Package: kwb.wtaq (via r-universe)

August 30, 2024

Title Interface to WTAQ Drawdown Model (http://water.usgs.gov/ogw/wtaq/)

Version 0.3.0

Description Functions enabling the writing of WTAQ input files, running of WTAQ and reading of WTAQ output files.

License MIT + file LICENSE

URL https://github.com/KWB-R/kwb.wtaq

BugReports https://github.com/KWB-R/kwb.wtaq/issues

Depends R (>= 3.0.0)

Imports kwb.utils, lattice, hydroGOF, plotrix, reshape

Suggests rmarkdown, knitr, testthat

VignetteBuilder knitr

Remotes github::KWB-R/kwb.utils

Biarch TRUE

BinaryFiles extdata/wtaq.2.1.exe

Encoding UTF-8

LazyLoad yes

RoxygenNote 7.2.0

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

RemoteUrl https://github.com/KWB-R/kwb.wtaq

RemoteRef HEAD

RemoteSha f05c8cb30c48051c7f4b856c40571bae08facc27

Contents

checkModelFit 3
drawdowns 3
fitnessAquifer 4
modelFitness 4

modelFitnessAggregated 5
owConfigurationToWtConfiguration 5
owConfigure 6
owConfigureWell 7
owGetDrawdowns 8
owPlotConfiguration 9
owPlotDrawdowns 10
owRandomConfiguration 10
owRandomWellfield 11
owSuperposeDrawdowns 12
owWellDistances 13
print.wtaqConfiguration 13
wtCalibrate 14
wtCheckConfiguration 15
wtConfigurationExample2 16
wtConfigurationExample3 18
wtConfigurationToOwConfiguration 19
wtConfigure 20
wtConfigureAquifer 21
wtConfiguredDistances 22
wtConfigureDrainage 22
wtConfiguredWellnames 23
wtConfigureGeneral 24
wtConfigureObservationWell 25
wtConfigurePumpwell 27
wtConfigureSolution 28
wtConfigureTimes 30
wtDefaultConfiguration 31
wtDefaultConfigurationSolution 31
wtInputFileLines 32
wtPlotConfiguration 33
wtPlotResult 34
wtReadInputFile 35
wtReadPlotFile 35
wtRepairConfiguration 38
wtRunConfiguration 38
wtRunInputFile 39
wtSetParameter 40
wtSetParameters 41

Index **42**

`checkModelFit` *Calibration of aquifer parameter "hkr"*

Description

Calibration of aquifer parameter "hkr"

Usage

```
checkModelFit(wtaqResult, wells = ".*")
```

Arguments

`wtaqResult` # in WTAQ result in "long format"

`wells` regular expression of wells to be included (e.g. ".*" for all, PW: only production well, OW: all observation wells)

Author(s)

Hauke Sonnenberg

`drawdowns` *Example data of measured drawdowns in a drinking water well field*

Description

Example data of measured drawdowns in a drinking water well field.

Usage

```
drawdowns
```

Format

List of five data frames each of which represents the timeseries of drawdowns measured at five different drinking water wells. The column *time.in.seconds* is the time of operation in seconds and the columns *W1*, *W2*, *W3*, *W4*, and *W5* correspond to the drawdowns at the different Wells W1 to W5.

fitnessAquifer *fitnessAquifer*

Description

fitnessAquifer

Usage

```
fitnessAquifer(parameterValue, parameterName, configuration,


    wellPattern = "*")
```

Arguments

```
parameterValue
parameterName
configuration
wellPattern
```

Author(s)

Hauke Sonnenberg

modelFitness *modelFitness*

Description

modelFitness

Usage

```
modelFitness(wtaqResult, wells = "*")
```

Arguments

`wtaqResult`	# in "long format"
`wells`	regular expression of wells to be included (e.g. * for all, PW: only production well, OW: all observation wells)

Author(s)

Hauke Sonnenberg

modelFitnessAggregated *modelFitnessAggregated*

Description

modelFitnessAggregated

Usage

```
modelFitnessAggregated(wtaqResult, wellPattern = "*")
```

Arguments

`wtaqResult`
`wellPattern`

Author(s)

Hauke Sonnenberg

owConfigurationToWtConfiguration *WTAQ configuration from Optiwells configuration*

Description

convert Optiwells configuration to WTAQ configuration with one well
being selected as pumping well

Usage

```
owConfigurationToWtConfiguration(owConfiguration, activeWell,


    Q, times = NULL, solution = wtConfigureSolution())
```

Arguments

`owConfiguration`	Optiwells configuration as created with `owConfigure`
`activeWell`	number of active well = row number in *owConfiguration$wellfield*
`Q`	discharge of active well
`times`	vector of times for which drawdowns are to be calculated
`solution`	List of solution parameters as retrieved by `wtConfigureSolution`.

Value

WTAQ configuration as e.g. returned by `wtConfigure`

Author(s)

Hauke Sonnenberg

See Also

`wtConfigurationToOwConfiguration`

`owConfigure` *Definition of an Optiwells well field configuration*

Description

Returns a user-defined Optiwells configuration on which e.g. `owGetDrawdowns` can be run.

Usage

```
owConfigure(wellfield = NULL, aquifer = NULL, drainage = NULL)
```

Arguments

`wellfield`	Optiwells wellfield configuration. Data frame
`aquifer`	WTAQ aquifer configuration as returned by `wtConfigureAquifer`
`drainage`	WTAQ drainage configuration as returned by `wtConfigureDrainage`

Value

list with elements *wellfield* (configuration of well characteristics), *aquifer* (WTAQ aquifer configuration) and *drainage* (WTAQ drainage configuration).

