

# Package: sdmEvalToolCore (via r-universe)

October 1, 2026

**Title** SDM Evaluation Tool Core Functions

**Version** 0.0.1

**Description** The goal of sdmEvalToolCore is to implement the data model and common non-interactive functionality for the SDM Evaluation Tool packages.

**License** Apache License 2.0

**Encoding** UTF-8

**Roxygen** list(markdown = TRUE)

**RoxygenNote** 7.3.3

**Imports** arrow, DBI, dbplyr, dplyr, glue, jsonlite, lorem, sf, suntools, RSQLite, terra, rlang, withr

**Suggests** rmarkdown

**LazyData** true

**Depends** R (>= 3.5)

**Config/pak/sysreqs** libabsl-dev cmake libgdal-dev gdal-bin libgeos-dev libicu-dev libssl-dev libproj-dev libsqlite3-dev libudunits2-dev

**Repository** <https://psolymos.r-universe.dev>

**Date/Publication** 2026-07-21 14:45:11 UTC

**RemoteUrl** <https://github.com/LandSciTech/sdmEvaluationTool>

**RemoteRef** HEAD

**RemoteSha** 32444f2dd7ef4af0025b8324a37b4f796d62fe6f

**RemoteSubdir** sdmEvalToolCore

## Contents

|                             |   |
|-----------------------------|---|
| check_table . . . . .       | 2 |
| combine_questions . . . . . | 3 |
| components . . . . .        | 4 |
| db_connect . . . . .        | 4 |
| db_create_tables . . . . .  | 5 |

|                                        |           |
|----------------------------------------|-----------|
| db_disconnect . . . . .                | 5         |
| db_path . . . . .                      | 6         |
| db_read_comments . . . . .             | 6         |
| db_read_deployment_materials . . . . . | 7         |
| db_read_evaluations . . . . .          | 7         |
| db_read_models . . . . .               | 8         |
| db_read_species . . . . .              | 8         |
| db_read_table . . . . .                | 9         |
| db_read_user_info . . . . .            | 9         |
| db_write_table . . . . .               | 10        |
| default_questions . . . . .            | 11        |
| default_species_table_canada . . . . . | 11        |
| fake . . . . .                         | 12        |
| fields . . . . .                       | 13        |
| followup_questions . . . . .           | 13        |
| fromBoolean . . . . .                  | 13        |
| generate_password . . . . .            | 14        |
| get_comp_ready . . . . .               | 14        |
| get_comp_rule . . . . .                | 15        |
| get_fields . . . . .                   | 15        |
| get_file_ext . . . . .                 | 16        |
| get_user_roles . . . . .               | 16        |
| make_dir . . . . .                     | 17        |
| make_material_id . . . . .             | 17        |
| make_target_path . . . . .             | 18        |
| make_user_db . . . . .                 | 19        |
| prepare_material_entry . . . . .       | 19        |
| read_file . . . . .                    | 20        |
| scaffold_table . . . . .               | 20        |
| sdmevaltool_options . . . . .          | 21        |
| tables . . . . .                       | 21        |
| timestamp . . . . .                    | 22        |
| upload-deployments . . . . .           | 23        |
| upload-materials . . . . .             | 23        |
| upload_material . . . . .              | 26        |
| user_roles . . . . .                   | 27        |
| write_file . . . . .                   | 27        |
| <b>Index</b>                           | <b>28</b> |

---

|             |                    |
|-------------|--------------------|
| check_table | <i>Check Table</i> |
|-------------|--------------------|

---

**Description**

Check Table

**Usage**

```
check_table(x, table_name, dryrun = FALSE)
```

**Arguments**

|            |                                                                        |
|------------|------------------------------------------------------------------------|
| x          | A data frame.                                                          |
| table_name | Table name.                                                            |
| dryrun     | Logical, if checks fail and error is produced, unless it is a dry run. |

**Value**

Invisible TRUE if all check passed.

