

# Package ‘phuse’

October 18, 2019

**Type** Package

**Title** Web Application Framework for 'PhUSE' Scripts

**Version** 0.2.0

**Author** Hanming Tu [aut, cre]

**Maintainer** Hanming Tu <hanming.tu@gmail.com>

**Description** Make it easy to review, download and execute scripts stored in Github 'phuse-scripts' repository <<https://github.com/phuse-org/phuse-scripts>>. Some examples included show the web application framework using the script metadata. The 'PhUSE' is Pharmaceutical Users Software Exchange <<http://www.phuse.eu>>.

**Depends** R (>= 3.0.1)

**License** MIT + file LICENSE

**Encoding** UTF-8

**LazyData** true

**RoxygenNote** 6.1.1

**Suggests** testthat

**Imports** yaml, shiny, RCurl, git2r, rlist, httr, stringr, jsonlite, SASxport

**NeedsCompilation** no

**Repository** CRAN

**Date/Publication** 2019-10-17 22:30:02 UTC

## R topics documented:

build_inputs . . . . .	2
build_script_df . . . . .	3
chk_workdir . . . . .	4
clone_github . . . . .	5
create_dir . . . . .	6
crt_workdir . . . . .	6
cvt_class2df . . . . .	7
cvt_list2df . . . . .	8

download_fns . . . . .	8
download_script . . . . .	9
download_script_files . . . . .	10
echo_msg . . . . .	11
extract_fns . . . . .	12
gen_simplified_ts . . . . .	12
get_inputs . . . . .	13
get_yaml_inputs . . . . .	14
init_cfg . . . . .	15
is_empty . . . . .	15
merge_lists . . . . .	16
read_yaml . . . . .	17
resolve . . . . .	17
run_example . . . . .	18
search_api . . . . .	19
search_github . . . . .	20
start_app . . . . .	21
start_phuse . . . . .	22
url.exists . . . . .	23
<b>Index</b>	<b>24</b>

---

 build\_inputs

*Build Inputs from Script Metadata for Phuse Web Framework*


---

### Description

Build R shiny code for Phuse Web Apps

### Usage

```
build_inputs(fn = NULL)
```

### Arguments

fn                    a file name or URL pointing to script metadata file

### Value

R shiny code for providing inputs to the script

### Author(s)

Hanming Tu

**Examples**

```
## Not run:
a <- "https://github.com/phuse-org/phuse-scripts/raw/master"
b <- "development/R/scripts"
c <- "Draw_Dist2_R.yml"
f1 <- paste(a,b,c, sep = '/')
r1 <- build_inputs(f1)

## End(Not run)
```

---

build_script_df	<i>Build Script Index Dataset</i>
-----------------	-----------------------------------

---

**Description**

Grep all the YML files, parse the metadata and build a data frame containing key metadata tags.

**Usage**

```
build_script_df(repo_url = "https://github.com/phuse-org/phuse-scripts.git",
  repo_base = "https://github.com/phuse-org/phuse-scripts/raw/master",
  repo_dir = NULL, work_dir = NULL, output_fn = NULL,
  days_to_update = 7, fn_only = FALSE, upd_opt = NULL)
```

**Arguments**

repo_url	a URL for a remote repository and default to 'https://github.com/phuse-org/phuse-scripts.git'
repo_base	a URL for repository base folder; default to "https://github.com/phuse-org/phuse-scripts/raw/master"
repo_dir	a local directory to host the repository; default to work_dir from crt_workdir if not specified
work_dir	a local directory to host the files containing a list of YML files; default to tempdir()/myRepo
output_fn	a CSV file name for outputting a list of YML files; default to "repo_name.yml.csv"
days_to_update	number of days before the output_fn is updated; default to 7 days. Set it to a negative number make it to update immediately.
fn_only	return file name only; default to FALSE
upd_opt	update option: File Repo Both

**Value**

a data frame containing a list of script metadata

**Author(s)**

Hanming Tu

**Examples**

```
## Not run:
r1 <- build_script_df()
r2 <- build_script_df(upd_opt = "file")
r3 <- build_script_df(upd_opt = "repo")
r4 <- build_script_df(upd_opt = "both")

## End(Not run)
```