Author(s)

Hauke Sonnenberg

See Also

`owRandomConfiguration`

owConfigureWell *Configure well(s) for Optiwells configuration*

Description

Configure well(s) for Optiwells configuration

Usage

```
owConfigureWell(wellName, x, y, r, z1, z2, sw)
```

Arguments

wellName	well identifier(s). Must be unique within a well field.
x	x coordinate(s) of well centre(s)
y	y coordinate(s) of well centre(s)
r	well radius(es)
z1	Depth(s) below top of aquifer or initial water table to the top of the screened interval of the well(s), in units of length.
z2	Depth(s) below top of aquifer or initial water table to the bottom of the screened interval of the well(s), in units of length.
sw	Well-bore skin parameter(s), dimensionless.

Value

data frame with one row (per well) and columns elements *wellName*,

x, *y*, *r*, *z1*, *z2*, *sw* defining a

well.

Author(s)

Hauke Sonnenberg

owGetDrawdowns *Drawdowns for Optiwells configuration and pump discharges*

Description

Calculate drawdowns for given Optiwells configuration and pump discharges

Usage

```
owGetDrawdowns(owConfiguration, Q, times = NULL, solution = NULL,


    to.matrix = TRUE, ...)
```

Arguments

`owConfiguration`
: Optiwells configuration as e.g. retrieved by `owRandomConfiguration`.

`Q`
: vector of discharges at the wells in the order of wells in owConfiguration$wellfield

`times`
: vector of times for which drawdowns are to be calculated

`solution`
: List of solution parameters as retrieved by `wtConfigureSolution`. If not specified, a default configuration, as retrieved by `wtDefaultConfigurationSolution` is used.

`to.matrix`
: if TRUE, the results returned by `wtRunConfiguration` that come in "long" form (the drawdowns for each well appear in blocks one below each other with a column "WELL" indicating the well name) are converted to "matrix" (= wide) form in which the calculated drawdowns appear in columns one beside the other.

`...`
: additional arguments passed to `wtRunConfiguration`, such as *show.results* or *dbg*, see there.

Value

list with as many elements as there are wells defined in the Optiwells configuration (list element *wellfield* of *owConfiguration*). For each pumping well (well with Q > 0) the list element at the corresponding position contains a data frame holding the drawdowns calculated by WTAQ with that well pumping alone and the other wells being observation wells. For each non-pumping well (well with Q = 0) the list element will be NULL. This list can then be used to calculate the superposition of drawdowns.

Author(s)

Hauke Sonnenberg

See Also

owSuperposeDrawdowns

---

owPlotConfiguration *Plot Optiwells configuration*

Description

Plot Optiwells configuration

Usage

```
owPlotConfiguration(owConfiguration, referenceWell = -1, view = c("top",

    "side"), col = rainbow(nrow(owConfiguration$wellfield)),

    onePage = TRUE, ...)
```

Arguments

`owConfiguration`	Optiwells configuration as retrieved by `owConfigure`
`referenceWell`	Number of reference well (according to row number in owConfiguration$wellfield or -1 (no reference well). If a reference well is specified, it will be plotted on the left-hand side of the sideview and the top view will be centered around that well.
`view`	vector of c("top", "side"). Determines whether to plot top view or side view or both.
`col`	colours. Default: rainbow colours
`onePage`	if TRUE, top view and side view appear below each other on one and the same page
`...`	

Author(s)

Hauke Sonnenberg

owPlotDrawdowns *plot drawdown (under construction!)*

Description

plot drawdown (under construction!)

Usage

```
owPlotDrawdowns(drawdownList, topview = TRUE, overlay = FALSE,


    fixview = FALSE)
```

Arguments

`drawdownList`

`topview`

`overlay`

`fixview`

Author(s)

Hauke Sonnenberg

owRandomConfiguration *Random Optiwells well field configuration*

Description

Returns a random Optiwells configuration on which e.g. `owGetDrawdowns` can be run.

Usage

```
owRandomConfiguration(numberOfWells, bb = 80, ...)
```

Arguments

`numberOfWells`	number of wells in the well field (must be a number between 1 and 25).
`bb`	Thickness or saturated thickness of aquifer at beginning of simulation, in units of length.
`...`	additional arguments that are passed to owRandomWellfield, such as: *rmean*, *z1mean*, *z2mean*, *swmean*, *rsd*, *z1sd*, *z2sd*, *swsd*, *digits*, see `owRandomWellfield`

Value

list with elements *wellfield* (configuration of well characteristics), *aquifer* (WTAQ aquifer configuration) and *drainage* (WTAQ drainage configuration), just as created by owConfigure

Author(s)

Hauke Sonnenberg

See Also

owRandomWellfield, owConfigure

---

owRandomWellfield *Generation of a random wellfield*

Description

Generation of a random wellfield

Usage

```
owRandomWellfield(numberOfWells, rmean, z1mean, z2mean, swmean,

    rsd = 0.1 * rmean, z1sd = 0.1 * z1mean, z2sd = 0.1 * z2mean,

    swsd = 0.1 * swmean, digits = 1)
```

Arguments

numberOfWells	number of wells in the well field (must be a number between 1 and 25).
rmean	mean value of normal distribution used to generate random values for 'r' (well radius)
z1mean	mean value of normal distribution used to generate random values for 'z1' (depth of begin of screen)
z2mean	mean value of normal distribution used to generate random values for 'z2' (depth of end of screen)
swmean	mean value of normal distribution used to generate random values for 'sw' (well-bore skin parameter)
rsd	standard deviation of normal distribution used to generate random values for 'r' (well radius). Default: 0.1 * rmean

`z1sd` standard deviation of normal distribution used to generate random values for 'z1'-values' (depth of begin of screen). Default: 0.1 * z1mean

`z2sd` standard deviation of normal distribution used to generate random values for 'z2'-values' (depth of end of screen). Default: 0.1 * z2mean

`swsd` standard deviation of normal distribution used to generate random values for 'sw' (well-bore skin parameter). Default: 0.1 * swmean

`digits` number of decimal digits for values of r, z1, z2. Default: 1

Value

data frame with each row representing a well of the well field and the columns representing the well properties: *wellName* (well name), *x* (x-coordinate of well), *y-coordinate of well*, *r* (well radius), *z1* (depth of begin of well screen), *z2* (depth of end of well screen), *sw* (well-bore skin parameter)

Author(s)

Hauke Sonnenberg

`owSuperposeDrawdowns` *superpose drawdowns*

Description

superpose drawdowns in list of drawdowns as returned by `owGetDrawdowns`.