**Examples**

```
users <- data.frame(
  user_id = c("holden", "draper", "okoye"),
  user_name = c("James Holden", "Bobbie Draper", "Elvi Okoye"),
  user_email = c(
    "jim@rocinante.org",
    "bdraper@mcrn.gov",
    "okoye@rce.com"
  ),
  user_affiliation = c("Rocinante", "MCRN", "RCE"),
  admin = c(TRUE, FALSE, FALSE),
  password = c("pass1", "pass2", "pass3")
)
check_table(users, "users")

models <- data.frame(
  model_id = "bam_v5_can71",
  model_name = "BAM v5 Can 71",
  model_description = "BAM version 5 Canada model in BCR 71"
)
check_table(models, "models")

check_table(models, "users", dryrun = TRUE)
```

---

|                   |                                      |
|-------------------|--------------------------------------|
| combine_questions | <i>Join questions and follow-ups</i> |
|-------------------|--------------------------------------|

---

**Description**

Join questions and follow-ups

**Usage**

```
combine_questions(q)
```

Arguments

q                      A data frame with questions.

Examples

```
q <- sdmEvalToolCore::default_questions
q$followup_level[5] <- 3
combine_questions(q)
```

---

|            |                   |
|------------|-------------------|
| components | <i>Components</i> |
|------------|-------------------|

---

Description

Components

Usage

components

Format

A data frame with component configuration (some columns are lists).

---

|            |                            |
|------------|----------------------------|
| db_connect | <i>Connect to Database</i> |
|------------|----------------------------|

---

Description

Connect to Database

Usage

```
db_connect(dbname = NULL, ...)
```

Arguments

dbname                DB name.  
...                    Arguments passed to [DBI::dbConnect\(\)](#).

Value

A database connection.

---

|                  |                  |
|------------------|------------------|
| db_create_tables | Create DB tables |
|------------------|------------------|

---

**Description**

Create DB tables

**Usage**

```
db_create_tables(con, tables = NULL, force = FALSE)
```

**Arguments**

|        |                                             |
|--------|---------------------------------------------|
| con    | DB connection.                              |
| tables | Character, table name.                      |
| force  | Logical, force (create even if not exists). |

**Examples**

```
con <- db_connect(":memory:")
db_create_tables(con, "models")
db_create_tables(con)
db_disconnect(con)
```

---

|               |                          |
|---------------|--------------------------|
| db_disconnect | Disconnect from Database |
|---------------|--------------------------|

---

**Description**

Disconnect from Database

**Usage**

```
db_disconnect(con, ...)
```

**Arguments**

|     |                                                           |
|-----|-----------------------------------------------------------|
| con | DB connection.                                            |
| ... | Arguments passed to <a href="#">DBI::dbDisconnect()</a> . |

---

|         |                         |
|---------|-------------------------|
| db_path | <i>Path to Database</i> |
|---------|-------------------------|

---

**Description**

Path to Database

**Usage**

db\_path()

**Value**

Path to default database

---

|                  |                             |
|------------------|-----------------------------|
| db_read_comments | <i>Get Comments from DB</i> |
|------------------|-----------------------------|

---

**Description**

Get Comments from DB

**Usage**

db\_read\_comments(con, deployment\_id)

**Arguments**

|               |                |
|---------------|----------------|
| con           | DB connection. |
| deployment_id | Deployment ID. |

**Value**

A data frame with comments for the deployment.

---

`db_read_deployment_materials`*Get Deployment Materials from DB*

---

**Description**

Get Deployment Materials from DB

**Usage**

```
db_read_deployment_materials(con, deployment_id = NULL, user_id = NULL)
```

**Arguments**

|                            |                                      |
|----------------------------|--------------------------------------|
| <code>con</code>           | DB connection.                       |
| <code>deployment_id</code> | Deployment IDs (multiple permitted). |
| <code>user_id</code>       | User ID.                             |

**Value**

A data frame with joined deployment materials.

---

`db_read_evaluations`     *Get Evaluations from DB*

---

**Description**

Get Evaluations from DB

**Usage**

```
db_read_evaluations(con, deployment_id = NULL, user_id = NULL)
```

**Arguments**

|                            |                                                |
|----------------------------|------------------------------------------------|
| <code>con</code>           | DB connection.                                 |
| <code>deployment_id</code> | Deployment ID.                                 |
| <code>user_id</code>       | Evaluation create user IDs (multiple allowed). |

**Value**

A data frame with evaluations for the deployment.

---

|                |                           |
|----------------|---------------------------|
| db_read_models | <i>Get models from DB</i> |
|----------------|---------------------------|

---

**Description**

Get models from DB

**Usage**

```
db_read_models(con, model_id = NULL)
```

**Arguments**

|          |                      |
|----------|----------------------|
| con      | DB connection.       |
| model_id | Model ID (optional). |

**Value**

A data frame with models. Optionally filtered to model\_id.

