Package 'epanet2toolkit'

February 9, 2023

Type Package

Title Call 'EPANET' Functions to Simulate Pipe Networks

Version 0.7.0

Date 2023-02-08

Maintainer Bradley Eck <brad@bradeck.net>

Suggests testthat, epanetReader

Description Enables simulation of water piping networks using 'EPANET'.
The package provides functions from the 'EPANET' programmer's toolkit as R functions so that basic or customized simulations can be carried out from R.
The package uses 'EPANET' version 2.2 from Open Water Analytics <https://github.com/OpenWaterAnalytics/EPANET/releases/tag/v2.2>.

License MIT + file LICENSE

Copyright COPYRIGHTS

URL https://github.com/bradleyjeck/epanet2toolkit

RoxygenNote 7.1.1

NeedsCompilation yes

Author Ernesto Arandia [aut],
Bradley Eck [aut, cre],
Lew Rossman [aut],
Michael Tryby [ctb],
Sam Hatchett [ctb],
Feng Shang [ctb],
James Uber [ctb],
Tom Taxon [ctb],
Hyoungmin Woo [ctb],
Jinduan Chen [ctb],
Yunier Soad [ctb],
Mike Kane [ctb],
Demetrios Eliades [ctb],
Will Furnass [ctb],
Steffen Macke [ctb],
Marios Kyriakou [ctb],

Elad Salomons [ctb],
Maurizio Cingi [ctb],
Bryant McDonnell [ctb],
Angela Marchi [ctb],
Markus Sunela [ctb],
Milad Ghiami [ctb],
IBM Corp. [cph],
Open Water Analytics [cph]

Repository CRAN

Date/Publication 2023-02-09 09:50:07 UTC

R topics documented:

ENclose . . . 3
ENcloseH . . . 4
ENcloseQ . . . 4
ENepanet . . . 5
ENgetcontrol . . . 5
ENgetcoord . . . 6
ENgetcount . . . 7
ENgetflowunits . . . 8
ENgetlinkid . . . 9
ENgetlinkindex . . . 9
ENgetlinknodes . . . 10
ENgetlinktype . . . 11
ENgetlinkvalue . . . 12
ENgetnodeid . . . 13
ENgetnodeindex . . . 14
ENgetnodetype . . . 15
ENgetnodevalue . . . 16
ENgetoption . . . 17
ENgetpatternid . . . 18
ENgetpatternindex . . . 19
ENgetpatternlen . . . 19
ENgetpatternvalue . . . 20
ENgetqualinfo . . . 21
ENgetqualtype . . . 21
ENgettimeparam . . . 22
ENgetversion . . . 23
ENinitH . . . 24
ENinitQ . . . 25
ENnextH . . . 26
ENnextQ . . . 27
ENopen . . . 28
ENopenH . . . 29
ENopenQ . . . 30
ENreport . . . 30

ENrunH 31
ENrunQ 32
ENsaveH 32
ENsaveinpfile 33
ENsetcontrol 33
ENsetcoord 35
ENsetlinkvalue 36
ENsetnodevalue 37
ENsetoption 38
ENsetpattern 39
ENsetpatternvalue 40
ENsetqualtype 40
ENsettimeparam 41
ENsolveH 42
ENsolveQ 43
ENstepQ 44
epanet2toolkit 44

Index **45**

`ENclose` *Close down the EPANET Toolkit system.*

Description

`ENclose` closes the EPANET Toolkit system (including all files being processed).

Usage

```
ENclose()
```

Value

Returns NULL invisibly; called for the side effect of closing EPANET.

Note

`ENclose` must be called when all processing has been completed, even if an error condition was encountered.

See Also

`ENopen`

Examples

```
# path to Net1.inp example file included with this package
inp <- file.path( find.package("epanet2toolkit"), "extdata","Net1.inp")
ENopen( inp, "Net1.rpt")
ENclose()
```

ENcloseH *close hydraulics engine*

Description

ENcloseH closes the hydraulic analysis system, freeing all allocated memory

Usage

```
ENcloseH()
```

Details

Call ENcloseH after all hydraulics analyses have been made using ENinitH-ENrunH-ENnextH. Do not call this function if ENsolveH is being used.

Value

Returns NULL invisibly; called for side effect

See Also

ENopenH, ENinitH, ENrunH, ENnextH

ENcloseQ *Close water quality analysis and free allocated memory*

Description

Close water quality analysis and free allocated memory

Usage

```
ENcloseQ()
```

Details

Do not call this function if ENsolveQ is being used.

Value

Returns NULL invisibly; called for side effect

ENepanet *ENepanet*

Description

runs a complete EPANET simulation

Usage

```
ENepanet(inpFile, rptFile, binOutFile = "")
```

Arguments

inpFile	name of input file
rptFile	name of report file (to be created)
binOutFile	name of optional binary output file

Value

Returns NULL invisibly; called for side effect

Examples

```
# path to Net1.inp example file included with this package
inp <- file.path( find.package("epanet2toolkit"), "extdata","Net1.inp")
print(inp)
ENepanet( inp, "Net1.rpt")
# try opening Net1.rpt in a text editor or reading it back
# into R with the read.rpt() function in package epanetReader
myRpt <- epanetReader::read.rpt("Net1.rpt")
summary(myRpt)
# clean-up the created file
file.remove("Net1.rpt")
```

ENgetcontrol *ENgetcontrol*

Description

Retrieve the parameters of a simple control statement.

Usage

```
ENgetcontrol(controlindex)
```

Arguments

controlindex	An integer specifying the control statement index.

Value

list of parameters of the control statement: ctype, lindex, setting, nindex, level

Note

Controls are indexed starting from 1 in the order in which they were entered into the [CONTROLS] section of the EPANET input file.

