Package: kwb.mia.iw (via r-universe)

August 20, 2024

Title InfoWorks related functions in KWB project MIA-CSO

Version 0.2.4

Description Calculation of file sizes of InfoWorks result csv-files
exported from InfoWorks.

License MIT + file LICENSE

Imports kwb.base, kwb.datetime, kwb.db, kwb.event, kwb.plot, kwb.utils

Remotes github::kwb-r/kwb.base, github::kwb-r/kwb.datetime,
github::kwb-r/kwb.db, github::kwb-r/kwb.event,
github::kwb-r/kwb.plot, github::kwb-r/kwb.utils

Encoding UTF-8

RoxygenNote 7.1.1

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

RemoteUrl https://github.com/KWB-R/kwb.mia.iw

RemoteRef HEAD

RemoteSha ec5b61ebd51023340001223d32fbd8275a0b9cae

Contents

`hsCreateGerrisInputConstParsCsv`
Create Gerris Input File for Constant Parameters

Description

Create Gerris input file containing boundary conditions for non-simulated water quality parameters

Usage

```
hsCreateGerrisInputConstParsCsv(
  csvExample,
  csvOut = NULL,
  constParVals = hsGerrisConstParsMia(),
  dbg = TRUE
)
```

Arguments

`csvExample`	path to example csv file containing names of Gerris boundary conditions in first row, starting in second column (first column is timestamp)
`csvOut`	full path to csv file to which non-simulated parameters are to be written. If missing, this function returns a character matrix containing the file content.
`constParVals`	Data frame with columns *VO2*, *SI*, *VPH*, *VX0*, *VX02*, *ZOOIND*, *VKIGR*, *LF*, *MW*, *VNO2*, *VNO3*, *CA*, *CHLA*, *ANTBL* containing the values for non-simulated parameters in the first and only row. Default: result of hsGerrisConstParsMia().
`dbg`	if `TRUE`, debug messages are shown

hsCreateGerrisInputFile *Write File for Gerris Import*

Description

Write a file for Gerris import based on a database table generated by StatAnalysis-evaluation "IwToQSim".

Usage

```
hsCreateGerrisInputFile(
  mdb,
  tbl,
  gerrisParID,
  csv.dir,
  csv.file = paste0("forGerrisImport_", gerrisParID, ".csv"),
  writeFile = TRUE,
  tsFormat = .defaultTimeFormat(),
  subst.na = "",
  dbg = FALSE
)
```

Arguments

mdb	full path to MS Access database containing table with required data
tbl	name of table to be exported to csv file
gerrisParID	Gerris parameter ID: one of "OBSB", "OCSB", "VNH4", "GESN", "GELP", "GESP", "SS", "Q"
csv.dir	path to directory to which output file is to be written
csv.file	optional name for output file, default: "forGerrisImport_<gerrisParID>.csv"
writeFile	only if TRUE, a file is written, otherwise the corresponding data is only returned by this function but not written to file.
tsFormat	format of timestamp to be used in output file (%d = day, %m = month, %Y = year, %H = hour, %M = minute, %S = second)
subst.na	string value by which NULL values in table are subsubstituted in the output file
dbg	if TRUE, debug messages are shown

hsCreateGerrisInputFiles *Create Gerris Input Files*

Description

Create Gerris input files from database containing tables prepared by StatAna-Evaluation "eIwToQSim"

Usage

```
hsCreateGerrisInputFiles(
  mdb,
  csv.dir,
  tblQ = "tbl_Q_m3_s",
  tblBOD = "tbl_c_BOD_tot_mg_L",
  tblCOD = "tbl_c_COD_tot_mg_L",
  tblNH4N = "tbl_c_NH4N_mg_L",
  tblNges = "tbl_c_NGES_mg_L",
  tblTPdis = "tbl_c_TP_dis_mg_L",
  tblTPtot = "tbl_c_TP_tot_mg_L",
  tblTSS = "tbl_c_TSS_mg_L",
  csv.basename = "forGerrisImport",
  separate = TRUE,
  overall = TRUE,
  tsFormat = .defaultTimeFormat(),
  subst.na = "",
  dbg = FALSE
)
```