---

|                 |                            |
|-----------------|----------------------------|
| db_read_species | <i>Get species from DB</i> |
|-----------------|----------------------------|

---

**Description**

Get species from DB

**Usage**

```
db_read_species(con, species_id = NULL)
```

**Arguments**

|            |                        |
|------------|------------------------|
| con        | DB connection.         |
| species_id | Species ID (optional). |

**Value**

A data frame with species and display names. Optionally filtered to species\_id.

---

|               |                          |
|---------------|--------------------------|
| db_read_table | <i>Get Table from DB</i> |
|---------------|--------------------------|

---

**Description**

Get Table from DB

**Usage**

```
db_read_table(con, table_name)
```

**Arguments**

|            |                |
|------------|----------------|
| con        | DB connection. |
| table_name | Table name.    |

**Value**

A data frame with the table data.

---

|                   |                              |
|-------------------|------------------------------|
| db_read_user_info | <i>Get User Info from DB</i> |
|-------------------|------------------------------|

---

**Description**

Get User Info from DB

**Usage**

```
db_read_user_info(con, user_id, deployment_id)
```

**Arguments**

|               |                |
|---------------|----------------|
| con           | DB connection. |
| user_id       | User ID.       |
| deployment_id | Deployment ID. |

**Value**

A data frame with user info and access roles. The "user\_roles" attribute gives the user roles as a vector.

---

|                |                                        |
|----------------|----------------------------------------|
| db_write_table | <i>Write Data to an Existing Table</i> |
|----------------|----------------------------------------|

---

### Description

Data will be inserted/updated/upserted in an existing DB table.

### Usage

```
db_write_table(  
  con,  
  table,  
  data,  
  mode = c("insert", "update", "upsert"),  
  check = TRUE,  
  dryrun = FALSE  
)
```

### Arguments

|        |                                                                                                                                       |
|--------|---------------------------------------------------------------------------------------------------------------------------------------|
| con    | A database connection.                                                                                                                |
| table  | Table name.                                                                                                                           |
| data   | Data frame to append to the DB table.                                                                                                 |
| mode   | How to execute the write operation: insert (append) new record, update existing record, or upsert (update existing or insert if not). |
| check  | Logical, should data be validated?                                                                                                    |
| dryrun | Logical, write to a text file when TRUE and to the database when FALSE.                                                               |

### Value

Invisible TRUE.

### Examples

```
con <- db_connect(":memory:")  
db_create_tables(con, "models")  
models <- data.frame(  
  model_id = "bam_v5_can71",  
  model_name = "BAM v5 Can 71",  
  model_description = "BAM version 5 Canada model in BCR 71"  
)  
db_write_table(con, "models", models, mode = "insert")  
db_disconnect(con)
```

---

|                   |                          |
|-------------------|--------------------------|
| default_questions | <i>Default Questions</i> |
|-------------------|--------------------------|

---

**Description**

Default Questions

**Usage**

```
default_questions
```

**Format**

A data frame with default questions.

---

|                              |                              |
|------------------------------|------------------------------|
| default_species_table_canada | <i>Default Species Table</i> |
|------------------------------|------------------------------|

---

**Description**

Default Species Table

**Usage**

```
default_species_table_canada()
```

**Value**

Data frame.

**Examples**

```
str(default_species_table_canada())
```

---

|      |                        |
|------|------------------------|
| fake | <i>Fake DB Entries</i> |
|------|------------------------|

---

**Description**

Fake DB Entries

**Usage**

```
fake_comment(deployment_id, model_id, species_id, user_id, time = NULL)

fake_evaluation(
  deployment_id,
  material_id,
  component_id,
  user_id,
  questions = NULL,
  time = NULL
)

fake_response(question)
```

**Arguments**

|               |                                           |
|---------------|-------------------------------------------|
| deployment_id | Deployment ID.                            |
| model_id      | Model ID.                                 |
| species_id    | Species ID.                               |
| user_id       | User ID.                                  |
| time          | Date-time.                                |
| material_id   | Material ID.                              |
| component_id  | Component ID.                             |
| questions     | A questions data frame.                   |
| question      | A questions data frame with a single row. |

**Examples**

```
fake_comment(
  deployment_id = "deployment1",
  model_id = "model1",
  species_id = "MOTR",
  user_id = "draper")

str(fake_response(sdmEvalToolCore::default_questions[1,]))

fake_evaluation(
  deployment_id = "deployment1",
```

```
material_id = "material1",
component_id = "model_fit",
user_id = "draper")
```

---

|        |               |
|--------|---------------|
| fields | <i>Fields</i> |
|--------|---------------|

---

**Description**

Fields

**Usage**

fields

**Format**

A data frame.

