Package: subfoldr2 (via r-universe)

February 18, 2025

Title Save and Load R Objects

Version 1.0.0

Description Facilitates saving and loading R objects, data frames, tables, plots, text blocks and numbers to subfolders.

License MIT + file LICENSE

Imports chk, data.table, dplyr, DBI, dttr2, fs, ggplot2, graphics, grDevices, hms, lifecycle, png, readr, readwritesqlite, rlang, RSQLite, snakecase, stats, tibble, tidyplus, tools, usethis, utils, writexl, yesno

Suggests blob, config, covr, daff, dbflobr, flobr, glue, readxl, RPostgres, sf, testthat (>= 3.0.0), units, waldo, withr

RdMacros lifecycle

Remotes edwindj/daff, poissonconsulting/readwritesqlite

Config/Needs/website poissonconsulting/poissontemplate

Config/testthat/edition 3

Encoding UTF-8

Roxygen list(markdown = TRUE)

RoxygenNote 7.3.2.9000

Config/pak/sysreqs libgdal-dev gdal-bin libgeos-dev git make libgit2-dev libicu-dev libpng-dev libssl-dev libproj-dev libsqlite3-dev libudunits2-dev libx11-dev zlib1g-dev

Repository https://poissonconsulting.r-universe.dev

RemoteUrl https://github.com/poissonconsulting/subfoldr2

RemoteRef HEAD

RemoteSha bee9936529d75192543f6401ac73d5267bfbf14f

Contents

`sbf_add_blob_column_to_db` *Add blob column*

Description

Add named empty blob column to database

Usage

```
sbf_add_blob_column_to_db(
  column_name,
  table_name,
  db_name = sbf_get_db_name(),
  sub = sbf_get_sub(),
  main = sbf_get_main()
)
```

Arguments

`column_name`	A string of the name of the BLOB column.
`table_name`	A string of the name of the existing table.
`db_name`	A string of the database name.
`sub`	A string specifying the path to the sub folder (by default the current sub folder).
`main`	A string specifying the path to the main folder (by default the current main folder)

Value

Invisible TRUE.

See Also

Other flob: `sbf_save_flobs_from_db()`, `sbf_upload_flobs_to_db()`

Other database functions: `sbf_close_db()`, `sbf_copy_db()`, `sbf_create_db()`, `sbf_execute_db()`, `sbf_open_db()`, `sbf_query_db()`, `sbf_upload_flobs_to_db()`

sbf_add_sub *Add Sub Folder*

Description

Add to existing sub folder.

Usage

```
sbf_add_sub(..., rm = FALSE, ask = getOption("sbf.ask", TRUE))
```

Arguments

...	One or more character vectors which are combined together.
rm	A flag specifying whether to remove the folder and all its contents if it already exists.
ask	A flag specifying whether to ask before removing the existing folder.

Value

An invisible string specifying the new sub folder.

See Also

Other directory functions: sbf_get_archive(), sbf_get_db_name(), sbf_get_main(), sbf_get_sub(), sbf_get_workbook_name(), sbf_reset(), sbf_reset_db_name(), sbf_reset_main(), sbf_reset_sub(), sbf_set_db_name(), sbf_set_main(), sbf_set_sub(), sbf_up_sub()

Examples

```
sbf_set_sub("nameofsub")
sbf_add_sub("anothername")
sbf_get_sub()
sbf_reset_sub()
```

sbf_archive_main *Archive Main Folder*

Description

Archives main folder by copy to a director of the same name with the current date and time appended.

Usage

```
sbf_archive_main(
  main = sbf_get_main(),
  ask = getOption("sbf.ask", TRUE),
  tz = dtt_default_tz()
)
```

Arguments

main	A string specifying the path to the main folder (by default the current main folder)
ask	A flag specifying whether to ask before removing the existing folder.
tz	A string specifying the time zone for the current date and time.

Value

An invisible string of the path to the archived folder.

See Also

Other archive: sbf_get_archive(), sbf_unarchive_main()

Other housekeeping functions: sbf_rm_flobs(), sbf_rm_main(), sbf_unarchive_main()

---

sbf_backup_pg *Save PostgreSQL backup*

Description

[Deprecated]

sbf_backup_pg() was moved to subfoldr2ext::sbfx_backup_pg().

Save a copy of your database in a plain text format. This saves all the SQL code to recreate the structure and data.

Usage

```
sbf_backup_pg(
  db_dump_name = sbf_get_db_name(),
  sub = sbf_get_sub(),
  main = sbf_get_main(),
  config_path = getOption("psql.config_path", NULL),
  config_value = getOption("psql.config_value", "default")
)
```

Arguments

db_dump_name	A string of the name for the database backup file
sub	A string specifying the path to the sub folder (by default the current sub folder).
main	A string specifying the path to the main folder (by default the current main folder)
config_path	A string of a file path to the yaml configuration file. The default value grabs the file path from the psql.config_path option and uses NULL if no value supplied.
config_value	A string of the name of value. The default value grabs the value from the psql.config_value option and uses "default" if no value is supplied.

Details

Wrapper on psql::psql_backup()

Value

TRUE (or errors)

See Also

Other postgresql functions: sbf_close_pg(), sbf_create_pg(), sbf_execute_pg(), sbf_get_config_file(), sbf_get_config_value(), sbf_get_schema(), sbf_list_tables_pg(), sbf_load_data_from_pg(), sbf_load_datas_from_pg(), sbf_open_pg(), sbf_reset_config_file(), sbf_reset_config_value(), sbf_reset_schema(), sbf_save_data_to_pg(), sbf_set_config_file(), sbf_set_config_value(), sbf_set_schema()

Examples

```
## Not run:
sbf_backup_pg()

sbf_backup_pg("database-22")

## End(Not run)
```

---

sbf_basename_sans_ext *Extension-less Base File Names*

Description

Just a wrapper on basename() and tools::file_path_sans_ext().

Usage

```
sbf_basename_sans_ext(x)
```

Arguments

x	A character vector of file paths.

Value

A character vector of extension-less base file names.

See Also

Other save functions: sbf_save_aws_files(), sbf_save_block(), sbf_save_data(), sbf_save_data_to_db(), sbf_save_data_to_pg(), sbf_save_datas(), sbf_save_datas_to_db(), sbf_save_db_metatable_descriptions(), sbf_save_db_to_workbook(), sbf_save_excel(), sbf_save_excels(), sbf_save_gpkg(), sbf_save_gpkgs(), sbf_save_number(), sbf_save_numbers(), sbf_save_object(), sbf_save_objects(), sbf_save_plot(), sbf_save_png(), sbf_save_spatial(), sbf_save_spatials(), sbf_save_string(), sbf_save_strings(), sbf_save_table(), sbf_save_window(), sbf_save_workbook()

Examples

```
sbf_basename_sans_ext("path/file.ext")
```

---

sbf_block_exists *Code Block Exists*

Description

this function is now deprecated as of version 0.0.0.9045

Usage

```
sbf_block_exists(x_name, sub = sbf_get_sub(), main = sbf_get_main())
```

Arguments

x_name	A string of the name.
sub	A string specifying the path to the sub folder (by default the current sub folder).
main	A string specifying the path to the main folder (by default the current main folder)

Value

A flag specifying whether the block exists.

See Also

Other exists functions: sbf_data_exists(), sbf_number_exists(), sbf_object_exists(), sbf_plot_exists(), sbf_string_exists(), sbf_table_exists()

sbf_close_db *Close Database Connection*

Description

Closes the database connection.

Usage

```
sbf_close_db(conn)
```

Arguments

conn	A DBIConnection object, as returned by dbConnect().

Details

The function is just a wrapper on [dbDisconnect][DBI::dbDisconnect](conn).

See Also

Other database functions: sbf_add_blob_column_to_db(), sbf_copy_db(), sbf_create_db(), sbf_execute_db(), sbf_open_db(), sbf_query_db(), sbf_upload_flobs_to_db()

sbf_close_pdf *Close PDF Device*

Description

Closes the current graphics device.

Usage

```
sbf_close_pdf()
```

Details

The function is just a wrapper on grDevices::dev.off().

See Also

Other graphic functions: sbf_close_window(), sbf_close_windows(), sbf_open_pdf(), sbf_open_window()

sbf_close_pg *Close PostgreSQL Connection*

Description

[Deprecated]

`sbf_close_pg()` was moved to `subfoldr2ext::sbfx_close_pg()`.

Close the PostgreSQL connection when you are done using a database.

Usage

```
sbf_close_pg(conn)
```

Arguments

conn	A DBIConnection object, as returned by `dbConnect()`.

Details

Wrapper on `DBI::dbDisconnect()`. It is important to remember to close connections or your database performance can decrease over time.

Value

TRUE (or errors).

See Also

Other postgresql functions: `sbf_backup_pg()`, `sbf_create_pg()`, `sbf_execute_pg()`, `sbf_get_config_file()`, `sbf_get_config_value()`, `sbf_get_schema()`, `sbf_list_tables_pg()`, `sbf_load_data_from_pg()`, `sbf_load_datas_from_pg()`, `sbf_open_pg()`, `sbf_reset_config_file()`, `sbf_reset_config_value()`, `sbf_reset_schema()`, `sbf_save_data_to_pg()`, `sbf_set_config_file()`, `sbf_set_config_value()`, `sbf_set_schema()`

Examples

```
## Not run:
conn <- sbf_open_pg()
sbf_close_pg(conn)

## End(Not run)
```

sbf_close_window *Close Window*

Description

Closes the current graphics device.

Usage

```
sbf_close_window()
```

Details

The function is just a wrapper on grDevices::dev.off().

See Also

Other graphic functions: sbf_close_pdf(), sbf_close_windows(), sbf_open_pdf(), sbf_open_window()

sbf_close_windows *Close Windows*

Description

Closes all current graphics device.

Usage

```
sbf_close_windows()
```

Details

The function is just a wrapper on grDevices::graphics.off().

See Also

Other graphic functions: sbf_close_pdf(), sbf_close_window(), sbf_open_pdf(), sbf_open_window()

sbf_compare_data *Compare Data*

Description

Compares data using waldo::compare.

Usage

```
sbf_compare_data(
  x,
  x_name = substitute(x),
  sub = sbf_get_sub(),
  main = sbf_get_main(),
  tolerance = sqrt(.Machine$double.eps),
  ignore_attr = TRUE
)
```

Arguments

x	The object to save.
x_name	A string of the name.
sub	A string specifying the path to the sub folder (by default the current sub folder).
main	A string specifying the path to the main folder (by default the current main folder)
tolerance	numeric ≥ 0. Differences smaller than `tolerance` are not reported. The default value is close to `1.5e-8`.
ignore_attr	Ignore differences in specified attributes? Supply a character vector to ignore differences in named attributes. By default the `"waldo_opts"` attribute is listed in `ignore_attr` so that changes to it are not reported; if you customize `ignore_attr`, you will probably want to do this yourself. For backward compatibility with `all.equal()`, you can also use `TRUE`, to all ignore differences in all attributes. This is not generally recommended as it is a blunt tool that will ignore many important functional differences.

Value

A character vector with class "waldo_compare".

See Also

Other compare functions: `sbf_compare_data_archive()`, `sbf_diff_data()`, `sbf_diff_data_archive()`, `sbf_diff_table()`, `sbf_is_equal_data()`, `sbf_is_equal_data_archive()`

`sbf_compare_data_archive` *Compare Data Archive*

Description

Compares existing data to archived data using using waldo::compare.

Usage

```
sbf_compare_data_archive(
  x_name = ".*",
  sub = sbf_get_sub(),
  main = sbf_get_main(),
  archive = 1L,
  recursive = FALSE,
  include_root = TRUE,
  tolerance = sqrt(.Machine$double.eps),
  ignore_attr = TRUE
)
```

Arguments

`x_name`	A string of the name.
`sub`	A string specifying the path to the sub folder (by default the current sub folder).
`main`	A string specifying the path to the main folder (by default the current main folder)
`archive`	A positive whole number specifying the folder archived folder where 1L (the default) indicates the most recently archived folder or a character string of the path to the archived folder.
`recursive`	A flag specifying whether to recurse into subfolders.
`include_root`	A flag indicating whether to include objects in the top sub folder.
`tolerance`	numeric ≥ 0. Differences smaller than `tolerance` are not reported. The default value is close to `1.5e-8`.
`ignore_attr`	Ignore differences in specified attributes? Supply a character vector to ignore differences in named attributes. By default the `"waldo_opts"` attribute is listed in `ignore_attr` so that changes to it are not reported; if you customize `ignore_attr`, you will probably want to do this yourself. For backward compatibility with `all.equal()`, you can also use `TRUE`, to all ignore differences in all attributes. This is not generally recommended as it is a blunt tool that will ignore many important functional differences.

Value

A named list of character vectors.

See Also

Other compare functions: sbf_compare_data(), sbf_diff_data(), sbf_diff_data_archive(), sbf_diff_table(), sbf_is_equal_data(), sbf_is_equal_data_archive()

---

sbf_copy_db *Copy SQLite Database*

Description

Copys an existing SQLite database to the subfolder.

Usage

```
sbf_copy_db(
  path,
  db_name = sbf_get_db_name(),
  sub = sbf_get_sub(),
  main = sbf_get_main(),
  exists = FALSE,
  ask = getOption("sbf.ask", TRUE)
)
```

Arguments

path	A string of the path to the database to copy (with the extension).
db_name	A string of the name for the new database (without the extension).
sub	A string specifying the path to the sub folder (by default the current sub folder).
main	A string specifying the path to the main folder (by default the current main folder)
exists	A logical scalar specifying whether the new database must already exist.
ask	A flag specifying whether to ask before deleting an existing database (if exists = FALSE).

Value

A flag indicating whether successfully copied.

See Also

Other database functions: sbf_add_blob_column_to_db(), sbf_close_db(), sbf_create_db(), sbf_execute_db(), sbf_open_db(), sbf_query_db(), sbf_upload_flobs_to_db()

sbf_create_db *Create SQLite Database*

Description

Create SQLite Database

Usage

```
sbf_create_db(
  db_name = sbf_get_db_name(),
  sub = sbf_get_sub(),
  main = sbf_get_main(),
  ask = getOption("sbf.ask", TRUE)
)
```

Arguments

db_name	A string of the database name.
sub	A string specifying the path to the sub folder (by default the current sub folder).
main	A string specifying the path to the main folder (by default the current main folder)
ask	A flag specifying whether to ask before deleting an existing database.

See Also

Other database functions: sbf_add_blob_column_to_db(), sbf_close_db(), sbf_copy_db(), sbf_execute_db(), sbf_open_db(), sbf_query_db(), sbf_upload_flobs_to_db()

sbf_create_pg *Create PostgreSQL database*

Description

[Deprecated]

sbf_create_pg() was moved to subfoldr2ext::sbfx_create_pg().

Create a new PostgreSQL database.

Usage

```
sbf_create_pg(
  dbname,
  config_path = getOption("psql.config_path", NULL),
  config_value = getOption("psql.config_value", "default")
)
```

Arguments

dbname	A string of the name of the new database to create.
config_path	A string of a file path to the yaml configuration file. The default value grabs the file path from the psql.config_path option and uses NULL if no value supplied.
config_value	A string of the name of value. The default value grabs the value from the psql.config_value option and uses "default" if no value is supplied.

Details

Wrapper on psql::psql_create_db()

Value

TRUE (or errors).

See Also

Other postgresql functions: sbf_backup_pg(), sbf_close_pg(), sbf_execute_pg(), sbf_get_config_file(), sbf_get_config_value(), sbf_get_schema(), sbf_list_tables_pg(), sbf_load_data_from_pg(), sbf_load_datas_from_pg(), sbf_open_pg(), sbf_reset_config_file(), sbf_reset_config_value(), sbf_reset_schema(), sbf_save_data_to_pg(), sbf_set_config_file(), sbf_set_config_value(), sbf_set_schema()

Examples

```
## Not run:
sbf_create_pg("new_database")
sbf_create_pg("new_database", config_path = "keys/config.yml")

## End(Not run)
```

---

sbf_data_exists *Data Exists*

Description

this function is now deprecated as of version 0.0.0.9045

Usage

```
sbf_data_exists(x_name, sub = sbf_get_sub(), main = sbf_get_main())
```

Arguments

x_name	A string of the name.
sub	A string specifying the path to the sub folder (by default the current sub folder).
main	A string specifying the path to the main folder (by default the current main folder)

Value

A flag specifying whether the data exists.

See Also

Other exists functions: sbf_block_exists(), sbf_number_exists(), sbf_object_exists(), sbf_plot_exists(), sbf_string_exists(), sbf_table_exists()

`sbf_diff_data` *Diff Data*

Description

Find differences with existing data. If doesn't exist, x is compared to itself.

Usage

```
sbf_diff_data(
  x,
  x_name = substitute(x),
  sub = sbf_get_sub(),
  main = sbf_get_main()
)
```

Arguments

x	The object to save.
x_name	A string of the name.
sub	A string specifying the path to the sub folder (by default the current sub folder).
main	A string specifying the path to the main folder (by default the current main folder)

Value

A daff difference object.

See Also

Other compare functions: sbf_compare_data(), sbf_compare_data_archive(), sbf_diff_data_archive(), sbf_diff_table(), sbf_is_equal_data(), sbf_is_equal_data_archive()

`sbf_diff_data_archive` *Diff Data Archive*

Description

Find differences with existing data and archived data. If doesn't exist (exists = NA) x is compared to itself.

Usage

```
sbf_diff_data_archive(
  x_name = ".*",
  sub = sbf_get_sub(),
  main = sbf_get_main(),
  archive = 1L,
  recursive = FALSE,
  include_root = TRUE,
  exists = NA
)
```

Arguments

`x_name`	A string of the name.
`sub`	A string specifying the path to the sub folder (by default the current sub folder).
`main`	A string specifying the path to the main folder (by default the current main folder)
`archive`	A positive whole number specifying the folder archived folder where 1L (the default) indicates the most recently archived folder or a character string of the path to the archived folder.
`recursive`	A flag specifying whether to recurse into subfolders.
`include_root`	A flag indicating whether to include objects in the top sub folder.
`exists`	A logical scalar specifying whether the file should exist.

Value

A named list of character vectors.

See Also

Other compare functions: `sbf_compare_data()`, `sbf_compare_data_archive()`, `sbf_diff_data()`, `sbf_diff_table()`, `sbf_is_equal_data()`, `sbf_is_equal_data_archive()`

sbf_diff_table *Diff Table*

Description

Find differences with existing table data. If doesn't exist (exists = NA) x is compared to itself.

Usage

```
sbf_diff_table(
  x,
  x_name = substitute(x),
  sub = sbf_get_sub(),
  main = sbf_get_main(),
  exists = NA
)
```

Arguments

x	The object to save.
x_name	A string of the name.
sub	A string specifying the path to the sub folder (by default the current sub folder).
main	A string specifying the path to the main folder (by default the current main folder)
exists	A logical scalar specifying whether the file should exist.

Value

A daff difference object.

See Also

Other compare functions: sbf_compare_data(), sbf_compare_data_archive(), sbf_diff_data(), sbf_diff_data_archive(), sbf_is_equal_data(), sbf_is_equal_data_archive()

sbf_execute_db *Execute SQL Statement on Existing Database*

Description

Really just a wrapper on DBI::dbExecute().

Usage

```
sbf_execute_db(
  sql,
  db_name = sbf_get_db_name(),
  sub = sbf_get_sub(),
  main = sbf_get_main()
)
```

Arguments

sql	A string of the SQL statement to execute.
db_name	A string of the database name.
sub	A string specifying the path to the sub folder (by default the current sub folder).
main	A string specifying the path to the main folder (by default the current main folder)

Value

A scalar numeric of the number of rows affected by the statement.

See Also

Other database functions: sbf_add_blob_column_to_db(), sbf_close_db(), sbf_copy_db(), sbf_create_db(), sbf_open_db(), sbf_query_db(), sbf_upload_flobs_to_db()

---

sbf_execute_pg *Execute SQL statement for PostgreSQL database*

Description

[Deprecated]

sbf_execute_pg() was moved to subfoldr2ext::sbfx_execute_pg().

Execute PostgreSQL statements.

Usage

```
sbf_execute_pg(
  sql,
  config_path = getOption("psql.config_path", NULL),
  config_value = getOption("psql.config_value", "default")
)
```

Arguments

sql	A string of the SQL statement to execute.
config_path	A string of a file path to the yaml configuration file. The default value grabs the file path from the psql.config_path option and uses NULL if no value supplied.
config_value	A string of the name of value. The default value grabs the value from the psql.config_value option and uses "default" if no value is supplied.

Details

Wrapper on psql::psql_execute_db()

Value

A scalar numeric of the number of rows affected by the statement.

See Also

Other postgresql functions: sbf_backup_pg(), sbf_close_pg(), sbf_create_pg(), sbf_get_config_file(), sbf_get_config_value(), sbf_get_schema(), sbf_list_tables_pg(), sbf_load_data_from_pg(), sbf_load_datas_from_pg(), sbf_open_pg(), sbf_reset_config_file(), sbf_reset_config_value(), sbf_reset_schema(), sbf_save_data_to_pg(), sbf_set_config_file(), sbf_set_config_value(), sbf_set_schema()

Examples

```
## Not run:
sbf_execute_pg(
  "CREATE SCHEMA boat_count"
)
sbf_execute_pg(
  "CREATE TABLE boat_count.input (
  file_name TEXT NOT NULL,
  comment TEXT)"
)

## End(Not run)
```

---

sbf_get_archive *Get Archive Directory*

Description

Get Archive Directory

Usage

```
sbf_get_archive(main = sbf_get_main(), archive = 1L)
```

Arguments

main	A string specifying the path to the main folder (by default the current main folder)
archive	A positive whole number specifying the folder archived folder where 1L (the default) indicates the most recently archived folder or a character string of the path to the archived folder.

Value

A string of the path to the archived directory.

See Also

Other archive: sbf_archive_main(), sbf_unarchive_main()

Other directory functions: sbf_add_sub(), sbf_get_db_name(), sbf_get_main(), sbf_get_sub(), sbf_get_workbook_name(), sbf_reset(), sbf_reset_db_name(), sbf_reset_main(), sbf_reset_sub(), sbf_set_db_name(), sbf_set_main(), sbf_set_sub(), sbf_up_sub()

---

sbf_get_config_file *Get the Config File Path*

Description

[Deprecated]

sbf_get_config_file() was moved to subfoldr2ext::sbfx_get_config_file().

Get the option set for psql.config_path

Usage

```
sbf_get_config_file()
```

Value

A string of the config file path.