Arguments

mdb	full path to MS Access database (.mdb) containing tables prepared by StatAna-Evaluation "eIwToQSim"
csv.dir	output directory to which database tables are to be exported in CSV format
tblQ	name of table containing flows Q in m3/s; default: "tbl_Q_m3_s"
tblBOD	name of table containing BOD concentrations in mg/L; default: "tbl_c_BOD_tot_mg_L"
tblCOD	name of table containing COD concentrations in mg/L; default: "tbl_c_COD_tot_mg_L"
tblNH4N	name of table containing NH4-N concentrations in mg/L; default: "tbl_c_NH4N_mg_L"
tblNges	name of table containing N total concentrations in mg/L; default: "tbl_c_NGES_mg_L"
tblTPdis	name of table containing total P (dissolved) concentrations in mg/L; default: "tbl_c_TP_dis_mg_L"
tblTPtot	name of table containing total P concentrations in mg/L; default: "tbl_c_TP_tot_mg_L"
tblTSS	name of table containing TSS concentrations in mg/L; default: "tbl_c_TSS_mg_L"

csv.basename	basename of file(s) to be created in *csv.dir*; default: "forGerrisImport". For flows, "_Hydrax" will be appended to the basename, and for concentrations of water quality parameters "_QSim_<par>", where <par> is one of "OBSB", "OCSB", "VNH4", "GESN", "GELP", "GESP", "SS". To the file containing concentrations of all the water quality parameters "_QSim_all" is appended.
separate	if TRUE, one file per parameter is created.
overall	if TRUE, one file containing all parameters is created.
tsFormat	format of timestamp to be used in output file (%d = day, %m = month, %Y = year, %H = hour, %M = minute, %S = second)
subst.na	substitution value for NA values
dbg	if TRUE, debug messages are shown

---

hsDaysInFile *Number of Days in InfoWorks Result CSV File*

Description

Number of Days in InfoWorks Result CSV File

Usage

```
hsDaysInFile(bytesFile, bytesHeader, bytesRow, timestep)
```

Arguments

bytesFile	file length in bytes
bytesHeader	length of header line in bytes
bytesRow	length of data row in bytes
timestep	result timestep in seconds

Value

number of days "contained" in an InfoWorks result CSV file of size *bytesFile* with a header line of *bytesHeader* bytes length, each data row being *bytesRow* bytes long and a result timestep of *timestep* seconds.

hsFileSize *Size of InfoWorks Result CSV File*

Description

Size of an InfoWorks result CSV file in bytes

Usage

```
hsFileSize(nDays, bytesHeader, bytesRow, timestep, dbg = FALSE)
```

Arguments

nDays	number of days
bytesHeader	length of header line in bytes
bytesRow	length of data row in bytes
timestep	result timestep in seconds
dbg	if TRUE, debug messages are shown

Value

Size of an InfoWorks result CSV file over *nDays* with a result timestep of *timestep* seconds in bytes if the header file is *bytesHeader* and each data row is *bytesRow* bytes long.

hsGerrisConstBoundMatrix *Gerris Constant Boundary Matrix*

Description

Character matrix containing default values for non-simulated parameters

Usage

```
hsGerrisConstBoundMatrix(boundNames, constPars, tstamp.1, tstamp.2, tstamp.n)
```

Arguments

boundNames	names of locations at which boundary conditions are given
constPars	values of parameters that are treated as constant boundary conditions
tstamp.1	first timestamp
tstamp.2	second timestamp
tstamp.n	last timestamp

hsGerrisConstParsMia *Gerris Constant Parameters*

Description

Gerris/QSim boundary conditions that are treated as constant within MIA-CSO project. Parameter names according to definition in Gerris configuration file "GerrisParam.xml"

Usage

```
hsGerrisConstParsMia(
  VO2 = 0,
  SI = 0,
  VPH = 7.44,
  VX0 = 0,
  VX02 = 0,
  ZOOIND = 0,
  VKIGR = 0.33,
  LF = 293.33,
  MW = 1.39,
  VNO2 = 0.21,
  VNO3 = 1.1,
  CA = 36.04,
  CHLA = 0,
  ANTBL = 0.33
)
```

