

# Package: swmmr (via r-universe)

June 24, 2026

**Type** Package

**Title** R Interface for US EPA's SWMM

**Description** Functions to connect the widely used Storm Water Management Model (SWMM) of the United States Environmental Protection Agency (US EPA)  
<<https://www.epa.gov/water-research/storm-water-management-model-swmm>>  
to R with currently two main goals: (1) Run a SWMM simulation from R and (2) provide fast access to simulation results, i.e. SWMM's binary '.out'-files. High performance is achieved with help of Rcpp. Additionally, reading SWMM's '.inp' and '.rpt' files is supported to glance model structures and to get direct access to simulation summaries.

**Version** 0.9.1.9000

**URL** <https://github.com/dleutnant/swmmr>

**License** GPL-3

**LazyData** TRUE

**Encoding** UTF-8

**BugReports** <https://github.com/dleutnant/swmmr/issues>

**ByteCompile** TRUE

**Imports** dplyr (>= 0.7.4), purrr (>= 0.2.4), readr (>= 1.1.1), Rcpp, tibble (>= 1.2.4), tidyr (>= 1.0.0), utils, xts (>= 0.10-1), zoo

**LinkingTo** Rcpp

**Suggests** DEoptim, ggplot2, sf (>= 0.6-1), knitr, rmarkdown, testthat

**SystemRequirements** SWMM (>=5.1.012)

**RoxygenNote** 7.1.2

**Roxygen** list(markdown = TRUE, roclets = c("`rd", "` namespace", "` collate"))

**VignetteBuilder** knitr

**Config/pak/sysreqs** libicu-dev libx11-dev

**Repository** https://kwb-r.r-universe.dev  
**Date/Publication** 2022-02-02 09:43:32 UTC  
**RemoteUrl** https://github.com/hsonne/swmmr  
**RemoteRef** HEAD  
**RemoteSha** 820fe76a47cf7a0ca2547ce35920aa30de94207b

## Contents

autoplot.inp . . . . .	2
convert_to_sf . . . . .	3
get_out_content . . . . .	4
get_out_version . . . . .	5
inp_to_files . . . . .	5
read_inp . . . . .	6
read_lid_rpt . . . . .	7
read_out . . . . .	7
read_rpt . . . . .	9
run_swmm . . . . .	10
shp_to_inp . . . . .	11
summary.inp . . . . .	12
swmmr . . . . .	13
write_inp . . . . .	13

<b>Index</b>	<b>15</b>
--------------	-----------

---

autoplot.inp	<i>Plot a swmm model structure using ggplot2</i>
--------------	--

---

## Description

This function reads an object of class 'inp'. All objects are converted to simple feature geometries via [inp\\_to\\_sf](#) and finally passed to [geom\\_sf](#). It allows to quickly visualize a model structure.

## Usage

```
## S3 method for class 'inp'
autoplot(x, ...)
```

## Arguments

x	An inp object
...	currently ignored

## Note

Lifecycle: experimental

## Examples

```
## Not run:  
inp <- read_inp("model.inp")  
autoplot(inp)  
  
## End(Not run)
```

---

convert\_to\_sf

*Convert swmm objects to simple feature geometries*

---

## Description

- `junctions_to_sf()`: converts junctions to simple features (required sections: junctions and coordinates)
- `outfalls_to_sf()`: converts junctions to simple features (required sections: outfalls and coordinates)
- `links_to_sf()`: converts links to simple features (required sections: conduits and coordinates)
- `subcatchments_to_sf()`: converts subcatchments to simple features (required sections: subcatchments, subareas, infiltration and polygons)
- `raingages_to_sf()`: converts raingages to simple features (required sections: raingages and symbols)
- `storages_to_sf()`: converts storages to simple features (required sections: storage and coordinates)
- `weirs_to_sf()`: converts weirs to simple features (required sections: weirs and coordinates)
- `orifices_to_sf()`: converts orifices to simple features (required sections: orifices and coordinates)
- `pumps_to_sf()`: converts pumps to simple features (required sections: pumps and coordinates)
- `inp_to_sf()`: converts junctions, outfalls, links, storages, weirs, orifices, pumps, subcatchments and raingages to a list of simple features

## Usage

`raingages_to_sf(x)`

`junctions_to_sf(x)`

`outfalls_to_sf(x)`

`storages_to_sf(x)`

`subcatchments_to_sf(x)`

`links_to_sf(x)`

```
weirs_to_sf(x)
orifices_to_sf(x)
pumps_to_sf(x)
weirs_to_sf(x)
orifices_to_sf(x)
pumps_to_sf(x)
inp_to_sf(x, remove_invalid = TRUE)
```

### Arguments

`x` An object of class 'inp', created by [read\\_inp](#).  
`remove_invalid` Should invalid sf geometries be removed?