See Also

Other postgresql functions: sbf_backup_pg(), sbf_close_pg(), sbf_create_pg(), sbf_execute_pg(), sbf_get_config_value(), sbf_get_schema(), sbf_list_tables_pg(), sbf_load_data_from_pg(), sbf_load_datas_from_pg(), sbf_open_pg(), sbf_reset_config_file(), sbf_reset_config_value(), sbf_reset_schema(), sbf_save_data_to_pg(), sbf_set_config_file(), sbf_set_config_value(), sbf_set_schema()

Examples

```
## Not run:
sbf_get_config_file()

## End(Not run)
```

---

sbf_get_config_value *Get the Config File Value*

Description

[Deprecated]

`sbf_get_config_value()` was moved to `subfoldr2ext::sbfx_get_config_value()`.

Get the value set for the `psql.config_value` options parameter.

Usage

```
sbf_get_config_value()
```

Value

A string of the config file value

See Also

Other postgresql functions: `sbf_backup_pg()`, `sbf_close_pg()`, `sbf_create_pg()`, `sbf_execute_pg()`, `sbf_get_config_file()`, `sbf_get_schema()`, `sbf_list_tables_pg()`, `sbf_load_data_from_pg()`, `sbf_load_datas_from_pg()`, `sbf_open_pg()`, `sbf_reset_config_file()`, `sbf_reset_config_value()`, `sbf_reset_schema()`, `sbf_save_data_to_pg()`, `sbf_set_config_file()`, `sbf_set_config_value()`, `sbf_set_schema()`

Examples

```
## Not run:
sbf_get_config_value()

## End(Not run)
```

sbf_get_db_name *Get Database Name*

Description

Gets database name (without the extension or path). By default (ie if not set) 'database'.

Usage

```
sbf_get_db_name()
```

Value

A string specifying the current database name (without the extension or path).

See Also

Other db_name: `sbf_set_db_name()`

Other directory functions: `sbf_add_sub()`, `sbf_get_archive()`, `sbf_get_main()`, `sbf_get_sub()`, `sbf_get_workbook_name()`, `sbf_reset()`, `sbf_reset_db_name()`, `sbf_reset_main()`, `sbf_reset_sub()`, `sbf_set_db_name()`, `sbf_set_main()`, `sbf_set_sub()`, `sbf_up_sub()`

Examples

```
sbf_get_db_name()
```

sbf_get_main *Get Main*

Description

Get Main

Usage

```
sbf_get_main()
```

Value

A string specifying the main directory.

See Also

Other directory functions: `sbf_add_sub()`, `sbf_get_archive()`, `sbf_get_db_name()`, `sbf_get_sub()`, `sbf_get_workbook_name()`, `sbf_reset()`, `sbf_reset_db_name()`, `sbf_reset_main()`, `sbf_reset_sub()`, `sbf_set_db_name()`, `sbf_set_main()`, `sbf_set_sub()`, `sbf_up_sub()`

Examples

```
sbf_get_main()
```

`sbf_get_schema` *Get Schema Name*

Description

[Deprecated]

`sbf_get_schema()` was moved to `subfoldr2ext::sbfx_get_schema()`.

Usage

```
sbf_get_schema()
```

Value

A string of the schema name.

See Also

Other postgresql functions: `sbf_backup_pg()`, `sbf_close_pg()`, `sbf_create_pg()`, `sbf_execute_pg()`, `sbf_get_config_file()`, `sbf_get_config_value()`, `sbf_list_tables_pg()`, `sbf_load_data_from_pg()`, `sbf_load_datas_from_pg()`, `sbf_open_pg()`, `sbf_reset_config_file()`, `sbf_reset_config_value()`, `sbf_reset_schema()`, `sbf_save_data_to_pg()`, `sbf_set_config_file()`, `sbf_set_config_value()`, `sbf_set_schema()`

Examples

```
## Not run:
sbf_get_schema()

## End(Not run)
```

`sbf_get_sub` *Get Sub Folder*

Description

Get Sub Folder

Usage

```
sbf_get_sub()
```

Value

A string specifying the current sub folder.

See Also

Other directory functions: sbf_add_sub(), sbf_get_archive(), sbf_get_db_name(), sbf_get_main(), sbf_get_workbook_name(), sbf_reset(), sbf_reset_db_name(), sbf_reset_main(), sbf_reset_sub(), sbf_set_db_name(), sbf_set_main(), sbf_set_sub(), sbf_up_sub()

Examples

```
sbf_set_sub("nameofsub")
sbf_get_sub()
sbf_reset_sub()
```

---

sbf_get_workbook_name *Get Workbook Name*

Description

Gets the basename of the current working directory

Usage

```
sbf_get_workbook_name()
```

Value

A string specifying the name of the current working directory

See Also

Other excel: sbf_save_db_to_workbook(), sbf_save_excel(), sbf_save_excels(), sbf_save_workbook()

Other directory functions: sbf_add_sub(), sbf_get_archive(), sbf_get_db_name(), sbf_get_main(), sbf_get_sub(), sbf_reset(), sbf_reset_db_name(), sbf_reset_main(), sbf_reset_sub(), sbf_set_db_name(), sbf_set_main(), sbf_set_sub(), sbf_up_sub()

Examples

```
sbf_get_workbook_name()
```

`sbf_is_equal_data` *Is Equal Data*

Description

Test if data is equal using `all.equal()`. If doesn't exist returns FALSE, unless exists = FALSE in which case returns TRUE or exists = NA in which case returns NA.

Usage

```
sbf_is_equal_data(
  x,
  x_name = substitute(x),
  sub = sbf_get_sub(),
  main = sbf_get_main(),
  exists = TRUE,
  tolerance = sqrt(.Machine$double.eps),
  check.attributes = TRUE
)
```

Arguments

`x`	The object to save.
`x_name`	A string of the name.
`sub`	A string specifying the path to the sub folder (by default the current sub folder).
`main`	A string specifying the path to the main folder (by default the current main folder)
`exists`	A logical scalar specifying whether the file should exist.
`tolerance`	numeric ≥ 0. Differences smaller than `tolerance` are not reported. The default value is close to `1.5e-8`.
`check.attributes`	logical indicating if the `attributes` of `target` and `current` (other than the names) should be compared.

Value

A named flag.

See Also

Other compare functions: `sbf_compare_data()`, `sbf_compare_data_archive()`, `sbf_diff_data()`, `sbf_diff_data_archive()`, `sbf_diff_table()`, `sbf_is_equal_data_archive()`

`sbf_is_equal_data_archive` *Is Equal Data Archive*

Description

Test if existing data are equal to archived data using `all.equal()`. If doesn't exist in both returns FALSE, unless exists = FALSE in which case returns TRUE or exists = NA in which case returns NA.

Usage

```
sbf_is_equal_data_archive(
  x_name = ".*",
  sub = sbf_get_sub(),
  main = sbf_get_main(),
  archive = 1L,
  recursive = FALSE,
  include_root = TRUE,
  exists = TRUE,
  tolerance = sqrt(.Machine$double.eps),
  check.attributes = TRUE
)
```

Arguments

`x_name`	A string of the name.
`sub`	A string specifying the path to the sub folder (by default the current sub folder).
`main`	A string specifying the path to the main folder (by default the current main folder)
`archive`	A positive whole number specifying the folder archived folder where 1L (the default) indicates the most recently archived folder or a character string of the path to the archived folder.
`recursive`	A flag specifying whether to recurse into subfolders.
`include_root`	A flag indicating whether to include objects in the top sub folder.
`exists`	A logical scalar specifying whether the file should exist.
`tolerance`	numeric ≥ 0. Differences smaller than `tolerance` are not reported. The default value is close to `1.5e-8`.
`check.attributes`	logical indicating if the `attributes` of `target` and `current` (other than the names) should be compared.

Value

A named logical vector.

See Also

Other compare functions: sbf_compare_data(), sbf_compare_data_archive(), sbf_diff_data(), sbf_diff_data_archive(), sbf_diff_table(), sbf_is_equal_data()

---

sbf_list_blocks *Gets List of Block Files as a Character Vector*

Description

Returns file paths for all block files matching regular expression x_name.

Usage

```
sbf_list_blocks(
  x_name = ".*",
  sub = sbf_get_sub(),
  main = sbf_get_main(),
  recursive = FALSE,
  include_root = TRUE,
  ext = "rds"
)
```

Arguments

x_name	A string of the name.
sub	A string specifying the path to the sub folder (by default the current sub folder).
main	A string specifying the path to the main folder (by default the current main folder)
recursive	A flag specifying whether to recurse into subfolders.
include_root	A flag indicating whether to include objects in the top sub folder.
ext	A string of the file extension.

See Also

Other list functions: sbf_list_datas(), sbf_list_dbs(), sbf_list_numbers(), sbf_list_objects(), sbf_list_plots(), sbf_list_strings(), sbf_list_tables(), sbf_list_windows()

sbf_list_datas *Gets List of Data Files as a Character Vector*

Description

Returns file paths for all data files matching regular expression x_name.

Usage

```
sbf_list_datas(
  x_name = ".*",
  sub = sbf_get_sub(),
  main = sbf_get_main(),
  recursive = FALSE,
  include_root = TRUE,
  ext = "rds"
)
```

Arguments

x_name	A string of the name.
sub	A string specifying the path to the sub folder (by default the current sub folder).
main	A string specifying the path to the main folder (by default the current main folder)
recursive	A flag specifying whether to recurse into subfolders.
include_root	A flag indicating whether to include objects in the top sub folder.
ext	A string of the file extension.

See Also

Other list functions: sbf_list_blocks(), sbf_list_dbs(), sbf_list_numbers(), sbf_list_objects(), sbf_list_plots(), sbf_list_strings(), sbf_list_tables(), sbf_list_windows()

sbf_list_dbs *Gets List of Database Files as a Character Vector*

Description

Returns file paths for all database files matching regular expression x_name.

Usage

```
sbf_list_dbs(
  x_name = ".*",
  sub = sbf_get_sub(),
  main = sbf_get_main(),
  recursive = FALSE,
  include_root = TRUE,
  ext = "sqlite"
)
```

Arguments

x_name	A string of the name.
sub	A string specifying the path to the sub folder (by default the current sub folder).
main	A string specifying the path to the main folder (by default the current main folder)
recursive	A flag specifying whether to recurse into subfolders.
include_root	A flag indicating whether to include objects in the top sub folder.
ext	A string of the file extension.

See Also

Other list functions: sbf_list_blocks(), sbf_list_datas(), sbf_list_numbers(), sbf_list_objects(), sbf_list_plots(), sbf_list_strings(), sbf_list_tables(), sbf_list_windows()

---

sbf_list_numbers *Gets List of Number Files as a Character Vector*

Description

Returns file paths for all number .rds files matching regular expression x_name.

Usage

```
sbf_list_numbers(
  x_name = ".*",
  sub = sbf_get_sub(),
  main = sbf_get_main(),
  recursive = FALSE,
  include_root = TRUE,
  ext = "rds"
)
```

Arguments

x_name	A string of the name.
sub	A string specifying the path to the sub folder (by default the current sub folder).
main	A string specifying the path to the main folder (by default the current main folder)
recursive	A flag specifying whether to recurse into subfolders.
include_root	A flag indicating whether to include objects in the top sub folder.
ext	A string of the file extension.

See Also

Other list functions: sbf_list_blocks(), sbf_list_datas(), sbf_list_dbs(), sbf_list_objects(), sbf_list_plots(), sbf_list_strings(), sbf_list_tables(), sbf_list_windows()

---

sbf_list_objects *Gets List of Object Files as a Character Vector*

Description

Returns file paths for all object files matching regular expression x_name.

Usage

```
sbf_list_objects(
  x_name = ".*",
  sub = sbf_get_sub(),
  main = sbf_get_main(),
  recursive = FALSE,
  include_root = TRUE,
  ext = "rds"
)
```

Arguments

x_name	A regular expression of the object names to match.
sub	A string specifying the path to the sub folder (by default the current sub folder).
main	A string specifying the path to the main folder (by default the current main folder)
recursive	A flag specifying whether to recurse into subfolders.
include_root	A flag indicating whether to include objects in the top sub folder.
ext	A string of the file extension.

See Also

Other list functions: sbf_list_blocks(), sbf_list_datas(), sbf_list_dbs(), sbf_list_numbers(), sbf_list_plots(), sbf_list_strings(), sbf_list_tables(), sbf_list_windows()

sbf_list_plots *Gets List of Plot Files as a Character Vector*

Description

Returns file paths for all plot files matching regular expression x_name.

Usage

```
sbf_list_plots(
  x_name = ".*",
  sub = sbf_get_sub(),
  main = sbf_get_main(),
  recursive = FALSE,
  include_root = TRUE,
  ext = "rds"
)
```

Arguments

x_name	A string of the name.
sub	A string specifying the path to the sub folder (by default the current sub folder).
main	A string specifying the path to the main folder (by default the current main folder)
recursive	A flag specifying whether to recurse into subfolders.
include_root	A flag indicating whether to include objects in the top sub folder.
ext	A string of the file extension.

See Also

Other list functions: sbf_list_blocks(), sbf_list_datas(), sbf_list_dbs(), sbf_list_numbers(), sbf_list_objects(), sbf_list_strings(), sbf_list_tables(), sbf_list_windows()

sbf_list_strings *Gets List of String Files as a Character Vector*

Description

Returns file paths for all string .rds files matching regular expression x_name.

Usage

```
sbf_list_strings(
  x_name = ".*",
  sub = sbf_get_sub(),
  main = sbf_get_main(),
  recursive = FALSE,
  include_root = TRUE,
  ext = "rds"
)
```

Arguments

x_name	A string of the name.
sub	A string specifying the path to the sub folder (by default the current sub folder).
main	A string specifying the path to the main folder (by default the current main folder)
recursive	A flag specifying whether to recurse into subfolders.
include_root	A flag indicating whether to include objects in the top sub folder.
ext	A string of the file extension.

See Also

Other list functions: sbf_list_blocks(), sbf_list_datas(), sbf_list_dbs(), sbf_list_numbers(), sbf_list_objects(), sbf_list_plots(), sbf_list_tables(), sbf_list_windows()

---

sbf_list_tables *Gets List of Data Files as a Character Vector*

Description

Returns file paths for table files matching regular expression x_name.

Usage

```
sbf_list_tables(
  x_name = ".*",
  sub = sbf_get_sub(),
  main = sbf_get_main(),
  recursive = FALSE,
  include_root = TRUE,
  ext = "rds"
)
```

Arguments

x_name	A string of the name.
sub	A string specifying the path to the sub folder (by default the current sub folder).
main	A string specifying the path to the main folder (by default the current main folder)
recursive	A flag specifying whether to recurse into subfolders.
include_root	A flag indicating whether to include objects in the top sub folder.
ext	A string of the file extension.

See Also

Other list functions: sbf_list_blocks(), sbf_list_datas(), sbf_list_dbs(), sbf_list_numbers(), sbf_list_objects(), sbf_list_plots(), sbf_list_strings(), sbf_list_windows()

---

sbf_list_tables_pg *List tables in a schema*

Description

[Deprecated]

sbf_list_tables_pg() was moved to subfoldr2ext::sbfx_list_tables_pg().

This function lists all the tables in a schema.

Usage

```
sbf_list_tables_pg(
  schema = getOption("psql.schema", "public"),
  config_path = getOption("psql.config_path", NULL),
  config_value = getOption("psql.config_value", "default")
)
```

Arguments

schema	A string of the schema name. Default value is "public".
config_path	A string of a file path to the yaml configuration file. The default value grabs the file path from the psql.config_path option and uses NULL if no value supplied.
config_value	A string of the name of value. The default value grabs the value from the psql.config_value option and uses "default" if no value is supplied.

Details

Wrapper on psql::psql_list_tables()

Value

A vector of table names

See Also

Other postgresql functions: sbf_backup_pg(), sbf_close_pg(), sbf_create_pg(), sbf_execute_pg(), sbf_get_config_file(), sbf_get_config_value(), sbf_get_schema(), sbf_load_data_from_pg(), sbf_load_datas_from_pg(), sbf_open_pg(), sbf_reset_config_file(), sbf_reset_config_value(), sbf_reset_schema(), sbf_save_data_to_pg(), sbf_set_config_file(), sbf_set_config_value(), sbf_set_schema()

Examples

```
## Not run:
sbf_list_tables_pg(
  "boat_count"
)
sbf_list_tables_pg()

## End(Not run)
```

---

sbf_list_windows *Gets List of Window Files as a Character Vector*

Description

Returns file paths for all window files matching regular expression x_name.

Usage

```
sbf_list_windows(
  x_name = ".*",
  sub = sbf_get_sub(),
  main = sbf_get_main(),
  recursive = FALSE,
  include_root = TRUE,
  ext = "png"
)
```

Arguments

x_name	A string of the name.
sub	A string specifying the path to the sub folder (by default the current sub folder).
main	A string specifying the path to the main folder (by default the current main folder)
recursive	A flag specifying whether to recurse into subfolders.
include_root	A flag indicating whether to include objects in the top sub folder.
ext	A string of the file extension.

See Also

Other list functions: sbf_list_blocks(), sbf_list_datas(), sbf_list_dbs(), sbf_list_numbers(), sbf_list_objects(), sbf_list_plots(), sbf_list_strings(), sbf_list_tables()

---

sbf_load_block *Load Code Block*

Description

Load Code Block

Usage

```
sbf_load_block(
  x_name,
  sub = sbf_get_sub(),
  main = sbf_get_main(),
  exists = TRUE
)
```

Arguments

x_name	A string of the name.
sub	A string specifying the path to the sub folder (by default the current sub folder).
main	A string specifying the path to the main folder (by default the current main folder)
exists	A logical scalar specifying whether the file should exist.

Value

A code block or NULL if doesn't exist.

See Also

Other load functions: sbf_load_blocks(), sbf_load_blocks_recursive(), sbf_load_data(), sbf_load_data_from_db(), sbf_load_data_from_pg(), sbf_load_datas(), sbf_load_datas_from_db(), sbf_load_datas_from_pg(), sbf_load_datas_recursive(), sbf_load_db_metatable(), sbf_load_number(), sbf_load_numbers(), sbf_load_numbers_recursive(), sbf_load_object(), sbf_load_objects(), sbf_load_objects_recursive(), sbf_load_plot(), sbf_load_plot_data(), sbf_load_plots_data(), sbf_load_plots_data_recursive(), sbf_load_plots_recursive(), sbf_load_spatial(), sbf_load_spatials(), sbf_load_string(), sbf_load_strings(), sbf_load_strings_recursive(), sbf_load_table(), sbf_load_tables(), sbf_load_tables_recursive(), sbf_load_windows_recursive(), sbf_subs_block_recursive(), sbf_subs_data_recursive(), sbf_subs_number_recursive(), sbf_subs_object_recursive(), sbf_subs_plot_recursive(), sbf_subs_string_recursive(), sbf_subs_table_recursive(), sbf_subs_window_recursive()

sbf_load_blocks *Load Blocks*

Description

Load Blocks

Usage

```
sbf_load_blocks(
  sub = sbf_get_sub(),
  main = sbf_get_main(),
  rename = identity,
  env = parent.frame()
)
```

Arguments

sub	A string specifying the path to the sub folder (by default the current sub folder).
main	A string specifying the path to the main folder (by default the current main folder)
rename	A single function argument which takes a character vector and returns a character vector of the same length. Used to rename objects before they are loaded into the environment.
env	The environment to the objects into

Value

A invisble character vector of the blocks' names.

See Also

Other load functions: sbf_load_block(), sbf_load_blocks_recursive(), sbf_load_data(), sbf_load_data_from_db(), sbf_load_data_from_pg(), sbf_load_datas(), sbf_load_datas_from_db(), sbf_load_datas_from_pg(), sbf_load_datas_recursive(), sbf_load_db_metatable(), sbf_load_number(), sbf_load_numbers(), sbf_load_numbers_recursive(), sbf_load_object(), sbf_load_objects(), sbf_load_objects_recursive(), sbf_load_plot(), sbf_load_plot_data(), sbf_load_plots_data(), sbf_load_plots_data_recursive(), sbf_load_plots_recursive(), sbf_load_spatial(), sbf_load_spatials(), sbf_load_string(), sbf_load_strings(), sbf_load_strings_recursive(), sbf_load_table(), sbf_load_tables(), sbf_load_tables_recursive(), sbf_load_windows_recursive(), sbf_subs_block_recursive(), sbf_subs_data_recursive(), sbf_subs_number_recursive(), sbf_subs_object_recursive(), sbf_subs_plot_recursive(), sbf_subs_string_recursive(), sbf_subs_table_recursive(), sbf_subs_window_recursive()

`sbf_load_blocks_recursive` *Load Blocks as Column in Data Frame*

Description

Recursively loads all the code blocks with names matching the regular expression x_name as the first character column (named blocks) in a data frame. Subsequent character vector columns specify the object names (named name) and sub folders (named sub1, sub2 etc).

Usage

```
sbf_load_blocks_recursive(
  x_name = ".*",
  sub = sbf_get_sub(),
  main = sbf_get_main(),
  include_root = TRUE,
  tag = ".*",
  meta = FALSE
)
```

Arguments

`x_name`	A string of the name.
`sub`	A string specifying the path to the sub folder (by default the current sub folder).
`main`	A string specifying the path to the main folder (by default the current main folder)
`include_root`	A flag indicating whether to include objects in the top sub folder.
`tag`	A string of the regular expression that the tag must match to be included.
`meta`	A flag specifying whether to include the report, caption and any other metadata as columns.

See Also

Other load functions: `sbf_load_block()`, `sbf_load_blocks()`, `sbf_load_data()`, `sbf_load_data_from_db()`, `sbf_load_data_from_pg()`, `sbf_load_datas()`, `sbf_load_datas_from_db()`, `sbf_load_datas_from_pg()`, `sbf_load_datas_recursive()`, `sbf_load_db_metatable()`, `sbf_load_number()`, `sbf_load_numbers()`, `sbf_load_numbers_recursive()`, `sbf_load_object()`, `sbf_load_objects()`, `sbf_load_objects_recursive()`, `sbf_load_plot()`, `sbf_load_plot_data()`, `sbf_load_plots_data()`, `sbf_load_plots_data_recursive()`, `sbf_load_plots_recursive()`, `sbf_load_spatial()`, `sbf_load_spatials()`, `sbf_load_string()`, `sbf_load_strings()`, `sbf_load_strings_recursive()`, `sbf_load_table()`, `sbf_load_tables()`, `sbf_load_tables_recursive()`, `sbf_load_windows_recursive()`, `sbf_subs_block_recursive()`, `sbf_subs_data_recursive()`, `sbf_subs_number_recursive()`, `sbf_subs_object_recursive()`, `sbf_subs_plot_recursive()`, `sbf_subs_string_recursive()`, `sbf_subs_table_recursive()`, `sbf_subs_window_recursive()`

sbf_load_data *Load Data*

Description

Load Data

Usage

```
sbf_load_data(
  x_name,
  sub = sbf_get_sub(),
  main = sbf_get_main(),
  exists = TRUE
)
```

Arguments

x_name	A string of the name.
sub	A string specifying the path to the sub folder (by default the current sub folder).
main	A string specifying the path to the main folder (by default the current main folder)
exists	A logical scalar specifying whether the file should exist.

