_chr = "skip",
  abbreviations_lup = NULL
)
```

Arguments

name_stub_1L_chr	Name stub (a character vector of length one)
name_pfx_1L_chr	Name prefix (a character vector of length one), Default: 'ready4_'
output_dir_1L_chr	Output directory (a character vector of length one), Default: 'data-raw'
class_desc_1L_chr	Class description (a character vector of length one), Default: ''
parent_cls_nm_1L_chr	Parent class name (a character vector of length one), Default: NULL
type_1L_chr	Type (a character vector of length one)
pt_ns_1L_chr	Prototype namespace (a character vector of length one), Default: ''
pt_chkr_pfx_1L_chr	Prototype checker prefix (a character vector of length one), Default: 'is.'
vals_ls	Values (a list), Default: NULL
ordered_1L_lgl	Ordered (a logical vector of length one), Default: FALSE
allowed_vals_ls	Allowed values (a list), Default: NULL
min_max_vals_dbl	Min max values (a double vector), Default: NULL
start_end_vals_dbl	Start end values (a double vector), Default: NULL
prototype_lup	Prototype (a lookup table)
nss_to_ignore_chr	Namespaces to ignore (a character vector), Default: 'NA'
file_exists_cdn_1L_chr	File exists condition (a character vector of length one), Default: 'skip'
abbreviations_lup	Abbreviations (a lookup table), Default: NULL

write_scripts_to_mk_r4_cls

Write scripts to make readyforwhatsnext S4 class

Description

write_scripts_to_mk_r4_cls() is a Write function that writes a file to a specified local directory. Specifically, this function implements an algorithm to write scripts to make readyforwhatsnext s4 class. 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_scripts_to_mk_r4_cls(
  name_stub_1L_chr,
  name_pfx_1L_chr = "ready4_",
  output_dir_1L_chr = "data-raw",
  outp_sub_dir_1L_chr = NULL,
  class_desc_1L_chr = "",
  parent_cls_nm_1L_chr = NULL,
  slots_chr,
  type_chr,
  meaningful_nms_ls = NULL,
  vals_ls = NULL,
  allowed_vals_ls = NULL,
  clss_to_inc_chr = NULL,
  prototype_lup,
  nss_to_ignore_chr = NA_character_,
  req_pkgs_chr = NA_character_,
  names_must_match_ls = NULL,
  slots_of_dif_lnts_chr = NULL,
  print_set_cls_1L_lgl = TRUE,
  print_helper = TRUE,
  print_gtrs_strs_1L_lgl = TRUE,
  print_validator_1L_lgl = TRUE,
  print_meaningful_nms_ls_1L_lgl = TRUE,
  class_in_cache_cdn_1L_chr = "stop"
)

```

Arguments

name_stub_1L_chr	Name stub (a character vector of length one)
name_pfx_1L_chr	Name prefix (a character vector of length one), Default: 'ready4_'
output_dir_1L_chr	Output directory (a character vector of length one), Default: 'data-raw'
outp_sub_dir_1L_chr	Output sub directory (a character vector of length one), Default: NULL
class_desc_1L_chr	Class description (a character vector of length one), Default: ''
parent_cls_nm_1L_chr	Parent class name (a character vector of length one), Default: NULL
slots_chr	Slots (a character vector)
type_chr	Type (a character vector)
meaningful_nms_ls	Meaningful names (a list), Default: NULL
vals_ls	Values (a list), Default: NULL
allowed_vals_ls	Allowed values (a list), Default: NULL
clss_to_inc_chr	Classes to include (a character vector), Default: NULL

```

prototype_lup    Prototype (a lookup table)
nss_to_ignore_chr
                  Namespaces to ignore (a character vector), Default: 'NA'
req_pkgs_chr     Req packages (a character vector), Default: 'NA'
names_must_match_ls
                  Names must match (a list), Default: NULL
slots_of_dif_lnts_chr
                  Slots of different lengths (a character vector), Default: NULL
print_set_cls_1L_lgl
                  Print set class (a logical vector of length one), Default: TRUE
print_helper     PARAM_DESCRIPTION, Default: TRUE
print_gtrs_strs_1L_lgl
                  Print getters setters (a logical vector of length one), Default: TRUE
print_validator_1L_lgl
                  Print validator (a logical vector of length one), Default: TRUE
print_meaningful_nms_ls_1L_lgl
                  Print meaningful names list (a logical vector of length one), Default: TRUE
class_in_cache_cdn_1L_chr
                  Class in cache condition (a character vector of length one), Default: 'stop'