---

|                    |                            |
|--------------------|----------------------------|
| followup_questions | <i>Follow-up Questions</i> |
|--------------------|----------------------------|

---

**Description**

Follow-up Questions

**Usage**

followup\_questions

**Format**

A data frame with follow-up questions.

---

|             |                         |
|-------------|-------------------------|
| fromBoolean | <i>Handling Boolean</i> |
|-------------|-------------------------|

---

**Description**

Handling Boolean

**Usage**

fromBoolean(x)

**Arguments**

x                      An R vector.

---

|                   |                          |
|-------------------|--------------------------|
| generate_password | <i>Generate password</i> |
|-------------------|--------------------------|

---

**Description**

Generate password

**Usage**

```
generate_password(length = 12, start_with_letter = TRUE)
```

**Arguments**

|                   |                                                                                 |
|-------------------|---------------------------------------------------------------------------------|
| length            | Length of the password. Default is 12.                                          |
| start_with_letter | Logical indicating if the password should start with a letter. Default is TRUE. |

**Value**

A randomly generated password.

**Examples**

```
generate_password()
```

---

|                |                                   |
|----------------|-----------------------------------|
| get_comp_ready | <i>Which Components Are Ready</i> |
|----------------|-----------------------------------|

---

**Description**

Which Components Are Ready

**Usage**

```
get_comp_ready(ready = character(0L))
```

**Arguments**

|       |                                            |
|-------|--------------------------------------------|
| ready | Character, component names that are ready. |
|-------|--------------------------------------------|

**Value**

A data frame with components and a column telling which ones are ready. The "percent\_ready" attribute gives a percentage value of the mandatory components that are ready.

**Examples**

```
r <- get_comp_ready(c("observations", "model_metadata"))
r
attr(,"percent_ready")
```

---

|               |                                 |
|---------------|---------------------------------|
| get_comp_rule | <i>Get Rule for a Component</i> |
|---------------|---------------------------------|

---

**Description**

Get Rule for a Component

**Usage**

```
get_comp_rule(component_id, rule_type = c("upload", "display", "evaluation"))
```

**Arguments**

|              |                                       |
|--------------|---------------------------------------|
| component_id | Character, component name (length 1). |
| rule_type    | Character, the type of rule.          |

**Value**

A list with the rules, or NULL.

**Examples**

```
str(get_comp_rule("observations", "upload"))
str(get_comp_rule("observations", "evaluation"))
```

---

|            |                         |
|------------|-------------------------|
| get_fields | <i>Get Table Fields</i> |
|------------|-------------------------|

---

**Description**

Get Table Fields

**Usage**

```
get_fields(table_name)
```

**Arguments**

|            |             |
|------------|-------------|
| table_name | Table name. |
|------------|-------------|

## Value

A data frame.

## Examples

```
get_fields("users")
get_fields("components")
```

---

|              |                           |
|--------------|---------------------------|
| get_file_ext | <i>Get File Extension</i> |
|--------------|---------------------------|

### Description

## Get File Extension

## Usage

```
get_file_ext(path)
```

## Arguments

|      |                         |
|------|-------------------------|
| path | Character, a file path. |
|------|-------------------------|

## Value

Lower case file extension as string.

## Examples

```
get_file_ext("bam_v5/oven/observations.parquet")
```

|                |                       |
|----------------|-----------------------|
| get_user_roles | <i>Get User Roles</i> |
|----------------|-----------------------|

### Description

## Get User Roles

## Usage

```
get_user_roles(role)
```

## Arguments

|      |                                      |
|------|--------------------------------------|
| role | Character, 1 or multiple user roles. |
|------|--------------------------------------|

**Value**

A data frame with the permissible user actions (TRUE/FALSE).