Value

A data frame or NULL if doesn't exist.

See Also

Other load functions: sbf_load_block(), sbf_load_blocks(), sbf_load_blocks_recursive(), sbf_load_data_from_db(), sbf_load_data_from_pg(), sbf_load_datas(), sbf_load_datas_from_db(), sbf_load_datas_from_pg(), sbf_load_datas_recursive(), sbf_load_db_metatable(), sbf_load_number(), sbf_load_numbers(), sbf_load_numbers_recursive(), sbf_load_object(), sbf_load_objects(), sbf_load_objects_recursive(), sbf_load_plot(), sbf_load_plot_data(), sbf_load_plots_data(), sbf_load_plots_data_recursive(), sbf_load_plots_recursive(), sbf_load_spatial(), sbf_load_spatials(), sbf_load_string(), sbf_load_strings(), sbf_load_strings_recursive(), sbf_load_table(), sbf_load_tables(), sbf_load_tables_recursive(), sbf_load_windows_recursive(), sbf_subs_block_recursive(sbf_subs_data_recursive(), sbf_subs_number_recursive(), sbf_subs_object_recursive(), sbf_subs_plot_recursive(), sbf_subs_string_recursive(), sbf_subs_table_recursive(), sbf_subs_window_recursive()

sbf_load_datas *Load Datas*

Description

Load Datas

Usage

```
sbf_load_datas(
  sub = sbf_get_sub(),
  main = sbf_get_main(),
  rename = identity,
  env = parent.frame()
)
```

Arguments

sub	A string specifying the path to the sub folder (by default the current sub folder).
main	A string specifying the path to the main folder (by default the current main folder)
rename	A single function argument which takes a character vector and returns a character vector of the same length. Used to rename objects before they are loaded into the environment.
env	The environment to the objects into

Value

A invisble character vector of the data frames' names.

See Also

Other load functions: sbf_load_block(), sbf_load_blocks(), sbf_load_blocks_recursive(), sbf_load_data(), sbf_load_data_from_db(), sbf_load_data_from_pg(), sbf_load_datas_from_db(), sbf_load_datas_from_pg(), sbf_load_datas_recursive(), sbf_load_db_metatable(), sbf_load_number(), sbf_load_numbers(), sbf_load_numbers_recursive(), sbf_load_object(), sbf_load_objects(), sbf_load_objects_recursive(), sbf_load_plot(), sbf_load_plot_data(), sbf_load_plots_data(), sbf_load_plots_data_recursive(), sbf_load_plots_recursive(), sbf_load_spatial(), sbf_load_spatials(), sbf_load_string(), sbf_load_strings(), sbf_load_strings_recursive(), sbf_load_table(), sbf_load_tables(), sbf_load_tables_recursive(), sbf_load_windows_recursive(), sbf_subs_block_recursive(), sbf_subs_data_recursive(), sbf_subs_number_recursive(), sbf_subs_object_recursive(), sbf_subs_plot_recursive(), sbf_subs_string_recursive(), sbf_subs_table_recursive(), sbf_subs_window_recursive()

sbf_load_datas_from_db *Load Data Frames from Database*

Description

Load Data Frames from Database

Usage

```
sbf_load_datas_from_db(
  db_name = sbf_get_db_name(),
  sub = sbf_get_sub(),
  main = sbf_get_main(),
  rename = identity,
  env = parent.frame()
)
```

Arguments

db_name	A string of the database name.
sub	A string specifying the path to the sub folder (by default the current sub folder).
main	A string specifying the path to the main folder (by default the current main folder)
rename	A single function argument which takes a character vector and returns a character vector of the same length. Used to rename objects before they are loaded into the environment.
env	The environment to the objects into

Value

An invisible character vector of the paths to the saved objects.

See Also

Other load functions: sbf_load_block(), sbf_load_blocks(), sbf_load_blocks_recursive(), sbf_load_data(), sbf_load_data_from_db(), sbf_load_data_from_pg(), sbf_load_datas(), sbf_load_datas_from_pg(), sbf_load_datas_recursive(), sbf_load_db_metatable(), sbf_load_number(), sbf_load_numbers(), sbf_load_numbers_recursive(), sbf_load_object(), sbf_load_objects(), sbf_load_objects_recursive(), sbf_load_plot(), sbf_load_plot_data(), sbf_load_plots_data(), sbf_load_plots_data_recursive(), sbf_load_plots_recursive(), sbf_load_spatial(), sbf_load_spatials(), sbf_load_string(), sbf_load_strings(), sbf_load_strings_recursive(), sbf_load_table(), sbf_load_tables(), sbf_load_tables_recursive(), sbf_load_windows_recursive(), sbf_subs_block_recursive(sbf_subs_data_recursive(), sbf_subs_number_recursive(), sbf_subs_object_recursive(), sbf_subs_plot_recursive(), sbf_subs_string_recursive(), sbf_subs_table_recursive(), sbf_subs_window_recursive()

sbf_load_datas_from_pg

Load Data Frames from PostgreSQL Database

Description

[Deprecated]

`sbf_load_datas_from_pg()` was moved to `subfoldr2ext::sbfx_load_datas_from_pg()`.

Load all the tables in a schema as data frames into your environment from a PostgreSQL database.

Usage

```
sbf_load_datas_from_pg(
  schema = getOption("psql.schema", "public"),
  rename = identity,
  env = parent.frame(),
  config_path = getOption("psql.config_path", NULL),
  config_value = getOption("psql.config_value", "default")
)
```

Arguments

`schema`	A string of the schema name. Default value is `"public"`.
`rename`	A single function argument which takes a character vector and returns a character vector of the same length. Used to rename objects before they are loaded into the environment.
`env`	The environment to the objects into
`config_path`	A string of a file path to the yaml configuration file. The default value grabs the file path from the psql.config_path option and uses `NULL` if no value supplied.
`config_value`	A string of the name of value. The default value grabs the value from the psql.config_value option and uses `"default"` if no value is supplied.

Value

An invisible character vector of the paths to the saved objects.

See Also

Other postgresql functions: `sbf_backup_pg()`, `sbf_close_pg()`, `sbf_create_pg()`, `sbf_execute_pg()`, `sbf_get_config_file()`, `sbf_get_config_value()`, `sbf_get_schema()`, `sbf_list_tables_pg()`, `sbf_load_data_from_pg()`, `sbf_open_pg()`, `sbf_reset_config_file()`, `sbf_reset_config_value()`, `sbf_reset_schema()`, `sbf_save_data_to_pg()`, `sbf_set_config_file()`, `sbf_set_config_value()`, `sbf_set_schema()`

Other load functions: `sbf_load_block()`, `sbf_load_blocks()`, `sbf_load_blocks_recursive()`, `sbf_load_data()`, `sbf_load_data_from_db()`, `sbf_load_data_from_pg()`, `sbf_load_datas()`,

sbf_load_datas_from_db(), sbf_load_datas_recursive(), sbf_load_db_metatable(), sbf_load_number(), sbf_load_numbers(), sbf_load_numbers_recursive(), sbf_load_object(), sbf_load_objects(), sbf_load_objects_recursive(), sbf_load_plot(), sbf_load_plot_data(), sbf_load_plots_data(), sbf_load_plots_data_recursive(), sbf_load_plots_recursive(), sbf_load_spatial(), sbf_load_spatials(), sbf_load_string(), sbf_load_strings(), sbf_load_strings_recursive(), sbf_load_table(), sbf_load_tables(), sbf_load_tables_recursive(), sbf_load_windows_recursive(), sbf_subs_block_recursive(sbf_subs_data_recursive(), sbf_subs_number_recursive(), sbf_subs_object_recursive(), sbf_subs_plot_recursive(), sbf_subs_string_recursive(), sbf_subs_table_recursive(), sbf_subs_window_recursive()

Examples

```
## Not run:
sbf_load_datas_from_pg()
sbf_load_datas_from_pg(schema = "capture")
sbf_load_datas_from_pg(rename = toupper)

## End(Not run)
```

---

sbf_load_datas_recursive

Load Data Frames as List Column in Data Frame

Description

Recursively loads all the data frames with names matching the regular expression x_name as the first (list) column (named data) in a data frame. Subsequent character vector columns specify the object names (named name) and sub folders (named sub1, sub2 etc).

Usage

```
sbf_load_datas_recursive(
  x_name = ".*",
  sub = sbf_get_sub(),
  main = sbf_get_main(),
  include_root = TRUE
)
```

Arguments

x_name	A string of the name.
sub	A string specifying the path to the sub folder (by default the current sub folder).
main	A string specifying the path to the main folder (by default the current main folder)
include_root	A flag indicating whether to include objects in the top sub folder.

See Also

Other load functions: sbf_load_block(), sbf_load_blocks(), sbf_load_blocks_recursive(), sbf_load_data(), sbf_load_data_from_db(), sbf_load_data_from_pg(), sbf_load_datas(), sbf_load_datas_from_db(), sbf_load_datas_from_pg(), sbf_load_db_metatable(), sbf_load_number(), sbf_load_numbers(), sbf_load_numbers_recursive(), sbf_load_object(), sbf_load_objects(), sbf_load_objects_recursive(), sbf_load_plot(), sbf_load_plot_data(), sbf_load_plots_data(), sbf_load_plots_data_recursive(), sbf_load_plots_recursive(), sbf_load_spatial(), sbf_load_spatials(), sbf_load_string(), sbf_load_strings(), sbf_load_strings_recursive(), sbf_load_table(), sbf_load_tables(), sbf_load_tables_recursive(), sbf_load_windows_recursive(), sbf_subs_block_recursive(sbf_subs_data_recursive(), sbf_subs_number_recursive(), sbf_subs_object_recursive(), sbf_subs_plot_recursive(), sbf_subs_string_recursive(), sbf_subs_table_recursive(), sbf_subs_window_recursive()

---

sbf_load_data_from_db *Load Data Frame from Database*

Description

Load Data Frame from Database

Usage

```
sbf_load_data_from_db(
  x_name,
  db_name = sbf_get_db_name(),
  sub = sbf_get_sub(),
  main = sbf_get_main()
)
```

Arguments

x_name	A string of the name.
db_name	A string of the database name.
sub	A string specifying the path to the sub folder (by default the current sub folder).
main	A string specifying the path to the main folder (by default the current main folder)

Value

A data.frame of the table.

See Also

Other load functions: sbf_load_block(), sbf_load_blocks(), sbf_load_blocks_recursive(), sbf_load_data(), sbf_load_data_from_pg(), sbf_load_datas(), sbf_load_datas_from_db(), sbf_load_datas_from_pg(), sbf_load_datas_recursive(), sbf_load_db_metatable(), sbf_load_number(), sbf_load_numbers(), sbf_load_numbers_recursive(), sbf_load_object(), sbf_load_objects(), sbf_load_objects_recursive(), sbf_load_plot(), sbf_load_plot_data(), sbf_load_plots_data(), sbf_load_plots_data_recursive(), sbf_load_plots_recursive(), sbf_load_spatial(), sbf_load_spatials(), sbf_load_string(), sbf_load_strings(), sbf_load_strings_recursive(), sbf_load_table(), sbf_load_tables(), sbf_load_tables_recursive(), sbf_load_windows_recursive(), sbf_subs_block_recursive(), sbf_subs_data_recursive(), sbf_subs_number_recursive(), sbf_subs_object_recursive(), sbf_subs_plot_recursive(), sbf_subs_string_recursive(), sbf_subs_table_recursive(), sbf_subs_window_recursive()

---

sbf_load_data_from_pg *Load a table from a PostgreSQL database*

Description

[Deprecated]

`sbf_load_data_from_pg()` was moved to `subfoldr2ext::sbfx_load_data_from_pg()`.

Read/load a table from a PostgreSQL database as a data frame into R.

Usage

```
sbf_load_data_from_pg(
  x,
  schema = getOption("psql.schema", "public"),
  config_path = getOption("psql.config_path", NULL),
  config_value = getOption("psql.config_value", "default")
)
```

Arguments

x	A string of the table name
schema	A string of the schema name. Default value is "public".
config_path	A string of a file path to the yaml configuration file. The default value grabs the file path from the psql.config_path option and uses NULL if no value supplied.
config_value	A string of the name of value. The default value grabs the value from the psql.config_value option and uses "default" if no value is supplied.

Details

Wrapper on `psql::psql_read_table()`

Value

A data frame

See Also

Other postgresql functions: sbf_backup_pg(), sbf_close_pg(), sbf_create_pg(), sbf_execute_pg(), sbf_get_config_file(), sbf_get_config_value(), sbf_get_schema(), sbf_list_tables_pg(), sbf_load_datas_from_pg(), sbf_open_pg(), sbf_reset_config_file(), sbf_reset_config_value(), sbf_reset_schema(), sbf_save_data_to_pg(), sbf_set_config_file(), sbf_set_config_value(), sbf_set_schema()

Other load functions: sbf_load_block(), sbf_load_blocks(), sbf_load_blocks_recursive(), sbf_load_data(), sbf_load_data_from_db(), sbf_load_datas(), sbf_load_datas_from_db(), sbf_load_datas_from_pg(), sbf_load_datas_recursive(), sbf_load_db_metatable(), sbf_load_number(), sbf_load_numbers(), sbf_load_numbers_recursive(), sbf_load_object(), sbf_load_objects(), sbf_load_objects_recursive(), sbf_load_plot(), sbf_load_plot_data(), sbf_load_plots_data(), sbf_load_plots_data_recursive(), sbf_load_plots_recursive(), sbf_load_spatial(), sbf_load_spatials(), sbf_load_string(), sbf_load_strings(), sbf_load_strings_recursive(), sbf_load_table(), sbf_load_tables(), sbf_load_tables_recursive(), sbf_load_windows_recursive(), sbf_subs_block_recursive(sbf_subs_data_recursive(), sbf_subs_number_recursive(), sbf_subs_object_recursive(), sbf_subs_plot_recursive(), sbf_subs_string_recursive(), sbf_subs_table_recursive(), sbf_subs_window_recursive()

Examples

```
## Not run:
sbf_load_data_from_pg("capture")
sbf_load_data_from_pg("counts", "boat_count")

## End(Not run)
```

---

sbf_load_db_metatable *Load Data Frame of Meta Table from Database*

Description

Load Data Frame of Meta Table from Database

Usage

```
sbf_load_db_metatable(
  db_name = sbf_get_db_name(),
  sub = sbf_get_sub(),
  main = sbf_get_main()
)
```

Arguments

db_name	A string of the database name.
sub	A string specifying the path to the sub folder (by default the current sub folder).
main	A string specifying the path to the main folder (by default the current main folder)

Value

A data.frame of the table.

See Also

Other load functions: sbf_load_block(), sbf_load_blocks(), sbf_load_blocks_recursive(), sbf_load_data(), sbf_load_data_from_db(), sbf_load_data_from_pg(), sbf_load_datas(), sbf_load_datas_from_db(), sbf_load_datas_from_pg(), sbf_load_datas_recursive(), sbf_load_number(), sbf_load_numbers(), sbf_load_numbers_recursive(), sbf_load_object(), sbf_load_objects(), sbf_load_objects_recursive(), sbf_load_plot(), sbf_load_plot_data(), sbf_load_plots_data(), sbf_load_plots_data_recursive(), sbf_load_plots_recursive(), sbf_load_spatial(), sbf_load_spatials(), sbf_load_string(), sbf_load_strings(), sbf_load_strings_recursive(), sbf_load_table(), sbf_load_tables(), sbf_load_tables_recursive(), sbf_load_windows_recursive(), sbf_subs_block_recursive(), sbf_subs_data_recursive(), sbf_subs_number_recursive(), sbf_subs_object_recursive(), sbf_subs_plot_recursive(), sbf_subs_string_recursive(), sbf_subs_table_recursive(), sbf_subs_window_recursive()

---

sbf_load_number *Load Number*

Description

Load Number

Usage

```
sbf_load_number(
  x_name,
  sub = sbf_get_sub(),
  main = sbf_get_main(),
  exists = TRUE
)
```

Arguments

x_name	A string of the name.
sub	A string specifying the path to the sub folder (by default the current sub folder).
main	A string specifying the path to the main folder (by default the current main folder)
exists	A logical scalar specifying whether the file should exist.

Value

A number or NULL if doesn't exist.

See Also

Other load functions: sbf_load_block(), sbf_load_blocks(), sbf_load_blocks_recursive(), sbf_load_data(), sbf_load_data_from_db(), sbf_load_data_from_pg(), sbf_load_datas(), sbf_load_datas_from_db(), sbf_load_datas_from_pg(), sbf_load_datas_recursive(), sbf_load_db_metatable(), sbf_load_numbers(), sbf_load_numbers_recursive(), sbf_load_object(), sbf_load_objects(), sbf_load_objects_recursive(), sbf_load_plot(), sbf_load_plot_data(), sbf_load_plots_data(), sbf_load_plots_data_recursive(), sbf_load_plots_recursive(), sbf_load_spatial(), sbf_load_spatials(), sbf_load_string(), sbf_load_strings(), sbf_load_strings_recursive(), sbf_load_table(), sbf_load_tables(), sbf_load_tables_recursive(), sbf_load_windows_recursive(), sbf_subs_block_recursive(), sbf_subs_data_recursive(), sbf_subs_number_recursive(), sbf_subs_object_recursive(), sbf_subs_plot_recursive(), sbf_subs_string_recursive(), sbf_subs_table_recursive(), sbf_subs_window_recursive()

---

sbf_load_numbers *Load Numbers*

Description

Load Numbers

Usage

```
sbf_load_numbers(
  sub = sbf_get_sub(),
  main = sbf_get_main(),
  rename = identity,
  env = parent.frame()
)
```

Arguments

sub	A string specifying the path to the sub folder (by default the current sub folder).
main	A string specifying the path to the main folder (by default the current main folder)
rename	A single function argument which takes a character vector and returns a character vector of the same length. Used to rename objects before they are loaded into the environment.
env	The environment to the objects into

Value

A invisble character vector of the numbers' names.

See Also

Other load functions: sbf_load_block(), sbf_load_blocks(), sbf_load_blocks_recursive(), sbf_load_data(), sbf_load_data_from_db(), sbf_load_data_from_pg(), sbf_load_datas(), sbf_load_datas_from_db(), sbf_load_datas_from_pg(), sbf_load_datas_recursive(), sbf_load_db_metatable(), sbf_load_number(), sbf_load_numbers_recursive(), sbf_load_object(), sbf_load_objects(), sbf_load_objects_recursive(), sbf_load_plot(), sbf_load_plot_data(), sbf_load_plots_data(), sbf_load_plots_data_recursive(), sbf_load_plots_recursive(), sbf_load_spatial(), sbf_load_spatials(), sbf_load_string(), sbf_load_strings(), sbf_load_strings_recursive(), sbf_load_table(), sbf_load_tables(), sbf_load_tables_recursive(), sbf_load_windows_recursive(), sbf_subs_block_recursive(), sbf_subs_data_recursive(), sbf_subs_number_recursive(), sbf_subs_object_recursive(), sbf_subs_plot_recursive(), sbf_subs_string_recursive(), sbf_subs_table_recursive(), sbf_subs_window_recursive()

---

sbf_load_numbers_recursive

Load Numbers as Column in Data Frame

Description

Recursively loads all the numbers with names matching the regular expression x_name as the first double column (named numbers) in a data frame. Subsequent character vector columns specify the object names (named name) and sub folders (named sub1, sub2 etc).

Usage

```
sbf_load_numbers_recursive(
  x_name = ".*",
  sub = sbf_get_sub(),
  main = sbf_get_main(),
  include_root = TRUE
)
```

Arguments

x_name	A string of the name.
sub	A string specifying the path to the sub folder (by default the current sub folder).
main	A string specifying the path to the main folder (by default the current main folder)
include_root	A flag indicating whether to include objects in the top sub folder.

See Also

Other load functions: sbf_load_block(), sbf_load_blocks(), sbf_load_blocks_recursive(), sbf_load_data(), sbf_load_data_from_db(), sbf_load_data_from_pg(), sbf_load_datas(), sbf_load_datas_from_db(), sbf_load_datas_from_pg(), sbf_load_datas_recursive(), sbf_load_db_metatable(), sbf_load_number(), sbf_load_numbers(), sbf_load_object(), sbf_load_objects(), sbf_load_objects_recursive

sbf_load_plot(), sbf_load_plot_data(), sbf_load_plots_data(), sbf_load_plots_data_recursive(), sbf_load_plots_recursive(), sbf_load_spatial(), sbf_load_spatials(), sbf_load_string(), sbf_load_strings(), sbf_load_strings_recursive(), sbf_load_table(), sbf_load_tables(), sbf_load_tables_recursive(), sbf_load_windows_recursive(), sbf_subs_block_recursive(), sbf_subs_data_recursive(), sbf_subs_number_recursive(), sbf_subs_object_recursive(), sbf_subs_plot_recursive(), sbf_subs_string_recursive(), sbf_subs_table_recursive(), sbf_subs_window_recursive()

`sbf_load_object` *Load Object*

Description

Load Object

Usage

```
sbf_load_object(
  x_name,
  sub = sbf_get_sub(),
  main = sbf_get_main(),
  exists = TRUE
)
```

Arguments

x_name	A string of the name.
sub	A string specifying the path to the sub folder (by default the current sub folder).
main	A string specifying the path to the main folder (by default the current main folder)
exists	A logical scalar specifying whether the file should exist.

Value

An R object or NULL if doesn't exist.

See Also

Other load functions: sbf_load_block(), sbf_load_blocks(), sbf_load_blocks_recursive(), sbf_load_data(), sbf_load_data_from_db(), sbf_load_data_from_pg(), sbf_load_datas(), sbf_load_datas_from_db(), sbf_load_datas_from_pg(), sbf_load_datas_recursive(), sbf_load_db_metatable(), sbf_load_number(), sbf_load_numbers(), sbf_load_numbers_recursive(), sbf_load_objects(), sbf_load_objects_recursive(), sbf_load_plot(), sbf_load_plot_data(), sbf_load_plots_data(), sbf_load_plots_data_recursive(), sbf_load_plots_recursive(), sbf_load_spatial(), sbf_load_spatials(), sbf_load_string(), sbf_load_strings(), sbf_load_strings_recursive(), sbf_load_table(), sbf_load_tables(), sbf_load_tables_recursive(), sbf_load_windows_recursive(), sbf_subs_block_recursive(), sbf_subs_data_recursive(), sbf_subs_number_recursive(), sbf_subs_object_recursive(),

sbf_subs_plot_recursive(), sbf_subs_string_recursive(), sbf_subs_table_recursive(), sbf_subs_window_recursive()

---

sbf_load_objects *Load Objects*

Description

Load Objects

Usage

```
sbf_load_objects(
  sub = sbf_get_sub(),
  main = sbf_get_main(),
  rename = identity,
  env = parent.frame()
)
```

Arguments

sub	A string specifying the path to the sub folder (by default the current sub folder).
main	A string specifying the path to the main folder (by default the current main folder)
rename	A single function argument which takes a character vector and returns a character vector of the same length. Used to rename objects before they are loaded into the environment.
env	The environment to the objects into

Value

A invisble character vector of the objects' names.