---

chk\_workdir

*Verify work directory*

---

**Description**

Verify if the dir is the work directory

**Usage**

```
chk_workdir(dir, top_dir = NULL, sub_dir = "myRepo")
```

**Arguments**

dir	a work directory; default to '/Users/user for Mac; "c:/tmp" for Windows
top_dir	a top or root directory; default to '/Users/user for Mac or getwd for other OS
sub_dir	a sub directory

**Value**

TRUE or FALSE

**Author(s)**

Hanming Tu

**Examples**

```
f1 <- tempdir()
r1 <- chk_workdir(f1)
```

---

clone_github	<i>Clone a GitHub repository</i>
--------------	----------------------------------

---

## Description

Clone a specified GitHub.

## Usage

```
clone_github(repo_url = "https://github.com/phuse-org/phuse-scripts.git",  
             repo_dir = NULL, repo_name = NULL, upd_opt = NULL)
```

## Arguments

repo_url	a URL for a remote repository and default to 'https://github.com/phuse-org/phuse-scripts.git'
repo_dir	a local directory to host the repository; default to work_dir from crt_workdir if not specified
repo_name	repo name; default to the repo name in repo_url.
upd_opt	update option: Repo

## Value

nothing.

## Author(s)

Hanming Tu

## Examples

```
## Not run:  
r1 <- clone_github()  
  
## End(Not run)
```

---

create_dir	<i>Create a directory</i>
------------	---------------------------

---

**Description**

create a directory

**Usage**

```
create_dir(r_dir, s_dir = NULL)
```

**Arguments**

r_dir	root directory
s_dir	sub directory

**Value**

directory name

**Author(s)**

Hanming Tu

**Examples**

```
## Not run:  
s1 <- tempdir()  
s2 <- "myRepo"  
# create dir /{s1}/myRepo if it does not exist  
d1 <- create_dir(s1, s2)  
  
## End(Not run)
```

---

crt_workdir	<i>Create work directory</i>
-------------	------------------------------

---

**Description**

define and create a work directory.

**Usage**

```
crt_workdir(top_dir = NULL, sub_dir = "myRepo", to_crt_dir = TRUE)
```

**Arguments**

top\_dir            a top or root directory; default to '/Users/user for Mac or getwd for other OS  
sub\_dir            a sub directory  
to\_crt\_dir        whether to create the dir; default to TRUE. If FALSE, just return the dir name

**Value**

the created directory

**Author(s)**

Hanming Tu

**Examples**

```
## Not run:  
d1 <- tempdir()  
r1 <- crt_workdir(d1)  
r2 <- crt_workdir(d1, to_crt_dir = FALSE) # just return the dir  
  
## End(Not run)
```

---

cvt\_class2df

*Convert a class to data fram*

---

**Description**

Convert class or list to a data frame

**Usage**

```
cvt_class2df(x, exc = "^_@", condition = FALSE)
```

**Arguments**

x                    a class or list  
exc                  exclude pattern  
condition            condition for excluding

**Author(s)**

Hanming Tu

**Examples**

```
r1 <- Sys.getenv()  
r2 <- cvt_class2df(r1)
```

cvt\_list2df

*Convert list to data frame*

---

**Description**

convert list to a data frame with the following structure: variable, level, type, value

**Usage**

```
cvt_list2df(a)
```

**Arguments**

a                    a list returned by read\_yaml or any list

**Value**

data frame

**Author(s)**

Hanming Tu

**Examples**

```
## Not run:
a <- "https://github.com/phuse-org/phuse-scripts/raw/master"
b <- "development/R/scripts"
c <- "Draw_Dist2_R.yml"
f1 <- paste(a,b,c, sep = '/')
r1 <- read_yaml(f1)
r2 <- cvt_list2df(r1)

## End(Not run)
```

---

download\_fns*Download files from a repository*

---

**Description**

download files defined in the input data frame.

