

# Package: **kwb.swmm** (via **r-universe**)

October 15, 2024

**Title** R Package with Functions for Working with EPA`s Storm Water Management Model (SWMM)

**Version** 0.1.1

**Description** R package with functions for working with EPA`s Storm Water Management Model [SWMM](<https://www.epa.gov/water-research/storm-water-management-model-swmm>).

**License** MIT + file LICENSE

**URL** <https://github.com/KWB-R/kwb.swmm>

**BugReports** <https://github.com/KWB-R/kwb.swmm/issues>

**Encoding** UTF-8

**LazyData** true

**Roxygen** list(markdown = TRUE)

**RoxygenNote** 7.1.2

**Imports** archive, data.table, dplyr, fs, kwb.event, kwb.utils, readr, rlang, swmmr, tibble, tidyr, tidyselect, magrittr

**Suggests** covr, knitr, lubridate, reticulate, rmarkdown, sessioninfo, withr

**Remotes** github::kwb-r/kwb.event, github::kwb-r/kwb.nextcloud, github::kwb-r/kwb.python, github::kwb-r/kwb.utils, github::hsonne/swmmr@dev

**VignetteBuilder** knitr

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

**RemoteUrl** <https://github.com/KWB-R/kwb.swmm>

**RemoteRef** HEAD

**RemoteSha** 62ccd4dda08033ab094aa38aadeb733dca76fb4c

## Contents

calculate_rainevent_stats . . . . .	2
download_swmm_executable . . . . .	3
extdata_file . . . . .	4
get_lid_para_types . . . . .	4
get_meta . . . . .	5
get_results . . . . .	5
hectar_to_squaremeter . . . . .	6
lps_to_mmPerHour . . . . .	6
run_swmm . . . . .	7
squaremeter_to_hectar . . . . .	7
<b>Index</b>	<b>9</b>

---

calculate\_rainevent\_stats  
*Calculate Rain Events*

---

## Description

Calculate Rain Events

## Usage

```
calculate_rainevent_stats(  
  results_system,  
  col_eventsep = "total_runoff",  
  aggregation_function = "sum",  
  signalThreshold = 0,  
  eventSeparationTime = 6 * 3600,  
  ...  
)
```

## Arguments

`results_system` data frame as retrieved by [get\\_results](#)

`col_eventsep` column to be used for event separation (default: "total\_runoff")

`aggregation_function` function to be used for aggregation and passed to [getEventStatistics](#) (default: "sum")

`signalThreshold` passed to [getEvents](#), Value that needs to be exceeded (signalComparisonOperator == "gt") or reached (signalComparisonOperator == "ge") by the rain heights (or intensities) in order to be counted as a "signal". default: 0

`eventSeparationTime` eventSeparationTime passed to [getEvents](#) (default: 6\*3600)

... additional arguments passed to [getEvents](#)

**Value**

tibble with statistics for all rain events

**Examples**

```
## Not run:
path_out_file <- "path-to-my-swmm-output-file"
results_system <- kwb.swmm::get_results(path_out = path_out_file)
rainevent_stats <- calculate_rainevent_stats(results_system,
      aggregation_function = "sum",
      signalThreshold = 0,
      eventSeparationTime = 6*3600)

## End(Not run)
```

---

download\_swmm\_executable

*Download SWMM executable (currently only for windows only!)*

---

**Description**

downloads SWMM executable from USEPA's GitHub repo of <https://github.com/USEPA/Stormwater-Management-Model/releases>

**Usage**

```
download_swmm_executable(tdir = tempdir(), swmm_version = "5.2.0")
```

**Arguments**

tdir	target directory for unzipping zip files
swmm_version	desired SWMM version, one of: "5.2.0", "5.1.15", "5.1.14", "5.1.13" (default: "5.2.0")

**Value**

path to download "runswmm.exe"

**Examples**

```
download_swmm_executable()
```

---

extdata_file	<i>Get Path to File in This Package</i>
--------------	---

---

**Description**

Get Path to File in This Package

**Usage**

```
extdata_file(...)
```

**Arguments**

... parts of path passed to [system.file](#)