See Also

Other load functions: sbf_load_block(), sbf_load_blocks(), sbf_load_blocks_recursive(), sbf_load_data(), sbf_load_data_from_db(), sbf_load_data_from_pg(), sbf_load_datas(), sbf_load_datas_from_db(), sbf_load_datas_from_pg(), sbf_load_datas_recursive(), sbf_load_db_metatable(), sbf_load_number(), sbf_load_numbers(), sbf_load_numbers_recursive(), sbf_load_object(), sbf_load_objects_recursive(), sbf_load_plot(), sbf_load_plot_data(), sbf_load_plots_data(), sbf_load_plots_data_recursive(), sbf_load_plots_recursive(), sbf_load_spatial(), sbf_load_spatials(), sbf_load_string(), sbf_load_strings(), sbf_load_strings_recursive(), sbf_load_table(), sbf_load_tables(), sbf_load_tables_recursive(), sbf_load_windows_recursive(), sbf_subs_block_recursive(), sbf_subs_data_recursive(), sbf_subs_number_recursive(), sbf_subs_object_recursive(), sbf_subs_plot_recursive(), sbf_subs_string_recursive(), sbf_subs_table_recursive(), sbf_subs_window_recursive()

sbf_load_objects_recursive *Load Objects as List Column in Data Frame*

Description

Recursively loads all the objects with names matching the regular expression x_name as the the first (list) column (named objects) in a data frame. Subsequent character vector columns specify the object names (named name) and sub folders (named sub1, sub2 etc).

Usage

```
sbf_load_objects_recursive(
  x_name = ".*",
  sub = sbf_get_sub(),
  main = sbf_get_main(),
  include_root = TRUE
)
```

Arguments

x_name	A string of the regular expression to match.
sub	A string specifying the path to the sub folder (by default the current sub folder).
main	A string specifying the path to the main folder (by default the current main folder)
include_root	A flag indicating whether to include objects in the top sub folder.

See Also

Other load functions: sbf_load_block(), sbf_load_blocks(), sbf_load_blocks_recursive(), sbf_load_data(), sbf_load_data_from_db(), sbf_load_data_from_pg(), sbf_load_datas(), sbf_load_datas_from_db(), sbf_load_datas_from_pg(), sbf_load_datas_recursive(), sbf_load_db_metatable(), sbf_load_number(), sbf_load_numbers(), sbf_load_numbers_recursive(), sbf_load_object(), sbf_load_objects(), sbf_load_plot(), sbf_load_plot_data(), sbf_load_plots_data(), sbf_load_plots_data_re, sbf_load_plots_recursive(), sbf_load_spatial(), sbf_load_spatials(), sbf_load_string(), sbf_load_strings(), sbf_load_strings_recursive(), sbf_load_table(), sbf_load_tables(), sbf_load_tables_recursive(), sbf_load_windows_recursive(), sbf_subs_block_recursive(), sbf_subs_data_recursive(), sbf_subs_number_recursive(), sbf_subs_object_recursive(), sbf_subs_plot_recursive(), sbf_subs_string_recursive(), sbf_subs_table_recursive(), sbf_subs_window_recursive()

sbf_load_plot *Load Plot*

Description

Load Plot

Usage

```
sbf_load_plot(
  x_name,
  sub = sbf_get_sub(),
  main = sbf_get_main(),
  exists = TRUE
)
```

Arguments

x_name	A string of the name.
sub	A string specifying the path to the sub folder (by default the current sub folder).
main	A string specifying the path to the main folder (by default the current main folder)
exists	A logical scalar specifying whether the file should exist.

Value

A ggplot object or NULL if doesn't exist.

See Also

Other load functions: sbf_load_block(), sbf_load_blocks(), sbf_load_blocks_recursive(), sbf_load_data(), sbf_load_data_from_db(), sbf_load_data_from_pg(), sbf_load_datas(), sbf_load_datas_from_db(), sbf_load_datas_from_pg(), sbf_load_datas_recursive(), sbf_load_db_metatable(), sbf_load_number(), sbf_load_numbers(), sbf_load_numbers_recursive(), sbf_load_object(), sbf_load_objects(), sbf_load_objects_recursive(), sbf_load_plot_data(), sbf_load_plots_data(), sbf_load_plots_data_recursive(), sbf_load_plots_recursive(), sbf_load_spatial(), sbf_load_spatials(), sbf_load_string(), sbf_load_strings(), sbf_load_strings_recursive(), sbf_load_table(), sbf_load_tables(), sbf_load_tables_recursive(), sbf_load_windows_recursive(), sbf_subs_block_recursive(), sbf_subs_data_recursive(), sbf_subs_number_recursive(), sbf_subs_object_recursive(), sbf_subs_plot_recursive(), sbf_subs_string_recursive(), sbf_subs_table_recursive(), sbf_subs_window_recursive()

`sbf_load_plots_data` *Load Plots Data*

Description

Load Plots Data

Usage

```
sbf_load_plots_data(
  sub = sbf_get_sub(),
  main = sbf_get_main(),
  rename = identity,
  env = parent.frame()
)
```

Arguments

`sub`	A string specifying the path to the sub folder (by default the current sub folder).
`main`	A string specifying the path to the main folder (by default the current main folder)
`rename`	A single function argument which takes a character vector and returns a character vector of the same length. Used to rename objects before they are loaded into the environment.
`env`	The environment to the objects into

Value

A invisble character vector of the plots' names.

See Also

Other load functions: `sbf_load_block()`, `sbf_load_blocks()`, `sbf_load_blocks_recursive()`, `sbf_load_data()`, `sbf_load_data_from_db()`, `sbf_load_data_from_pg()`, `sbf_load_datas()`, `sbf_load_datas_from_db()`, `sbf_load_datas_from_pg()`, `sbf_load_datas_recursive()`, `sbf_load_db_metatable()`, `sbf_load_number()`, `sbf_load_numbers()`, `sbf_load_numbers_recursive()`, `sbf_load_object()`, `sbf_load_objects()`, `sbf_load_objects_recursive()`, `sbf_load_plot()`, `sbf_load_plot_data()`, `sbf_load_plots_data_recursive()`, `sbf_load_plots_recursive()`, `sbf_load_spatial()`, `sbf_load_spatials()`, `sbf_load_string()`, `sbf_load_strings()`, `sbf_load_strings_recursive()`, `sbf_load_table()`, `sbf_load_tables()`, `sbf_load_tables_recursive()`, `sbf_load_windows_recursive()`, `sbf_subs_block_recursive()`, `sbf_subs_data_recursive()`, `sbf_subs_number_recursive()`, `sbf_subs_object_recursive()`, `sbf_subs_plot_recursive()`, `sbf_subs_string_recursive()`, `sbf_subs_table_recursive()`, `sbf_subs_window_recursive()`

sbf_load_plots_data_recursive *Load Plots Data as List Column in Data Frame*

Description

Recursively loads all the default data from the plots with names matching the regular expression x_name as the first (list) column (named plots_data) in a data frame. Subsequent character vector columns specify the object names (named name) and sub folders (named sub1, sub2 etc).

Usage

```
sbf_load_plots_data_recursive(
  x_name = ".*",
  sub = sbf_get_sub(),
  main = sbf_get_main(),
  include_root = TRUE,
  tag = ".*",
  meta = FALSE
)
```

Arguments

x_name	A string of the name.
sub	A string specifying the path to the sub folder (by default the current sub folder).
main	A string specifying the path to the main folder (by default the current main folder)
include_root	A flag indicating whether to include objects in the top sub folder.
tag	A string of the regular expression that the tag must match to be included.
meta	A flag specifying whether to include the report, caption and any other metadata as columns.

See Also

Other load functions: sbf_load_block(), sbf_load_blocks(), sbf_load_blocks_recursive(), sbf_load_data(), sbf_load_data_from_db(), sbf_load_data_from_pg(), sbf_load_datas(), sbf_load_datas_from_db(), sbf_load_datas_from_pg(), sbf_load_datas_recursive(), sbf_load_db_metatable(), sbf_load_number(), sbf_load_numbers(), sbf_load_numbers_recursive(), sbf_load_object(), sbf_load_objects(), sbf_load_objects_recursive(), sbf_load_plot(), sbf_load_plot_data(), sbf_load_plots_data(), sbf_load_plots_recursive(), sbf_load_spatial(), sbf_load_spatials(), sbf_load_string(), sbf_load_strings(), sbf_load_strings_recursive(), sbf_load_table(), sbf_load_tables(), sbf_load_tables_recursive(), sbf_load_windows_recursive(), sbf_subs_block_recursive(), sbf_subs_data_recursive(), sbf_subs_number_recursive(), sbf_subs_object_recursive(), sbf_subs_plot_recursive(), sbf_subs_string_recursive(), sbf_subs_table_recursive(), sbf_subs_window_recursive()

sbf_load_plots_recursive *Load Plots as List Column in Data Frame*

Description

Recursively loads all the plots with names matching the regular expression x_name as the first (list) column (named plots) in a data frame. Subsequent character vector columns specify the object names (named name) and sub folders (named sub1, sub2 etc).

Usage

```
sbf_load_plots_recursive(
  x_name = ".*",
  sub = sbf_get_sub(),
  main = sbf_get_main(),
  include_root = TRUE,
  tag = ".*",
  meta = FALSE
)
```

Arguments

x_name	A string of the name.
sub	A string specifying the path to the sub folder (by default the current sub folder).
main	A string specifying the path to the main folder (by default the current main folder)
include_root	A flag indicating whether to include objects in the top sub folder.
tag	A string of the regular expression that the tag must match to be included.
meta	A flag specifying whether to include the report, caption and any other metadata as columns.

See Also

Other load functions: sbf_load_block(), sbf_load_blocks(), sbf_load_blocks_recursive(), sbf_load_data(), sbf_load_data_from_db(), sbf_load_data_from_pg(), sbf_load_datas(), sbf_load_datas_from_db(), sbf_load_datas_from_pg(), sbf_load_datas_recursive(), sbf_load_db_metatable(), sbf_load_number(), sbf_load_numbers(), sbf_load_numbers_recursive(), sbf_load_object(), sbf_load_objects(), sbf_load_objects_recursive(), sbf_load_plot(), sbf_load_plot_data(), sbf_load_plots_data(), sbf_load_plots_data_recursive(), sbf_load_spatial(), sbf_load_spatials(), sbf_load_string(), sbf_load_strings(), sbf_load_strings_recursive(), sbf_load_table(), sbf_load_tables(), sbf_load_tables_recursive(), sbf_load_windows_recursive(), sbf_subs_block_recursive(), sbf_subs_data_recursive(), sbf_subs_number_recursive(), sbf_subs_object_recursive(), sbf_subs_plot_recursive(), sbf_subs_string_recursive(), sbf_subs_table_recursive(), sbf_subs_window_recursive()

sbf_load_plot_data *Load Plot Data*

Description

Load Plot Data

Usage

```
sbf_load_plot_data(
  x_name,
  sub = sbf_get_sub(),
  main = sbf_get_main(),
  exists = TRUE
)
```

Arguments

x_name	A string of the name.
sub	A string specifying the path to the sub folder (by default the current sub folder).
main	A string specifying the path to the main folder (by default the current main folder)
exists	A logical scalar specifying whether the file should exist.

Value

A data.frame or NULL if doesn't exist.

See Also

Other load functions: sbf_load_block(), sbf_load_blocks(), sbf_load_blocks_recursive(), sbf_load_data(), sbf_load_data_from_db(), sbf_load_data_from_pg(), sbf_load_datas(), sbf_load_datas_from_db(), sbf_load_datas_from_pg(), sbf_load_datas_recursive(), sbf_load_db_metatable(), sbf_load_number(), sbf_load_numbers(), sbf_load_numbers_recursive(), sbf_load_object(), sbf_load_objects(), sbf_load_objects_recursive(), sbf_load_plot(), sbf_load_plots_data(), sbf_load_plots_data_recursive(), sbf_load_plots_recursive(), sbf_load_spatial(), sbf_load_spatials(), sbf_load_string(), sbf_load_strings(), sbf_load_strings_recursive(), sbf_load_table(), sbf_load_tables(), sbf_load_tables_recursive(), sbf_load_windows_recursive(), sbf_subs_block_recursive(), sbf_subs_data_recursive(), sbf_subs_number_recursive(), sbf_subs_object_recursive(), sbf_subs_plot_recursive(), sbf_subs_string_recursive(), sbf_subs_table_recursive(), sbf_subs_window_recursive()

sbf_load_spatial *Load Spatial Data*

Description

Loads an sf tbl that must meet the same requirements as sbf_save_spatial.

Usage

```
sbf_load_spatial(
  x_name,
  sub = sbf_get_sub(),
  main = sbf_get_main(),
  exists = TRUE
)
```

Arguments

x_name	A string of the name.
sub	A string specifying the path to the sub folder (by default the current sub folder).
main	A string specifying the path to the main folder (by default the current main folder)
exists	A logical scalar specifying whether the file should exist.

Value

An sf tbl or NULL if doesn't exist.

See Also

Other load functions: sbf_load_block(), sbf_load_blocks(), sbf_load_blocks_recursive(), sbf_load_data(), sbf_load_data_from_db(), sbf_load_data_from_pg(), sbf_load_datas(), sbf_load_datas_from_db(), sbf_load_datas_from_pg(), sbf_load_datas_recursive(), sbf_load_db_metatable(), sbf_load_number(), sbf_load_numbers(), sbf_load_numbers_recursive(), sbf_load_object(), sbf_load_objects(), sbf_load_objects_recursive(), sbf_load_plot(), sbf_load_plot_data(), sbf_load_plots_data(), sbf_load_plots_data_recursive(), sbf_load_plots_recursive(), sbf_load_spatials(), sbf_load_string(), sbf_load_strings(), sbf_load_strings_recursive(), sbf_load_table(), sbf_load_tables(), sbf_load_tables_recursive(), sbf_load_windows_recursive(), sbf_subs_block_recursive(), sbf_subs_data_recursive(), sbf_subs_number_recursive(), sbf_subs_object_recursive(), sbf_subs_plot_recursive(), sbf_subs_string_recursive(), sbf_subs_table_recursive(), sbf_subs_window_recursive()

`sbf_load_spatials` *Load Spatial Datas*

Description

Loads sf tbls that must meet the same requirements as `sbf_save_spatials`.

Usage

```
sbf_load_spatials(
  sub = sbf_get_sub(),
  main = sbf_get_main(),
  rename = identity,
  env = parent.frame()
)
```

Arguments

`sub`	A string specifying the path to the sub folder (by default the current sub folder).
`main`	A string specifying the path to the main folder (by default the current main folder)
`rename`	A single function argument which takes a character vector and returns a character vector of the same length. Used to rename objects before they are loaded into the environment.
`env`	The environment to the objects into

Value

A invisble character vector of the data frames' names.

See Also

Other load functions: `sbf_load_block()`, `sbf_load_blocks()`, `sbf_load_blocks_recursive()`, `sbf_load_data()`, `sbf_load_data_from_db()`, `sbf_load_data_from_pg()`, `sbf_load_datas()`, `sbf_load_datas_from_db()`, `sbf_load_datas_from_pg()`, `sbf_load_datas_recursive()`, `sbf_load_db_metatable()`, `sbf_load_number()`, `sbf_load_numbers()`, `sbf_load_numbers_recursive()`, `sbf_load_object()`, `sbf_load_objects()`, `sbf_load_objects_recursive()`, `sbf_load_plot()`, `sbf_load_plot_data()`, `sbf_load_plots_data()`, `sbf_load_plots_data_recursive()`, `sbf_load_plots_recursive()`, `sbf_load_spatial()`, `sbf_load_string()`, `sbf_load_strings()`, `sbf_load_strings_recursive()`, `sbf_load_table()`, `sbf_load_tables()`, `sbf_load_tables_recursive()`, `sbf_load_windows_recursive()`, `sbf_subs_block_recursive()`, `sbf_subs_data_recursive()`, `sbf_subs_number_recursive()`, `sbf_subs_object_recursive()`, `sbf_subs_plot_recursive()`, `sbf_subs_string_recursive()`, `sbf_subs_table_recursive()`, `sbf_subs_window_recursive()`

sbf_load_string *Load String*

Description

Load String

Usage

```
sbf_load_string(
  x_name,
  sub = sbf_get_sub(),
  main = sbf_get_main(),
  exists = TRUE
)
```

Arguments

x_name	A string of the name.
sub	A string specifying the path to the sub folder (by default the current sub folder).
main	A string specifying the path to the main folder (by default the current main folder)
exists	A logical scalar specifying whether the file should exist.

Value

A string or NULL if doesn't exist.

See Also

Other load functions: sbf_load_block(), sbf_load_blocks(), sbf_load_blocks_recursive(), sbf_load_data(), sbf_load_data_from_db(), sbf_load_data_from_pg(), sbf_load_datas(), sbf_load_datas_from_db(), sbf_load_datas_from_pg(), sbf_load_datas_recursive(), sbf_load_db_metatable(), sbf_load_number(), sbf_load_numbers(), sbf_load_numbers_recursive(), sbf_load_object(), sbf_load_objects(), sbf_load_objects_recursive(), sbf_load_plot(), sbf_load_plot_data(), sbf_load_plots_data(), sbf_load_plots_data_recursive(), sbf_load_plots_recursive(), sbf_load_spatial(), sbf_load_spatials(), sbf_load_strings(), sbf_load_strings_recursive(), sbf_load_table(), sbf_load_tables(), sbf_load_tables_recursive(), sbf_load_windows_recursive(), sbf_subs_block_recursive(), sbf_subs_data_recursive(), sbf_subs_number_recursive(), sbf_subs_object_recursive(), sbf_subs_plot_recursive(), sbf_subs_string_recursive(), sbf_subs_table_recursive(), sbf_subs_window_recursive()

sbf_load_strings *Load Strings*

Description

Load Strings

Usage

```
sbf_load_strings(
  sub = sbf_get_sub(),
  main = sbf_get_main(),
  rename = identity,
  env = parent.frame()
)
```

Arguments

sub	A string specifying the path to the sub folder (by default the current sub folder).
main	A string specifying the path to the main folder (by default the current main folder)
rename	A single function argument which takes a character vector and returns a character vector of the same length. Used to rename objects before they are loaded into the environment.
env	The environment to the objects into

Value

A invisble character vector of the string' names.

See Also

Other load functions: sbf_load_block(), sbf_load_blocks(), sbf_load_blocks_recursive(), sbf_load_data(), sbf_load_data_from_db(), sbf_load_data_from_pg(), sbf_load_datas(), sbf_load_datas_from_db(), sbf_load_datas_from_pg(), sbf_load_datas_recursive(), sbf_load_db_metatable(), sbf_load_number(), sbf_load_numbers(), sbf_load_numbers_recursive(), sbf_load_object(), sbf_load_objects(), sbf_load_objects_recursive(), sbf_load_plot(), sbf_load_plot_data(), sbf_load_plots_data(), sbf_load_plots_data_recursive(), sbf_load_plots_recursive(), sbf_load_spatial(), sbf_load_spatials(), sbf_load_string(), sbf_load_strings_recursive(), sbf_load_table(), sbf_load_tables(), sbf_load_tables_recursive(), sbf_load_windows_recursive(), sbf_subs_block_recursive(), sbf_subs_data_recursive(), sbf_subs_number_recursive(), sbf_subs_object_recursive(), sbf_subs_plot_recursive(), sbf_subs_string_recursive(), sbf_subs_table_recursive(), sbf_subs_window_recursive()

sbf_load_strings_recursive *Load Strings as Column in Data Frame*

Description

Recursively loads all the numbers with names matching the regular expression x_name as the first character column (named strings) in a data frame. Subsequent character vector columns specify the object names (named name) and sub folders (named sub1, sub2 etc).

Usage

```
sbf_load_strings_recursive(
  x_name = ".*",
  sub = sbf_get_sub(),
  main = sbf_get_main(),
  include_root = TRUE,
  tag = ".*",
  meta = FALSE
)
```

Arguments

x_name	A string of the name.
sub	A string specifying the path to the sub folder (by default the current sub folder).
main	A string specifying the path to the main folder (by default the current main folder)
include_root	A flag indicating whether to include objects in the top sub folder.
tag	A string of the regular expression that the tag must match to be included.
meta	A flag specifying whether to include the report, caption and any other metadata as columns.

See Also

Other load functions: sbf_load_block(), sbf_load_blocks(), sbf_load_blocks_recursive(), sbf_load_data(), sbf_load_data_from_db(), sbf_load_data_from_pg(), sbf_load_datas(), sbf_load_datas_from_db(), sbf_load_datas_from_pg(), sbf_load_datas_recursive(), sbf_load_db_metatable(), sbf_load_number(), sbf_load_numbers(), sbf_load_numbers_recursive(), sbf_load_object(), sbf_load_objects(), sbf_load_objects_recursive(), sbf_load_plot(), sbf_load_plot_data(), sbf_load_plots_data(), sbf_load_plots_data_recursive(), sbf_load_plots_recursive(), sbf_load_spatial(), sbf_load_spatials(), sbf_load_string(), sbf_load_strings(), sbf_load_table(), sbf_load_tables(), sbf_load_tables_recursive(), sbf_load_windows_recursive(), sbf_subs_block_recursive(), sbf_subs_data_recursive(), sbf_subs_number_recursive(), sbf_subs_object_recursive(), sbf_subs_plot_recursive(), sbf_subs_string_recursive(), sbf_subs_table_recursive(), sbf_subs_window_recursive()

sbf_load_table *Load Table*

Description

Load Table

Usage

```
sbf_load_table(
  x_name,
  sub = sbf_get_sub(),
  main = sbf_get_main(),
  exists = TRUE
)
```

Arguments

x_name	A string of the name.
sub	A string specifying the path to the sub folder (by default the current sub folder).
main	A string specifying the path to the main folder (by default the current main folder)
exists	A logical scalar specifying whether the file should exist.

Value

A data frame or NULL if doesn't exist.