**Usage**

```
download_fns(df, tgtDir = NULL,
             baseDir = "https://github.com/phuse-org/phuse-scripts/raw/master")
```



**Arguments**

df	a data frame containing file names produced from extract_fns
tgtDir	target directory for storing the files
baseDir	base directory in the repository including the repo URL. Default to "https://github.com/phuse-org/phuse-scripts/raw/master"

**Author(s)**

Hanming Tu

**Examples**

```
## Not run:
a <- "https://github.com/phuse-org/phuse-scripts/raw/master"
b <- "development/R/scripts"
c <- "Draw_Dist2_R.yml"
f1 <- paste(a,b,c, sep = '/')
f2 <- read_yaml(f1)
f3 <- extract_fns(f2)
f4 <- download_fns(f3)

## End(Not run)
```

---

download_script	<i>Download files defined in script metadata</i>
-----------------	--

---

**Description**

download scripts, data or any file defined in the script metadata.

**Usage**

```
download_script(cfg, wkDir = "workdir", source_lib = TRUE)
```

**Arguments**

cfg	a list containing script metadata
wkDir	work directory where the files will be downloaded to
source_lib	whether to source the library defined for the script in the metadata

**Value**

target directory name

**Author(s)**

Hanming Tu

## Examples

```
## Not run:
a <- "https://github.com/phuse-org/phuse-scripts/raw/master"
b <- "development/R/scripts"
c <- "Draw_Dist2_R.yml"
f1 <- paste(a,b,c, sep = '/')
f2 <- read_yaml(f1)
f3 <- download_script(f2)

## End(Not run)
```

---

download\_script\_files *Download files from a repository*

---

## Description

download files defined in the input list from a repository.

## Usage

```
download_script_files(fns, tgtDir,
  baseDir = "https://github.com/phuse-org/phuse-scripts/raw/master",
  scriptDir = "data/send/PDS/Xpt")
```

## Arguments

fns	a list containing file names
tgtDir	target directory for storing the files
baseDir	base directory in the repository including the repo URL. Default to "https://github.com/phuse-org/phuse-scripts/raw/master"
scriptDir	script directory in the repository

## Author(s)

Hanming Tu

## Examples

```
## Not run:
fns <- c("dm.xpt", "ex.xpt")
dir <- tempdir()
a <- download_script_files(fns, dir)

## End(Not run)
```

---

`echo_msg`*Echo message*

---

**Description**

This method displays or writes the message based on debug level. The filehandler is provided through environment variable 'log\_fn', and the outputs are written to the file. This method will display message or a hash array based on debug level ('d\_level'). If 'd\_level' is set to '0', no message or array will be displayed. If 'd\_level' is set to '2', it will only display the message level (lvl) is less than or equal to '2'. If you call this method without providing a message level, the message level (lvl) is default to '0'. Of course, if no message is provided to the method, it will be quietly returned. If 'd\_level' is set to '1', all the messages with default message level, i.e., 0, and '1' will be displayed. The higher level messages will not be displayed.

**Usage**

```
echo_msg(prg, step, msg, lvl = 0, fn = NULL)
```

**Arguments**

<code>prg</code>	program name calling from
<code>step</code>	step in the program
<code>msg</code>	the message to be displayed. No newline is needed in the end of the message. It will add the newline code at the end of the message.
<code>lvl</code>	the message level is assigned to the message. If it is higher than the debug level, then the message will not be displayed.
<code>fn</code>	log file name

**Value**

message

**Author(s)**

Hanming Tu

**Examples**

```
NULL;
```

extract\_fns

*Extract File Names from Script Metadata*

---

**Description**

extract folders and file names from a list containing script metadata.

**Usage**

```
extract_fns(lst)
```

**Arguments**

lst                    a list containing script metadata

**Value**

a data frame (subdir, filename) containing parsed file names

**Author(s)**

Hanming Tu

**Examples**

```
a <- "https://github.com/phuse-org/phuse-scripts/raw/master"
b <- "development/R/scripts"
c <- "Draw_Dist2_R.yml"
f1 <- paste(a,b,c, sep = '/')
f2 <- read_yaml(f1)
f3 <- extract_fns(f2)
```

---

gen\_simplified\_ts*Generate Simplified TS domain xpt file*

---

**Description**

This procedure creates a simplified trial summary SAS xpt file.