```

```
write_script_to_make_gnrc
```

```
    Write script to make generic
```

Description

write_script_to_make_gnrc() is a Write function that writes a file to a specified local directory. Specifically, this function implements an algorithm to write script to make generic. 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_script_to_make_gnrc(
  write_file_ls,
  gnrc_exists_1L_lgl,
  gen_mthd_pair_ls,
  fn_name_1L_chr,
  fn_type_1L_chr,
  fn_desc_1L_chr = NA_character_,
  fn_outp_type_1L_chr = NA_character_,
  fn_title_1L_chr = NA_character_,
  class_nm_1L_chr = NA_character_,
  output_dir_1L_chr = NA_character_,
  overwrite_1L_lgl = F,
  s3_1L_lgl = F,
  write_1L_lgl = T,
  doc_in_class_1L_lgl = F
)

```

Arguments

write_file_ls Write file (a list)
 gnrc_exists_1L_lgl
 Generic exists (a logical vector of length one)
 gen_mthd_pair_ls
 Generate method pair (a list)
 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_desc_1L_chr Function description (a character vector of length one), Default: 'NA'
 fn_outp_type_1L_chr
 Function output type (a character vector of length one), Default: 'NA'
 fn_title_1L_chr
 Function title (a character vector of length one), Default: 'NA'
 class_nm_1L_chr
 Class name (a character vector of length one), Default: 'NA'
 output_dir_1L_chr
 Output directory (a character vector of length one), Default: 'NA'
 overwrite_1L_lgl
 Overwrite (a logical vector of length one), Default: F
 s3_1L_lgl
 S3 (a logical vector of length one), Default: F
 write_1L_lgl Write (a logical vector of length one), Default: T
 doc_in_class_1L_lgl
 Document in class (a logical vector of length one), Default: F

 write_script_to_make_mthd

Write script to make method

Description

write_script_to_make_mthd() is a Write function that writes a file to a specified local directory. Specifically, this function implements an algorithm to write script to make method. 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_script_to_make_mthd(
  write_file_ls,
  gen_mthd_pair_ls,
  class_nm_1L_chr,
  fn_name_1L_chr,
  fn_type_1L_chr,
  fn_desc_1L_chr = NA_character_,
  fn_outp_type_1L_chr = NA_character_,
  imports_chr,
  write_1L_lgl = T,
  append_1L_lgl = T,
  doc_in_class_1L_lgl = F
)

```

Arguments

write_file_ls Write file (a list)
 gen_mthd_pair_ls
 Generate method pair (a list)
 class_nm_1L_chr
 Class name (a character vector of length one)
 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_desc_1L_chr Function description (a character vector of length one), Default: 'NA'
 fn_outp_type_1L_chr
 Function output type (a character vector of length one), Default: 'NA'
 imports_chr Imports (a character vector)
 write_1L_lgl Write (a logical vector of length one), Default: T
 append_1L_lgl Append (a logical vector of length one), Default: T
 doc_in_class_1L_lgl
 Document in class (a logical vector of length one), Default: F

write_slot_gtr_str_mthds

Write slot getter setter methods

Description

write_slot_gtr_str_mthds() is a Write function that writes a file to a specified local directory. Specifically, this function implements an algorithm to write slot getter setter methods. 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_slot_gtr_str_mthds(
  slot_nm_1L_chr,
  set_only_1L_lgl,
  parent_cls_nm_1L_chr,
  class_nm_1L_chr,
  print_gtrs_strs_1L_lgl,
  output_dir_1L_chr,
  nss_to_ignore_chr,
  req_pkgs_chr
)

```

Arguments

slot_nm_1L_chr Slot name (a character vector of length one)
 set_only_1L_lgl
 Set only (a logical vector of length one)
 parent_cls_nm_1L_chr
 Parent class name (a character vector of length one)

class_nm_1L_chr	Class name (a character vector of length one)
print_gtrs_strs_1L_lgl	Print getters setters (a logical vector of length one)
output_dir_1L_chr	Output directory (a character vector of length one)
nss_to_ignore_chr	Namespaces to ignore (a character vector)
req_pkgs_chr	Req packages (a character vector)

write_std_mthd	<i>Write standard method</i>
----------------	------------------------------

Description

write_std_mthd() is a Write function that writes a file to a specified local directory. Specifically, this function implements an algorithm to write standard method. 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_mthd(
  fn,
  fn_name_1L_chr,
  class_nm_1L_chr,
  fn_desc_chr,
  fn_title_1L_chr,
  fn_outp_type_1L_chr,
  pkg_nm_1L_chr,
  output_dir_1L_chr,
  signature_1L_chr = NA_character_,
  append_1L_lgl = T,
  first_1L_lgl = T
)
```