See Also

Other load functions: sbf_load_block(), sbf_load_blocks(), sbf_load_blocks_recursive(), sbf_load_data(), sbf_load_data_from_db(), sbf_load_data_from_pg(), sbf_load_datas(), sbf_load_datas_from_db(), sbf_load_datas_from_pg(), sbf_load_datas_recursive(), sbf_load_db_metatable(), sbf_load_number(), sbf_load_numbers(), sbf_load_numbers_recursive(), sbf_load_object(), sbf_load_objects(), sbf_load_objects_recursive(), sbf_load_plot(), sbf_load_plot_data(), sbf_load_plots_data(), sbf_load_plots_data_recursive(), sbf_load_plots_recursive(), sbf_load_spatial(), sbf_load_spatials(), sbf_load_string(), sbf_load_strings(), sbf_load_strings_recursi, sbf_load_tables(), sbf_load_tables_recursive(), sbf_load_windows_recursive(), sbf_subs_block_recursive(, sbf_subs_data_recursive(), sbf_subs_number_recursive(), sbf_subs_object_recursive(), sbf_subs_plot_recursive(), sbf_subs_string_recursive(), sbf_subs_table_recursive(), sbf_subs_window_recursive()

`sbf_load_tables` *Load Tables*

Description

Load Tables

Usage

```
sbf_load_tables(
  sub = sbf_get_sub(),
  main = sbf_get_main(),
  rename = identity,
  env = parent.frame()
)
```

Arguments

`sub`	A string specifying the path to the sub folder (by default the current sub folder).
`main`	A string specifying the path to the main folder (by default the current main folder)
`rename`	A single function argument which takes a character vector and returns a character vector of the same length. Used to rename objects before they are loaded into the environment.
`env`	The environment to the objects into

Value

A invisble character vector of the data frames' names.

See Also

Other load functions: `sbf_load_block()`, `sbf_load_blocks()`, `sbf_load_blocks_recursive()`, `sbf_load_data()`, `sbf_load_data_from_db()`, `sbf_load_data_from_pg()`, `sbf_load_datas()`, `sbf_load_datas_from_db()`, `sbf_load_datas_from_pg()`, `sbf_load_datas_recursive()`, `sbf_load_db_metatable()`, `sbf_load_number()`, `sbf_load_numbers()`, `sbf_load_numbers_recursive()`, `sbf_load_object()`, `sbf_load_objects()`, `sbf_load_objects_recursive()`, `sbf_load_plot()`, `sbf_load_plot_data()`, `sbf_load_plots_data()`, `sbf_load_plots_data_recursive()`, `sbf_load_plots_recursive()`, `sbf_load_spatial()`, `sbf_load_spatials()`, `sbf_load_string()`, `sbf_load_strings()`, `sbf_load_strings_recursi`, `sbf_load_table()`, `sbf_load_tables_recursive()`, `sbf_load_windows_recursive()`, `sbf_subs_block_recursive()`, `sbf_subs_data_recursive()`, `sbf_subs_number_recursive()`, `sbf_subs_object_recursive()`, `sbf_subs_plot_recursive()`, `sbf_subs_string_recursive()`, `sbf_subs_table_recursive()`, `sbf_subs_window_recursive()`

sbf_load_tables_recursive *Load Data Frames as List Column in Data Frame*

Description

Recursively loads all the data frames with names matching the regular expression x_name as the first (list) column (named tables) in a data frame. Subsequent character vector columns specify the object names (named name) and sub folders (named sub1, sub2 etc).

Usage

```
sbf_load_tables_recursive(
  x_name = ".*",
  sub = sbf_get_sub(),
  main = sbf_get_main(),
  include_root = TRUE,
  tag = ".*",
  meta = FALSE
)
```

Arguments

x_name	A string of the name.
sub	A string specifying the path to the sub folder (by default the current sub folder).
main	A string specifying the path to the main folder (by default the current main folder)
include_root	A flag indicating whether to include objects in the top sub folder.
tag	A string of the regular expression that the tag must match to be included.
meta	A flag specifying whether to include the report, caption and any other metadata as columns.

See Also

Other load functions: sbf_load_block(), sbf_load_blocks(), sbf_load_blocks_recursive(), sbf_load_data(), sbf_load_data_from_db(), sbf_load_data_from_pg(), sbf_load_datas(), sbf_load_datas_from_db(), sbf_load_datas_from_pg(), sbf_load_datas_recursive(), sbf_load_db_metatable(), sbf_load_number(), sbf_load_numbers(), sbf_load_numbers_recursive(), sbf_load_object(), sbf_load_objects(), sbf_load_objects_recursive(), sbf_load_plot(), sbf_load_plot_data(), sbf_load_plots_data(), sbf_load_plots_data_recursive(), sbf_load_plots_recursive(), sbf_load_spatial(), sbf_load_spatials(), sbf_load_string(), sbf_load_strings(), sbf_load_strings_recursi, sbf_load_table(), sbf_load_tables(), sbf_load_windows_recursive(), sbf_subs_block_recursive(), sbf_subs_data_recursive(), sbf_subs_number_recursive(), sbf_subs_object_recursive(), sbf_subs_plot_recursive(), sbf_subs_string_recursive(), sbf_subs_table_recursive(), sbf_subs_window_recursive()

sbf_load_windows_recursive *Load Window Paths as Character Column in Data Frame*

Description

Recursively loads all the paths to the png files with names matching the regular expression x_name as the the first (list) column (named windows) in a data frame. Subsequent character vector columns specify the object names (named name) and sub folders (named sub1, sub2 etc).

Usage

```
sbf_load_windows_recursive(
  x_name = ".*",
  sub = sbf_get_sub(),
  main = sbf_get_main(),
  include_root = TRUE,
  tag = ".*",
  meta = FALSE
)
```

Arguments

x_name	A string of the name.
sub	A string specifying the path to the sub folder (by default the current sub folder).
main	A string specifying the path to the main folder (by default the current main folder)
include_root	A flag indicating whether to include objects in the top sub folder.
tag	A string of the regular expression that the tag must match to be included.
meta	A flag specifying whether to include the report, caption and any other metadata as columns.

See Also

Other load functions: sbf_load_block(), sbf_load_blocks(), sbf_load_blocks_recursive(), sbf_load_data(), sbf_load_data_from_db(), sbf_load_data_from_pg(), sbf_load_datas(), sbf_load_datas_from_db(), sbf_load_datas_from_pg(), sbf_load_datas_recursive(), sbf_load_db_metatable(), sbf_load_number(), sbf_load_numbers(), sbf_load_numbers_recursive(), sbf_load_object(), sbf_load_objects(), sbf_load_objects_recursive(), sbf_load_plot(), sbf_load_plot_data(), sbf_load_plots_data(), sbf_load_plots_data_recursive(), sbf_load_plots_recursive(), sbf_load_spatial(), sbf_load_spatials(), sbf_load_string(), sbf_load_strings(), sbf_load_strings_recursi, sbf_load_table(), sbf_load_tables(), sbf_load_tables_recursive(), sbf_subs_block_recursive(), sbf_subs_data_recursive(), sbf_subs_number_recursive(), sbf_subs_object_recursive(), sbf_subs_plot_recursive(), sbf_subs_string_recursive(), sbf_subs_table_recursive(), sbf_subs_window_recursive()

sbf_number_exists *Number Exists*

Description

this function is now deprecated as of version 0.0.0.9045

Usage

```
sbf_number_exists(x_name, sub = sbf_get_sub(), main = sbf_get_main())
```

Arguments

x_name	A string of the name.
sub	A string specifying the path to the sub folder (by default the current sub folder).
main	A string specifying the path to the main folder (by default the current main folder)

Value

A flag specifying whether the number exists.

See Also

Other exists functions: sbf_block_exists(), sbf_data_exists(), sbf_object_exists(), sbf_plot_exists(), sbf_string_exists(), sbf_table_exists()

sbf_object_exists *Object Exists*

Description

this function is now deprecated as of version 0.0.0.9045

Usage

```
sbf_object_exists(x_name, sub = sbf_get_sub(), main = sbf_get_main())
```

Arguments

x_name	A string of the name.
sub	A string specifying the path to the sub folder (by default the current sub folder).
main	A string specifying the path to the main folder (by default the current main folder)

Value

A flag specifying whether the object exists.

See Also

Other exists functions: sbf_block_exists(), sbf_data_exists(), sbf_number_exists(), sbf_plot_exists(), sbf_string_exists(), sbf_table_exists()

---

sbf_open_db *Open SQLite Database Connection*

Description

Opens a RSQLite::SQLiteConnection to a SQLite database. Foreign key constraints are enabled.

Usage

```
sbf_open_db(
  db_name = sbf_get_db_name(),
  sub = sbf_get_sub(),
  main = sbf_get_main(),
  exists = TRUE,
  caption = NULL,
  report = NA,
  tag = NULL,
  ask = getOption("sbf.ask", TRUE)
)
```

Arguments

db_name	A string of the database name.
sub	A string specifying the path to the sub folder (by default the current sub folder).
main	A string specifying the path to the main folder (by default the current main folder)
exists	A logical scalar specifying whether the database must already exist.
caption	A string specifying the database metadata table caption. If NULL the caption is unchanged. If the caption is not specified for a databases it is set to be "". Deprecated.
report	A logical scalar specifying whether to include the database metadata table in the report. If report = NA the setting is not changed. Soft-deprecated. If the report status is not specified for a databases it is included in the report. deprecated.
tag	A string of the tag. Deprecated.
ask	A flag specifying whether to ask before deleting an existing database (if exists = FALSE).

See Also

Other database functions: sbf_add_blob_column_to_db(), sbf_close_db(), sbf_copy_db(), sbf_create_db(), sbf_execute_db(), sbf_query_db(), sbf_upload_flobs_to_db()

---

sbf_open_pdf *Open PDF Device*

Description

Opens a pdf device in the current pdfs subfolder using grDevices::[pdf][grDevices::pdf]().

Usage

```
sbf_open_pdf(
  x_name = "plots",
  sub = sbf_get_sub(),
  main = sbf_get_main(),
  width = 6,
  height = width
)
```

Arguments

x_name	A string of the name.
sub	A string specifying the path to the sub folder (by default the current sub folder).
main	A string specifying the path to the main folder (by default the current main folder)
width	A positive number indicating the width in inches.
height	A positive number indicating the height in inches.

See Also

Other graphic functions: sbf_close_pdf(), sbf_close_window(), sbf_close_windows(), sbf_open_window()

sbf_open_pg *Open PostgreSQL Connection*

Description

[Deprecated]

`sbf_open_pg()` was moved to `subfoldr2ext::sbfx_open_pg()`.

Connect to a PostgreSQL database with a config.yml file.

Usage

```
sbf_open_pg(
  config_path = getOption("psql.config_path", NULL),
  config_value = getOption("psql.config_value", "default")
)
```

Arguments

`config_path`	A string of a file path to the yaml configuration file. The default value grabs the file path from the psql.config_path option and uses `NULL` if no value supplied.
`config_value`	A string of the name of value. The default value grabs the value from the psql.config_value option and uses `"default"` if no value is supplied.

Details

Wrapper on `psql::psql_connect()`

Value

An S4 object that inherits from DBIConnection.

See Also

Other postgresql functions: `sbf_backup_pg()`, `sbf_close_pg()`, `sbf_create_pg()`, `sbf_execute_pg()`, `sbf_get_config_file()`, `sbf_get_config_value()`, `sbf_get_schema()`, `sbf_list_tables_pg()`, `sbf_load_data_from_pg()`, `sbf_load_datas_from_pg()`, `sbf_reset_config_file()`, `sbf_reset_config_value()`, `sbf_reset_schema()`, `sbf_save_data_to_pg()`, `sbf_set_config_file()`, `sbf_set_config_value()`, `sbf_set_schema()`

Examples

```
## Not run:
conn <- sbf_open_pg()
sbf_close_pg(conn)

sbf_open_pg("config.yml")
sbf_close_pg(conn)
```

```
sbf_open_pg(config_path = "config.yml", config_value = "database")
sbf_close_pg(conn)

## End(Not run)
```

---

sbf_open_window *Open Graphics Window*

Description

Opens a graphics window on any platform. By default the window is 6 x 6 inches.

Usage

```
sbf_open_window(width = 6, height = width)
```

Arguments

width	A positive number of the plotting area width in inches.
height	A positive number of the plotting area height in inches.

See Also

Other graphic functions: sbf_close_pdf(), sbf_close_window(), sbf_close_windows(), sbf_open_pdf()

---

sbf_path_block *Path to Code Block*

Description

Path to Code Block

Usage

```
sbf_path_block(
  x_name,
  sub = sbf_get_sub(),
  main = sbf_get_main(),
  ext = "rds",
  exists = NA
)
```

Arguments

x_name	A string of the name.
sub	A string specifying the path to the sub folder (by default the current sub folder).
main	A string specifying the path to the main folder (by default the current main folder)
ext	A string specifying the extension.
exists	A logical scalar specifying whether the file should exist.

Value

A string indicating the path.

---

sbf_path_data *Path to Data*

Description

Path to Data

Usage

```
sbf_path_data(
  x_name,
  sub = sbf_get_sub(),
  main = sbf_get_main(),
  ext = "rds",
  exists = NA
)
```

Arguments

x_name	A string of the name.
sub	A string specifying the path to the sub folder (by default the current sub folder).
main	A string specifying the path to the main folder (by default the current main folder)
ext	A string specifying the extension.
exists	A logical scalar specifying whether the file should exist.

Value

A string indicating the path.

sbf_path_db *Path to Database*

Description

Path to Database

Usage

```
sbf_path_db(
  x_name = sbf_get_db_name(),
  sub = sbf_get_sub(),
  main = sbf_get_main(),
  ext = "sqlite",
  exists = NA
)
```

Arguments

x_name	A string of the name.
sub	A string specifying the path to the sub folder (by default the current sub folder).
main	A string specifying the path to the main folder (by default the current main folder)
ext	A string specifying the extension.
exists	A logical scalar specifying whether the file should exist.

Value

A string indicating the path.

sbf_path_number *Path to Number*

Description

Path to Number

Usage

```
sbf_path_number(
  x_name,
  sub = sbf_get_sub(),
  main = sbf_get_main(),
  ext = "rds",
  exists = NA
)
```

Arguments

x_name	A string of the name.
sub	A string specifying the path to the sub folder (by default the current sub folder).
main	A string specifying the path to the main folder (by default the current main folder)
ext	A string specifying the extension.
exists	A logical scalar specifying whether the file should exist.

Value

A string indicating the path.

---

sbf_path_object *Path to Object*

Description

Path to Object

Usage

```
sbf_path_object(
  x_name,
  sub = sbf_get_sub(),
  main = sbf_get_main(),
  ext = "rds",
  exists = NA
)
```

Arguments

x_name	A string of the name.
sub	A string specifying the path to the sub folder (by default the current sub folder).
main	A string specifying the path to the main folder (by default the current main folder)
ext	A string specifying the extension.
exists	A logical scalar specifying whether the file should exist.

Value

A string indicating the path.

sbf_path_plot *Path to Plot*

Description

Path to Plot

Usage

```
sbf_path_plot(
  x_name,
  sub = sbf_get_sub(),
  main = sbf_get_main(),
  ext = "rds",
  exists = NA
)
```

Arguments

x_name	A string of the name.
sub	A string specifying the path to the sub folder (by default the current sub folder).
main	A string specifying the path to the main folder (by default the current main folder)
ext	A string specifying the extension.
exists	A logical scalar specifying whether the file should exist.

Value

A string indicating the path.

sbf_path_string *Path to String*

Description

Path to String

Usage

```
sbf_path_string(
  x_name,
  sub = sbf_get_sub(),
  main = sbf_get_main(),
  ext = "rds",
  exists = NA
)
```

Arguments

x_name	A string of the name.
sub	A string specifying the path to the sub folder (by default the current sub folder).
main	A string specifying the path to the main folder (by default the current main folder)
ext	A string specifying the extension.
exists	A logical scalar specifying whether the file should exist.

Value

A string indicating the path.

---

sbf_path_table *Path to Table*

Description

Path to Table

Usage

```
sbf_path_table(
  x_name,
  sub = sbf_get_sub(),
  main = sbf_get_main(),
  ext = "rds",
  exists = NA
)
```

Arguments

x_name	A string of the name.
sub	A string specifying the path to the sub folder (by default the current sub folder).
main	A string specifying the path to the main folder (by default the current main folder)
ext	A string specifying the extension.
exists	A logical scalar specifying whether the file should exist.

Value

A string indicating the path.

sbf_path_window *Path to Window*

Description

Path to Window

Usage

```
sbf_path_window(
  x_name,
  sub = sbf_get_sub(),
  main = sbf_get_main(),
  ext = "png",
  exists = NA
)
```

Arguments

x_name	A string of the name.
sub	A string specifying the path to the sub folder (by default the current sub folder).
main	A string specifying the path to the main folder (by default the current main folder)
ext	A string specifying the extension.
exists	A logical scalar specifying whether the file should exist.

Value

A string indicating the path.

sbf_plot_exists *Plot Exists*

Description

this function is now deprecated as of version 0.0.0.9045

Usage

```
sbf_plot_exists(x_name, sub = sbf_get_sub(), main = sbf_get_main())
```

Arguments

x_name	A string of the name.
sub	A string specifying the path to the sub folder (by default the current sub folder).
main	A string specifying the path to the main folder (by default the current main folder)

Value

A flag specifying whether the plot exists.

See Also

Other exists functions: sbf_block_exists(), sbf_data_exists(), sbf_number_exists(), sbf_object_exists(), sbf_string_exists(), sbf_table_exists()

---

sbf_print *Print ggplot Object*

Description

Retries printing a ggplot object if grid errors occurs.

Usage

```
sbf_print(
  x,
  newpage = is.null(vp),
  vp = NULL,
  ntry = 3L,
  plot = getOption("sbf.plot", TRUE)
)
```

Arguments

x	An object to print.
newpage	draw new (empty) page first?
vp	viewport to draw plot in
ntry	A positive whole number specifying the number of tries.
plot	A flag indicating whether or not to print the plot.

Details

Grid errors include the text "cannot pop the top-level viewport" or "no applicable method for 'depth'"

sbf_query_db *Query Existing Database*

Description

Really just a wrapper on DBI::dbGetQuery().

Usage

```
sbf_query_db(
  sql,
  db_name = sbf_get_db_name(),
  sub = sbf_get_sub(),
  main = sbf_get_main()
)
```

Arguments

sql	A string of the SQL statement to execute.
db_name	A string of the database name.
sub	A string specifying the path to the sub folder (by default the current sub folder).
main	A string specifying the path to the main folder (by default the current main folder)

Value

A scalar numeric of the number of rows affected by the statement.

See Also

Other database functions: sbf_add_blob_column_to_db(), sbf_close_db(), sbf_copy_db(), sbf_create_db(), sbf_execute_db(), sbf_open_db(), sbf_upload_flobs_to_db()

sbf_reset *Reset Main, Sub and Database Name*

Description

Reset Main, Sub and Database Name

Usage

```
sbf_reset()
```

Value

An invisible NULL.

See Also

Other directory functions: sbf_add_sub(), sbf_get_archive(), sbf_get_db_name(), sbf_get_main(), sbf_get_sub(), sbf_get_workbook_name(), sbf_reset_db_name(), sbf_reset_main(), sbf_reset_sub(), sbf_set_db_name(), sbf_set_main(), sbf_set_sub(), sbf_up_sub()

Examples

```
sbf_reset()
```

---

sbf_reset_config_file *Reset the Config File Path*

Description

[Deprecated]

`sbf_reset_config_file()` was moved to `subfoldr2ext::sbfx_reset_config_file()`.

Reset the psql.config_path option to the default value.

Usage

```
sbf_reset_config_file()
```

Value

An invisible string of the default file path

See Also

Other postgresql functions: sbf_backup_pg(), sbf_close_pg(), sbf_create_pg(), sbf_execute_pg(), sbf_get_config_file(), sbf_get_config_value(), sbf_get_schema(), sbf_list_tables_pg(), sbf_load_data_from_pg(), sbf_load_datas_from_pg(), sbf_open_pg(), sbf_reset_config_value(), sbf_reset_schema(), sbf_save_data_to_pg(), sbf_set_config_file(), sbf_set_config_value(), sbf_set_schema()

Examples

```
## Not run:
sbf_reset_config_file()

## End(Not run)
```

sbf_reset_config_value *Reset the Config File Value*

Description

[Deprecated]

`sbf_reset_config_value()` was moved to `subfoldr2ext::sbfx_reset_config_value()`.

Reset the value for `psql.config_value` to the default value.

Usage

```
sbf_reset_config_value()
```

Value

An invisible string of the default file path

See Also

Other postgresql functions: `sbf_backup_pg()`, `sbf_close_pg()`, `sbf_create_pg()`, `sbf_execute_pg()`, `sbf_get_config_file()`, `sbf_get_config_value()`, `sbf_get_schema()`, `sbf_list_tables_pg()`, `sbf_load_data_from_pg()`, `sbf_load_datas_from_pg()`, `sbf_open_pg()`, `sbf_reset_config_file()`, `sbf_reset_schema()`, `sbf_save_data_to_pg()`, `sbf_set_config_file()`, `sbf_set_config_value()`, `sbf_set_schema()`

Examples

```
## Not run:
sbf_reset_config_value()

## End(Not run)
```

sbf_reset_db_name *Reset Database Name*

Description

Sets database name option to 'database'.

Usage

```
sbf_reset_db_name()
```

Value

An invisible string of the 'database'.

See Also

Other directory functions: sbf_add_sub(), sbf_get_archive(), sbf_get_db_name(), sbf_get_main(), sbf_get_sub(), sbf_get_workbook_name(), sbf_reset(), sbf_reset_main(), sbf_reset_sub(), sbf_set_db_name(), sbf_set_main(), sbf_set_sub(), sbf_up_sub()

Examples

```
sbf_reset_db_name()
```

---

sbf_reset_main *Reset Main*

Description

Reset Main

Usage

```
sbf_reset_main(rm = FALSE, ask = getOption("sbf.ask", TRUE))
```

Arguments

rm	A flag specifying whether to remove the folder and all its contents if it already exists.
ask	A flag specifying whether to ask before removing the existing folder.

Value

An invisible copy of the string "output".

See Also

Other reset: sbf_reset_sub(), sbf_rm_main()

Other directory functions: sbf_add_sub(), sbf_get_archive(), sbf_get_db_name(), sbf_get_main(), sbf_get_sub(), sbf_get_workbook_name(), sbf_reset(), sbf_reset_db_name(), sbf_reset_sub(), sbf_set_db_name(), sbf_set_main(), sbf_set_sub(), sbf_up_sub()

sbf_reset_schema *Reset Schema Name*

Description

[Deprecated]

`sbf_reset_schema()` was moved to `subfoldr2ext::sbfx_reset_schema()`.