See Also

ENsetcontrol

Examples

```
# path to Net1.inp example file included with this package
inp <- file.path( find.package("epanet2toolkit"), "extdata","Net1.inp")
ENopen( inp, "Net1.rpt")
ENgetcontrol(1)
ENclose()
```

ENgetcoord *Get coordinates for a node*

Description

Get coordinates for a node

Usage

```
ENgetcoord(nodeindex)
```

Arguments

nodeindex	of node

Value

vector of x,y coordinate

Examples

```
# path to Net1.inp example file included with this package
inp <- file.path( find.package("epanet2toolkit"), "extdata","Net1.inp")
ENopen( inp, "Net1.rpt")
ENgetcoord(3)
ENclose()
```

ENgetcount *Get number of network elements.*

Description

ENgetcount retrieves the number of network components of a specific type.

Usage

```
ENgetcount(compcode)
```

Arguments

compcode	A character string, integer or numeric specifying the component code(s) (see below).

Details

Component codes consist of the following:

EN_NODECOUNT	0	Nodes
EN_TANKCOUNT	1	Reservoirs and tank nodes
EN_LINKCOUNT	2	Links
EN_PATCOUNT	3	Time patterns
EN_CURVECOUNT	4	Curves
EN_CONTROLCOUNT	5	Simple controls

The number of junctions in a network equals the number of nodes minus the number of tanks and reservoirs.

Value

The number of network components.

Examples

```
# path to Net1.inp example file included with this package
inp <- file.path( find.package("epanet2toolkit"), "extdata","Net1.inp")
ENopen( inp, "Net1.rpt")
ENgetcount(0)
ENgetcount("EN_NODECOUNT")
ENclose()
```

ENgetflowunits *Retrieve a code number indicating the units used to express all flow rates.*

Description

ENgetflowunits retrieves a code number indicating the units used to express all flow rates.

Usage

```
ENgetflowunits()
```

Value

An integer, the code numnber indicating the flow units.

Note

Flow units codes are as follows:

0	= EN_CFS	cubic feet per second
1	= EN_GPM	gallons per minute
2	= EN_MGD	million gallons per day
3	= EN_IMGD	Imperial mgd
4	= EN_AFD	acre-feet per day
5	= EN_LPS	liters per second
6	= EN_LPM	liters per minute
7	= EN_MLD	million liters per day
8	= EN_CMH	cubic meters per hour
9	= EN_CMD	cubic meters per day

Flow units are specified in the [OPTIONS] section of the EPANET Input file.

Flow units in liters or cubic meters implies that metric units are used for all other quantities in addition to flow. Otherwise US units are employed. (See Units of Measurement).

Examples

```
# path to Net1.inp example file included with this package
inp <- file.path( find.package("epanet2toolkit"), "extdata","Net1.inp")
ENopen( inp, "Net1.rpt")
ENgetflowunits()
ENclose()
```

ENgetlinkid *Retrieve the ID label of a link*

Description

ENgetlinkid retrieves the ID label of the link given its index.

Usage

```
ENgetlinkid(linkindex)
```

Arguments

linkindex	integer specifying the link index.

Value

character ID

Note

Link indexes are consecutive integers starting from 1.

See Also

ENgetlinkindex

Examples

```
# path to Net1.inp example file included with this package
inp <- file.path( find.package("epanet2toolkit"), "extdata","Net1.inp")
ENopen( inp, "Net1.rpt")
ENgetlinkid(1)
ENgetlinkid(12)
ENclose()
```

ENgetlinkindex *Retrieve the index of a link*

Description

ENgetlinkindex retrieves the index of a link with specified ID.

Usage

```
ENgetlinkindex(linkid)
```

Arguments

linkid	character

Value

integer index of requested link

Note

Link indexes are consecutive integers starting from 1.

See Also

ENgetlinkid

Examples

```
# path to Net1.inp example file included with this package
inp <- file.path( find.package("epanet2toolkit"), "extdata","Net1.inp")
ENopen( inp, "Net1.rpt")
ENgetlinkindex("22")
ENclose()
```

ENgetlinknodes *Retrieve the index of the end nodes of a link*

Description

Retrieve the index of the end nodes of a link

Usage

```
ENgetlinknodes(linkindex)
```

Arguments

linkindex	integer specifying the link index

Value

integer vector of node indices for this link

Note

Node and link indexes are consecutive integers starting from 1.

The From and To nodes are as defined for the link in the EPANET input file. The actual direction of flow in the link is not considered.

See Also

ENgetlinkindex

Examples

```
# path to Net1.inp example file included with this package
inp <- file.path( find.package("epanet2toolkit"), "extdata","Net1.inp")
ENopen( inp, "Net1.rpt")
ENgetlinknodes(1)
ENgetlinknodes(11)
ENclose()
```

ENgetlinktype *Retrieve the type code for a link*

Description

Retrieve the type code for a link

Usage

```
ENgetlinktype(linkindex)
```

Arguments

linkindex	for which type code is requested

Value

integer type-code of the link

Note

Link indexes are consecutive integers starting from 1. Link type codes consist of the following constants:

EN_CVPIPE	0	Pipe with Check Valve
EN_PIPE	1	Pipe
EN_PUMP	2	Pump
EN_PRV	3	Pressure Reducing Valve
EN_PSV	4	Pressure Sustaining Valve
EN_PBV	5	Pressure Breaker Valve
EN_FCV	6	Flow Control Valve
EN_TCV	7	Throttle Control Valve
EN_GPV	8	General Purpose Valve

See Also

ENgetlinkindex

Examples

```
# path to Net1.inp example file included with this package
inp <- file.path( find.package("epanet2toolkit"), "extdata","Net1.inp")
ENopen( inp, "Net1.rpt")
ENgetlinktype(1)
ENgetlinktype(12)
ENclose()
```

ENgetlinkvalue *Retrieve parameter value for a link*

Description

ENgetlinkvalue retrieves the value of a specific link parameter for a link.