Arguments

fn	Function (a function)
fn_name_1L_chr	Function name (a character vector of length one)
class_nm_1L_chr	Class name (a character vector of length one)
fn_desc_chr	Function description (a character vector)
fn_title_1L_chr	Function title (a character vector of length one)
fn_outp_type_1L_chr	Function output type (a character vector of length one)
pkg_nm_1L_chr	Package name (a character vector of length one)

output_dir_1L_chr	Output directory (a character vector of length one)
signature_1L_chr	Signature (a character vector of length one), Default: 'NA'
append_1L_lgl	Append (a logical vector of length one), Default: T
first_1L_lgl	First (a logical vector of length one), Default: T

write_to_delete_fls_with_ptrn	<i>Write to delete files with pattern</i>
-------------------------------	---

Description

write_to_delete_fls_with_ptrn() is a Write function that writes a file to a specified local directory. Specifically, this function implements an algorithm to write to delete files with pattern. 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_delete_fls_with_ptrn(dir_1L_chr, pattern_1L_chr)
```

Arguments

dir_1L_chr	Directory (a character vector of length one)
pattern_1L_chr	Pattern (a character vector of length one)

write_to_delete_gnrc_fn_fls	<i>Write to delete generic function files</i>
-----------------------------	---

Description

write_to_delete_gnrc_fn_fls() is a Write function that writes a file to a specified local directory. Specifically, this function implements an algorithm to write to delete generic 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_to_delete_gnrc_fn_fls(x, output_dir_1L_chr)
```

Arguments

x	PARAM_DESCRIPTION
output_dir_1L_chr	Output directory (a character vector of length one)

write_to_mk_r4_cls	<i>Write to make readyforwhatsnext S4 class</i>
--------------------	---