Reset schema name back to public

Usage

```
sbf_reset_schema()
```

Value

An invisible string of the schema name the database is set to

See Also

Other postgresql functions: `sbf_backup_pg()`, `sbf_close_pg()`, `sbf_create_pg()`, `sbf_execute_pg()`, `sbf_get_config_file()`, `sbf_get_config_value()`, `sbf_get_schema()`, `sbf_list_tables_pg()`, `sbf_load_data_from_pg()`, `sbf_load_datas_from_pg()`, `sbf_open_pg()`, `sbf_reset_config_file()`, `sbf_reset_config_value()`, `sbf_save_data_to_pg()`, `sbf_set_config_file()`, `sbf_set_config_value()`, `sbf_set_schema()`

Examples

```
## Not run:
sbf_reset_schema()

## End(Not run)
```

sbf_reset_sub *Reset Sub Folder*

Description

Reset Sub Folder

Usage

```
sbf_reset_sub(rm = FALSE, ask = getOption("sbf.ask", TRUE))
```

Arguments

rm	A flag specifying whether to remove the folder and all its contents if it already exists.
ask	A flag specifying whether to ask before removing the existing folder.

Value

An invisible character vector of length 0.

See Also

Other reset: sbf_reset_main(), sbf_rm_main()

Other directory functions: sbf_add_sub(), sbf_get_archive(), sbf_get_db_name(), sbf_get_main(), sbf_get_sub(), sbf_get_workbook_name(), sbf_reset(), sbf_reset_db_name(), sbf_reset_main(), sbf_set_db_name(), sbf_set_main(), sbf_set_sub(), sbf_up_sub()

Examples

```
sbf_set_sub("nameofsub")
sbf_get_sub()
sbf_reset_sub()
```

---

sbf_rm_flobs *Delete Flobs Subdfolder*

Description

Delete Flobs Subdfolder

Usage

```
sbf_rm_flobs(
  sub = sbf_get_sub(),
  main = sbf_get_main(),
  ask = getOption("sbf.ask", TRUE)
)
```

Arguments

sub	A string specifying the path to the sub folder (by default the current sub folder).
main	A string specifying the path to the main folder (by default the current main folder)
ask	A flag specifying whether to ask before deleting the subfolder.

Value

A invisible string of the directory deleted.

See Also

Other housekeeping functions: sbf_archive_main(), sbf_rm_main(), sbf_unarchive_main()

---

sbf_rm_main *Remove Main*

Description

Remove Main

Usage

```
sbf_rm_main(main = sbf_get_main(), ask = getOption("sbf.ask", TRUE))
```

Arguments

main	A string specifying the path to the main folder (by default the current main folder)
ask	A flag specifying whether to ask before removing the existing folder.

Value

An invisible copy of the main folder.

See Also

Other reset: sbf_reset_main(), sbf_reset_sub()

Other housekeeping functions: sbf_archive_main(), sbf_rm_flobs(), sbf_unarchive_main()

---

sbf_save_aws_files *Download files from AWS S3*

Description

[Deprecated]

sbf_save_aws_files() was moved to subfoldr2ext::sbfx_save_aws_files().

Download files from an AWS S3 bucket into the analysis.

Usage

```
sbf_save_aws_files(
  bucket_name,
  sub = sbf_get_sub(),
  main = sbf_get_main(),
  data_type = NULL,
  year = NULL,
  month = NULL,
  day = NULL,
  file_name = NULL,
  file_extension = NULL,
  max_request_size = 1000,
  ask = getOption("sbf.ask", TRUE),
  silent = TRUE,
  aws_access_key_id = Sys.getenv("AWS_ACCESS_KEY_ID"),
  aws_secret_access_key = Sys.getenv("AWS_SECRET_ACCESS_KEY"),
  region = Sys.getenv("AWS_REGION", "ca-central-1")
)
```

Arguments

bucket_name	A string of the AWS S3 bucket name.
sub	A string specifying the path to the sub folder (by default the current sub folder).
main	A string specifying the path to the main folder (by default the current main folder)
data_type	A string (by default NULL) for which data type to return. Check the folder names within the shiny-upload in AWS for options common examples include punch-data, tracks, logger, image and pdf.
year	A whole number (by default NULL) indicating which year to return. Format YYYY.
month	A whole number (by default NULL) indicating which month to return. Format MM.
day	A whole number (by default NULL) indicating which day to return. Format DD.
file_name	A string (by default NULL) containing the name of the file to return. Do not include extension type.
file_extension	A string (by default NULL) with the file extension to return. Do not include period.
max_request_size	A whole number (by default 1000) indicating the maximum number of files to be returned.
ask	A flag specifying whether to ask before overwriting files.
silent	A flag (by default FALSE) to silence messages about number of files returned. Set to TRUE to silence messages.
aws_access_key_id	A string of your AWS user access key ID. The default is the environment variable named AWS_ACCESS_KEY_ID.

aws_secret_access_key
: A string of your AWS user secret access key. The default is the environment variable named AWS_SECRET_ACCESS_KEY.

region
: A string of the AWS region. The default is the environment variable named AWS_REGION.

See Also

Other save functions: sbf_basename_sans_ext(), sbf_save_block(), sbf_save_data(), sbf_save_data_to_db(), sbf_save_data_to_pg(), sbf_save_datas(), sbf_save_datas_to_db(), sbf_save_db_metatable_descriptions(), sbf_save_db_to_workbook(), sbf_save_excel(), sbf_save_excels(), sbf_save_gpkg(), sbf_save_gpkgs(), sbf_save_number(), sbf_save_numbers(), sbf_save_object(), sbf_save_objects(), sbf_save_plot(), sbf_save_png(), sbf_save_spatial(), sbf_save_spatials(), sbf_save_string(), sbf_save_strings(), sbf_save_table(), sbf_save_window(), sbf_save_workbook()

Examples

```
## Not run:
sbf_save_aws_files(
  bucket_name = "exploit-upload-poissonconsulting",
  data_type = "upload-recapture",
  year = 2021,
  file_name = "processed_data",
  file_extension = "csv"
)

## End(Not run)
```

---

sbf_save_block *Save Block*

Description

A block in this context is a character vector of length one of.

Usage

```
sbf_save_block(
  x,
  x_name = substitute(x),
  sub = sbf_get_sub(),
  main = sbf_get_main(),
  caption = "",
  report = TRUE,
  tag = ""
)
```

Arguments

x	A string of the block to save.
x_name	A string of the name.
sub	A string specifying the path to the sub folder (by default the current sub folder).
main	A string specifying the path to the main folder (by default the current main folder)
caption	A string of the caption.
report	A flag specifying whether to include in a report.
tag	A string of the tag.

Value

An invisible string of the path to the saved object.

See Also

Other save functions: sbf_basename_sans_ext(), sbf_save_aws_files(), sbf_save_data(), sbf_save_data_to_db(), sbf_save_data_to_pg(), sbf_save_datas(), sbf_save_datas_to_db(), sbf_save_db_metatable_descriptions(), sbf_save_db_to_workbook(), sbf_save_excel(), sbf_save_excels(), sbf_save_gpkg(), sbf_save_gpkgs(), sbf_save_number(), sbf_save_numbers(), sbf_save_object(), sbf_save_objects(), sbf_save_plot(), sbf_save_png(), sbf_save_spatial(), sbf_save_spatials(), sbf_save_string(), sbf_save_strings(), sbf_save_table(), sbf_save_window(), sbf_save_workbook()

---

sbf_save_data *Save Data*

Description

Save Data

Usage

```
sbf_save_data(
  x,
  x_name = substitute(x),
  sub = sbf_get_sub(),
  main = sbf_get_main()
)
```

Arguments

x	The data frame to save.
x_name	A string of the name.
sub	A string specifying the path to the sub folder (by default the current sub folder).
main	A string specifying the path to the main folder (by default the current main folder)

Value

An invisible string of the path to the saved data.frame

See Also

Other save functions: sbf_basename_sans_ext(), sbf_save_aws_files(), sbf_save_block(), sbf_save_data_to_db(), sbf_save_data_to_pg(), sbf_save_datas(), sbf_save_datas_to_db(), sbf_save_db_metatable_descriptions(), sbf_save_db_to_workbook(), sbf_save_excel(), sbf_save_excels(), sbf_save_gpkg(), sbf_save_gpkgs(), sbf_save_number(), sbf_save_numbers(), sbf_save_object(), sbf_save_objects(), sbf_save_plot(), sbf_save_png(), sbf_save_spatial(), sbf_save_spatials(), sbf_save_string(), sbf_save_strings(), sbf_save_table(), sbf_save_window(), sbf_save_workbook()

---

sbf_save_datas *Save Data Frames*

Description

Save Data Frames

Usage

```
sbf_save_datas(
  sub = sbf_get_sub(),
  main = sbf_get_main(),
  env = parent.frame()
)
```

Arguments

sub	A string specifying the path to the sub folder (by default the current sub folder).
main	A string specifying the path to the main folder (by default the current main folder)
env	An environment.

Value

An invisible character vector of the paths to the saved objects.

See Also

Other save functions: sbf_basename_sans_ext(), sbf_save_aws_files(), sbf_save_block(), sbf_save_data(), sbf_save_data_to_db(), sbf_save_data_to_pg(), sbf_save_datas_to_db(), sbf_save_db_metatable_descriptions(), sbf_save_db_to_workbook(), sbf_save_excel(), sbf_save_excels(), sbf_save_gpkg(), sbf_save_gpkgs(), sbf_save_number(), sbf_save_numbers(), sbf_save_object(), sbf_save_objects(), sbf_save_plot(), sbf_save_png(), sbf_save_spatial(), sbf_save_spatials(), sbf_save_string(), sbf_save_strings(), sbf_save_table(), sbf_save_window(), sbf_save_workbook()

`sbf_save_datas_to_db` *Save Data Frames to Existing Database*

Description

Save Data Frames to Existing Database

Usage

```
sbf_save_datas_to_db(
  db_name = sbf_get_db_name(),
  sub = sbf_get_sub(),
  main = sbf_get_main(),
  commit = TRUE,
  strict = TRUE,
  env = parent.frame(),
  silent = getOption("rws.silent", FALSE)
)
```

Arguments

db_name	A string of the database name.
sub	A string specifying the path to the sub folder (by default the current sub folder).
main	A string specifying the path to the main folder (by default the current main folder)
commit	A flag specifying whether to commit the operations (calling with commit = FALSE can be useful for checking data).
strict	A flag specifying whether to error if x has extraneous columns or if exists = TRUE extraneous data frames.
env	An environment.
silent	A flag specifying whether to suppress messages and warnings.

Value

An invisible character vector of the paths to the saved objects.

See Also

Other save functions: `sbf_basename_sans_ext()`, `sbf_save_aws_files()`, `sbf_save_block()`, `sbf_save_data()`, `sbf_save_data_to_db()`, `sbf_save_data_to_pg()`, `sbf_save_datas()`, `sbf_save_db_metatable_c`
`sbf_save_db_to_workbook()`, `sbf_save_excel()`, `sbf_save_excels()`, `sbf_save_gpkg()`, `sbf_save_gpkgs()`, `sbf_save_number()`, `sbf_save_numbers()`, `sbf_save_object()`, `sbf_save_objects()`, `sbf_save_plot()`, `sbf_save_png()`, `sbf_save_spatial()`, `sbf_save_spatials()`, `sbf_save_string()`, `sbf_save_strings()`, `sbf_save_table()`, `sbf_save_window()`, `sbf_save_workbook()`

sbf_save_data_to_db *Save Data Frame to Existing Database*

Description

Save Data Frame to Existing Database

Usage

```
sbf_save_data_to_db(
  x,
  x_name = substitute(x),
  db_name = sbf_get_db_name(),
  sub = sbf_get_sub(),
  main = sbf_get_main(),
  commit = TRUE,
  strict = TRUE,
  silent = getOption("rws.silent", FALSE)
)
```

Arguments

x	The object to save.
x_name	A string of the table name.
db_name	A string of the database name.
sub	A string specifying the path to the sub folder (by default the current sub folder).
main	A string specifying the path to the main folder (by default the current main folder)
commit	A flag specifying whether to commit the operations (calling with commit = FALSE can be useful for checking data).
strict	A flag specifying whether to error if x has extraneous columns or if exists = TRUE extraneous data frames.
silent	A flag specifying whether to suppress messages and warnings.

Value

An invisible character vector of the paths to the saved objects.

See Also

Other save functions: sbf_basename_sans_ext(), sbf_save_aws_files(), sbf_save_block(), sbf_save_data(), sbf_save_data_to_pg(), sbf_save_datas(), sbf_save_datas_to_db(), sbf_save_db_metatable_ sbf_save_db_to_workbook(), sbf_save_excel(), sbf_save_excels(), sbf_save_gpkg(), sbf_save_gpkgs(), sbf_save_number(), sbf_save_numbers(), sbf_save_object(), sbf_save_objects(), sbf_save_plot(), sbf_save_png(), sbf_save_spatial(), sbf_save_spatials(), sbf_save_string(), sbf_save_strings(), sbf_save_table(), sbf_save_window(), sbf_save_workbook()

`sbf_save_data_to_pg` *Add data frame to PostgreSQL database*

Description

[Deprecated]

`sbf_save_data_to_pg()` was moved to `subfoldr2ext::sbfx_save_data_to_pg()`.

Add data with a data frame to your PostgreSQL database. The data frame name must match the table name in your database, if not use the `tbl_name` argument to pass the table name.

Usage

```
sbf_save_data_to_pg(
  x,
  x_name = NULL,
  schema = getOption("psql.schema", "public"),
  config_path = getOption("psql.config_path", NULL),
  config_value = getOption("psql.config_value", "default")
)
```

Arguments

`x`	The data frame to save.
`x_name`	A string of the name.
`schema`	A string of the schema name. Default value is `"public"`.
`config_path`	A string of a file path to the yaml configuration file. The default value grabs the file path from the psql.config_path option and uses `NULL` if no value supplied.
`config_value`	A string of the name of value. The default value grabs the value from the psql.config_value option and uses `"default"` if no value is supplied.

Details

Wrapper on `psql::psql_add_data()`

Value

A scalar numeric.

See Also

Other postgresql functions: `sbf_backup_pg()`, `sbf_close_pg()`, `sbf_create_pg()`, `sbf_execute_pg()`, `sbf_get_config_file()`, `sbf_get_config_value()`, `sbf_get_schema()`, `sbf_list_tables_pg()`, `sbf_load_data_from_pg()`, `sbf_load_datas_from_pg()`, `sbf_open_pg()`, `sbf_reset_config_file()`, `sbf_reset_config_value()`, `sbf_reset_schema()`, `sbf_set_config_file()`, `sbf_set_config_value()`, `sbf_set_schema()`

Other save functions: sbf_basename_sans_ext(), sbf_save_aws_files(), sbf_save_block(), sbf_save_data(), sbf_save_data_to_db(), sbf_save_datas(), sbf_save_datas_to_db(), sbf_save_db_metatable_descriptions(), sbf_save_db_to_workbook(), sbf_save_excel(), sbf_save_excels(), sbf_save_gpkg(), sbf_save_gpkgs(), sbf_save_number(), sbf_save_numbers(), sbf_save_object(), sbf_save_objects(), sbf_save_plot(), sbf_save_png(), sbf_save_spatial(), sbf_save_spatials(), sbf_save_string(), sbf_save_strings(), sbf_save_table(), sbf_save_window(), sbf_save_workbook()

Examples

```
## Not run:
sbf_save_data_to_pg(outing, "creel")
sbf_save_data_to_pg(outing_new, "creel", "outing")

## End(Not run)
```

---

sbf_save_db_metatable_descriptions

Saves Meta Table Descriptions to Database

Description

Saves meta table descriptions to a database. Its important to note that if overwrite = TRUE and x includes blank descriptions then existing non-blank descriptions will be overwritten.

Usage

```
sbf_save_db_metatable_descriptions(
  x,
  db_name = sbf_get_db_name(),
  sub = sbf_get_sub(),
  main = sbf_get_main(),
  overwrite = FALSE,
  strict = TRUE
)
```

Arguments

x	A data.frame with Table, Column and Description columns.
db_name	A string of the database name.
sub	A string specifying the path to the sub folder (by default the current sub folder).
main	A string specifying the path to the main folder (by default the current main folder)
overwrite	A flag specifying whether to overwrite existing descriptions.
strict	A flag specifying whether to error if x has extraneous descriptions.

Value

A invisible data.frame of the altered descriptions.

See Also

Other save functions: sbf_basename_sans_ext(), sbf_save_aws_files(), sbf_save_block(), sbf_save_data(), sbf_save_data_to_db(), sbf_save_data_to_pg(), sbf_save_datas(), sbf_save_datas_to_db(), sbf_save_db_to_workbook(), sbf_save_excel(), sbf_save_excels(), sbf_save_gpkg(), sbf_save_gpkgs(), sbf_save_number(), sbf_save_numbers(), sbf_save_object(), sbf_save_objects(), sbf_save_plot(), sbf_save_png(), sbf_save_spatial(), sbf_save_spatials(), sbf_save_string(), sbf_save_strings(), sbf_save_table(), sbf_save_window(), sbf_save_workbook()

---

sbf_save_db_to_workbook

Save Database to Excel Workbook

Description

Converts a database to an single excel workbook where each table is its own spreadsheet.

Usage

```
sbf_save_db_to_workbook(
  workbook_name = sbf_get_workbook_name(),
  db_name = sbf_get_db_name(),
  exclude_tables = "^$",
  sub = sbf_get_sub(),
  main = sbf_get_main(),
  epgs = NULL
)
```

Arguments

workbook_name	The name of the excel workbook you are creating. Default is the base name of the current working directory.
db_name	A string of the database name.
exclude_tables	A regular expression listing tables to be excluded.
sub	A string specifying the path to the sub folder (by default the current sub folder).
main	A string specifying the path to the main folder (by default the current main folder)
epgs	The projection to convert to

See Also

Other excel: sbf_get_workbook_name(), sbf_save_excel(), sbf_save_excels(), sbf_save_workbook()

Other save functions: sbf_basename_sans_ext(), sbf_save_aws_files(), sbf_save_block(), sbf_save_data(), sbf_save_data_to_db(), sbf_save_data_to_pg(), sbf_save_datas(), sbf_save_datas_to_db(), sbf_save_db_metatable_descriptions(), sbf_save_excel(), sbf_save_excels(), sbf_save_gpkg(), sbf_save_gpkgs(), sbf_save_number(), sbf_save_numbers(), sbf_save_object(), sbf_save_objects(), sbf_save_plot(), sbf_save_png(), sbf_save_spatial(), sbf_save_spatials(), sbf_save_string(), sbf_save_strings(), sbf_save_table(), sbf_save_window(), sbf_save_workbook()

Examples

```
## Not run:
sbf_save_db_to_workbook()

# exclude the sites table
sbf_save_db_to_workbook(exclude_tables = "sites")

# exclude the sites and species table
sbf_save_db_to_workbook(exclude_tables = "sites|species")

## End(Not run)
```

---

sbf_save_excel *Save Dataframe to Excel Workbook*

Description

Save Dataframe to Excel Workbook

Usage

```
sbf_save_excel(
  x,
  x_name = substitute(x),
  max_sheets = 1L,
  sub = sbf_get_sub(),
  main = sbf_get_main(),
  epgs = NULL
)
```

Arguments

x	The data frame to save.
x_name	A string of the name.
max_sheets	An integer specifying the maximum number of sheets to split your table into for writing to excel. The default is 1.

sub	A string specifying the path to the sub folder (by default the current sub folder).
main	A string specifying the path to the main folder (by default the current main folder)
epgs	The projection to convert to

Details

This takes a data frame and saves it to their own excel workbook.

This function will split up large dataframes into smaller tables for writing to excel because excel only allows a maximum number of 1,048,576. For the max_sheets argument you can pass a number higher then the required and it will only return as many sheets as there is data.

Value

An invisible string of the path to the saved data.frame

See Also

Other save functions: sbf_basename_sans_ext(), sbf_save_aws_files(), sbf_save_block(), sbf_save_data(), sbf_save_data_to_db(), sbf_save_data_to_pg(), sbf_save_datas(), sbf_save_datas_to_db(), sbf_save_db_metatable_descriptions(), sbf_save_db_to_workbook(), sbf_save_excels(), sbf_save_gpkg(), sbf_save_gpkgs(), sbf_save_number(), sbf_save_numbers(), sbf_save_object(), sbf_save_objects(), sbf_save_plot(), sbf_save_png(), sbf_save_spatial(), sbf_save_spatials(), sbf_save_string(), sbf_save_strings(), sbf_save_table(), sbf_save_window(), sbf_save_workbook()

Other excel: sbf_get_workbook_name(), sbf_save_db_to_workbook(), sbf_save_excels(), sbf_save_workbook()

Examples

```
## Not run:
sbf_save_excel()

## End(Not run)
```

---

sbf_save_excels *Save Excels*

Description

Saves data frames from the environment to their own excel workbook. Each table will be its own excel workbook.

Usage

```
sbf_save_excels(
  sub = sbf_get_sub(),
  main = sbf_get_main(),
  env = parent.frame(),
  epgs = NULL
)
```

Arguments

sub	A string specifying the path to the sub folder (by default the current sub folder).
main	A string specifying the path to the main folder (by default the current main folder)
env	An environment.
epgs	The projection to convert to

Value

An invisible string of the path to the saved data.frame

See Also

Other excel: sbf_get_workbook_name(), sbf_save_db_to_workbook(), sbf_save_excel(), sbf_save_workbook()

Other save functions: sbf_basename_sans_ext(), sbf_save_aws_files(), sbf_save_block(), sbf_save_data(), sbf_save_data_to_db(), sbf_save_data_to_pg(), sbf_save_datas(), sbf_save_datas_to_db(), sbf_save_db_metatable_descriptions(), sbf_save_db_to_workbook(), sbf_save_excel(), sbf_save_gpkg(), sbf_save_gpkgs(), sbf_save_number(), sbf_save_numbers(), sbf_save_object(), sbf_save_objects(), sbf_save_plot(), sbf_save_png(), sbf_save_spatial(), sbf_save_spatials(), sbf_save_string(), sbf_save_strings(), sbf_save_table(), sbf_save_window(), sbf_save_workbook()

Examples

```
## Not run:
sbf_save_excels()

## End(Not run)
```

---

sbf_save_flobs_from_db

Save flobs

Description

Saves and systematically renames all blobbed files by default (dir = NULL) to flobs sub directory corresponding to database using dbflobr::save_all_flobs().

Usage

```
sbf_save_flobs_from_db(
  db_name = sbf_get_db_name(),
  sub = sbf_get_sub(),
  main = sbf_get_main(),
  dir = NULL,
  dbflobr_sub = FALSE,
  replace = FALSE
)
```

Arguments

db_name	A string of the database name.
sub	A logical scalar specifying whether to save all existing files in a subdirectory of the same name (sub = TRUE) or all possible files in a subdirectory of the same name (sub = NA) or not nest files within a subdirectory (sub = FALSE).
main	A string specifying the path to the main folder (by default the current main folder)
dir	A string of the path to the directory to save the files in.
dbflobr_sub	A logical specifying whether to save all existing files in a subdirectory of the same name (dbflobr_sub = TRUE) or all possible files in a subdirectory of the same name (dbflobr_sub = NA) or not nest files within a subdirectory (dbflobr_sub = FALSE).
replace	A flag specifying whether to replace existing files. If sub = TRUE (or sub = NA) and replace = TRUE then all existing files within a subdirectory are deleted.