Usage

```
owSuperposeDrawdowns(drawdownList, dbg = FALSE)
```

Arguments

`drawdownList` list of drawdowns as returned by `owGetDrawdowns`

`dbg` if TRUE, debug messages are shown. Default: FALSE

Value

data frame

Author(s)

Hauke Sonnenberg

See Also

owGetDrawdowns

---

owWellDistances — *Distances between wells*

Description

Calculates distances between wells based on coordinates given in Optiwells configuration of the wellfield

Usage

```
owWellDistances(wellfield, referenceWell = 1)
```

Arguments

`wellfield`	data frame as e.g. retrieved by `owRandomWellfield` with each row representing a well of the well field.
`referenceWell`	Number of reference well corresponding with row number in *wellfield*. For each well the distance to the reference well is calculated. Default: 1

Value

vector of numeric representing the distances of each well to a reference well (determined by *referenceWell*)

Author(s)

Hauke Sonnenberg

---

print.wtaqConfiguration — *Print WTAQ configuration*

Description

Print WTAQ configuration

Usage

```
## S3 method for class 'wtaqConfiguration'
print(x, ...)
```

Arguments

`x`

`...`

Author(s)

Hauke Sonnenberg

`wtCalibrate` *Simple calibration procedure*

Description

Simple calibration procedure

Usage

```
wtCalibrate(configuration,
      parameterName = "hkr",
      startvalue = 0.02,
      endvalue = 1e-05,
      changevalue = 1e-04,
      wells = "OW4",
      fitnessCriteria = data.frame(
      name = c("RMSE", "PBIAS", "PBIAS", "R2"),
      vals = c(0.03, 5, -5, 0.95),
      condition = c("<=", "<=", ">=", ">=")))
```

Arguments

`configuration`

`parameterName`

`startvalue`

`endvalue`

`changevalue`

`wells`

`fitnessCriteria`
stat. parameters for for model fit evaluation

Author(s)

Hauke Sonnenberg

`wtCheckConfiguration` *check WTAQ configuration*

Description

Checks a WTAQ configuration for its structure

Usage

```
wtCheckConfiguration(configuration, part = "complete", dbg = TRUE)
```

Arguments

`configuration`	WTAQ configuration (complete) or sub-configuration (general, aquifer, rainage, times, solution, pumpwell, obswell)
`part`	one of c("complete", "general", "aquifer", "drainage", "times", "solution", "pumpwell", "obswell")
`dbg`	if TRUE, debug messages are shown, else not.

Value

Error message or "" if no error occurred

Author(s)

Hauke Sonnenberg

Examples

```
# Generate default configuration


cconf <- wtConfigure()




# Check (complete) configuration. No error -> ok
```

```
wtCheckConfiguration(cconf)


# Generate default observation well configuration
oconf <- wtConfigureObservationWell()


# Check ”obswell” configuration. No error -> ok
wtCheckConfiguration(oconf, ”obswell”)


# Set obswells in complete configuration (forgetting that obswells must be
# a list!):
cconf$obswells <- oconf


# Check (complete) configuration again
# -> Error when ”Checking config of observation well 1”.
wtCheckConfiguration(cconf)
```

wtConfigurationExample2

WTAQ configuration representing WTAQ example 2

Description

WTAQ configuration corresponding to sample problem 2 of WTAQ distribution

Usage

```
wtConfigurationExample2()
```

Value

list with elements *general*, *aquifer*, *drainage*,
times, *solution*, *pumpwell*, *obswells*,
representing a full WTAQ configuration.

Author(s)

Hauke Sonnenberg

See Also

`wtConfigurationExample3`, `wtConfigure`

Examples

```
# Get configuration of sample problem 2 of WTAQ distribution

cnf <- wtConfigurationExample2()



# Print formatted output of configuration

print(cnf)



# Plot wellfield profile of configuration

wtPlotConfiguration(cnf)
```

`wtConfigurationExample3`

WTAQ configuration representing WTAQ example 3

Description

WTAQ configuration corresponding to sample problem 3 of WTAQ distribution

Usage

```
wtConfigurationExample3()
```

Value

list with elements *general*, *aquifer*, *drainage*,
times, *solution*, *pumpwell*, *obswells*,
representing a full WTAQ configuration.

Author(s)

Hauke Sonnenberg

See Also

wtConfigurationExample2, wtConfigure

Examples

```

# Get configuration of sample problem 3 of WTAQ distribution


cnf <- wtConfigurationExample3()




# Print formatted output of configuration


print(cnf)




# Plot wellfield profile of configuration
```

```
wtPlotConfiguration(cnf)
```

---

wtConfigurationToOwConfiguration
wtConfigurationToOwConfiguration

Description

Generate Optiwells configuration from WTAQ configuration

Usage

```
wtConfigurationToOwConfiguration(wtaqConfiguration)
```

Arguments

`wtaqConfiguration`
WTAQ configuration as retrieved by `wtConfigure`

Value

list with elements *wellfield* (configuration of well characteristics), *aquifer* (WTAQ aquifer configuration) and *drainage* (WTAQ drainage configuration), just as creaetd by `owConfigure`. The list elements *aquifer* and *drainage* are copied from the given WTAQ configuration. The wellfield configuration is generated from the information on the pumping well and on the observation wells given in that same configuration.