Usage

```
ENgetlinkvalue(linkindex, paramcode)
```

Arguments

linkindex	index of the link
paramcode	requested parameter type either as name or number

Value

The parameter value of a specified link.

Note

Link indexes are consecutive integers starting from 1. Link parameter codes consist of the following constants:

EN_DIAMETER	0	Diameter
EN_LENGTH	1	Length
EN_ROUGHNESS	2	Roughness coeff.
EN_MINORLOSS	3	Minor loss coeff.
EN_INITSTATUS	4	Initial link status (0 = closed, 1 = open)
EN_INITSETTING	5	Initial pipe roughness Initial pump speed Initial valve setting
EN_KBULK	6	Bulk reaction coeff.
EN_KWALL	7	Wall reaction coeff.
EN_FLOW	8	Flow rate

EN_VELOCITY	9	Flow velocity
EN_HEADLOSS	10	Head loss
EN_STATUS	11	Actual link status (0 = closed, 1 = open)
EN_SETTING	12	Pipe roughness Actual pump speed Actal valve setting
EN_ENERGY	13	Energy expended in kwatts

Parameters 8 - 13 (EN_FLOW through EN_ENERGY) are computed values. The others are design parameters.

Flow rate is positive if the direction of flow is from the designated start node of the link to its designated end node, and negative otherwise.

Values are returned in units which depend on the units used for flow rate in the EPANET input file.

See Also

ENgetlinkindex ENgetflowunits

Examples

```
# path to Net1.inp example file included with this package
inp <- file.path( find.package("epanet2toolkit"), "extdata","Net1.inp")
ENopen(inp, "Net1.rpt")
ENgetlinkvalue(1, "EN_DIAMETER")
ENgetlinkvalue(1, "EN_LENGTH")
ENgetlinkvalue(8, "EN_DIAMETER")
ENgetlinkvalue(8, "EN_LENGTH")
ENclose()
```

ENgetnodeid *Retrieve the ID label a node.*

Description

ENgetnodeid retrieves the ID label a node from its index

Usage

```
ENgetnodeid(nodeindex)
```

Arguments

nodeindex	An integer node index

Value

A character string, the ID label of the specified node.

Note

Node indexes are consecutive integers starting from 1.

See Also

ENgetnodeindex

Examples

```
# path to Net1.inp example file included with this package
inp <- file.path( find.package("epanet2toolkit"), "extdata","Net1.inp")
ENopen( inp, "Net1.rpt")
ENgetnodeid(1)
ENgetnodeid(5)
ENgetnodeid(9)
ENclose()
```

ENgetnodeindex *Retrieve the index of a node*

Description

Retrieve the index of a node

Usage

```
ENgetnodeindex(nodeid)
```

Arguments

nodeid	A character string specifying the node ID.

Value

An integer index of the specified node.

Note

Node indexes are consecutive integers starting from 1.

See Also

ENgetnodeid

Examples

```
# path to Net1.inp example file included with this package
inp <- file.path( find.package("epanet2toolkit"), "extdata","Net1.inp")
ENopen( inp, "Net1.rpt")
ENgetnodeindex("10")
ENgetnodeindex("23")
ENclose()
```

ENgetnodetype *Retrieve the node-type code*

Description

ENgetnodetype retrieves the node-type code

Usage

```
ENgetnodetype(nodeindex)
```

Arguments

nodeindex	An integer specifying the node index.

Value

integer type-code of the node.

Note

Node indexes are consecutive integers starting from 1.

Node type codes consist of the following constants:

EN_JUNCTION	0	Junction node
EN_RESERVOIR	1	Reservoir node
EN_TANK	2	Tank node

Examples

```
# path to Net1.inp example file included with this package
inp <- file.path( find.package("epanet2toolkit"), "extdata","Net1.inp")
ENopen( inp, "Net1.rpt")
ENgetnodetype(1)
ENgetnodetype(10)
ENgetnodetype(11)
ENclose()
```

ENgetnodevalue *Retrieve node parameter value.*

Description

ENgetnodevalue retrieves the values of specific node parameters.

Usage

```
ENgetnodevalue(nodeindex, paramcode)
```

Arguments

nodeindex	An integer vector specifying the node index.
paramcode	An integer or character string, the parameter codes (see below).

Value

parameter value

Note

Node indexes are consecutive integers starting from 1.

Node parameter codes consist of the following constants:

EN_ELEVATION	0	Elevation
EN_BASEDEMAND	1	Base demand
EN_PATTERN	2	Demand pattern index
EN_EMITTER	3	Emitter coeff.
EN_INITQUAL	4	Initial quality
EN_SOURCEQUAL	5	Source quality
EN_SOURCEPAT	6	Source pattern index
EN_SOURCETYPE	7	Source type (see note below)
EN_TANKLEVEL	8	Initial water level in tank
EN_DEMAND	9	Actual demand
EN_HEAD	10	Hydraulic head
EN_PRESSURE	11	Pressure
EN_QUALITY	12	Actual quality
EN_SOURCEMASS	13	Mass flow rate per minute of a chemical source

Parameters 9 - 13 (EN_DEMAND through EN_SOURCEMASS) are computed values. The others are input design parameters.

Source types are identified with the following constants:

EN_CONCEN	0
EN_MASS	1

EN_SETPOINT	2
EN_FLOWPACED	3

See [SOURCES] for a description of these source types.

Values are returned in units which depend on the units used for flow rate in the EPANET input file (see Units of Measurement).