Value

An invisible named list of named vectors of the file names and new file names saved.

See Also

Other flob: sbf_add_blob_column_to_db(), sbf_upload_flobs_to_db()

---

sbf_save_gpkg *Save sf data frame to Geopackage*

Description

Save sf data frame to Geopackage

Usage

```
sbf_save_gpkg(
  x,
  x_name = substitute(x),
  sub = sbf_get_sub(),
  main = sbf_get_main()
)
```

Arguments

x	The sf data frame to save.
x_name	A string of the name.
sub	A string specifying the path to the sub folder (by default the current sub folder).
main	A string specifying the path to the main folder (by default the current main folder)

Details

This takes an sf data frame and saves as geopackage.

Value

An invisible string of the path to the saved geopackage

See Also

Other save functions: sbf_basename_sans_ext(), sbf_save_aws_files(), sbf_save_block(), sbf_save_data(), sbf_save_data_to_db(), sbf_save_data_to_pg(), sbf_save_datas(), sbf_save_datas_to_db(), sbf_save_db_metatable_descriptions(), sbf_save_db_to_workbook(), sbf_save_excel(), sbf_save_excels(), sbf_save_gpkgs(), sbf_save_number(), sbf_save_numbers(), sbf_save_object(), sbf_save_objects(), sbf_save_plot(), sbf_save_png(), sbf_save_spatial(), sbf_save_spatials(), sbf_save_string(), sbf_save_strings(), sbf_save_table(), sbf_save_window(), sbf_save_workbook()

Examples

```
## Not run:
sbf_save_gpkg()

## End(Not run)
```

sbf_save_gpkgs *Save sf data frames to Geopackages*

Description

An sf object of file name file_name is saved as file_name.gpkg.

Usage

```
sbf_save_gpkgs(
  sub = sbf_get_sub(),
  main = sbf_get_main(),
  env = parent.frame(),
  all_sfcs = TRUE
)
```

Arguments

sub	A string specifying the path to the sub folder (by default the current sub folder).
main	A string specifying the path to the main folder (by default the current main folder)
env	An environment.
all_sfcs	A flag specifying whether to save non-active sfc columns as geopackages.

Details

By default (all_sfcs = TRUE) non-active sfc columns are saved as file_name_geometry_column_name.gpkg this includes data frames with no active sfc column.

Value

An invisible character vector of the paths to the saved objects.

See Also

Other save functions: sbf_basename_sans_ext(), sbf_save_aws_files(), sbf_save_block(), sbf_save_data(), sbf_save_data_to_db(), sbf_save_data_to_pg(), sbf_save_datas(), sbf_save_datas_to_db(), sbf_save_db_metatable_descriptions(), sbf_save_db_to_workbook(), sbf_save_excel(), sbf_save_excels(), sbf_save_gpkg(), sbf_save_number(), sbf_save_numbers(), sbf_save_object(), sbf_save_objects(), sbf_save_plot(), sbf_save_png(), sbf_save_spatial(), sbf_save_spatials(), sbf_save_string(), sbf_save_strings(), sbf_save_table(), sbf_save_window(), sbf_save_workbook()

sbf_save_number *Save Number*

Description

Save Number

Usage

```
sbf_save_number(
  x,
  x_name = substitute(x),
  sub = sbf_get_sub(),
  main = sbf_get_main()
)
```

Arguments

x	The number to save.
x_name	A string of the name.
sub	A string specifying the path to the sub folder (by default the current sub folder).
main	A string specifying the path to the main folder (by default the current main folder)

Value

An invisible string of the path to the saved object.

See Also

Other save functions: sbf_basename_sans_ext(), sbf_save_aws_files(), sbf_save_block(), sbf_save_data(), sbf_save_data_to_db(), sbf_save_data_to_pg(), sbf_save_datas(), sbf_save_datas_to_db(), sbf_save_db_metatable_descriptions(), sbf_save_db_to_workbook(), sbf_save_excel(), sbf_save_excels(), sbf_save_gpkg(), sbf_save_gpkgs(), sbf_save_numbers(), sbf_save_object(), sbf_save_objects(), sbf_save_plot(), sbf_save_png(), sbf_save_spatial(), sbf_save_spatials(), sbf_save_string(), sbf_save_strings(), sbf_save_table(), sbf_save_window(), sbf_save_workbook()

sbf_save_numbers *Save Numbers*

Description

Save Numbers

Usage

```
sbf_save_numbers(
  sub = sbf_get_sub(),
  main = sbf_get_main(),
  env = parent.frame()
)
```

Arguments

sub	A string specifying the path to the sub folder (by default the current sub folder).
main	A string specifying the path to the main folder (by default the current main folder)
env	An environment.

Value

An invisible character vector of the paths to the saved objects.

See Also

Other save functions: sbf_basename_sans_ext(), sbf_save_aws_files(), sbf_save_block(), sbf_save_data(), sbf_save_data_to_db(), sbf_save_data_to_pg(), sbf_save_datas(), sbf_save_datas_to_db(), sbf_save_db_metatable_descriptions(), sbf_save_db_to_workbook(), sbf_save_excel(), sbf_save_excels(), sbf_save_gpkg(), sbf_save_gpkgs(), sbf_save_number(), sbf_save_object(), sbf_save_objects(), sbf_save_plot(), sbf_save_png(), sbf_save_spatial(), sbf_save_spatials(), sbf_save_string(), sbf_save_strings(), sbf_save_table(), sbf_save_window(), sbf_save_workbook()

sbf_save_object *Save Object*

Description

Save Object

Usage

```
sbf_save_object(
  x,
  x_name = substitute(x),
  sub = sbf_get_sub(),
  main = sbf_get_main()
)
```

Arguments

x	The object to save.
x_name	A string of the name.
sub	A string specifying the path to the sub folder (by default the current sub folder).
main	A string specifying the path to the main folder (by default the current main folder)

Value

An invisible string of the path to the saved object.

See Also

Other save functions: sbf_basename_sans_ext(), sbf_save_aws_files(), sbf_save_block(), sbf_save_data(), sbf_save_data_to_db(), sbf_save_data_to_pg(), sbf_save_datas(), sbf_save_datas_to_db(), sbf_save_db_metatable_descriptions(), sbf_save_db_to_workbook(), sbf_save_excel(), sbf_save_excels(), sbf_save_gpkg(), sbf_save_gpkgs(), sbf_save_number(), sbf_save_numbers(), sbf_save_objects(), sbf_save_plot(), sbf_save_png(), sbf_save_spatial(), sbf_save_spatials(), sbf_save_string(), sbf_save_strings(), sbf_save_table(), sbf_save_window(), sbf_save_workbook()

---

sbf_save_objects *Save Objects*

Description

Save Objects

Usage

```
sbf_save_objects(
  sub = sbf_get_sub(),
  main = sbf_get_main(),
  env = parent.frame()
)
```

Arguments

sub	A string specifying the path to the sub folder (by default the current sub folder).
main	A string specifying the path to the main folder (by default the current main folder)
env	An environment.

Value

An invisible character vector of the paths to the saved objects.

See Also

Other save functions: sbf_basename_sans_ext(), sbf_save_aws_files(), sbf_save_block(), sbf_save_data(), sbf_save_data_to_db(), sbf_save_data_to_pg(), sbf_save_datas(), sbf_save_datas_to_db(), sbf_save_db_metatable_descriptions(), sbf_save_db_to_workbook(), sbf_save_excel(), sbf_save_excels(), sbf_save_gpkg(), sbf_save_gpkgs(), sbf_save_number(), sbf_save_numbers(), sbf_save_object(), sbf_save_plot(), sbf_save_png(), sbf_save_spatial(), sbf_save_spatials(), sbf_save_string(), sbf_save_strings(), sbf_save_table(), sbf_save_window(), sbf_save_workbook()

---

sbf_save_plot *Save Plot*

Description

Saves a ggplot object. By default it saves the last plot to be modified or created.

Usage

```
sbf_save_plot(
  x = ggplot2::last_plot(),
  x_name = substitute(x),
  sub = sbf_get_sub(),
  main = sbf_get_main(),
  caption = "",
  report = TRUE,
  tag = "",
  units = "in",
  width = NA,
  height = width,
  dpi = 300,
  limitsize = TRUE,
  csv = 1000L
)
```

Arguments

x	The ggplot object to save.
x_name	A string of the name.
sub	A string specifying the path to the sub folder (by default the current sub folder).
main	A string specifying the path to the main folder (by default the current main folder)
caption	A string of the caption.
report	A flag specifying whether to include in a report.
tag	A string of the tag.
units	A string of the units. Can be "in" (default) or "mm" or "cm".
width	A number of the plot width in inches.
height	A number of the plot width in inches.
dpi	A number of the resolution in dots per inch.
limitsize	When TRUE (the default), ggsave() will not save images larger than 50x50 inches, to prevent the common error of specifying dimensions in pixels.
csv	A count specifying the maximum number of rows to save as a csv file.

See Also

Other save functions: sbf_basename_sans_ext(), sbf_save_aws_files(), sbf_save_block(), sbf_save_data(), sbf_save_data_to_db(), sbf_save_data_to_pg(), sbf_save_datas(), sbf_save_datas_to_db(), sbf_save_db_metatable_descriptions(), sbf_save_db_to_workbook(), sbf_save_excel(), sbf_save_excels(), sbf_save_gpkg(), sbf_save_gpkgs(), sbf_save_number(), sbf_save_numbers(), sbf_save_object(), sbf_save_objects(), sbf_save_png(), sbf_save_spatial(), sbf_save_spatials(), sbf_save_string(), sbf_save_strings(), sbf_save_table(), sbf_save_window(), sbf_save_workbook()

---

sbf_save_png *Save png File*

Description

Saves a png file to the windows.

Usage

```
sbf_save_png(
  x,
  x_name = sbf_basename_sans_ext(x),
  sub = sbf_get_sub(),
  main = sbf_get_main(),
  caption = "",
  report = TRUE,
  tag = "",
  width = NA,
  units = "in"
)
```

Arguments

x	A string of the path to the png file to save.
x_name	A string of the name.
sub	A string specifying the path to the sub folder (by default the current sub folder).
main	A string specifying the path to the main folder (by default the current main folder)
caption	A string of the caption.
report	A flag specifying whether to include in a report.
tag	A string of the tag.
width	A number of the plot width in inches.
units	A string of the units. Can be "in" (default) or "mm" or "cm".

See Also

Other save functions: sbf_basename_sans_ext(), sbf_save_aws_files(), sbf_save_block(), sbf_save_data(), sbf_save_data_to_db(), sbf_save_data_to_pg(), sbf_save_datas(), sbf_save_datas_to_db(), sbf_save_db_metatable_descriptions(), sbf_save_db_to_workbook(), sbf_save_excel(), sbf_save_excels(), sbf_save_gpkg(), sbf_save_gpkgs(), sbf_save_number(), sbf_save_numbers(), sbf_save_object(), sbf_save_objects(), sbf_save_plot(), sbf_save_spatial(), sbf_save_spatials(), sbf_save_string(), sbf_save_strings(), sbf_save_table(), sbf_save_window(), sbf_save_workbook()

---

sbf_save_spatial *Save Spatial Data*

Description

Saves an sf tbl with at least one row for which the first column (not a geometry) is unique with no missing values and only one geometry column which must have a defined projection.

Usage

```
sbf_save_spatial(x, x_name = NULL, sub = sbf_get_sub(), main = sbf_get_main())
```

Arguments

x	The sf tbl to save.
x_name	A string of the name.
sub	A string specifying the path to the sub folder (by default the current sub folder).
main	A string specifying the path to the main folder (by default the current main folder)

Value

An invisible string of the path to the saved data.frame

See Also

Other save functions: sbf_basename_sans_ext(), sbf_save_aws_files(), sbf_save_block(), sbf_save_data(), sbf_save_data_to_db(), sbf_save_data_to_pg(), sbf_save_datas(), sbf_save_datas_to_db(), sbf_save_db_metatable_descriptions(), sbf_save_db_to_workbook(), sbf_save_excel(), sbf_save_excels(), sbf_save_gpkg(), sbf_save_gpkgs(), sbf_save_number(), sbf_save_numbers(), sbf_save_object(), sbf_save_objects(), sbf_save_plot(), sbf_save_png(), sbf_save_spatials(), sbf_save_string(), sbf_save_strings(), sbf_save_table(), sbf_save_window(), sbf_save_workbook()

---

sbf_save_spatials *Save Spatial Data Frames*

Description

Saves sf tbls each with at least one row for which the first column (not a geometry) is unique with no missing values and only one geometry column which must have a defined projection. The functions expects that all data frames in the environment meet these requirements.

Usage

```
sbf_save_spatials(
  sub = sbf_get_sub(),
  main = sbf_get_main(),
  env = parent.frame()
)
```

Arguments

sub	A string specifying the path to the sub folder (by default the current sub folder).
main	A string specifying the path to the main folder (by default the current main folder)
env	An environment.

Value

An invisible character vector of the paths to the saved objects.

See Also

Other save functions: sbf_basename_sans_ext(), sbf_save_aws_files(), sbf_save_block(), sbf_save_data(), sbf_save_data_to_db(), sbf_save_data_to_pg(), sbf_save_datas(), sbf_save_datas_to_db(), sbf_save_db_metatable_descriptions(), sbf_save_db_to_workbook(), sbf_save_excel(), sbf_save_excels(), sbf_save_gpkg(), sbf_save_gpkgs(), sbf_save_number(), sbf_save_numbers(), sbf_save_object(), sbf_save_objects(), sbf_save_plot(), sbf_save_png(), sbf_save_spatial(), sbf_save_string(), sbf_save_strings(), sbf_save_table(), sbf_save_window(), sbf_save_workbook()

sbf_save_string *Save String*

Description

A string in this context is a character vector of length one of inline text.

Usage

```
sbf_save_string(
  x,
  x_name = substitute(x),
  sub = sbf_get_sub(),
  main = sbf_get_main(),
  report = TRUE,
  tag = ""
)
```

Arguments

x	The string to save.
x_name	A string of the name.
sub	A string specifying the path to the sub folder (by default the current sub folder).
main	A string specifying the path to the main folder (by default the current main folder)
report	A flag specifying whether to include in a report.
tag	A string of the tag.

Value

An invisible string of the path to the saved object.

See Also

Other save functions: sbf_basename_sans_ext(), sbf_save_aws_files(), sbf_save_block(), sbf_save_data(), sbf_save_data_to_db(), sbf_save_data_to_pg(), sbf_save_datas(), sbf_save_datas_to_db(), sbf_save_db_metatable_descriptions(), sbf_save_db_to_workbook(), sbf_save_excel(), sbf_save_excels(), sbf_save_gpkg(), sbf_save_gpkgs(), sbf_save_number(), sbf_save_numbers(), sbf_save_object(), sbf_save_objects(), sbf_save_plot(), sbf_save_png(), sbf_save_spatial(), sbf_save_spatials(), sbf_save_strings(), sbf_save_table(), sbf_save_window(), sbf_save_workbook()

sbf_save_strings *Save Strings*

Description

Save Strings

Usage

```
sbf_save_strings(
  sub = sbf_get_sub(),
  main = sbf_get_main(),
  env = parent.frame()
)
```

Arguments

sub	A string specifying the path to the sub folder (by default the current sub folder).
main	A string specifying the path to the main folder (by default the current main folder)
env	An environment.

Value

An invisible character vector of the paths to the saved objects.

See Also

Other save functions: sbf_basename_sans_ext(), sbf_save_aws_files(), sbf_save_block(), sbf_save_data(), sbf_save_data_to_db(), sbf_save_data_to_pg(), sbf_save_datas(), sbf_save_datas_to_db(), sbf_save_db_metatable_descriptions(), sbf_save_db_to_workbook(), sbf_save_excel(), sbf_save_excels(), sbf_save_gpkg(), sbf_save_gpkgs(), sbf_save_number(), sbf_save_numbers(), sbf_save_object(), sbf_save_objects(), sbf_save_plot(), sbf_save_png(), sbf_save_spatial(), sbf_save_spatials(), sbf_save_string(), sbf_save_table(), sbf_save_window(), sbf_save_workbook()

sbf_save_table *Save Table*

Description

Save Table

Usage

```
sbf_save_table(
  x,
  x_name = substitute(x),
  sub = sbf_get_sub(),
  main = sbf_get_main(),
  caption = "",
  report = TRUE,
  tag = ""
)
```

Arguments

x	The data frame to save.
x_name	A string of the name.
sub	A string specifying the path to the sub folder (by default the current sub folder).
main	A string specifying the path to the main folder (by default the current main folder)
caption	A string of the caption.
report	A flag specifying whether to include in a report.
tag	A string of the tag.

Value

An invisible string of the path to the saved object.

See Also

Other save functions: sbf_basename_sans_ext(), sbf_save_aws_files(), sbf_save_block(), sbf_save_data(), sbf_save_data_to_db(), sbf_save_data_to_pg(), sbf_save_datas(), sbf_save_datas_to_db(), sbf_save_db_metatable_descriptions(), sbf_save_db_to_workbook(), sbf_save_excel(), sbf_save_excels(), sbf_save_gpkg(), sbf_save_gpkgs(), sbf_save_number(), sbf_save_numbers(), sbf_save_object(), sbf_save_objects(), sbf_save_plot(), sbf_save_png(), sbf_save_spatial(), sbf_save_spatials(), sbf_save_string(), sbf_save_strings(), sbf_save_window(), sbf_save_workbook()

---

sbf_save_window *Save Window*

Description

Saves the current graphics device to a png file.

Usage

```
sbf_save_window(
  x_name = "window",
  sub = sbf_get_sub(),
  main = sbf_get_main(),
  caption = "",
  report = TRUE,
  tag = "",
  width = NA,
  height = width,
  units = "in",
  dpi = 300
)
```

Arguments

x_name	A string of the name.
sub	A string specifying the path to the sub folder (by default the current sub folder).
main	A string specifying the path to the main folder (by default the current main folder)
caption	A string of the caption.
report	A flag specifying whether to include in a report.
tag	A string of the tag.
width	A number of the plot width in inches.
height	A number of the plot width in inches.
units	A string of the units. Can be "in" (default) or "mm" or "cm".
dpi	A number of the resolution in dots per inch.

See Also

Other save functions: sbf_basename_sans_ext(), sbf_save_aws_files(), sbf_save_block(), sbf_save_data(), sbf_save_data_to_db(), sbf_save_data_to_pg(), sbf_save_datas(), sbf_save_datas_to_db(), sbf_save_db_metatable_descriptions(), sbf_save_db_to_workbook(), sbf_save_excel(), sbf_save_excels(), sbf_save_gpkg(), sbf_save_gpkgs(), sbf_save_number(), sbf_save_numbers(), sbf_save_object(), sbf_save_objects(), sbf_save_plot(), sbf_save_png(), sbf_save_spatial(), sbf_save_spatials(), sbf_save_string(), sbf_save_strings(), sbf_save_table(), sbf_save_workbook()

---

sbf_save_workbook *Save Dataframes to Excel Workbook*

Description

This takes the data frames from the environment and saves them to a single excel workbook where each table is its own spreadsheet.

Usage

```
sbf_save_workbook(
  workbook_name = basename(getwd()),
  sub = sbf_get_sub(),
  main = sbf_get_main(),
  env = parent.frame(),
  epgs = NULL
)
```

Arguments

workbook_name	The name of the excel workbook you are creating. Default is the base name of the current working directory.
sub	A string specifying the path to the sub folder (by default the current sub folder).
main	A string specifying the path to the main folder (by default the current main folder)
env	An environment.
epgs	The projection to convert to

Value

An invisible string of the path to the saved data.frame

See Also

Other excel: sbf_get_workbook_name(), sbf_save_db_to_workbook(), sbf_save_excel(), sbf_save_excels()

Other save functions: sbf_basename_sans_ext(), sbf_save_aws_files(), sbf_save_block(), sbf_save_data(), sbf_save_data_to_db(), sbf_save_data_to_pg(), sbf_save_datas(), sbf_save_datas_to_db(), sbf_save_db_metatable_descriptions(), sbf_save_db_to_workbook(), sbf_save_excel(), sbf_save_excels(), sbf_save_gpkg(), sbf_save_gpkgs(), sbf_save_number(), sbf_save_numbers(), sbf_save_object(), sbf_save_objects(), sbf_save_plot(), sbf_save_png(), sbf_save_spatial(), sbf_save_spatials(), sbf_save_string(), sbf_save_strings(), sbf_save_table(), sbf_save_window()

Examples

```
## Not run:
sbf_save_workbook()

## End(Not run)
```

sbf_set_config_file *Set the Config File path*

Description

[Deprecated]

`sbf_set_config_file()` was moved to `subfoldr2ext::sbfx_set_config_file()`.

A wrapper to quickly set the `psql.config_path` options parameter.

Usage

```
sbf_set_config_file(path = "config.yml")
```

Arguments

`path`	A file path to the location of the yaml file containing your connection details.

Details

This function is recommended to be added to your header when used.

Value

An invisible string of the file path given

See Also

Other postgresql functions: `sbf_backup_pg()`, `sbf_close_pg()`, `sbf_create_pg()`, `sbf_execute_pg()`, `sbf_get_config_file()`, `sbf_get_config_value()`, `sbf_get_schema()`, `sbf_list_tables_pg()`, `sbf_load_data_from_pg()`, `sbf_load_datas_from_pg()`, `sbf_open_pg()`, `sbf_reset_config_file()`, `sbf_reset_config_value()`, `sbf_reset_schema()`, `sbf_save_data_to_pg()`, `sbf_set_config_value()`, `sbf_set_schema()`

Examples

```
## Not run:
sbf_set_config_file()
sbf_set_config_file("Keys/config-captures.yml")

## End(Not run)
```

`sbf_set_config_value` *Set the Config Value*

Description

[Deprecated]

`sbf_set_config_value()` was moved to `subfoldr2ext::sbfx_set_config_value()`.

Wrapper for setting the `psql.config_value` options parameter.

Usage

```
sbf_set_config_value(value = NULL)
```

Arguments

`value`	A string of the config file value to grab.

Details

This function is recommended to be added to your header when used.