Description

write_to_mk_r4_cls() is a Write function that writes a file to a specified local directory. Specifically, this function implements an algorithm to write to make readyforwhatsnext s4 class. 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_mk_r4_cls(
  class_nm_1L_chr,
  slots_chr,
  type_chr,
  pt_ls,
  parent_cls_nm_1L_chr,
  print_set_cls_1L_lgl,
  class_desc_1L_chr,
  output_file_class,
  clss_to_inc_chr,
  prototype_lup,
  helper_lgl = F,
  parent_ns_ls
)
```

Arguments

class_nm_1L_chr	Class name (a character vector of length one)
slots_chr	Slots (a character vector)
type_chr	Type (a character vector)
pt_ls	Prototype (a list)
parent_cls_nm_1L_chr	Parent class name (a character vector of length one)
print_set_cls_1L_lgl	Print set class (a logical vector of length one)
class_desc_1L_chr	Class description (a character vector of length one)
output_file_class	PARAM_DESCRIPTION
clss_to_inc_chr	Classes to include (a character vector)
prototype_lup	Prototype (a lookup table)
helper_lgl	Helper (a logical vector), Default: F
parent_ns_ls	Parent namespace (a list)

Index

* datasets

- abbreviations_lup, [5](#)
- fn_type_lup_tb, [7](#)
- prototype_lup, [36](#)

- abbreviations_lup, [5](#)
- add_class, [5](#)
- add_class.ready4_class_pt_lup, [6](#)

- fn_type_lup_tb, [7](#)

- get_class_fl_nms, [7](#)
- get_class_ns, [8](#)
- get_nms_of_cls_to_inc, [8](#)
- get_nms_of_curr_gnrcls, [9](#)
- get_parent_cls_ns, [10](#)
- get_parent_cls_pt_fn, [11](#)
- get_parent_cls_pts, [10](#)
- get_parent_cls_slot_nms, [11](#)

- is_ready4_class_pt_lup, [12](#)
- is_ready4_constructor_tbl, [12](#)

- make_alg_to_gen_ref_to_cls, [13](#)
- make_alg_to_get_pt_val, [13](#)
- make_alg_to_set_gnrc, [14](#)
- make_alg_to_set_mthd, [15](#)
- make_alg_to_set_old_cls, [15](#)
- make_alg_to_set_validity_of_r4_cls, [16](#)
- make_alg_to_write_gtr_str_mthds, [17](#)
- make_child_cls_fn_body, [18](#)
- make_class_pt_tb_for_r3_and_r4_cls, [19](#)
- make_class_pts_tb, [18](#)
- make_dmt_inc_tag, [19](#)
- make_fn_pt_to_check_r3_cls_inhtc, [20](#)
- make_fn_pt_to_make_r3_cls_pt, [20](#)
- make_fn_pt_to_make_unvld_r3_cls_inst, [21](#)
- make_fn_pt_to_make_vld_r3_cls_inst, [22](#)
- make_fn_pt_to_make_vld_r3_cls_inst, [22](#)
- make_gnrc_fn, [23](#)
- make_gnrc_mthd_pair_ls, [24](#)
- make_helper_fn, [24](#)

- make_lines_for_writing_dmt_d_fn, [25](#)
- make_ls_of_pkgs_to_imp, [26](#)
- make_ls_of_tfd_nms_of_curr_gnrcls, [26](#)
- make_lup, [27](#)
- make_lup.ready4_constructor_tbl, [27](#)
- make_new_ready4_class_pt_lup, [28](#)
- make_new_ready4_constructor_tbl, [29](#)
- make_one_row_class_pt_tb, [29](#)
- make_one_row_pt_tb_for_new_r3_cls, [30](#)
- make_one_row_pt_tb_for_new_r4_cls, [30](#)
- make_prototype_ready4_class_pt_lup, [31](#)
- make_prototype_ready4_constructor_tbl, [31](#)
- make_pt_ls, [32](#)
- make_pt_ls_for_new_r3_cls, [32](#)
- make_pt_tb_for_new_r3_cls, [33](#)
- make_pt_tb_for_new_r4_cls, [34](#)
- make_show_mthd_fn, [34](#)

- order_tb, [35](#)
- order_tb.ready4_constructor_tbl, [35](#)

- prototype_lup, [36](#)

- ready4_class_pt_lup, [36](#)
- ready4_constructor_tbl, [37](#)
- ready4class (ready4class-package), [4](#)
- ready4class-package, [4](#)
- remake_ls_cols, [37](#)
- remake_ls_cols.ready4_constructor_tbl, [38](#)
- remove_class_from_cache, [38](#)
- replace_NA_in_fn, [39](#)

- transform_alg_to_ref_cls_nm, [39](#)
- transform_class_ns, [40](#)
- transform_fn_into_chr, [40](#)
- transform_parent_ns_ls, [41](#)
- transform_pt_ls_for_new_cls, [41](#)

- update_lup_for_ns, [42](#)
- update_lup_for_ns.ready4_class_pt_lup, [42](#)

- validate_ready4_class_pt_lup, [43](#)

validate_ready4_constructor_tbl, [43](#)

write_classes, [44](#)

write_classes.ready4_constructor_tbl,
[44](#)

write_classes_and_make_lup, [45](#)

write_classes_and_make_lup.ready4_constructor_tbl,
[45](#)

write_gtr_str_mthds_for_r4, [47](#)

write_gtr_str_mthds_for_slots, [47](#)

write_mthds_for_r3_or_r4_cls, [48](#)

write_script_to_make_gnrc, [54](#)

write_script_to_make_mthd, [55](#)

write_scripts_to_make_gnrc_and_mthd,
[49](#)

write_scripts_to_mk_cls, [50](#)

write_scripts_to_mk_r3_cls, [51](#)

write_scripts_to_mk_r4_cls, [52](#)

write_slot_gtr_str_mthds, [56](#)

write_std_mthd, [57](#)

write_to_delete_fls_with_ptrn, [58](#)

write_to_delete_gnrc_fn_fls, [58](#)

write_to_mk_r4_cls, [59](#)