Arguments

VO2	"Sauerstoffgehalt" in mg/l
SI	"Silizium" in mg/l
VPH	"pH-Wert"
VX0	"Nitrosomonas" in mg/l
VX02	"Nitrobacter" in mg/l
ZOOIND	"Rotatorien/Zooplanktondichte" in Ind/l
VKIGR	"Kieselalgen/Anteil der Kieselalgen am Gesamt-Chlorophyll-a", 0..1
LF	Leitfaehigkeit" in mikro-S/cm
MW	"m-Wert" in mmol/l
VNO2	"Nitrit-N" in mg/l
VNO3	"Nitrat-N" in mg/l
CA	"Calcium" in mg/l
CHLA	"Chlorophyll-a" in mikro-g/l
ANTBL	"Blaualgen/Anteil der Blaualgen am Gesamt-Chlorophyll-a", 0..1

Value

Data frame with columns *VO2*, *SI*, *VPH*, *VX0*, *VX02*, *ZOOIND*, *VKIGR*, *LF*, *MW*, *VNO2*, *VNO3*, *CA*, *CHLA*, *ANTBL* containing the default values for non-simulated parameters in the first and only row

---

hsGetIwResult *Get Infoworks Result*

Description

Get Infoworks Result

Usage

```
hsGetIwResult(file, columns = NULL, skip.columns = "Seconds", dbg = FALSE)
```

Arguments

file	file
columns	defalut: NULL
skip.columns	default: "Seconds"
dbg	if TRUE, debug messages are shown

Value

data frame

---

hsGetIwResultAvgAboveZero *Get Infoworks Result Average Above Zero*

Description

Get Infoworks Result Average Above Zero

Usage

```
hsGetIwResultAvgAboveZero(src, type)
```

Arguments

src	either full path to mdb or to directory in which InfoWorks result files (csv) are located
type	one of "flow", "BOD_tot", "COD_tot", "NH4N", "TKN_tot", "TP_dis", "TP_tot", "TSS"

hsGetIwResultAvgFromCsv *Get Infoworks Result Average From CSV*

Description

Get Infoworks Result Average From CSV

Usage

```
hsGetIwResultAvgFromCsv(csvdir, type, qthreshold = 0.003, dbg = TRUE)
```

Arguments

csvdir	directory in which InfoWorks result files (csv) are located
type	one of "flow", "BOD_tot", "COD_tot", "NH4N", "TKN_tot", "TP_dis", "TP_tot", "TSS"
qthreshold	threshold for Q values
dbg	if TRUE, debug messages are shown

hsGetIwResultAvgFromMdb *Get Infoworks Result Average From MS Access Database*

Description

Get Infoworks Result Average From MS Access Database

Usage

```
hsGetIwResultAvgFromMdb(
  mdb,
  type,
  tblQ = "tbl_05_Q_bei_Ueberlauf_m3_s",
  tblL = paste("tbl_02_15minMittel_L", type, "kg_s", sep = "_"),
  skipCols = c("^kug$", "^overall$")
)
```

Arguments

mdb	mdb
type	type
tblQ	name of database table containing flows
tblL	name of database table containing loads
skipCols	vector of patterns matching names of columns to be skipped

hsIntegrals *Integrals*

Description

Calculates the integrals (sum of all value columns within event's time interval, multiplied with "signal width" of events)

Usage

```
hsIntegrals(data, evts, dbg = FALSE)
```

Arguments

data	time-series data with timestamp in first column
evts	event data describing events contained in *data*
dbg	if TRUE, debug messages are shown

Value

vector of integral values with one value per event. Length of vector corresponds to number of rows in *evts*

hsIwEventSummary *Volume and Mass Load per Event*

Description

Volume and Mass Load per Event

Usage

```
hsIwEventSummary(
  src,
  mfpars = c("flow", "BOD_tot", "COD_tot", "NH4N", "TKN_tot", "TP_dis", "TP_tot",
    "TSS")
)
```

Arguments

src	Data source; can be either path to mdb (filled by StatAna) or directory containing original csv files exported from InfoWorks
mfpars	Vector of parameter acronyms; Default: c("flow", "BOD_tot", "COD_tot", "NH4N", "TKN_tot", "TP_dis", "TP_tot", "TSS")

Value

list with event list *evts*, matrix *allIntegrals* containing volume and mass loads per event and data.frame *iwdata* containing all result data

---

`hsIwParNameMap` *Parameter Name Mapping*

Description

Mapping between parameter names used in table names in temporary mdb (written to by StatAna) and parameter acronyms used in InfoWorks result csv files.