**Examples**

```
str(get_user_roles(c("modeler")))
str(get_user_roles(c("evaluator")))
str(get_user_roles(c("evaluator", "modeler")))
```

---

|          |                       |
|----------|-----------------------|
| make_dir | <i>Make Directory</i> |
|----------|-----------------------|

---

**Description**

Make Directory

**Usage**

```
make_dir(path, ...)
```

**Arguments**

- path           Character, a file path.
- ...            Arguments passed to `dir.create()`.

**Value**

Create a directory recursively as a side effect, returning logical indicating the result ().

---

|                  |                         |
|------------------|-------------------------|
| make_material_id | <i>Make Material ID</i> |
|------------------|-------------------------|

---

**Description**

Make Material ID

**Usage**

```
make_material_id(model_id, species_id, component_id, sep = "_")
```

**Arguments**

- model\_id       Model ID.
- species\_id     Species ID.
- component\_id   Component ID.
- sep            Character. Separator.

Examples

```
make_material_id("model1", NA, "predictor_metadata")
make_material_id("model1", "CAWA", "predictor_raster")
```

---

|                  |                         |
|------------------|-------------------------|
| make_target_path | <i>Make Target Path</i> |
|------------------|-------------------------|

---

Description

Make Target Path

Usage

```
make_target_path(path, data = list(), base = NULL)
```

Arguments

|      |                                                                                                                   |
|------|-------------------------------------------------------------------------------------------------------------------|
| path | Character, a file path.                                                                                           |
| data | List with name value pairs, the values are substituted into the path according to glue rules (with curly braces). |
| base | Character, base path, defaults to sdmevaltool_options()\$base when NULL.                                          |

Value

The path string with data values substituted.

Examples

```
make_target_path("bam_v5/oven/observations.parquet")
make_target_path("{model}/{species}/observations.parquet",
  list(model = "bam_v5", species = "oven"))
make_target_path("{model}/{species}/observations.parquet",
  list(model = "bam_v5", species = "oven"), ".")
```

---

|              |                                                     |
|--------------|-----------------------------------------------------|
| make_user_db | <i>Make user database suitable for shinymanager</i> |
|--------------|-----------------------------------------------------|

---

**Description**

Make user database suitable for shinymanager

**Usage**

```
make_user_db()
```

**Value**

A data frame with user credentials.

---

|                        |                               |
|------------------------|-------------------------------|
| prepare_material_entry | <i>Prepare Material Entry</i> |
|------------------------|-------------------------------|

---

**Description**

Prepare Material Entry

**Usage**

```
prepare_material_entry(
  model_id,
  species_id,
  component_id,
  user_id,
  material_settings = "[]"
)
```

**Arguments**

|                   |                    |
|-------------------|--------------------|
| model_id          | Model ID.          |
| species_id        | Species ID.        |
| component_id      | Component ID.      |
| user_id           | User ID.           |
| material_settings | Material settings. |

**Examples**

```
prepare_material_entry("model1", "CAWA", "predictor_raster", "testuser")
```

---

`read_file`*Read File*

---

**Description**

Read File

**Usage**

```
read_file(path, ...)
```

**Arguments**

`path`                Character, a file path.

`...`                Arguments passed to the reader functions.

**Value**

Return the result from the file.

---

`scaffold_table`*Scaffold Table*

---

**Description**

Scaffold Table

**Usage**

```
scaffold_table(table_name)
```

**Arguments**

`table_name`        Table name.

**Value**

A data frame returned invisibly, called for the printed side effects.

**Examples**

```
scaffold_table("users")  
scaffold_table("materials")
```

---

|                     |                        |
|---------------------|------------------------|
| sdmevaltool_options | <i>Package Options</i> |
|---------------------|------------------------|

---

**Description**

Package Options

**Usage**

sdmevaltool\_options(...)

**Arguments**

...                   Names list of options.

**Examples**

(sdmevaltool\_options())

---

|        |               |
|--------|---------------|
| tables | <i>Tables</i> |
|--------|---------------|

---

**Description**

Tables

**Usage**

tables

**Format**

A data frame.