Author(s)

Hauke Sonnenberg

See Also

`owConfigurationToWtConfiguration`

wtConfigure *Define parameter values for WTAQ model run*

Description

Define parameter values for WTAQ model run

Usage

```
wtConfigure(general = wtConfigureGeneral(), aquifer = wtConfigureAquifer(),


    drainage = wtConfigureDrainage(), times = wtConfigureTimes(),


    solution = wtConfigureSolution(), pumpwell = wtConfigurePumpwell(),


    obswells = list(wtConfigureObservationWell(obname = "obs1"),


        wtConfigureObservationWell(obname = "obs2", r = 50)))
```

Arguments

`general`	List of general parameters as retrieved by `wtConfigureGeneral`.
`aquifer`	List of aquifer parameters as retrieved by `wtConfigureAquifer`.
`drainage`	List of drainage parameters as retrieved by `wtConfigureDrainage`.
`times`	List of simulation time parameters as retrieved by `wtConfigureTimes`.
`solution`	List of solution parameters as retrieved by `wtConfigureSolution`.
`pumpwell`	List of pumped well parameters as retrieved by `wtConfigurePumpwell`.
`obswells`	List of lists of observation well parameters each of which can be retrieved by `wtConfigureObservationWell`.

Value

list with elements *general*, *aquifer*, *drainage*,
times, *solution*, *pumpwell*, *obswells*,
representing a full WTAQ configuration. For the different elements see the
descriptions in the Arguments section.

Author(s)

Hauke Sonnenberg

See Also

wtConfigurationExample2, wtConfigurationExample3, wtRunConfiguration

---

wtConfigureAquifer *Aquifer parameter values*

Description

Define aquifer-related WTAQ parameters

Usage

```
wtConfigureAquifer(aqtype = "WATER TABLE", bb = 20, hkr = 0.0041,


    hkz = 0.00174, ss = 3.76e-05, sy = 0.25)
```

Arguments

aqtype — Type of aquifer being simulated. Two options are provided:
AQTYPE = CONFINED or
AQTYPE = WATER TABLE

bb — Thickness or saturated thickness of aquifer at beginning of simulation, in units of length.

hkr — Horizontal hydraulic conductivity of aquifer, in units of length per time.

hkz — Vertical hydraulic conductivity of aquifer, in units of length per time.

ss — Specific storage of aquifer, in units of inverse length.

sy — Specific yield of aquifer, dimensionless. Enter 0 if
AQTYPE = CONFINED.

Value

list with elements *aqtype*, *bb*, *hkr*, *hkz*,
ss, *sy*, representing WTAQ parameters related to the aquifer.
See descriptions in Arguments section.

Author(s)

Hauke Sonnenberg

See Also

wtConfigure

wtConfiguredDistances *distances of observation wells*

Description

Vector of distances between pumping well and observation wells (including 0 as first distance!)

Usage

```
wtConfiguredDistances(configuration)
```

Arguments

`configuration`	WTAQ configuration as retrieved by `wtConfigure`.

Value

named vector of numeric containing zero as first element and the distances between pumping well and observation wells as following elements, in the order or their definition in the WTAQ configuration. The names of the elements correspond to the well names as retrieved also with `wtConfiguredWellnames`

Author(s)

Hauke Sonnenberg

wtConfigureDrainage *Drainage parameter values*

Description

Define drainage-related WTAQ parameters

Usage

```
wtConfigureDrainage(idra = 2, alpha = c(), acc = 5, akk = 31.7,

    amm = 100, axmm = 10)
```

Arguments

`idra` Type of drainage at water table. Enter 0 if AQTYPE = CONFINED.
Three options are provided:
IDRA = 0: Instantaneous drainage.
IDRA = 1: Gradual drainage.
IDRA = 2: Drainage with unsaturated-zone characterization.

`alpha` Only used if idra = 1. Drainage constants, in units of inverse time.
Maximum of 5 values is allowed.

`acc` Only used if idra = 2. Soil-moisture retention exponent, in units of inverse length.

`akk` Only used if idra = 2. Relative hydraulic-conductivity exponent, in units of inverse length. The value specified must be greater than or equal to that specified for ACC.

`amm` Only used if idra = 2. Initial unsaturated-zone thickness above the capillary fringe, in units of length.

`axmm` Only used if idra = 2. The unsaturated-zone thickness above the capillary fringe above which an assumption of an infinitely thick unsaturated-zone thickness is assumed, in units of length.

Value

list with elements *idra*, *alpha*, *acc*, *akk*,
amm, *axmm*, representing WTAQ parameters related to drainage.
See descriptions in Arguments section.

Author(s)

Hauke Sonnenberg

See Also

`wtConfigure`

`wtConfiguredWellnames` *wtConfiguredWellnames*

Description

Read all well names (pumping well “PW” plus observation wells) from
WTAQ configuration *configuration*

Usage

```
wtConfiguredWellnames(configuration)
```

Arguments

`configuration` WTAQ configuration as generated by `wtConfigure`

Value

vector of character containing the name of the pumping well (as the first element) and the names of the observation wells as following elements (according to the order of their definition in *configuration*

Author(s)

Hauke Sonnenberg

`wtConfigureGeneral` *General parameter values*

Description

Define general WTAQ parameters

Usage

```
wtConfigureGeneral(title = paste("Input file generated by kwb.wtaq-functions on",

    date()), format = "DIMENSIONAL")
```

Arguments

`title` Title of simulation; up to 70 characters in length. Leave this line blank if no title is specified.