---

get_lid_para_types	<i>get_lid_para_types</i>
--------------------	---------------------------

---

**Description**

get\_lid\_para\_types

**Usage**

```
get_lid_para_types()
```

**Value**

returns tidy data frame with required parameteration for LIDs based on SWMM documentation

**Examples**

```
get_lid_para_types()
```

---

get_meta	<i>Helper function: get metadata for selected elements</i>
----------	--

---

**Description**

Helper function: get metadata for selected elements

**Usage**

```
get_meta(name, path_out)
```

**Arguments**

name	of element to select. one of: c("subcatchments", "nodes", "links", "system")
path_out	path to SWMM output file

**Value**

tibble with columns "name", "itype", "vindex" and corresponding IDs required for [read\\_out](#)

---

get_results	<i>Get Results</i>
-------------	--------------------

---

**Description**

Get Results

**Usage**

```
get_results(path_out, type = "system", object_name = NULL, vIndex = NULL)
```

**Arguments**

path_out	path to SWMM output file
type	output type, select from c("subcatchments" "nodes", "links" or "system"), default: "system"
object_name	Sets the objects of which time series data is returned. if NULL all objects will be returned (default: NULL)
vIndex	indexes of system wide indexes to be imported. By default all indexes will be imported (default: NULL). If only selected vIndex's should be imported have a look at the documentation of <a href="#">read_out</a>

**Value**

for "system" a tibble with selected system variables, for all other "type" a list with sublists for each "object\_name"

**Examples**

```
## Not run:
path_out_file <- "path-to-my-swmm-output-file"
results_subcatchments <- kwb.swmm::get_results(path_out_file, type = "subcatchments")
results_links <- kwb.swmm::get_results(path_out_file, type = "links")
results_nodes <- kwb.swmm::get_results(path_out_file, type = "nodes")
results_system <- kwb.swmm::get_results(path_out_file, type = "system")

## End(Not run)
```

---

hectar\_to\_squaremeter *Helper function: hectar to squaremeter*

---

**Description**

Helper function: hectar to squaremeter

**Usage**

```
hectar_to_squaremeter(hectar)
```

**Arguments**

hectar            hectar

**Value**

area in hectar

**Examples**

```
squaremeter_to_hectar(100)
```

---

lps\_to\_mmPerHour        *Helper function: converts l/s into mm/h*

---

**Description**

Helper function: converts l/s into mm/h

**Usage**

```
lps_to_mmPerHour(values)
```

**Arguments**

values            values in liter per second

**Value**

values in mm per hour

**Examples**

```
lps_to_mmPerHour(1)
```

---

 run\_swmm

*Run SWMM*


---

**Description**

allows to run SWMM model in model folder, i.e. it is possible to define paths to climate data in swmm\_model.inp with relative paths (if these are contained in the same folder or a subfolder of the model\_dir)

**Usage**

```
run_swmm(model_dir, model_inp, exe, ...)
```

**Arguments**

model_dir	model directory with 'model.inp'
model_inp	name of model input file 'model.inp'
exe	path to SWMM executable
...	additional arguments passed to shell

**Value**

runs SWMM

---

 squaremeter\_to\_hectar *Helper function: squaremeter to hectar*


---

**Description**

Helper function: squaremeter to hectar

**Usage**

```
squaremeter_to_hectar(squaremeter)
```

**Arguments**

squaremeter	squaremeter
-------------	-------------

8

*squaremeter\_to\_hectar*

**Value**

area in hectar

**Examples**

`squaremeter_to_hectar(100)`



# Index

`calculate_rainevent_stats`, 2

`download_swmm_executable`, 3

`extdata_file`, 4

`get_lid_para_types`, 4

`get_meta`, 5

`get_results`, 2, 5

`getEvents`, 2

`getEventStatistics`, 2

`hectar_to_squaremeter`, 6

`lps_to_mmPerHour`, 6

`read_out`, 5

`run_swmm`, 7

`squaremeter_to_hectar`, 7

`system.file`, 4