---

|           |                            |
|-----------|----------------------------|
| timestamp | <i>Handling Timestamps</i> |
|-----------|----------------------------|

---

**Description**

Handling Timestamps

**Usage**

```
timestamp_from(ms, tz = NULL, ...)
```

```
timestamp_to(dt)
```

```
now()
```

**Arguments**

|     |                                                                              |
|-----|------------------------------------------------------------------------------|
| ms  | Unix time in seconds.                                                        |
| tz  | Time zone, when NULL it defaults to <code>sdmevaltool_options()\$tz</code> . |
| ... | Other optional arguments passed to <a href="#">as.POSIXct()</a> .            |
| dt  | POSIXct coercible date/time value.                                           |

**Value**

`timestamp_from` returns [POSIXct\(\)](#).

`timestamp_to` returns Unix time as character .

**See Also**

[sdmevaltool\\_options\(\)](#) to set tz globally.

**Examples**

```
dt1 <- now()
timestamp_to(dt1)
timestamp_from(timestamp_to(dt1))
```

---

|                    |                                     |
|--------------------|-------------------------------------|
| upload-deployments | <i>Prepare Deployment Materials</i> |
|--------------------|-------------------------------------|

---

**Description**

Prepare Deployment Materials

**Usage**

```
prep_deployment_questions(deployment_id, x = NULL, ...)

prep_deployment_settings(deployment_id, x, ...)

prep_deployment_subunits(
  deployment_id,
  x = NULL,
  reference_raster,
  reference_points,
  n = c(20, 20),
  square = TRUE,
  ...
)
```

**Arguments**

- deployment\_id    Deployment ID.
- x                An object to upload.
- ...              Arguments passed to the write function.
- reference\_raster        Reference raster layer.
- reference\_points        Reference spatial points object.
- n                Number of grid cells in x and y direction.
- square           Logical. If 'FALSE', create hexagonal grid.

---

|                  |                                 |
|------------------|---------------------------------|
| upload-materials | <i>Prepare Upload Materials</i> |
|------------------|---------------------------------|

---

**Description**

Prepare Upload Materials

**Usage**

```
prep_predictor_raster(  
  x,  
  resolution = 5,  
  model_id,  
  user_id,  
  material_settings = NULL,  
  con = NULL,  
  update = FALSE,  
  ...  
)  
  
prep_predictor_metadata(  
  x,  
  model_id,  
  user_id,  
  material_settings = NULL,  
  con = NULL,  
  update = FALSE,  
  ...  
)  
  
prep_model_metadata(  
  x,  
  model_id,  
  user_id,  
  material_settings = NULL,  
  con = NULL,  
  update = FALSE,  
  ...  
)  
  
prep_spatial_prediction(  
  x,  
  model_id,  
  species_id,  
  user_id,  
  material_settings = NULL,  
  con = NULL,  
  update = FALSE,  
  ...  
)  
  
prep_observations(  
  x,  
  species,  
  model_id,  
  user_id,
```

```

    material_settings = NULL,
    con = NULL,
    update = FALSE,
    ...
)

prep_model_summary(
  x,
  species,
  model_id,
  user_id,
  material_settings = NULL,
  con = NULL,
  update = FALSE,
  ...
)

prep_model_fit(
  x,
  species,
  model_id,
  user_id,
  material_settings = NULL,
  con = NULL,
  update = FALSE,
  ...
)

```

### Arguments

|                                |                                                               |
|--------------------------------|---------------------------------------------------------------|
| <code>x</code>                 | An object to upload.                                          |
| <code>resolution</code>        | Raster resampling resolution                                  |
| <code>model_id</code>          | Model ID.                                                     |
| <code>user_id</code>           | User ID.                                                      |
| <code>material_settings</code> | Material settings.                                            |
| <code>con</code>               | Database connection (use NULL to avoid inserting new record). |
| <code>update</code>            | Logical. Is this a new entry or an update to an existing one. |
| <code>...</code>               | Arguments passed to the write function.                       |
| <code>species_id</code>        | Species ID.                                                   |
| <code>species</code>           | Reference species table.                                      |

---

|                 |                                                       |
|-----------------|-------------------------------------------------------|
| upload_material | <i>Write File and Insert/Update Material DB Entry</i> |
|-----------------|-------------------------------------------------------|

---

**Description**

Write File and Insert/Update Material DB Entry

**Usage**

```
upload_material(  
  component_id,  
  x,  
  model_id,  
  species_id,  
  user_id,  
  deployment_id = NULL,  
  material_settings = NULL,  
  con = NULL,  
  update = FALSE,  
  ...  
)
```