`format` Analysis format. Enter DIMENSIONAL.

Value

list with elements *title* and *format*, representing general WTAQ parameters. See descriptions in Arguments section.

Author(s)

Hauke Sonnenberg

See Also

`wtConfigure`

wtConfigureObservationWell *Observation well parameter values*

Description

Define WTAQ parameters related to an obervation well

Usage

```
wtConfigureObservationWell(obname = "obs1", r = 30, iows = 0,

    idpr = 0, rp = ifelse(idpr == 0, 0, 0.1), z1 = ifelse(iows ==

        2, 0, 5), z2 = ifelse(iows == 2, 0, 10), xll = ifelse(idpr ==

        0, 0, z2 - z1), zp = ifelse(iows == 2, 3, 0), tsobs = data.frame(t = 60 *

        1:3, dd = c(0.4, 0.6, 0.7)), irun = 1)
```

Arguments

`obname` Name of observation well or piezometer; up to 10 characters in length.

`r` Radial distance from axis of pumped well to observation well or piezometer, in units of length.

`iows` Type of observation well or piezometer:
IOWS = 0: Partially penetrating observation well.
IOWS = 1: Fully penetrating observation well.
IOWS = 2: Observation piezometer. Default: 0

`idpr` Options for delayed response of observation well.
IDPR = 0: No delayed response.
IDPR = 1: Delayed response. Default: 0

`rp` Inside radius of the observation well (or piezometer) standpipe in the interval over which water levels are changing during pumping, in units of length. Enter 0 if IDPR = 0. Default: 0.1, but 0 if idpr = 0.

`z1` Depth below top of aquifer or initial water table to the top of screened interval of observation well, in units of length. Use for IOWS = 0 or 1. Enter 0 if IOWS = 2. Default: 5.0, but 0 if iows = 2.

`z2` Depth below top of aquifer or initial water table to the bottom of screened interval of observation well, in units of length. Use for IOWS = 0 or 1. Enter 0 if IOWS = 2. Default: 10.0, but 0 if iows = 2.

`xll` Length of screened interval of observation well or piezometer, in units of length. Must be 0 if IDPR = 0. Default: z2 - z1, but 0 if idpr = 0.

`zp` Depth below top of aquifer or initial water table to center of piezometer, in units of length. Use for IOWS = 2. Enter 0.0 if IOWS = 0 or 1. Default: 3.0, but 0.0 if iows not equal to 2.

`tsobs` data.frame with column *t* containing the user-specified times for which drawdown at the observation well or piezometer will be calculated. If the data.frame has no rows, no drawdowns are calculated for the observation well or piezometer. The data.frame can (optionally) contain the measured drawdown for the observation well or piezometer at the corresponding times in an additional column *dd*.

`irun` Option to suppress drawdown calculations for the observation well or piezometer. Allows user to specify time-drawdown data, but those data are ignored during the simulation. Options are:
IRUN = 0: Drawdowns not calculated.
IRUN = 1: Drawdowns calculated.

Value

list with elements *irun*, *obname*, *iows*, *idpr*,

r, *z1*, *z2*, *zp*, *rp*, *xll*, *tsobs*,

representing WTAQ parameters related to an observation well.

See description in Arguments section.

Author(s)

Hauke Sonnenberg

See Also

`wtConfigure`

wtConfigurePumpwell *Pumped well parameter values*

Description

Define WTAQ parameters related to the pumped well

Usage

```
wtConfigurePumpwell(ipws = 0, ipwd = 1, rw = 1, rc = ifelse(ipwd ==

    0, 0, rw), zpd = 5, zpl = 10, sw = 0, qq = 0.05, tspw = data.frame(t = 60 *

    1:3, dd = 2 * c(0.4, 0.6, 0.7)), pwname = "PW", ipump = 1,

    irun = 1)
```

Arguments

`ipws` Type of pumped well:
IPWS = 0: Partially penetrating pumped well.
IPWS = 1: Fully penetrating pumped well. Default: 0

`ipwd` Type of diameter of pumped well:
IPWD = 0: Infinitesimal diameter (line-source theory).
IPWD = 1: Finite diameter. Default: 1

`rw` Radius of pumped well screen, in units of length. Default: 1.0

`rc` Inside radius of pumped well in the interval where water levels are changing during pumping, in units of length. Must be 0 if IPWD = 0. Default: value of rw, but 0 if ipwd = 0.

`zpd` Depth below top of aquifer or initial water table to the top of the screened interval of the pumped well, in units of length. Default: 5.0

`zpl` Depth below top of aquifer or initial water table to the bottom of the screened interval of the pumped well, in units of length. Default: 10.0

`sw` Well-bore skin parameter, dimensionless. Default: 0.0

`qq` Pumping rate of well, in units of cubic length per time. Default: 0.05

`tspw` data.frame with column *t* containing the user-specified times for which drawdown at the pumped well will be calculated. If the data.frame has no rows, no drawdowns are calculated for the pumped well. The data.frame can (optionally) contain the drawdowns that have been measured for the pumped well at the corresponding times in an additional column *dd*.

`pwname` Name of pumping well. This parameter is not evaluated by WTAQ but will be used in the result data frame returned by `wtRunConfiguration`. We introduce it in order to be able to identify the well within a wellfield. Default: "PW"

`ipump` Option to suppress calculations of drawdown at pumped well:
IPUMP = 0: Drawdown is not calculated at pumped well.
IPUMP = 1: Drawdown is calculated at pumped well.

`irun` Option to suppress drawdown calculations for the pumped well. Allows user to specify time-drawdown data, but those data are ignored during the simulation. Options are:
IRUN = 0: Drawdowns not calculated.
IRUN = 1: Drawdowns calculated.