Examples

```
# path to Net1.inp example file included with this package
inp <- file.path( find.package("epanet2toolkit"), "extdata","Net1.inp")
ENopen( inp, "Net1.rpt")
ENgetnodevalue(1, "EN_ELEVATION")
ENgetnodevalue(5, "EN_BASEDEMAND")
ENclose()
```

ENgetoption *Retrieve the value of an analysis option.*

Description

ENgetoption retrieves the value of one or more particular analysis options.

Usage

```
ENgetoption(optioncode)
```

Arguments

optioncode	A character or integer specifying the option code (see below).

Details

Option codes consist of the following constants:

EN_TRIALS	0
EN_ACCURACY	1
EN_TOLERANCE	2
EN_EMITEXPON	3
EN_DEMANDMULT	4

Value

numeric value of the specified analysis option(s).

Examples

```
# path to Net1.inp example file included with this package
```

```
inp <- file.path( find.package("epanet2toolkit"), "extdata","Net1.inp")
ENopen( inp, "Net1.rpt")
ENgetoption(0)
ENgetoption("EN_TRIALS")
ENclose()
```

ENgetpatternid *Retrieve the ID label a time pattern*

Description

ENgetpatternid retrieves the ID label of a particular time pattern.

Usage

```
ENgetpatternid(patternindex)
```

Arguments

patternindex	An integer specifying the time pattern index.

Value

A character string, the pattern ID label of the specified time pattern.

Note

Pattern indexes are consecutive integers starting from 1.

Examples

```
# path to Net1.inp example file included with this package
inp <- file.path( find.package("epanet2toolkit"), "extdata","Net1.inp")
ENopen( inp, "Net1.rpt")
ENgetpatternid(1)
ENclose()
```

ENgetpatternindex *Retrieve the index a time pattern.*

Description

ENgetpatternindex retrieves the index of a time pattern.

Usage

```
ENgetpatternindex(patternid)
```

Arguments

patternid	A character string specifying the pattern ID

Value

An integer, the index of the specified time pattern.

Note

Pattern indexes are consecutive integers starting from 1.

Examples

```
# path to Net1.inp example file included with this package
inp <- file.path( find.package("epanet2toolkit"), "extdata","Net1.inp")
ENopen( inp, "Net1.rpt")
ENgetpatternindex("1")
ENclose()
```

ENgetpatternlen *Retrieve the number of time periods in a time pattern.*

Description

ENgetpatternlen retrieves the number of time periods in a specific time pattern.

Usage

```
ENgetpatternlen(patternindex)
```

Arguments

patternindex	An integer specifying a time pattern index.

Value

An integer, the time pattern length.

Note

Pattern indexes are consecutive integers starting from 1.

Examples

```
# path to Net1.inp example file included with this package
inp <- file.path( find.package("epanet2toolkit"), "extdata","Net1.inp")
ENopen( inp, "Net1.rpt")
ENgetpatternlen(1)
ENclose()
```

ENgetpatternvalue *Retrieve the multiplier factor for a specific time period*

Description

ENgetpatternvalue retrieves the multiplier factor for specific time periods in a pattern.

Usage

```
ENgetpatternvalue(index, period)
```

Arguments

index	An integer specifying the time pattern index.
period	An integer or integer vector of the periods within the time pattern.

Value

A numeric or numeric vector, the multiplier factor for the specific time pattern and period.

Note

Pattern indexes and periods are consecutive integers starting from 1.

See Also

ENgetpatternindex, ENgetpatternlen, ENsetpatternvalue

Examples

```
# path to Net1.inp example file included with this package
inp <- file.path( find.package("epanet2toolkit"), "extdata","Net1.inp")
ENopen( inp, "Net1.rpt")
ENgetpatternvalue(1,1)
ENgetpatternvalue(1,2)
ENgetpatternvalue(1,3)
ENclose()
```

ENgetqualinfo *Get quality analysis information*

Description

Get quality analysis information

Usage

```
ENgetqualinfo()
```

Value

list with elements: qualcode, chemname, chemunits, tracenode

Examples

```
inp <- file.path( find.package("epanet2toolkit"), "extdata","Net1.inp")
ENopen( inp, "Net1.rpt")
ENgetqualinfo()
ENclose()
# clean-up the created files
file.remove("Net1.rpt")
```

ENgetqualtype *Retrieve the type of water quality analysis called for.*

Description

ENgetqualtype retrieves the type of water quality analysis called for.

Usage

```
ENgetqualtype()

ENgetqualtype()
```

Value

A named integer vector, the water quality analysis code (see below) and the index of node traced in a source tracing analysis.

list of qualcode and trace node

Note

Water quality analysis codes are as follows:

EN_NONE	0	No quality analysis
EN_CHEM	1	Chemical analysis
EN_AGE	2	Water age analysis
EN_TRACE	3	Source tracing

The tracenode value will be 0 when the quality code is not EN_TRACE.

See Also

ENsetqualtype

Examples

```
# path to Net1.inp example file included with this package
inp <- file.path( find.package("epanet2toolkit"), "extdata","Net1.inp")
ENopen( inp, "Net1.rpt")
ENgetqualtype()
ENclose()
inp <- file.path( find.package("epanet2toolkit"), "extdata","Net1.inp")
ENopen( inp, "Net1.rpt")
ENgetqualtype()
ENclose()
# clean-up the created files
file.remove("Net1.rpt")
```

ENgettimeparam *Get the value of one or more specific analysis time parameters.*

Description

ENgettimeparam retrieves the value of one or more specific analysis time parameters.

Usage

```
ENgettimeparam(paramcode)
```

Arguments

paramcode	A character string or integer specifying the parameter code (see below).