**Arguments**

|                   |                                                                                                  |
|-------------------|--------------------------------------------------------------------------------------------------|
| component_id      | Component ID.                                                                                    |
| x                 | Object.                                                                                          |
| model_id          | Model ID.                                                                                        |
| species_id        | Species ID.                                                                                      |
| user_id           | User ID.                                                                                         |
| deployment_id     | Deployment ID.                                                                                   |
| material_settings | List with the material settings.                                                                 |
| con               | Database connection (use NULL to avoid inserting a record).                                      |
| update            | Logical. Is this a new entry (FALSE, default) or an update (TRUE) to an existing material entry. |
| ...               | Arguments passed to the write function.                                                          |

---

|            |                   |
|------------|-------------------|
| user_roles | <i>User Roles</i> |
|------------|-------------------|

---

**Description**

User Roles

**Usage**

user\_roles

**Format**

A data frame.

---

|            |                   |
|------------|-------------------|
| write_file | <i>Write File</i> |
|------------|-------------------|

---

**Description**

Write File

**Usage**

write\_file(x, path, ...)

**Arguments**

- |      |                                           |
|------|-------------------------------------------|
| x    | Object to write.                          |
| path | Character, a file path.                   |
| ...  | Arguments passed to the writer functions. |

**Value**

Writes file as a side effect.

# Index

- \* **datasets**
  - components, [4](#)
  - default\_questions, [11](#)
  - fields, [13](#)
  - followup\_questions, [13](#)
  - tables, [21](#)
  - user\_roles, [27](#)
- as.POSIXct(), [22](#)
- check\_table, [2](#)
- combine\_questions, [3](#)
- components, [4](#)
- db\_connect, [4](#)
- db\_create\_tables, [5](#)
- db\_disconnect, [5](#)
- db\_path, [6](#)
- db\_read\_comments, [6](#)
- db\_read\_deployment\_materials, [7](#)
- db\_read\_evaluations, [7](#)
- db\_read\_models, [8](#)
- db\_read\_species, [8](#)
- db\_read\_table, [9](#)
- db\_read\_user\_info, [9](#)
- db\_write\_table, [10](#)
- DBI::dbConnect(), [4](#)
- DBI::dbDisconnect(), [5](#)
- default\_questions, [11](#)
- default\_species\_table\_canada, [11](#)
- dir.create(), [17](#)
- fake, [12](#)
- fake\_comment(fake), [12](#)
- fake\_evaluation(fake), [12](#)
- fake\_response(fake), [12](#)
- fields, [13](#)
- followup\_questions, [13](#)
- fromBoolean, [13](#)
- generate\_password, [14](#)
- get\_comp\_ready, [14](#)
- get\_comp\_rule, [15](#)
- get\_fields, [15](#)
- get\_file\_ext, [16](#)
- get\_user\_roles, [16](#)
- make\_dir, [17](#)
- make\_material\_id, [17](#)
- make\_target\_path, [18](#)
- make\_user\_db, [19](#)
- now(timestamp), [22](#)
- POSIXct(), [22](#)
- prep\_deployment\_questions
  - (upload-deployments), [23](#)
- prep\_deployment\_settings
  - (upload-deployments), [23](#)
- prep\_deployment\_subunits
  - (upload-deployments), [23](#)
- prep\_model\_fit(upload-materials), [23](#)
- prep\_model\_metadata(upload-materials),  
[23](#)
- prep\_model\_summary(upload-materials),  
[23](#)
- prep\_observations(upload-materials), [23](#)
- prep\_predictor\_metadata
  - (upload-materials), [23](#)
- prep\_predictor\_raster
  - (upload-materials), [23](#)
- prep\_spatial\_prediction
  - (upload-materials), [23](#)
- prepare\_material\_entry, [19](#)
- read\_file, [20](#)
- scaffold\_table, [20](#)
- sdmevaltool\_options, [21](#)
- sdmevaltool\_options(), [22](#)
- tables, [21](#)

timestamp, [22](#)  
timestamp\_from(timestamp), [22](#)  
timestamp\_to(timestamp), [22](#)  
  
upload-deployments, [23](#)  
upload-materials, [23](#)  
upload\_material, [26](#)  
user\_roles, [27](#)  
  
write\_file, [27](#)