**Usage**

```
gen_simplified_ts(studyid, tsparmcd = "SSTDTC",
  tsval = format(Sys.time(), "%Y-%m-%d"), tsvalnf = " ",
  ofn = "ts.xpt")
```

**Arguments**

studyid	is Study Identifier(STUDYID); required.
tsparmcd	is Trial Summary Parameter Short Name(TSPARMCD); defaults to 'SSTDTC'
tsval	is Parameter Value(TSVAL); defaults to current date in format of "YYYY-MM-DD"
tsvalnf	is Parameter Null Flavor(TSVALNF); default to blank
ofn	is output file name

**Author(s)**

Hanming Tu

**Examples**

```
## Not run:
  library(phuse)
  fn <- gen_simplified_ts();

## End(Not run)
```

---

get\_inputs

*Get Inputs from Input Sources*

---

**Description**

Get inputs from interactive session (shiny webpage), command line or script metadata.

**Usage**

```
get_inputs(fn = NULL, input = NULL, cmd = NULL)
```

**Arguments**

fn	a file name or URL pointing to script metadata file
input	the input parameter from shiny webpage
cmd	the commandArgs

**Value**

a list of input values provided for the script

**Author(s)**

Hanming Tu

**Examples**

```
a <- "https://github.com/phuse-org/phuse-scripts/raw/master"  
b <- "development/R/scripts"  
c <- "Draw_Dist2_R.yml"  
f1 <- paste(a,b,c, sep = '/')  
r1 <- get_inputs(f1)
```

---

`get_yml_inputs`*Get Inputs from Script Metadata File*

---

**Description**

Get inputs from script metadata.

**Usage**

```
get_yml_inputs(fn = NULL)
```

**Arguments**

`fn` a file name or URL pointing to script metadata file

**Value**

a list of input values provided for the script

**Author(s)**

Hanming Tu

**Examples**

```
a <- "https://github.com/phuse-org/phuse-scripts/raw/master"  
b <- "development/R/scripts"  
c <- "Draw_Dist2_R.yml"  
f1 <- paste(a,b,c, sep = '/')  
r1 <- get_inputs(f1)
```

---

init_cfg	<i>Initialize configuration for phuse</i>
----------	---

---

**Description**

read script metadata file in the repository and merged it with a local script metadata file if it exists.

**Usage**

```
init_cfg(cfg)
```

**Arguments**

cfg                    a list containing script metadata information

**Value**

a list containing the merged configuration

**Author(s)**

Hanming Tu

**Examples**

```
a <- "https://github.com/phuse-org/phuse-scripts/raw/master"  
b <- "development/R/scripts"  
c <- "Draw_Dist2_R.yml"  
f1 <- paste(a,b,c, sep = '/')  
f2 <- read_yaml(f1)  
r1 <- init_cfg(f2)
```

---

is_empty	<i>Check if a variable is na or null or space</i>
----------	---

---

**Description**

check if string or list is empty (na, null or blank spaces).

**Usage**

```
is_empty(x)
```

**Arguments**

x                    a list or string

**Value**

true or false

**Author(s)**

Hanming Tu

**Examples**

```
is_empty(NULL);  
is_empty(' ');  
is_empty(NA);
```

---

merge\_lists

*Compare and merge two lists*

---

**Description**

compare two lists using the first list as a base; update the values of the first list if the second one has different values; add variables to the first if they do not exist in the first list.

**Usage**

```
merge_lists(a, b)
```

**Arguments**

a	the 1st list
b	the 2nd list

**Value**

a list containing the merged configuration

**Author(s)**

Hanming Tu

**Examples**

```
a <- "https://github.com/phuse-org/phuse-scripts/raw/master"  
b <- "development/R/scripts"  
c <- "Draw_Dist1_R.yml"  
f1 <- paste(a,b,c, sep = '/')  
dr <- resolve(system.file("examples", package = "phuse"), "02_display")  
f2 <- paste(dr, "www", "Draw_Dist_R.yml", sep = '/')  
r1 <- read_yaml(f1)  
r2 <- read_yaml(f2)  
r3 <- merge_lists(r1, r2)
```



---

read_yaml	<i>Read YAML file into a list</i>
-----------	-----------------------------------

---

**Description**

read script metadata file in the repository or in a local folder.