### Value

A simple feature or a list of simple features

### See Also

[sf](#)

---

get_out_content	<i>Get the content of an .out file.</i>
-----------------	---

---

### Description

This function opens an .out file and lists all available time series data. Currently, the list is returned 'as is' which might change in future. It belongs to a set of helper functions which aim to simplify the work with .out files. The lifecycle of this function is considered experimental.

### Usage

```
get_out_content(file = "")
```

### Arguments

`file` The file to be read.

### Value

A list showing the available content.

**Examples**

```
## Not run:  
content <- get_out_content("model.out")  
  
## End(Not run)
```

---

get_out_version	<i>Get the swmm version the .out file was generated with</i>
-----------------	--

---

**Description**

This function opens an .out file and extract the swmm version the file was generated with. It belongs to a set of helper functions which aim to simplify the work with .out files. The lifecycle of this function is considered experimental.

**Usage**

```
get_out_version(file = "")
```

**Arguments**

file	The file to be read.
------	----------------------

**Value**

A vector of type integer

**Examples**

```
## Not run:  
version <- get_out_version("model.out")  
  
## End(Not run)
```

---

inp_to_files	<i>Convert SWMM's .inp to .shp and txt files</i>
--------------	--

---

**Description**

Convert SWMM's .inp to .shp and txt files

**Usage**

```
inp_to_files(x, name, path_out = getwd())
```

**Arguments**

x	An object of class inp.
name	Give a name for the current model, e.g. "Example1".
path_out	Writeable directory name where to save the converted files. Folders: dat, shp and txt will be created if not existent. Default is the current working directory of the R process.

**Value**

.dat, .shp and/or .txt files.

---

read_inp	<i>Read SWMM's .inp file</i>
----------	------------------------------

---

**Description**

Reads a SWMM .inp file and creates a list with corresponding SWMM sections.

**Usage**

```
read_inp(x, rm.comment = TRUE, ...)
```

**Arguments**

x	Name (incl. path) to an input file.
rm.comment	Should lines with comments starting with a ";" be discarded?
...	optional arguments passed to <a href="#">read_lines</a> .

**Value**

An object of class inp

**Examples**

```
## Not run:
list_of_inp_sections <- read_inp("model.inp")

## End(Not run)
```

---

read_lid_rpt	<i>Read SWMM's LID Report File</i>
--------------	------------------------------------

---

**Description**

Reads a SWMM's LID Report File and returns a tibble

**Usage**

```
read_lid_rpt(x, return_xts = TRUE, ...)
```

**Arguments**

x	Name (incl. path) to a LID report file.
return_xts	logical. Sets the return type. If set to TRUE, xts objects are returned, FALSE gives tibbles.
...	optional arguments passed to <a href="#">read_table2</a>

**Value**

A tibble or xts object

**Examples**

```
## Not run:  
tbl_lid_rpt <- read_lid_rpt("lid_rpt.txt")  
  
## End(Not run)
```

---

read_out	<i>Read time series data from SWMM's .out file</i>
----------	--

---

**Description**

Reads the binary output ('.out') generated by the stormwater management model 'SWMM' and creates a list of xts-objects.

**Usage**

```
read_out(file = "", iType = NULL, object_name = NULL, vIndex = NULL)
```

**Arguments**

file	The file to be read.
iType	Sets the result type: 0 for Subcatchments, 1 for nodes, 2 for links, 3 for system variables. Leave empty for retrieving elements available.
object_name	Sets the objects of which time series data is returned. Leave empty for retrieving elements available.
vIndex	Sets the variables to be read (s. Details). Leave empty for retrieving elements available.

**Details**

vIndex depends on the result type. Choices are...

for each **subcatchment** variable:

- 0 for rainfall rate (in/hr or mm/hr),
- 1 for snow depth (inches or millimeters),
- 2 for evaporation loss (in/day or mm/day),
- 3 for infiltration loss (in/hr or mm/hr),
- 4 for runoff flow (flow units),
- 5 for groundwater flow into the drainage network (flow units),
- 6 for groundwater elevation (ft or m),
- 7 for soil moisture in the unsaturated groundwater zone (volume fraction),
- 7 + N for washoff concentration of each pollutant (mass/liter).

for each **node** variable:

- 0 for water depth (ft or m above the node invert elevation),
- 1 for hydraulic head (ft or m, absolute elevation per vertical datum),
- 2 for stored water volume (including ponded water, ft<sup>3</sup> or m<sup>3</sup>),
- 3 for lateral inflow (runoff + all other external inflows, in flow units),
- 4 for total inflow (lateral inflow + upstream inflows, in flow units),
- 5 for surface flooding (excess overflow when the node is at full depth, in flow units),
- 5 + N for concentration of each pollutant after any treatment (mass/liter),

for each **link** variable:

- 0 for flow rate (flow units),
- 1 for average water depth (ft or m),
- 2 for flow velocity (ft/s or m/s),
- 3 for volume of water (ft<sup>3</sup> or m<sup>3</sup>),
- 4 for capacity (fraction of full area filled by flow for conduits; control setting for pumps and regulators),
- 4 + N for concentration of each pollutant (mass/liter),

for each **system-wide** variable:

- 0 for air temperature (deg. F or deg. C),
- 1 for total rainfall (in/hr or mm/hr),
- 2 for total snow depth (inches or millimeters),
- 3 for average losses (in/hr or mm/hr),
- 4 for total runoff (flow units),
- 5 for total dry weather inflow (flow units),
- 6 for total groundwater inflow (flow units),
- 7 for total RDII inflow (flow units),
- 8 for total external inflow (flow units),
- 9 for total direct inflow (flow units),
- 10 for total external flooding (flow units),
- 11 for total outflow from outfalls (flow units),
- 12 for total nodal storage volume ( ft<sup>3</sup> or m<sup>3</sup>),
- 13 for potential evaporation (in/day or mm/day),
- 14 for actual evaporation (in/day or mm/day).

### Value

A list of xts-objects.

### See Also

[xts](#).

### Examples

```
## Not run:  
xts_list_of_results <- read_out("model.out")  
  
## End(Not run)
```

---

read\_rpt

*Read SWMM's .rpt file*

---

### Description

Reads a SWMM .rpt file and creates a list with corresponding results sections.

### Usage

```
read_rpt(x, ...)
```

**Arguments**

x                    Name (incl. path) to an report file.  
 ...                  optional arguments passed to `read_lines`

**Value**

An object of class `rpt`

**Examples**

```
## Not run:
list_of_rpt_results <- read_rpt("model.rpt")

## End(Not run)
```

---

run\_swmm

*Initiate a simulation run*

---

**Description**

This function runs a swmm inp file. If rpt and out files are not specified files are automatically created in the same directory of the inp file.

**Usage**

```
run_swmm(inp, rpt = NULL, out = NULL, exec = NULL, stdout = "", wait = TRUE)
```

**Arguments**

inp                  Name and path to an input file.  
 rpt                  Name and path to a report file.  
 out                  Name and path to an out file.  
 exec                Name and path to swmm5 executable. If not manually set, the following paths are looked up when package gets loaded: windows: "C:/Program Files (x86)/EPA SWMM 5.X.XXX/swmm5.exe" not windows: "/usr/local/bin/swmm5", "/usr/bin/swmm5"  
 stdout              where output to 'stdout' or 'stderr' should be sent. Possible values are "", to the R console (the default), NULL or FALSE (discard output), TRUE (capture the output in a character vector) or a character string naming a file.  
 wait                a logical (not NA) indicating whether the R interpreter should wait for the command to finish, or run it asynchronously. This will be ignored (and the interpreter will always wait) if stdout = TRUE or stderr = TRUE. When running the command asynchronously, no output will be displayed on the Rgui console in Windows (it will be dropped, instead).

**Details**

The path to a swmm5 executable is read by calling `'getOption("swmmr.exec")'`.

**Examples**

```
## Not run:
result <- run_swmm("model.inp")

## End(Not run)
```

shp\_to\_inp

*Convert \*.shp files to SWMM's \*.inp file***Description**

Reads \*.shp files and other information needed for SWMM's \*.inp file and returns a list of class inp. If paths are not specified default values are taken.

**Usage**

```
shp_to_inp(
  path_options = NULL,
  path_polygon = NULL,
  subcatchment_typologies = NULL,
  path_point = NULL,
  junction_parameters = NULL,
  path_outfall = NULL,
  path_line = NULL,
  conduit_material = NULL,
  path_timeseries = NULL,
  infiltration = NULL,
  path_pumps = NULL,
  path_pump_curve = NULL,
  path_weirs = NULL,
  path_storage = NULL,
  path_storage_curve = NULL
)
```