Value

list with elements *irun*, *ipws*, *ipwd*, *ipump*,
qq, *rw*, *rc*, *zpd*, *zpl*, *sw*,
tspw, representing WTAQ parameters related to the pumped well. See
descriptions in Arguments section.

Author(s)

Hauke Sonnenberg

See Also

`wtConfigure`

`wtConfigureSolution` *Solver-related parameter values*

Description

Define WTAQ parameters related to the solver algorithms

Usage

```
wtConfigureSolution(isoln = 2, rerrnr = 1e-10, rerrsum = 0, nmax = 0,


    ntms = 30, ns = 8, error = 1e-04, nnn = 6, method = 3)
```

Arguments

`isoln` Numerical-inversion solution type:
ISOLN = 1: Solution by the Stehfest algorithm (must use this option for confined aquifers).
ISOLN = 2: Solution by the de Hoog algorithm (must use this option for drainage with unsaturated-zone characterization [wtConfigureDrainage()$idra == 2]).

`rerrnr` Relative error for Newton-Raphson iteration and finite summations of drawdown for water-table aquifers. A value of 1.0E-10 is suggested. Enter 0.0D0 for AQTYPE = CONFINED.

`rerrsum` Only used if isoln = 1. Relative error for finite summations of drawdown for confined aquifers. Suggested value is 1.0E-07 to 1.0E-08. Enter 0 if AQTYPE = WATER TABLE.

`nmax` Only used if isoln = 1. Maximum number of terms permitted in the finite summations of drawdown for confined aquifers. Suggested value is 200. Enter 0 if AQTYPE = WATER TABLE.

`ntms` Factor used to determine number of terms in the finite summations for drawdown for water-table aquifers. Suggested values are 20 or 30. Enter 0 if AQTYPE = CONFINED.

`ns` Only used if isoln = 1. Number of terms used in the Stehfest algorithm. This must be an even integer, the value of which depends upon computer precision. If the computer holds 16 significant figures in double precision, let NS = 6 to 12. A value of 8 is recommended.

`error` Only used if isoln = 2. Relative error sought for the accuracy of the numerical inversion. A value of 1.0E-04 is suggested.

`nnn` Only used if isoln = 2. Number of terms used in the summation of the Fourier series of the approximation to the inverse Laplace transform. A value of 6 is suggested.

`method` Only used if isoln = 2. Indicates which method will be used to accelerate convergence of the Fourier series. Options are 1, 2, or 3. Only METHOD = 3 has been tested and was found to be satisfactory. Users can consult de Hoog and others (1982) and John Knight's subroutine LAPADC for additional details if needed.

Value

list with elements *isoln*, *rerrnr*, *rerrsum*, *nmax*,

ntms, *ns*, *error*, *nnn*, *method*,

representing WTAQ parameters related to the solving algorithms used in WTAQ.

See descriptions in Arguments section.

Author(s)

Hauke Sonnenberg

See Also

`wtConfigure`

`wtConfigureTimes` *Simulation time parameter values*

Description

Define WTAQ parameters related to the times to be simulated

Usage

```
wtConfigureTimes(its = 1, tlast = 0, nlc = 0, nox = 0)
```

Arguments

`its`	Time specification: ITS = 0: Log-cycle time steps (use to generate theoretical curves). ITS = 1: User-specified times.
`tlast`	Largest value of time. Enter 0 if ITS = 1.
`nlc`	Number of logarithmic cycles on the time scale for which drawdown will be calculated. Enter 0 if ITS = 1.
`nox`	Number of equally spaced times per logarithmic cycle for which drawdown will be calculated. Enter 0 if ITS = 1.

Value

list with elements *its*, *tlast*, *nlc*, *nox*,

representing WTAQ parameters defining the times to be simulated.

See descriptions in Arguments section.

Author(s)

Hauke Sonnenberg

See Also

`wtConfigure`

`wtDefaultConfiguration` *Default WTAQ configuration*

Description

Default WTAQ configuration.

Usage

```
wtDefaultConfiguration()
```

Value

Currently: configuration for WTAQ sample problem 2, as also returned by `wtConfigurationExample2`

Author(s)

Hauke Sonnenberg

See Also

`wtConfigurationExample2`, `wtConfigurationExample3`

`wtDefaultConfigurationSolution` *default WTAQ solver configuration*

Description

default configuration for the WTAQ solver

Usage

```
wtDefaultConfigurationSolution(aqtype, idra = ifelse(aqtype ==

    "CONFINED", 0, 1))
```

Arguments

`aqtype`	Type of aquifer being simulated. Two options are provided:
	AQTYPE = CONFINED or
	AQTYPE = WATER TABLE

`idra`	Type of drainage at water table. Enter 0 if AQTYPE = CONFINED.
	Three options are provided:
	IDRA = 0: Instantaneous drainage.
	IDRA = 1: Gradual drainage.
	IDRA = 2: Drainage with unsaturated-zone characterization.

Author(s)

Hauke Sonnenberg

`wtInputFileLines` *text lines for WTAQ input file*

Description

This function transforms a WTAQ configuration as generated with `wtConfigure` into a vecotor of text lines. These text lines, written to a file, can be used as input file to the WTAQ drawdown modelling software.

Usage

```
wtInputFileLines(configuration = wtConfigure(), sep = "\t\t",


    dbg = FALSE)
```

Arguments

`configuration`	WTAQ configuration as generated by `wtConfigure`.
`sep`	Separator to be placed between parameter values and parameter names. Default: two tab characters.
`dbg`	if TRUE, debug message are shown, else not. Default: FALSE

Value

character vector with each element representing one row of the input file.