**Usage**

```
read_yaml(fn)
```

**Arguments**

fn                    a URL or file name containing script metadata

**Value**

a list containing the parsed metadata tags

**Author(s)**

Hanming Tu

**Examples**

```
a <- "https://github.com/phuse-org/phuse-scripts/raw/master"
b <- "development/R/scripts"
c <- "Draw_Dist2_R.yml"
f1 <- paste(a,b,c, sep = '/')
r1 <- get_inputs(f1)
```

---

resolve	<i>Resolve absolute path</i>
---------	------------------------------

---

**Description**

Resolve absolute directory

**Usage**

```
resolve(dir, relpath)
```

**Arguments**

dir                    directory  
relpath                relative path

**Author(s)**

Hanming Tu

**Examples**

```
resolve("/Users/htu/myRepo", "scripts")
# get "/Users/htu/myRepo/scripts"
```

run\_example

*Run example***Description**

run examples stored in the example folder.

**Usage**

```
run_example(example = NA, pkg = "phuse", port = NULL,
  launch.browser = getOption("shiny.launch.browser", interactive()),
  host = getOption("shiny.host", "127.0.0.1"), display.mode = c("auto",
  "normal", "showcase"))
```

**Arguments**

example	Example name
pkg	package name
port	Port number
launch.browser	define the browser- shiny.launch.browser
host	define the host or ip address
display.mode	modes are auto, normal or showcase

**Author(s)**

Hanming Tu

**Examples**

```
## Not run:
library(phuse)
run_example("02_display")

## End(Not run)
```

---

 search\_api

*Search GitHub and build a script index data frame*


---

**Description**

Use GitHub search API to search for YAML files in phuse-scripts repository and output the list.

**Usage**

```
search_api(file_ext = "yaml",
  gh_api = "https://api.github.com/search/code",
  rp_name = "phuse-org/phuse-scripts",
  rep_url = "https://github.com/phuse-org/phuse-scripts",
  rep_dir = "tree/master",
  rep_base = "https://raw.githubusercontent.com/phuse-org/phuse-scripts/master",
  loc_base = "C:/myCodes/phuse-org/phuse-scripts", filename = NULL,
  search_for = NULL, size = NULL, path = NULL)
```

**Arguments**

file_ext	file extension; default to 'yaml'
gh_api	GitHub API URL; default to 'https://api.github.com/search/code'
rp_name	repository name; default to 'phuse-org/phuse-scripts'
rep_url	a URL for a remote repository and default to 'https://github.com/phuse-org/phuse-scripts'
rep_dir	rep dir for file name; default to 'tree/master'
rep_base	a URL for repository base folder; default to "https://github.com/phuse-org/phuse-scripts/raw/master"
loc_base	a URL for repository base folder; default to "C:/myCodes/phuse-org/phuse-scripts"
filename	file names to be searched; default to null
search_for	text or key word to be searched in the files; default to null.
size	file size; default to null
path	file path; default to null

**Value**

a list of YAML files

**Author(s)**

Hanming Tu

**Examples**

```
## Not run:
r1 <- search_api('yml')

## End(Not run)
```

---

search\_github

*Search GitHub and build a script index data frame*


---

**Description**

Use GitHub search API to search for YAML files in phuse-scripts repository and output the list.