Value

An invisible string of the value given

See Also

Other postgresql functions: `sbf_backup_pg()`, `sbf_close_pg()`, `sbf_create_pg()`, `sbf_execute_pg()`, `sbf_get_config_file()`, `sbf_get_config_value()`, `sbf_get_schema()`, `sbf_list_tables_pg()`, `sbf_load_data_from_pg()`, `sbf_load_datas_from_pg()`, `sbf_open_pg()`, `sbf_reset_config_file()`, `sbf_reset_config_value()`, `sbf_reset_schema()`, `sbf_save_data_to_pg()`, `sbf_set_config_file()`, `sbf_set_schema()`

Examples

```
## Not run:
sbf_set_config_value("shinyapp")

## End(Not run)
```

sbf_set_db_name *Set Database Name*

Description

Sets database name option (without the extension or path).

Usage

```
sbf_set_db_name(db_name = "database")
```

Arguments

db_name	A string specifying the new database name (without the extension or path).

Value

An invisible string specifying the new database name (without the extension or path).

See Also

Other db_name: sbf_get_db_name()

Other directory functions: sbf_add_sub(), sbf_get_archive(), sbf_get_db_name(), sbf_get_main(), sbf_get_sub(), sbf_get_workbook_name(), sbf_reset(), sbf_reset_db_name(), sbf_reset_main(), sbf_reset_sub(), sbf_set_main(), sbf_set_sub(), sbf_up_sub()

Examples

```
sbf_set_db_name("database")
```

sbf_set_main *Set Main*

Description

The directory is created when needed if it doesn't already exist.

Usage

```
sbf_set_main(..., rm = FALSE, ask = getOption("sbf.ask", TRUE))
```

Arguments

...	One or more character vectors which are combined together.
rm	A flag specifying whether to remove the folder and all its contents if it already exists.
ask	A flag specifying whether to ask before removing the existing folder.

Value

An invisible string of the path to the main folder.

See Also

Other directory functions: sbf_add_sub(), sbf_get_archive(), sbf_get_db_name(), sbf_get_main(), sbf_get_sub(), sbf_get_workbook_name(), sbf_reset(), sbf_reset_db_name(), sbf_reset_main(), sbf_reset_sub(), sbf_set_db_name(), sbf_set_sub(), sbf_up_sub()

---

sbf_set_schema *Set Schema Name*

Description

[Deprecated]

`sbf_set_schema()` was moved to `subfoldr2ext::sbfx_set_schema()`.

Usage

```
sbf_set_schema(schema = "public")
```

Arguments

schema	A string of the schema name. Default value is "public".

Value

An invisible schema name

See Also

Other postgresql functions: sbf_backup_pg(), sbf_close_pg(), sbf_create_pg(), sbf_execute_pg(), sbf_get_config_file(), sbf_get_config_value(), sbf_get_schema(), sbf_list_tables_pg(), sbf_load_data_from_pg(), sbf_load_datas_from_pg(), sbf_open_pg(), sbf_reset_config_file(), sbf_reset_config_value(), sbf_reset_schema(), sbf_save_data_to_pg(), sbf_set_config_file(), sbf_set_config_value()

Examples

```
## Not run:
sbf_set_schema("capture")

## End(Not run)
```

sbf_set_sub *Set Sub Folder*

Description

Set Sub Folder

Usage

```
sbf_set_sub(..., rm = FALSE, ask = getOption("sbf.ask", TRUE))
```

Arguments

...	One or more character vectors which are combined together.
rm	A flag specifying whether to remove the folder and all its contents if it already exists.
ask	A flag specifying whether to ask before removing the existing folder.

Value

An invisible string specifying the new sub folder.

See Also

Other directory functions: sbf_add_sub(), sbf_get_archive(), sbf_get_db_name(), sbf_get_main(), sbf_get_sub(), sbf_get_workbook_name(), sbf_reset(), sbf_reset_db_name(), sbf_reset_main(), sbf_reset_sub(), sbf_set_db_name(), sbf_set_main(), sbf_up_sub()

Examples

```
sbf_set_sub("nameofsub")
sbf_get_sub()
sbf_reset_sub()
```

sbf_string_exists *String Exists*

Description

this function is now deprecated as of version 0.0.0.9045

Usage

```
sbf_string_exists(x_name, sub = sbf_get_sub(), main = sbf_get_main())
```

Arguments

x_name	A string of the name.
sub	A string specifying the path to the sub folder (by default the current sub folder).
main	A string specifying the path to the main folder (by default the current main folder)

Value

A flag specifying whether the string exists.

See Also

Other exists functions: sbf_block_exists(), sbf_data_exists(), sbf_number_exists(), sbf_object_exists(), sbf_plot_exists(), sbf_table_exists()

---

sbf_subs_block_recursive

Gets Subs of a Block as a Character Vector

Description

Recursively returns all the subs of block with name x_name.

Usage

```
sbf_subs_block_recursive(
  x_name,
  sub = sbf_get_sub(),
  main = sbf_get_main(),
  include_root = TRUE
)
```

Arguments

x_name	A string of the name.
sub	A string specifying the path to the sub folder (by default the current sub folder).
main	A string specifying the path to the main folder (by default the current main folder)
include_root	A flag indicating whether to include objects in the top sub folder.

See Also

Other load functions: sbf_load_block(), sbf_load_blocks(), sbf_load_blocks_recursive(), sbf_load_data(), sbf_load_data_from_db(), sbf_load_data_from_pg(), sbf_load_datas(), sbf_load_datas_from_db(), sbf_load_datas_from_pg(), sbf_load_datas_recursive(), sbf_load_db_metatable(), sbf_load_number(), sbf_load_numbers(), sbf_load_numbers_recursive(), sbf_load_object(), sbf_load_objects(), sbf_load_objects_recursive(), sbf_load_plot(), sbf_load_plot_data(), sbf_load_plots_data(), sbf_load_plots_data_recursive(), sbf_load_plots_recursive(), sbf_load_spatial(), sbf_load_spatials(), sbf_load_string(), sbf_load_strings(), sbf_load_strings_recursi, sbf_load_table(), sbf_load_tables(), sbf_load_tables_recursive(), sbf_load_windows_recursive(), sbf_subs_data_recursive(), sbf_subs_number_recursive(), sbf_subs_object_recursive(), sbf_subs_plot_recursive(), sbf_subs_string_recursive(), sbf_subs_table_recursive(), sbf_subs_window_recursive()

`sbf_subs_data_recursive`

Gets Subs of a Data Frame as a Character Vector

Description

Recursively returns all the subs of data frames of objects with name x_name.

Usage

```
sbf_subs_data_recursive(
  x_name,
  sub = sbf_get_sub(),
  main = sbf_get_main(),
  include_root = TRUE
)
```

Arguments

x_name	A string of the name.
sub	A string specifying the path to the sub folder (by default the current sub folder).
main	A string specifying the path to the main folder (by default the current main folder)
include_root	A flag indicating whether to include objects in the top sub folder.

See Also

Other load functions: sbf_load_block(), sbf_load_blocks(), sbf_load_blocks_recursive(), sbf_load_data(), sbf_load_data_from_db(), sbf_load_data_from_pg(), sbf_load_datas(), sbf_load_datas_from_db(), sbf_load_datas_from_pg(), sbf_load_datas_recursive(), sbf_load_db_metatable(), sbf_load_number(), sbf_load_numbers(), sbf_load_numbers_recursive(), sbf_load_object(), sbf_load_objects(), sbf_load_objects_recursive(), sbf_load_plot(), sbf_load_plot_data(), sbf_load_plots_data(), sbf_load_plots_data_recursive(), sbf_load_plots_recursive(),

sbf_load_spatial(), sbf_load_spatials(), sbf_load_string(), sbf_load_strings(), sbf_load_strings_recursi
sbf_load_table(), sbf_load_tables(), sbf_load_tables_recursive(), sbf_load_windows_recursive(),
sbf_subs_block_recursive(), sbf_subs_number_recursive(), sbf_subs_object_recursive(),
sbf_subs_plot_recursive(), sbf_subs_string_recursive(), sbf_subs_table_recursive(),
sbf_subs_window_recursive()

---

sbf_subs_number_recursive

Gets Subs of a Number as a Character Vector

Description

Recursively returns all the subs of number with name x_name.

Usage

```
sbf_subs_number_recursive(
  x_name,
  sub = sbf_get_sub(),
  main = sbf_get_main(),
  include_root = TRUE
)
```

Arguments

x_name	A string of the name.
sub	A string specifying the path to the sub folder (by default the current sub folder).
main	A string specifying the path to the main folder (by default the current main folder)
include_root	A flag indicating whether to include objects in the top sub folder.

See Also

Other load functions: sbf_load_block(), sbf_load_blocks(), sbf_load_blocks_recursive(),
sbf_load_data(), sbf_load_data_from_db(), sbf_load_data_from_pg(), sbf_load_datas(),
sbf_load_datas_from_db(), sbf_load_datas_from_pg(), sbf_load_datas_recursive(), sbf_load_db_metatable()
sbf_load_number(), sbf_load_numbers(), sbf_load_numbers_recursive(), sbf_load_object(),
sbf_load_objects(), sbf_load_objects_recursive(), sbf_load_plot(), sbf_load_plot_data(),
sbf_load_plots_data(), sbf_load_plots_data_recursive(), sbf_load_plots_recursive(),
sbf_load_spatial(), sbf_load_spatials(), sbf_load_string(), sbf_load_strings(), sbf_load_strings_recursi
sbf_load_table(), sbf_load_tables(), sbf_load_tables_recursive(), sbf_load_windows_recursive(),
sbf_subs_block_recursive(), sbf_subs_data_recursive(), sbf_subs_object_recursive(),
sbf_subs_plot_recursive(), sbf_subs_string_recursive(), sbf_subs_table_recursive(),
sbf_subs_window_recursive()

sbf_subs_object_recursive *Gets Subs of an Object as a Character Vector*

Description

Recursively returns all the subs of objects with name x_name.

Usage

```
sbf_subs_object_recursive(
  x_name,
  sub = sbf_get_sub(),
  main = sbf_get_main(),
  include_root = TRUE
)
```

Arguments

x_name	A string of the name.
sub	A string specifying the path to the sub folder (by default the current sub folder).
main	A string specifying the path to the main folder (by default the current main folder)
include_root	A flag indicating whether to include objects in the top sub folder.

See Also

Other load functions: sbf_load_block(), sbf_load_blocks(), sbf_load_blocks_recursive(), sbf_load_data(), sbf_load_data_from_db(), sbf_load_data_from_pg(), sbf_load_datas(), sbf_load_datas_from_db(), sbf_load_datas_from_pg(), sbf_load_datas_recursive(), sbf_load_db_metatable(), sbf_load_number(), sbf_load_numbers(), sbf_load_numbers_recursive(), sbf_load_object(), sbf_load_objects(), sbf_load_objects_recursive(), sbf_load_plot(), sbf_load_plot_data(), sbf_load_plots_data(), sbf_load_plots_data_recursive(), sbf_load_plots_recursive(), sbf_load_spatial(), sbf_load_spatials(), sbf_load_string(), sbf_load_strings(), sbf_load_strings_recursi, sbf_load_table(), sbf_load_tables(), sbf_load_tables_recursive(), sbf_load_windows_recursive(), sbf_subs_block_recursive(), sbf_subs_data_recursive(), sbf_subs_number_recursive(), sbf_subs_plot_recursive(), sbf_subs_string_recursive(), sbf_subs_table_recursive(), sbf_subs_window_recursive()

sbf_subs_plot_recursive *Gets Subs of a Plot as a Character Vector*

Description

Recursively returns all the subs of plot with name x_name.

Usage

```
sbf_subs_plot_recursive(
  x_name,
  sub = sbf_get_sub(),
  main = sbf_get_main(),
  include_root = TRUE
)
```

Arguments

x_name	A string of the name.
sub	A string specifying the path to the sub folder (by default the current sub folder).
main	A string specifying the path to the main folder (by default the current main folder)
include_root	A flag indicating whether to include objects in the top sub folder.

See Also

Other load functions: sbf_load_block(), sbf_load_blocks(), sbf_load_blocks_recursive(), sbf_load_data(), sbf_load_data_from_db(), sbf_load_data_from_pg(), sbf_load_datas(), sbf_load_datas_from_db(), sbf_load_datas_from_pg(), sbf_load_datas_recursive(), sbf_load_db_metatable(), sbf_load_number(), sbf_load_numbers(), sbf_load_numbers_recursive(), sbf_load_object(), sbf_load_objects(), sbf_load_objects_recursive(), sbf_load_plot(), sbf_load_plot_data(), sbf_load_plots_data(), sbf_load_plots_data_recursive(), sbf_load_plots_recursive(), sbf_load_spatial(), sbf_load_spatials(), sbf_load_string(), sbf_load_strings(), sbf_load_strings_recursi, sbf_load_table(), sbf_load_tables(), sbf_load_tables_recursive(), sbf_load_windows_recursive(), sbf_subs_block_recursive(), sbf_subs_data_recursive(), sbf_subs_number_recursive(), sbf_subs_object_recursive(), sbf_subs_string_recursive(), sbf_subs_table_recursive(), sbf_subs_window_recursive()

sbf_subs_string_recursive *Gets Subs of a String as a Character Vector*

Description

Recursively returns all the subs of string with name x_name.

Usage

```
sbf_subs_string_recursive(
  x_name,
  sub = sbf_get_sub(),
  main = sbf_get_main(),
  include_root = TRUE
)
```

Arguments

x_name	A string of the name.
sub	A string specifying the path to the sub folder (by default the current sub folder).
main	A string specifying the path to the main folder (by default the current main folder)
include_root	A flag indicating whether to include objects in the top sub folder.

See Also

Other load functions: sbf_load_block(), sbf_load_blocks(), sbf_load_blocks_recursive(), sbf_load_data(), sbf_load_data_from_db(), sbf_load_data_from_pg(), sbf_load_datas(), sbf_load_datas_from_db(), sbf_load_datas_from_pg(), sbf_load_datas_recursive(), sbf_load_db_metatable(), sbf_load_number(), sbf_load_numbers(), sbf_load_numbers_recursive(), sbf_load_object(), sbf_load_objects(), sbf_load_objects_recursive(), sbf_load_plot(), sbf_load_plot_data(), sbf_load_plots_data(), sbf_load_plots_data_recursive(), sbf_load_plots_recursive(), sbf_load_spatial(), sbf_load_spatials(), sbf_load_string(), sbf_load_strings(), sbf_load_strings_recursi, sbf_load_table(), sbf_load_tables(), sbf_load_tables_recursive(), sbf_load_windows_recursive(), sbf_subs_block_recursive(), sbf_subs_data_recursive(), sbf_subs_number_recursive(), sbf_subs_object_recursive(), sbf_subs_plot_recursive(), sbf_subs_table_recursive(), sbf_subs_window_recursive()

sbf_subs_table_recursive *Gets Subs of a Table as a Character Vector*

Description

Recursively returns all the subs of table with name x_name.

Usage

```
sbf_subs_table_recursive(
  x_name,
  sub = sbf_get_sub(),
  main = sbf_get_main(),
  include_root = TRUE
)
```

Arguments

x_name	A string of the name.
sub	A string specifying the path to the sub folder (by default the current sub folder).
main	A string specifying the path to the main folder (by default the current main folder)
include_root	A flag indicating whether to include objects in the top sub folder.

See Also

Other load functions: sbf_load_block(), sbf_load_blocks(), sbf_load_blocks_recursive(), sbf_load_data(), sbf_load_data_from_db(), sbf_load_data_from_pg(), sbf_load_datas(), sbf_load_datas_from_db(), sbf_load_datas_from_pg(), sbf_load_datas_recursive(), sbf_load_db_metatable(), sbf_load_number(), sbf_load_numbers(), sbf_load_numbers_recursive(), sbf_load_object(), sbf_load_objects(), sbf_load_objects_recursive(), sbf_load_plot(), sbf_load_plot_data(), sbf_load_plots_data(), sbf_load_plots_data_recursive(), sbf_load_plots_recursive(), sbf_load_spatial(), sbf_load_spatials(), sbf_load_string(), sbf_load_strings(), sbf_load_strings_recursi, sbf_load_table(), sbf_load_tables(), sbf_load_tables_recursive(), sbf_load_windows_recursive(), sbf_subs_block_recursive(), sbf_subs_data_recursive(), sbf_subs_number_recursive(), sbf_subs_object_recursive(), sbf_subs_plot_recursive(), sbf_subs_string_recursive(), sbf_subs_window_recursive()

sbf_subs_window_recursive *Gets Subs of a Window as a Character Vector*

Description

Recursively returns all the subs of window with name x_name.

Usage

```
sbf_subs_window_recursive(
  x_name,
  sub = sbf_get_sub(),
  main = sbf_get_main(),
  include_root = TRUE
)
```

Arguments

x_name	A string of the name.
sub	A string specifying the path to the sub folder (by default the current sub folder).
main	A string specifying the path to the main folder (by default the current main folder)
include_root	A flag indicating whether to include objects in the top sub folder.

See Also

Other load functions: sbf_load_block(), sbf_load_blocks(), sbf_load_blocks_recursive(), sbf_load_data(), sbf_load_data_from_db(), sbf_load_data_from_pg(), sbf_load_datas(), sbf_load_datas_from_db(), sbf_load_datas_from_pg(), sbf_load_datas_recursive(), sbf_load_db_metatable(), sbf_load_number(), sbf_load_numbers(), sbf_load_numbers_recursive(), sbf_load_object(), sbf_load_objects(), sbf_load_objects_recursive(), sbf_load_plot(), sbf_load_plot_data(), sbf_load_plots_data(), sbf_load_plots_data_recursive(), sbf_load_plots_recursive(), sbf_load_spatial(), sbf_load_spatials(), sbf_load_string(), sbf_load_strings(), sbf_load_strings_recursi, sbf_load_table(), sbf_load_tables(), sbf_load_tables_recursive(), sbf_load_windows_recursive(), sbf_subs_block_recursive(), sbf_subs_data_recursive(), sbf_subs_number_recursive(), sbf_subs_object_recursive(), sbf_subs_plot_recursive(), sbf_subs_string_recursive(), sbf_subs_table_recursive()

sbf_table_exists *Table Exists*

Description

this function is now deprecated as of version 0.0.0.9045

Usage

```
sbf_table_exists(x_name, sub = sbf_get_sub(), main = sbf_get_main())
```

Arguments

x_name	A string of the name.
sub	A string specifying the path to the sub folder (by default the current sub folder).
main	A string specifying the path to the main folder (by default the current main folder)

Value

A flag specifying whether the table exists.

See Also

Other exists functions: sbf_block_exists(), sbf_data_exists(), sbf_number_exists(), sbf_object_exists(), sbf_plot_exists(), sbf_string_exists()

sbf_unarchive_main *Unarchive Main Folder*

Description

Unarchives an archived main folder.

Usage

```
sbf_unarchive_main(
  main = sbf_get_main(),
  archive = 1L,
  ask = getOption("sbf.ask", TRUE)
)
```

Arguments

main	A string specifying the path to the main folder (by default the current main folder)
archive	A positive whole number specifying the folder archived folder where 1L (the default) indicates the most recently archived folder or a character string of the path to the archived folder.
ask	A flag specifying whether to ask before removing the existing folder.

Value

An invisible string of the path to the previously archived folder.

See Also

Other archive: sbf_archive_main(), sbf_get_archive()

Other housekeeping functions: sbf_archive_main(), sbf_rm_flobs(), sbf_rm_main()

---

sbf_upload_flobs_to_db

Upload flobs

Description

Uploads all files to database by default dir = NULL then uploads flobs subdirectory corresponding to database using dbflobr::import_all_flobs().

Usage

```
sbf_upload_flobs_to_db(
  db_name = sbf_get_db_name(),
  sub = sbf_get_sub(),
  main = sbf_get_main(),
  dir = NULL,
  dbflobr_sub = FALSE,
  exists = FALSE,
  replace = FALSE
)
```

Arguments

db_name	A string of the database name.
sub	A logical scalar specifying whether to import flobs based on their filename (sub = FALSE) or the name of their subdirectory (sub = TRUE) which must only contain 1 file. If sub = NA and replace = TRUE then the names of the subdirectories are used irrespective of whether they include files and existing flobs are deleted if the corresponding subdirectory is empty. If sub = TRUE or sub = NA then recursion is just one subfolder deep.

main	A string specifying the path to the main folder (by default the current main folder)
dir	A string of the path to the directory to import the files from. Files need to be within nested folders like 'table1/column1/a.csv'. This structure is created automatically if save_all_flobs() function is used.
dbflobr_sub	A logical scalar specifying whether to import flobs based on their filename (sub = FALSE) or the name of their subdirectory (sub = TRUE) which must only contain 1 file. If sub = NA and replace = TRUE then the names of the subdirectories are used irrespective of whether they include files and existing flobs are deleted if the corresponding subdirectory is empty. If sub = TRUE or sub = NA then recursion is just one subfolder deep.
exists	A logical scalar specifying whether the column must (TRUE) or mustn't (FALSE) already exist or whether it doesn't matter (NA). IF FALSE, a new BLOB column is created.
replace	A flag indicating whether to replace existing flobs (TRUE) or not (FALSE).

Value

An invisible named list indicating directory path, file names and whether files were successfully written to database.

See Also

Other flob: sbf_add_blob_column_to_db(), sbf_save_flobs_from_db()

Other database functions: sbf_add_blob_column_to_db(), sbf_close_db(), sbf_copy_db(), sbf_create_db(), sbf_execute_db(), sbf_open_db(), sbf_query_db()

---

sbf_up_sub *Move Up Sub Folder*

Description

Moves up the sub folder hierarchy.

Usage

```
sbf_up_sub(n = 1L, rm = FALSE, ask = getOption("sbf.ask", TRUE))
```

Arguments

n	A positive int of the number of subfolders to move up.
rm	A flag specifying whether to remove the folder and all its contents if it already exists.
ask	A flag specifying whether to ask before removing the existing folder.

Value

An invisible string specifying the new sub folder.

See Also

Other directory functions: sbf_add_sub(), sbf_get_archive(), sbf_get_db_name(), sbf_get_main(), sbf_get_sub(), sbf_get_workbook_name(), sbf_reset(), sbf_reset_db_name(), sbf_reset_main(), sbf_reset_sub(), sbf_set_db_name(), sbf_set_main(), sbf_set_sub()

Examples

```
sbf_set_sub("nameofsub/othersub/yetothersub")
sbf_up_sub()
sbf_get_sub()
sbf_reset_sub()
```

---

sbf_write_datas_to_xlsx

Write Datas to Excel File

Description

Writes all the data frames in the environment to an xlsx file. Each data frame is saved to a sheet with the same name.

Usage

```
sbf_write_datas_to_xlsx(
  path,
  exists = NA,
  env = parent.frame(),
  ask = getOption("sbf.ask", TRUE)
)
```

Arguments

path	A string of the path to the xlsx file (with the extension).
exists	A logical scalar specifying whether the file should exist.
env	An environment.
ask	A flag specifying whether to ask before deleting an existing database (if exists = FALSE).

Value

An invisible character vector of the names of the data frames.

Index