Details

Time parameter codes consist of the following constants:

EN_DURATION	0	Simulation duration
EN_HYDSTEP	1	Hydraulic time step
EN_QUALSTEP	2	Water quality time step
EN_PATTERNSTEP	3	Time pattern time step
EN_PATTERNSTART	4	Time pattern start time
EN_REPORTSTEP	5	Reporting time step
EN_REPORTSTART	6	Report starting time
EN_RULESTEP	7	Time step for evaluating rule-based controls
EN_STATISTIC	8	Type of time series post-processing used: 0 = none 1 = averaged 2 = minimums 3 = maximums 4 = ranges
EN_PERIODS	9	Number of reporting periods saved to binary output file

Value

A named integer with the value of the specified time parameter.

Examples

```
# path to Net1.inp example file included with this package
inp <- file.path( find.package("epanet2toolkit"), "extdata","Net1.inp")
ENopen(inp, "Net1.rpt")
ENgettimeparam("EN_DURATION")
ENgettimeparam("EN_HYDSTEP")
ENclose()
```

ENgetversion *Retrieve the current version number of the EPANET Toolkit.*

Description

ENgetversion retrieves the current version number of the EPANET Toolkit.

Usage

```
ENgetversion()
```

Value

An integer, the Toolkit version number.

Note

The version number is a 5-digit integer that increases sequentially from 20001 with each new update of the Toolkit.

Examples

```
ENgetversion()
```

ENinitH *Initialize hydraulic engine*

Description

ENinitH Initializes storage tank levels, link status and settings, and the simulation clock time prior to running a hydraulic analysis.

Usage

```
ENinitH(flag)
```

Arguments

flag	A two-digit flag indicating if hydraulic results will be saved to the hydraulics file (rightmost digit) and if link flows should be re-initialized.

Details

Call ENinitH prior to running a hydraulic analysis using ENrunH and ENnextH.ENopenH must have been called prior to calling ENinitH. Do not call ENinitH if a complete hydraulic analysis is being made with a call to ENsolveH. Values of flag have the following meanings:

00	do not re-initialize flows, do not save results to file
01	do not re-initialize flows, save results to file
10	re-initialize flows, do not save results to file
11	re-initialize flows, save results to file

Set flag to 1 (or 11) if you will be making a subsequent water quality run, using ENreport to generate a report, or using ENsavehydfile to save the binary hydraulics file.

Value

Returns NULL invisibly; called for side effect

See Also

ENopenH, ENrunH, ENnextH, ENcloseH

Examples

```
# path to Net1.inp example file included with this package
inp <- file.path( find.package("epanet2toolkit"), "extdata","Net1.inp")
ENopen( inp, "Net1.rpt")
ENopenH()
ENinitH(0)
ENrunH()
ENcloseH()
ENclose()
# clean-up the created files
file.remove("Net1.rpt")
```

ENinitQ *Initialize water quality analysis*

Description

Initialize water quality analysis

Usage

```
ENinitQ(saveFlag)
```

Arguments

saveFlag	boolean or integer indicating whether to save quality results to a file

Details

Call ENinitQ before running quality analysis using ENrunQ with ENnextQ or ENstepQ. ENopenQ must have been called prior to calling ENinitQ. Do not call ENinitQ with ENsolveQ.

Value

Returns NULL invisibly on success or throws an error or warning

Examples

```
inp <- file.path( find.package("epanet2toolkit"), "extdata","Net1.inp")
ENopen( inp, "Net1.rpt")
ENsolveH()
ENsetqualtype("EN_CHEM", "Chlorine", "mg/L", "")
ENopenQ()
ENinitQ(0)
ENrunQ()
ENcloseQ()
ENclose()
# clean-up the created files
file.remove("Net1.rpt")
```

ENnextH *determine the next hydraulic step*

Description

ENnextH determines the length of time until the next hydraulic event occurs in an extended period simulation.

Usage

```
ENnextH()
```

Details

This function is used in conjunction with ENrunH to perform an extended period hydraulic analysis (see example below).

The return value is automatically computed as the smaller of:

- the time interval until the next hydraulic time step begins
- the time interval until the next reporting time step begins
- the time interval until the next change in demands occurs
- the time interval until a tank becomes full or empty
- the time interval until a control or rule fires

Value

An integer, the time (in seconds) until next hydraulic event occurs or 0 if at the end of the simulation period.

See Also

ENopenH, ENinitH, ENrunH, ENcloseH, ENsettimeparam

Examples

```
# path to Net1.inp example file included with this package
inp <- file.path( find.package("epanet2toolkit"), "extdata","Net1.inp")
ENopen( inp, "Net1.rpt")
  # store simulation times
  t = NULL
  ENopenH()
  ENinitH(11)
  repeat {
    t <- c(t, ENrunH())
    tstep <- ENnextH()
    if (tstep == 0) {
      break
```

```
    }
  }
  ENcloseH()
  ENclose()
# clean-up the created files
file.remove("Net1.rpt")
```

ENnextQ *Advances WQ simulation to start of the next hydraulic time period.*

Description

Advances WQ simulation to start of the next hydraulic time period.

Usage

```
ENnextQ()
```

Value

seconds until next hydraulic event occurs or 0 if at the end of the simulation period.

Examples

```
inp <- file.path( find.package("epanet2toolkit"), "extdata","Net1.inp")
ENopen( inp, "Net1.rpt")
ENsolveH()
ENsetqualtype("EN_CHEM", "Chlorine", "mg/L", "")
ENopenQ()
ENinitQ(0)
ENrunQ()
ENnextQ()
ENrunQ()
ENcloseQ()
ENclose()
# clean-up the created files
file.remove("Net1.rpt")
```

ENopen *Open the EPANET Toolkit.*

Description

ENopen opens the EPANET Toolkit to analyze a particular water distribution system.

Usage

```
ENopen(inpFileName, rptFileName, outFileName)
```

Arguments

inpFileName	A string, the name of the EPANET Input file.
rptFileName	A string, the name of the EPANET Report file.
outFileName	A string, the name of an optional binary Output file.