Usage

```
hsIwParNameMap()
```

`hsIwPlot1` *Infoworks Plot 1*

Description

Infoworks Plot 1

Usage

```
hsIwPlot1(
  data,
  allIntegrals,
  evts,
  type = 2,
  basemain,
  legend.sort = FALSE,
  ylog = FALSE,
  yBottom = ifelse(ylog, 1e-04, 0),
  cex.legend = 0.55
)
```

Arguments

`data`	`data`
`allIntegrals`	`allIntegrals`
`evts`	`evts`
`type`	default: 2
`basemain`	`basemain`

legend.sort	default: FALSE
ylog	default: FALSE
yBottom	default: ifelse(ylog, 0.0001, 0)
cex.legend	expansion factor for legend texts

`hsIwPlot2` *Infoworks Plot 2*

Description

Infoworks Plot 2

Usage

```
hsIwPlot2(
  allIntegrals,
  evts,
  plotTotal = TRUE,
  plotEvents = FALSE,
  pars = c("TSS", "COD_tot", "BOD_tot", "TKN_tot", "NH4N", "TP_tot", "TP_dis")
)
```

Arguments

allIntegrals	allIntegrals
evts	evts
plotTotal	default: TRUE
plotEvents	default: FALSE
pars	default: c("TSS", "COD_tot", "BOD_tot", "TKN_tot", "NH4N", "TP_tot", "TP_dis")

`hsIwPlot3` *Infoworks Plot 3*

Description

Infoworks Plot 3

Usage

```
hsIwPlot3(allIntegrals, boxplot.range = 0)
```

Arguments

allIntegrals	allIntegrals
boxplot.range	this determines how far the plot whiskers extend out from the box. See argument range of barplot.

hsIwPlotAll *Infoworks Plot All*

Description

Infoworks Plot All

Usage

```
hsIwPlotAll(
  src,
  subtitle = "",
  mfpars = c("flow", "BOD_tot", "COD_tot", "NH4N", "TKN_tot", "TP_dis", "TP_tot",
    "TSS"),
  ylog = FALSE,
  type = 2,
  outpdf = "",
  outdir = "",
  legend.sort = FALSE,
  cex.legend = 0.55
)
```

Arguments

src	Data source; can be either path to mdb (filled by StatAna) or directory containing original csv files exported from InfoWorks
subtitle	subtitle
mfpars	Vector of parameter acronyms; Default: c("flow", "BOD_tot", "COD_tot", "NH4N", "TKN_tot", "TP_dis", "TP_tot", "TSS")
ylog	y axis logarithmic?
type	default: 2
outpdf	Path to output file (.pdf). Default: ""
outdir	Path to output directory. Default: ""
legend.sort	default: FALSE
cex.legend	expansion factor for legend texts

`hsIwResultFileSize` *InfoWorks Result CSV File Size*

Description

Size of an InfoWorks result CSV file depending on the simulated time period between *dateFirst* and *dateLast*, the result `timestep` *timestep* applied and the number *nDataCol* of data columns in the file.

Usage

```
hsIwResultFileSize(
  dateFirst,
  dateLast,
  timestep,
  nDataCol,
  bytesHeader = -1,
  tstamp = "yyyy-mm-dd hh:nn:ss",
  colWidth = 12,
  dbg = FALSE
)
```

Arguments

`dateFirst`	first date (day) to be simulated in ISO-format: *yyyy-mm-dd*
`dateLast`	last date (day) to be simulated in ISO-format: *yyyy-mm-dd*
`timestep`	result `timestep` in seconds
`nDataCol`	number of data columns (time-columns excluded) in the InfoWorks result CSV file
`bytesHeader`	length of header line in bytes
`tstamp`	string representing an example timestamp
`colWidth`	width of a data column in bytes
`dbg`	if `TRUE`, debug messages are shown

Value

List with elements *Bytes*, *kB*, *MB*, *GB* giving the requested file size in the according unit.