Author(s)

Hauke Sonnenberg

See Also

wtReadInputFile

---

wtPlotConfiguration *plot wellfield profile*

Description

plot wellfield profile

Usage

```
wtPlotConfiguration(configuration = wtConfigurationExample3(),
    col.pw = "black", col.ow = rainbow(length(configuration$obswells)),
    main = "WTAQ Configuration with Pumping Well on the Left",
    asp = 1, ...)
```

Arguments

configuration	configuration as retrieved by wtConfigure, default: wtConfigurationExample3()
col.pw	pumping well colour, default: "black"
col.ow	observation well colours. Default: rainbow colours
main	title for the plot
asp	aspect ratio, see ?plot
...	further arguments passed to .plotSideView

Author(s)

Hauke Sonnenberg

wtPlotResult *Plot WTAQ results*

Description

Plot WTAQ results

Usage

```
wtPlotResult(wtaqResult, main = "", plottype = "w", showMeasurements = TRUE,


    auto.key = NULL, asp = NA, PDF = FALSE, PNG = FALSE, pumpingWellName = "PW",


    xlim = NULL, ylim = NULL, ...)
```

Arguments

`wtaqResult`	data frame as returned by `wtRunConfiguration`
`main`	plot title, default: ""
`plottype`	vector of plot types ("s" = superposed, "w" = one plot per well, "d" = along distance to pump well, "t" each time along distance to well). Default: "w" (one plot per well).
`showMeasurements`	if TRUE, measurements are shown
`auto.key`	given to xyplot, see there. If NULL, a default key with as many columns as there are wells is used. Default: NULL.
`asp`	aspect ratio between x and y axis. Default: 1. Set to NA if aspect ratio does not matter.
`PDF`	if TRUE, a pdf file is created in tempdir() and opened in a PDF viewer
`PNG`	if TRUE, all plots made with plot type "t" are saved to png files in tempdir()/wtaqResult.
`pumpingWellName`	name of pumping well in wtaqResult
`xlim`	
`ylim`	
`...`	additional arguments given to xyplot

Author(s)

Hauke Sonnenberg

wtReadInputFile *Read WTAQ configuration from input file*

Description

Reads a WTAQ configuration (as e.g. required by `wtRunConfiguration`) from an existing WTAQ input file.

Usage

```
wtReadInputFile(inputFile, dbg = FALSE)
```

Arguments

`inputFile`	full path to an existing WTAQ input file
`dbg`	if TRUE, debug message are shown, else not.

Value

list with elements *general*, *aquifer*, *drainage*, *times*, *solution*, *pumpwell*, *obswell*, representing a WTAQ model run configuration.

Author(s)

Hauke Sonnenberg

See Also

`wtRunInputFile`, `wtConfigure`

wtReadPlotFile *Read WTAQ plot file*

Description

Read WTAQ plot file

Usage

```
wtReadPlotFile(plotfile, logtimes = NULL, toListView = FALSE,


    dbg = FALSE)
```

Arguments

`plotfile` plot file as produced by WTAQ model, either in “list view”, i.e. in which data is organised in blocks being written one after the other or in “matrix view”, i.e. in which data is organised in one table only

`logtimes` if TRUE, time steps are supposed to be logarithmic, i.e. it is assumed that the result plot file is in “matrix form”. Otherwise (logtimes == FALSE) it is assumed that the result plot file is in “list form” as each data block can contain different timestamps. Setting this argument accelerates the reading of the results as the result plot file does not have to be read twice (once for the format, once for the actual data).

`toListView` if TRUE, results are always returned in “list view” in which drawdowns of different wells do not appear in different columns but all in the same column *CALCDD*. A column *WELL* is added to indicate the well to which the drawdown belongs. Default: FALSE.

`dbg` if TRUE, debug messages are shown. Default: FALSE

Author(s)

Hauke Sonnenberg

Examples

```
### Read plot files as provided with "sample problems" 1 to 3 in WTAQ

### installation files

pfile <- system.file("extdata", sprintf("plt.sp%d", 1:3),

                     package = "kwb.wtaq")



dat1 <- wtReadPlotFile(pfile[1])

dat2 <- wtReadPlotFile(pfile[2])

dat3 <- wtReadPlotFile(pfile[3])
```

```
### Plot HDPW + HDOB1 + HDOB2 + HDOB3 + HDOB4 over dimensionless time (TDRDSQ)
tr1 <- lattice::xyplot(PW + OB1 + OB2 + OB3 + OB4 ~ TDRDSQ,
              data = dat1, type = ”l”, auto.key = list(columns = 5),
              main = ”Example 1”)

### Plot CALCDD over TIME
tr2 <- lattice::xyplot(CALCDD ~ TIME, groups = WELL,
              data = dat2, type = ”l”, auto.key = list(columns = 5),
              main = ”Example 2”)

tr3 <- lattice::xyplot(CALCDD ~ TIME, groups = WELL,
              data = dat3, type = ”l”, auto.key = list(columns = 5),
              main = ”Example 3”)
print(tr1)
print(tr2)
print(tr3)
```

wtRepairConfiguration *repair configuration*

Description

repair configuration by resetting parameter values that depend on others

Usage

```
wtRepairConfiguration(configuration, dbg = FALSE)
```

Arguments

configuration	WTAQ configuration, as retrieved by wtConfigure.
dbg	if TRUE, debug messages are shown, else not

Author(s)

Hauke Sonnenberg

wtRunConfiguration *Run WTAQ with given configuration*

Description

Run a WTAQ simulation with the given configuration

Usage

```
wtRunConfiguration(configuration, wtaq.exe = .wtaq_path(), targetDirectory = tempdir(),

    show.results = FALSE, fileExtension = "", dbg = FALSE, ...)
```

Arguments

configuration	WTAQ configuration, as retrieved by wtConfigure.
wtaq.exe	full path to WTAQ executable (default: compiled executable in package subfolder "extdata" as defined in helper function .wtaq_path())
targetDirectory	optional. Target directory. If no target directory is given, a temporary directory will be used.
show.results	if TRUE, the content of the results file will be shown in the R console. Default: FALSE