Value

returns NULL invisibly on success or raises an error or warning.

Note

If there is no need to save an EPANET's binary Output file, then outFileName can be an empty string ("").

If rptFileName is an empty string, reporting will be made to the operating system stdout device (which is usually the console/terminal).

enOpen must be called before any of the other toolkit functions are used. The only exception is enEpanet.

See Also

ENclose

Examples

```
# path to Net1.inp example file included with this package
inp <- file.path( find.package("epanet2toolkit"), "extdata","Net1.inp")
ENopen( inp, "Net1.rpt")
ENclose()
```

ENopenH *Open hydraulics analysis system.*

Description

ENopenH opens the EPANET hydraulics analysis system.

Usage

```
ENopenH()
```

Details

Call ENopenH prior to running the first hydraulic analysis using the ENinitH-ENrunH-ENnextH sequence. Multiple analyses can be made before calling ENcloseH to close the hydraulic analysis system.

Do not call this function if ENsolveH is being used to run a complete hydraulic analysis.

Value

Returns NULL invisibly; called for side effect

See Also

ENinitH, ENrunH, ENnextH, ENcloseH

Examples

```
# path to Net1.inp example file included with this package
inp <- file.path( find.package("epanet2toolkit"), "extdata","Net1.inp")
ENopen( inp, "Net1.rpt")
ENopenH()
ENinitH(0)
ENrunH()
ENcloseH()
ENclose()
# clean-up the created files
file.remove("Net1.rpt")
```

ENopenQ *Sets up for Water Quality analysis*

Description

Sets up for Water Quality analysis

Usage

```
ENopenQ()
```

Value

Returns NULL invisibly on success or throws an error or warning

Examples

```
inp <- file.path( find.package("epanet2toolkit"), "extdata","Net1.inp")
ENopen( inp, "Net1.rpt")
ENsolveH()
ENsetqualtype("EN_CHEM", "Chlorine", "mg/L", "")
ENopenQ()
ENinitQ(0)
ENrunQ()
ENcloseQ()
ENclose()
# clean-up the created files
file.remove("Net1.rpt")
```

ENreport *Write simulation report to the report file*

Description

Write simulation report to the report file

Usage

```
ENreport()
```

Value

Returns NULL invisibly; called for side effect

Examples

```
# path to Net1.inp example file included with this package
inp <- file.path( find.package("epanet2toolkit"), "extdata","Net1.inp")
ENopen( inp, "Net1.rpt", "Net1.bin")
ENsolveH()
ENsolveQ()
ENreport()
ENclose()
# clean up the created files
file.remove("Net1.rpt")
file.remove("Net1.bin")
```

ENrunH *run hydraulics engine*

Description

ENrunH Runs a single period hydraulic analysis, retrieving the current simulation clock time t.

Usage

```
ENrunH()
```

Details

Use ENrunH along with ENnextH in a while loop to analyze hydraulics in each period of an extended period simulation. This process automatically updates the simulation clock time so treat t as a read-only variable.

ENinitH must have been called prior to running the ENrunH-ENnextH loop.

See ENnextH for an example of using this function.

Value

Returns NULL invisibly; called for side effect

See Also

ENopenH, ENinitH, ENnextH, ENcloseH

Examples

```
# path to Net1.inp example file included with this package
inp <- file.path( find.package("epanet2toolkit"), "extdata","Net1.inp")
ENopen( inp, "Net1.rpt")
ENopenH()
ENinitH(0)
ENrunH()
ENcloseH()
```

```
ENclose()
# clean-up the created files
file.remove("Net1.rpt")
```

ENrunQ *Computs WQ results at current time .*

Description

Computs WQ results at current time .

Usage

```
ENrunQ()
```

Details

used in a loop with ENnextQ() to run an extended period WQ simulation.

Value

current simulation time in seconds

Examples

```
inp <- file.path( find.package("epanet2toolkit"), "extdata","Net1.inp")
ENopen( inp, "Net1.rpt")
ENsolveH()
ENsetqualtype("EN_CHEM", "Chlorine", "mg/L", "")
ENopenQ()
ENinitQ(0)
ENrunQ()
ENcloseQ()
ENclose()
# clean-up the created files
file.remove("Net1.rpt")
```

ENsaveH *ENsaveH Saves hydraulic results to binary file*

Description

ENsaveH

Saves hydraulic results to binary file

Usage

```
ENsaveH()
```

Details

Must be called before ENreport() if no WQ simulation has been made. Should not be called if ENsolveQ() will be used.

Value

Returns NULL invisibly; called for side effect

`ENsaveinpfile` *ENaveinpfile Saves current data to "INP" formatted text file.*

Description

ENaveinpfile

Saves current data to "INP" formatted text file.

Usage

```
ENsaveinpfile(filename)
```

Arguments

`filename`	The file path to create

Value

Returns NULL invisibly; called for side effect

`ENsetcontrol` *Set the parameters of a simple control statement*

Description

`ENsetcontrol` sets the parameters of a simple control statements.

Usage

```
ENsetcontrol(
  cindex,
  ctype = NULL,
  lindex = NULL,
  setting = NULL,
  nindex = NULL,
  level = NULL
)
```

Arguments

cindex	Integer, control statement index
ctype	Integer or character string, the control type code (see Details below).
lindex	Integer, index of the link being controlled.
setting	Numeric, value of the control setting.
nindex	Integer, the index of the controlling node.
level	value of controlling water level or pressure for level controls or of time of control action (in seconds) for time-based controls

Details

Controls are indexed starting from 1 in the order in which they were entered into the [CONTROLS] section of the EPANET input file. Control type codes consist of the following:

EN_LOWLEVEL	0	Control applied when tank level or node pressure drops below specified level
EN_HILEVEL	1	Control applied when tank level or node pressure rises above specified level
EN_TIMER	2	Control applied at specific time into simulation
EN_TIMEOFDAY	3	Control applied at specific time of day

For pipes, a setting of 0 means the pipe is closed and 1 means it is open. For a pump, the setting contains the pump's speed, with 0 meaning the pump is closed and 1 meaning it is open at its normal speed. For a valve, the setting refers to the valve's pressure, flow, or loss coefficient, depending on valve type.