`hsNameFilter` *Name Filter*

Description

Include / exclude elements from vector by patterns

Usage

```
hsNameFilter(x, posPtrn = NULL, negPtrn = NULL)
```

Arguments

`x`	vector of character
`posPtrn`	pattern matching elements to be included
`negPtrn`	pattern matching elements to be excluded

Value

elements of *x* matching *posPtrn* and not matching *negPtrn*

`hsPlotIwFileSizeVsTsAndPeriod`
IW File Size vs. Time Step, Period

Description

Plot showing InfoWorks result CSV file size vs. different combinations of timestep and simulated time period

Usage

```
hsPlotIwFileSizeVsTsAndPeriod(
  nCols,
  colWidth = 12,
  tsFormat = "01.01.2011 00:00:00",
  bytesHeader = -1
)
```

Arguments

`nCols`	number of data columns in the InfoWorks result CSV file
`colWidth`	width of a data column in bytes
`tsFormat`	string representing a timestamp
`bytesHeader`	length of header line in bytes

hsRep *Repeat* elements *n-times*

Description

Repeat elements n-times

Usage

```
hsRep(elements, n)
```

Arguments

elements	vector of which each element is to be repeated n-times
n	number by which each element of *elements* is repeated

Value

vector in which each element of *elements* is repeated *n*-times

hsRowLen *Row Length in InfoWorks Result CSV File*

Description

length of a data row in the InfoWorks result CSV file in bytes

Usage

```
hsRowLen(colWidth, colNum, tsFormat = "2011-12-31 23:59:59")
```

Arguments

colWidth	width of a data column in bytes
colNum	number of data columns
tsFormat	string representing a timestamp

Value

Number of bytes needed for one row of an InfoWorks result CSV file with *colNum* data columns of *colWidth* bytes each and a timestamp column of format according to the example timestamp *tstamp*.

`hsTimeBarPlot` *Plot Type 1*

Description

Plot type 1: boxes with mean flow as `height` and event duration as width

Usage

```
hsTimeBarPlot(
  t1,
  t2,
  height,
  tlim = c(min(t1), max(t2)),
  ylim = NULL,
  ylog = TRUE,
  ymult = c(1, 2, 3, 5),
  main = "hsTimeBarPlot",
  tlab = "time",
  ylab = "height",
  time.format = .defaultDateFormat(),
  tlab.mindist = 86400,
  col = "grey",
  cex.all = 0.8,
  cex.legend = cex.all,
  cex.text = cex.all,
  cex.axis = cex.all,
  legend.values = height,
  legend.sort = TRUE,
  legend.format = .defaultLegendFormat(),
  legend.title = "Legend:",
  mar = c(6, 5, 5, 10)
)
```

Arguments

`t1`	begin times of bars
`t2`	end times of bars
`height`	bar heights
`tlim`	limits of time axis. Default: c(min(`t1`), max(`t2`))
`ylim`	limits of y axis, Default: c(min(`height`), max(`height`))
`ylog`	if TRUE, y axis is scaled logarithmically, else linearly
`ymult`	numbers to be multiplied by exponents of ten for labelling of y axis when *ylog* is TRUE
`main`	plot title

tlab	label of time axis
ylab	label of y axis
time.format	format of time axis labels
tlab.mindist	minimum "time distance" between time labels, in seconds
col	bar colour
cex.all	general expansion factor to be applied to cex.legend, cex.text, cex.axis, if not given
cex.legend	expansion factor for legend texts
cex.text	expansion factor of texts within plot (bar numbers)
cex.axis	expansion factor of axis labels
legend.values	values to be shown in legend, default: height values
legend.sort	if TRUE, legend is sorted decreasingly by legend.values
legend.format	format string to be used by sprintf for generation of legend entries from legend values, e.g. "V = %8.0f m3"
legend.title	legend title
mar	plot margins

---

hsTsInFile *Result Time Step in InfoWorks Result CSV File*

Description

Result Time Step in InfoWorks Result CSV File

Usage

```
hsTsInFile(bytesFile, bytesHeader, bytesRow, nDays)
```

Arguments

bytesFile	file length in bytes
bytesHeader	length of header line in bytes
bytesRow	lengh of data row in bytes
nDays	number of days

Value

Returns the (possibly) applied result timestep of an InfoWorks simulation depending on the maximal allowed InfoWorks result CSV file size *bytesFile* in bytes, on the number *nDays* of days to simulate and on the size *bytesHeader* and *bytesRow* of the header and of a data line, respectively.

Index