**Arguments**

- path\_options** Name (incl. path) to a .txt file with SWMM sections. Write section name in lower case and in squared brackets. The following sections are allowed: options, report, raingages, evaporation, pollutant, landuse, buildup, washoff, coverages.
- path\_polygon** Name (incl. path) to a .shp file with polygons features. At least the following subcatchment related columns must be specified: Name, Outlet, Area, RouteTo.
- subcatchment\_typologies** R data.frame or tibble with further subcatchment related parameters. If subcatchment\_typologies is given, polygon feature has to include a column named Type. Parameters defined in subcatchment\_typologies parameters are merged to subcatchments by Type.

path_point	Name (incl. path) to a .shp file with point features. At least the following junction related columns must be specified: Name, Bottom and Top or Ymax.
junction_parameters	R data.frame or tibble with further junction related parameters (e.g. Surcharge depth).
path_outfall	Name (incl. path) to a .shp file with point features. At least the following outfall related columns must be specified: Name, Bottom, Type, Gated.
path_line	Name (incl. path) to a .shp file with line features. At least the following conduit related columns must be specified: Name, Length, Type, FromNode, ToNode, OutOffset, Geom1.
conduit_material	R data.frame or tibble with further conduit related parameters (e.g. roughness). If conduit_material is given, line feature has to include a column named Material. Parameters defined in conduit_material parameters are merged to conduits by Material.
path_timeseries	Name (incl. path) to a .dat file with a timeseries in SWMM format.
infiltration	R data.frame or tibble with infiltration parameters related to soil properties. If infiltration is given, polygon feature has to include a column named soil. Infiltration parameters are merged to subcatchments by soil name.
path_pumps	Name (incl. path) to a .shp file with line features. All parameters must be given: Name, FromNode, ToNode, Pcurve, status, Startup, Shutoff.
path_pump_curve	Name (incl. path) to a .txt file with pump curve information. Having the following structure: "Name of pump" "PUMP1-4" "x" "y", without header.
path_weirs	Name (incl. path) to a .shp file with line features. All parameters must be given: Name, FromNode, ToNode, Type, CrestHt, Cd, Gated, EC, Cd2, Sur.
path_storage	Name (incl. path) to a .txt file with storage curve information. Having the following structure: "Name of storage" "Storage" "x" "y", without header.
path_storage_curve	Name (incl. path) to a .txt file with storage curve information. Having the following structure: "Name of storage unit" "Storage" "x" "y", without header.

**Value**

A list of class inp.

---

summary.inp

*Show summary of a swmm model structure*

---

**Description**

Show summary of a swmm model structure

**Usage**

```
## S3 method for class 'inp'
summary(object, ...)
```

**Arguments**

```
object      An object of class 'inp', created by read\_inp.
...         Additional parameters (currently ignored).
```

**Value**

The input is returned invisibly.

**Examples**

```
## Not run:
x <- read_inp("model.inp")
summary(x)

## End(Not run)
```

---

swmmr

swmmr *package*


---

**Description**

R Interface for US EPA's SWMM

---

write\_inp

*Write SWMM's .inp file*


---

**Description**

Writes an inp object to disk which can be read and run by SWMM.

**Usage**

```
write_inp(x, file)
```

**Arguments**

```
x           An object of class 'inp', created by read\_inp.
file        either a character string naming a file or a connection open for writing. ""
            indicates output to the console.
```

**Examples**

```
## Not run:  
  write_inp(inp, "~/model.inp")  
  
## End(Not run)
```

# Index

autoplot.inp, [2](#)

connection, [13](#)  
convert\_to\_sf, [3](#)

geom\_sf, [2](#)  
get\_out\_content, [4](#)  
get\_out\_version, [5](#)

inp\_to\_files, [5](#)  
inp\_to\_sf, [2](#)  
inp\_to\_sf (convert\_to\_sf), [3](#)

junctions\_to\_sf (convert\_to\_sf), [3](#)

links\_to\_sf (convert\_to\_sf), [3](#)

orifices\_to\_sf (convert\_to\_sf), [3](#)  
outfalls\_to\_sf (convert\_to\_sf), [3](#)

pumps\_to\_sf (convert\_to\_sf), [3](#)

raingages\_to\_sf (convert\_to\_sf), [3](#)  
read\_inp, [4](#), [6](#), [13](#)  
read\_lid\_rpt, [7](#)  
read\_lines, [6](#), [10](#)  
read\_out, [7](#)  
read\_rpt, [9](#)  
read\_table2, [7](#)  
run\_swmm, [10](#)

sf, [4](#)  
shp\_to\_inp, [11](#)  
storages\_to\_sf (convert\_to\_sf), [3](#)  
subcatchments\_to\_sf (convert\_to\_sf), [3](#)  
summary.inp, [12](#)  
swmmr, [13](#)

weirs\_to\_sf (convert\_to\_sf), [3](#)  
write\_inp, [13](#)

xts, [9](#)