For Timer or Time-of-Day controls set the nindex parameter to 0.

For level controls, if the controlling node nindex is a tank then the level parameter should be a water level above the tank bottom (not an elevation). Otherwise level should be a junction pressure.

To remove a control on a particular link, set the lindex parameter to 0. Values for the other parameters in the function will be ignored.

Value

Returns NULL invisibly on success or raises an error or warning.

See Also

ENsetcontrol

Examples

```
# path to Net1.inp example file included with this package
inp <- file.path( find.package("epanet2toolkit"), "extdata","Net1.inp")
ENopen( inp, "Net1.rpt")
ENgetcontrol(1)
ENsetcontrol(1, ctype=2, lindex=3, setting=1, nindex=0, level=54)
ENgetcontrol(1)
ENclose()
```

ENsetcoord *Set coordinates for a node*

Description

Set coordinates for a node

Usage

```
ENsetcoord(nodeindex, x, y)
```

Arguments

nodeindex	index of nodes for which to set coords
x	coordinate
y	coordinate

Value

returns NULL invisibily on success or raises an error or warning

Examples

```
# path to Net1.inp example file included with this package
inp <- file.path( find.package("epanet2toolkit"), "extdata","Net1.inp")
ENopen( inp, "Net1.rpt")
ENgetcoord(3)
ENsetcoord(3,33,44)
ENgetcoord(3)
ENclose()
```

ENsetlinkvalue *Set a parameter value for a link*

Description

Set a parameter value for a link

Usage

```
ENsetlinkvalue(index, paramcode, value)
```

Arguments

index	of the link
paramcode	number or name of parameter code, see details
value	new value of the parameter.

Details

Links are indexed starting from 1.

Link parameter codes consist of the following constants:

EN_DIAMETER	0	Diameter
EN_LENGTH	1	Length
EN_ROUGHNESS	2	Roughness coeff.
EN_MINORLOSS	3	Minor loss coeff.
EN_INITSTATUS	4	Initial link status (0 = closed, 1 = open)
EN_INITSETTING	5	Pipe roughness Initial pump speed Initial valve setting
EN_KBULK	6	Bulk reaction coeff.
EN_KWALL	7	Wall reaction coeff.
EN_STATUS	11	Current pump or valve status (0 = closed, 1 = open)
EN_SETTING	12	Current pump speed of valve setting.

Values are supplied in units which depend on the units used for flow rate in the EPANET input file (see Units of Measurement). Use EN_INITSTATUS and EN_INITSETTING to set the design value for a link's status or setting that exists prior to the start of a simulation. Use EN_STATUS and EN_SETTING to change these values while a simulation is being run (within the ENrunH - ENnextH loop).

If a control valve has its status explicitly set to OPEN or CLOSED, then to make it active again during a simulation you must provide a new valve setting value using the EN_SETTING parameter.

For pipes, either EN_ROUGHNESS or EN_INITSETTING can be used to change roughness.

Value

Returns NULL invisibly on success or raises a warning or error.

Examples

```
# path to Net1.inp example file included with this package
inp <- file.path( find.package("epanet2toolkit"), "extdata","Net1.inp")
ENopen(inp, "Net1.rpt")
ENgetlinkvalue(8, "EN_LENGTH")
ENsetlinkvalue(8, "EN_LENGTH", 3333)
ENgetlinkvalue(8, "EN_DIAMETER")
ENclose()
```

ENsetnodevalue *Set the parameter value for a node.*

Description

ENsetnodevalue sets parameter value for one node.

Usage

```
ENsetnodevalue(index, paramcode = NULL, value = NULL)
```

Arguments

index	An integer vector, the node index.
paramcode	An integer vector, the parameter code (see Details below).
value	A numeric vector, the new value of the parameter.

Details

Nodes are indexed starting from 1 in the order in which they were entered into the [NODES] section of the EPANET input file.

Node parameter codes consist of the following constants:

EN_ELEVATION	0	Elevation
EN_BASEDEMAND	1	Base demand
EN_PATTERN	2	Demand pattern index
EN_EMITTER	3	Emitter coeff.
EN_INITQUAL	4	Initial quality
EN_SOURCEQUAL	5	Source quality
EN_SOURCEPAT	6	Source pattern index
EN_SOURCETYPE	7	Source type (see note below)
EN_TANKLEVEL	8	Initial water level in tank

Source types are identified with the following constants:

EN_CONCEN	0

EN_MASS	1
EN_SETPOINT	2
EN_FLOWPACED	3

See [SOURCES] for a description of these source types.

Values are supplied in units which depend on the units used for flow rate in the EPANET input file (see Units of Measurement).

Value

returns NULL invisibly on success or raises an error or warning.

Examples

```
# path to Net1.inp example file included with this package
inp <- file.path( find.package("epanet2toolkit"), "extdata","Net1.inp")
ENopen( inp, "Net1.rpt")
ENgetnodevalue(3, "EN_ELEVATION")
ENsetnodevalue(3, "EN_ELEVATION", 777)
ENgetnodevalue(3, "EN_ELEVATION")
ENclose()
```

ENsetoption *Set the value of a particular analysis option.*

Description

ENsetoption sets the value of a particular analysis option.