**Usage**

```
search_github(filename = "*.yaml",
  rep_url = "https://github.com/phuse-org/phuse-scripts",
  rep_dir = "tree/master",
  rep_base = "https://raw.githubusercontent.com/phuse-org/phuse-scripts/master",
  out_type = "fn", work_dir = NULL, output_fn = NULL,
  days_to_update = 7, fn_only = FALSE)
```

**Arguments**

filename	file names to be searched; default to *.yaml.
rep_url	a URL for a remote repository and default to 'https://github.com/phuse-org/phuse-scripts'
rep_dir	rep dir for file name; default to 'tree/master'
rep_base	a URL for repository base folder; default to "https://github.com/phuse-org/phuse-scripts/raw/master"
out_type	output type; default to 'fn' - just file names.
work_dir	a local directory to host the files containing a list of YAML files; default to tempdir()/myRepo
output_fn	a CSV file name for outputting a list of YAML files; default to "repo_name_yaml.csv"
days_to_update	number of days before the output_fn is updated; default to 7 days. Set it to a negative number make it to update immediately.
fn_only	return file name only; default to FALSE

**Value**

a list of YAML files

**Author(s)**

Hanming Tu

**Examples**

```
## Not run:
  r1 <- search_github('*.*.yml')

## End(Not run)
```

---

start_app	<i>Start Phuse Web Application</i>
-----------	------------------------------------

---

**Description**

start phuse web application framework. This includes all the functions in start\_phuse plus starting standalone application by name.

**Usage**

```
start_app(app_name = NULL, n = 2, pkg = "phuse", pt = NULL,
  lb = getOption("shiny.launch.browser", interactive()),
  ht = getOption("shiny.host", "127.0.0.1"), dm = "normal",
  msg_lvl = NULL, loc = "local")
```

**Arguments**

app_name	app or script name
n	Example number
pkg	package name
pt	Port number
lb	define the browser- shiny.launch.browser
ht	define the host or ip address
dm	display modes are auto, normal or showcase
msg_lvl	message level
loc	location of the script: local github; default to 'local'

**Author(s)**

Hanming Tu

**Examples**

```
## Not run:
  library(phuse)
  start_app() # default to "02_display"
  start_app(1) # start "01_html"

## End(Not run)
```

---

start_phuse	<i>Start Phuse Web Application</i>
-------------	------------------------------------

---

### Description

start phuse web application framework.

### Usage

```
start_phuse(n = 2, pkg = "phuse", pt = NULL,  
            lb = getOption("shiny.launch.browser", interactive()),  
            ht = getOption("shiny.host", "127.0.0.1"), dm = "normal",  
            msg_lvl = NULL)
```

### Arguments

n	Example number
pkg	package name
pt	Port number
lb	define the browser- shiny.launch.browser
ht	define the host or ip address
dm	display modes are auto, normal or showcase
msg_lvl	message level

### Author(s)

Hanming Tu

### Examples

```
## Not run:  
library(phuse)  
start_phusee() # default to "02_display"  
start_phuse(1) # start "01_html"  
  
## End(Not run)
```

---

url.exists	<i>Check URL based on http package</i>
------------	--

---

**Description**

Check if URL exists.

**Usage**

```
url.exists(url = "https://github.com/phuse-org/phuse-scripts.git",  
          show = FALSE)
```

**Arguments**

url	a URL for a remote repository and default to 'https://github.com/phuse-org/phuse-scripts.git'
show	boolean variable; default to FALSE

**Value**

TRUE or FALSE

**Author(s)**

Hanming Tu

**Examples**

```
url.exists('https://github.com/phuse-org/phuse-scripts.git')
```

# Index

[build\\_inputs](#), 2  
[build\\_script\\_df](#), 3

[chk\\_workdir](#), 4  
[clone\\_github](#), 5  
[create\\_dir](#), 6  
[crt\\_workdir](#), 6  
[cvt\\_class2df](#), 7  
[cvt\\_list2df](#), 8

[download\\_fns](#), 8  
[download\\_script](#), 9  
[download\\_script\\_files](#), 10

[echo\\_msg](#), 11  
[extract\\_fns](#), 12

[gen\\_simplified\\_ts](#), 12  
[get\\_inputs](#), 13  
[get\\_yaml\\_inputs](#), 14

[init\\_cfg](#), 15  
[is\\_empty](#), 15

[merge\\_lists](#), 16

[read\\_yaml](#), 17  
[resolve](#), 17  
[run\\_example](#), 18

[search\\_api](#), 19  
[search\\_github](#), 20  
[start\\_app](#), 21  
[start\\_phuse](#), 22

[url.exists](#), 23