`fileExtension`	extension given to files
`dbg`	if TRUE, debug message are shown, else not. Default: FALSE
`...`	further arguments passed to wtRunInputFile, e.g. *show.output.on.console*

Value

model result as read with `wtReadPlotFile` from the output

file generated by the WTAQ modelling software

Author(s)

Hauke Sonnenberg

See Also

`wtRunInputFile`

`wtRunInputFile` *Run WTAQ with given input file*

Description

Run a WTAQ simulation with the given input file

Usage

```
wtRunInputFile(inputFile, wtaq.exe = .wtaq_path(), targetDirectory = tempdir(),

    show.results = FALSE, copyToTarget = TRUE, configuration = NULL,

    batchRun = FALSE, dbg = FALSE)
```

Arguments

`inputFile`	Existing WTAQ input file
`wtaq.exe`	full path to WTAQ executable (default: compiled executable in package subfolder "extdata" as defined in helper function .wtaq_path())
`targetDirectory`	optional. Target directory. If no target directory is given, a temporary directory will be used.
`show.results`	if TRUE, the content of the results file will be shown in the R console. Default: FALSE

`copyToTarget`	if TRUE, the input file is copied to the target directory. Set this argument to FALSE if the input file already is in the target directory and does not need to be copied again.
`configuration`	WTAQ configuration object as retrieved by `wtConfigure`. If not given (default), it will be constructed from the input file
`batchRun`	if TRUE batch run (may require admin rights!), else using direct command, default: FALSE
`dbg`	if TRUE, debug messages are shown, else not. Default: FALSE

Value

model result as read with `wtReadPlotFile` from the output

file generated by the WTAQ modelling software

Author(s)

Hauke Sonnenberg

See Also

`wtInputFileLines`, `wtReadInputFile`,`wtRunConfiguration`

`wtSetParameter` *set parameter in WTAQ configuration*

Description

set numerical, scalar parameter in WTAQ configuration

Usage

```
wtSetParameter(configuration, parameterName, parameterValue)
```

Arguments

`configuration`	WTAQ configuration as returned by `wtConfigure`
`parameterName`	parameter name. Must be one of the Aquifer parameters "bb", "hkr", "hkz", "ss", "sy" (see `wtConfigureAquifer` or one of the Drainage parameters "acc", "akk", "amm", "axmm" (see `wtConfigureDrainage`) or one of the Pumpwell parameters "qq", "rw", "rc", "zpd", "zpl", "sw" (see `wtConfigurePumpwell`)
`parameterValue`	numeric value to which the parameter *parameterName* shall be set

Value

configuration with the parameter *parameterName* altered to
the value given in *parameterValue*

Author(s)

Hauke Sonnenberg

See Also

`wtSetParameters`

`wtSetParameters` *set parameters in WTAQ configuration*

Description

set numerical, scalar parameters in WTAQ configuration

Usage

```
wtSetParameters(configuration, assignments = NULL)
```

Arguments

`configuration`	WTAQ configuration as returned by `wtConfigure`
`assignments`	list of "name = value" pairs defining the parameter assignments, e.g. list(hkr = 0.001, hkz = 0.002) to set the horizontal hydraulic conductivity (hkr) to 0.0001 (length/time) and the vertical hydraulic conductivity (hkz) to 0.00005 (length/time). For the allowed parameter names see the description in `wtSetParameter`

Value

configuration with adapted parameter values as defined in
assignments

Author(s)

Hauke Sonnenberg

See Also

`wtSetParameter`

Index

∗ **drawdowns**
drawdowns, 3

checkModelFit, 3

drawdowns, 3

fitnessAquifer, 4

modelFitness, 4
modelFitnessAggregated, 5

owConfigurationToWtConfiguration, 5, *19*
owConfigure, *5*, 6, *9*, *11*, *19*
owConfigureWell, 7
owGetDrawdowns, *6*, 8, *10*, *12, 13*
owPlotConfiguration, 9
owPlotDrawdowns, 10
owRandomConfiguration, *6*, *8*, 10
owRandomWellfield, *10, 11*, 11, *13*
owSuperposeDrawdowns, *9*, 12
owWellDistances, 13

print.wtaqConfiguration, 13

wtCalibrate, 14
wtCheckConfiguration, 15
wtConfigurationExample2, 16, *18*, *21*, *31*
wtConfigurationExample3, *17*, 18, *21*, *31*
wtConfigurationToOwConfiguration, *6*, 19
wtConfigure, *6*, *17–19*, 20, *21–24*, *26*, *28*, *30*, *32, 33*, *35*, *38*, *40, 41*
wtConfigureAquifer, *6*, *20*, 21, *40*
wtConfiguredDistances, 22
wtConfigureDrainage, *6*, *20*, 22, *40*
wtConfiguredWellnames, *22*, 23
wtConfigureGeneral, *20*, 24
wtConfigureObservationWell, *20*, 25
wtConfigurePumpwell, *20*, 27, *40*
wtConfigureSolution, *5*, *8*, *20*, 28
wtConfigureTimes, *20*, 30
wtDefaultConfiguration, 31
wtDefaultConfigurationSolution, *8*, 31
wtInputFileLines, 32, *40*
wtPlotConfiguration, 33
wtPlotResult, 34
wtReadInputFile, *33*, 35, *40*
wtReadPlotFile, 35, *39, 40*
wtRepairConfiguration, 38
wtRunConfiguration, *8*, *21*, *28*, *34, 35*, 38, *40*
wtRunInputFile, *35*, *39*, 39
wtSetParameter, 40, *41*
wtSetParameters, *41*, 41