Usage

```
ENsetoption(optioncode, value)
```

Arguments

optioncode	An integer or character vector specifying the option
value	numeric

Details

Option codes consist of the following constants:

EN_TRIALS	0
EN_ACCURACY	1
EN_TOLERANCE	2
EN_EMITEXPON	3
EN_DEMANDMULT	4

Examples

```
# path to Net1.inp example file included with this package
inp <- file.path( find.package("epanet2toolkit"), "extdata","Net1.inp")
ENopen( inp, "Net1.rpt")
ENgetoption("EN_TRIALS")
ENsetoption("EN_TRIALS", 22)
ENgetoption("EN_TRIALS")
ENclose()
```

ENsetpattern *Set all of the multiplier factors for a specific time pattern.*

Description

ENsetpattern sets all of the multiplier factors for a specific time pattern.

Usage

```
ENsetpattern(index, factors)
```

Arguments

index	An integer, the pattern index.
factors	A numeric vector, the multiplier factors for the entire pattern.

Details

Pattern indexes are consecutive integers starting from 1.

Use this function to redefine (and resize) a time pattern all at once; use ENsetpatternvalue to revise pattern factors in specific time periods of a pattern.

See Also

ENgetpatternindex, ENgetpatternlen, ENgetpatternvalue, ENsetpatternvalue

Examples

```
# path to Net1.inp example file included with this package
inp <- file.path( find.package("epanet2toolkit"), "extdata","Net1.inp")
ENopen( inp, "Net1.rpt")
ENsetpattern(1, 1:10)
ENgetpatternvalue(1,1)
ENgetpatternvalue(1,10)
ENclose()
```

ENsetpatternvalue *set pattern value*

Description

set pattern value

Usage

```
ENsetpatternvalue(index, period, value)
```

Arguments

index	index of pattern
period	time period for setting the value
value	value to set

Value

returns NULL inivisbly on success

Examples

```
# path to Net1.inp example file included with this package
inp <- file.path( find.package("epanet2toolkit"), "extdata","Net1.inp")
ENopen( inp, "Net1.rpt")
ENgetpatternvalue(1,3)
ENsetpatternvalue(1,3, 9.876)
ENgetpatternvalue(1,3)
ENclose()
```

ENsetqualtype *Set the type of water quality analysis called for.*

Description

ENsetqualtype sets the type of water quality analysis called for.

Usage

```
ENsetqualtype(qualcode, chemname = "", chemunits = "", tracenode = "")
```

Arguments

qualcode	An integer or a character string, the water quality analysis code (see below).
chemname	A character string, the name of the chemical being analyzed.
chemunits	A character string, units that the chemical is measured in.
tracenode	A character string, ID of node traced in a source tracing analysis.

Details

Water quality analysis codes are as follows:

EN_NONE	0	No quality analysis
EN_CHEM	1	Chemical analysis
EN_AGE	2	Water age analysis
EN_TRACE	3	Source tracing

Chemical name and units can be an empty string if the analysis is not for a chemical. The same holds for the trace node if the analysis is not for source tracing. Note that the trace node is specified by ID and not by index.

Value

returns NULL invisibly on success

See Also

ENgetqualtype

Examples

```
inp <- file.path( find.package("epanet2toolkit"), "extdata","Net1.inp")
ENopen( inp, "Net1.rpt")
ENgetqualtype()
ENsetqualtype("EN_CHEM", "Chlorine", "mg/L", "")
ENgetqualtype()
ENclose()
# clean-up the created files
file.remove("Net1.rpt")
```

ENsettimeparam *Set the value of a time parameter.*

Description

ENsettimeparam sets the value of a time parameter.

Usage

```
ENsettimeparam(paramcode, timevalue)
```

Arguments

paramcode	An integer or character
timevalue	An integer or character value of the time parameters in seconds.

Details

Time parameter codes consist of the following constants:

EN_DURATION	0	Simulation duration
EN_HYDSTEP	1	Hydraulic time step
EN_QUALSTEP	2	Water quality time step
EN_PATTERNSTEP	3	Time pattern time step
EN_PATTERNSTART	4	Time pattern start time
EN_REPORTSTEP	5	Reporting time step
EN_REPORTSTART	6	Reporting starting time
EN_RULESTEP	7	Time step for evaluating rule-based controls
EN_STATISTIC	8	Type of time series post-processing to use: EN_NONE (0) = none EN_AVERAGE (1) = averaged EN_MINIMUM (2) = minimums EN_MAXIMUM (3) = maximums EN_RANGE (4) = ranges

Do not change time parameters after calling ENinitH in a hydraulic analysis or ENinitQ in a water quality analysis

Examples

```
# path to Net1.inp example file included with this package
inp <- file.path( find.package("epanet2toolkit"), "extdata","Net1.inp")
ENopen(inp, "Net1.rpt")
ENgettimeparam("EN_HYDSTEP")
ENsettimeparam("EN_HYDSTEP", 600)
ENgettimeparam("EN_HYDSTEP")
ENclose()
```

ENsolveH *ENsolveH*

Description

Solves the network hydraulics for all time periods

Usage

```
ENsolveH()
```

Value

Returns NULL invisibly; called for side effect

Examples

```
# path to Net1.inp example file included with this package
inp <- file.path( find.package("epanet2toolkit"), "extdata","Net1.inp")
ENopen( inp, "Net1.rpt","Net1.bin")
ENsolveH()
ENsolveQ()
ENgetnodevalue(2, "EN_PRESSURE")
ENclose()
# clean-up the created files
file.remove("Net1.rpt")
file.remove("Net1.bin")
```

ENsolveQ *Solve network water quality for all time periods*

Description

Solve network water quality for all time periods

Usage

```
ENsolveQ()
```

Value

Returns NULL invisibly on success or throws an error or warning

Examples

```
inp <- file.path( find.package("epanet2toolkit"), "extdata","Net1.inp")
ENopen( inp, "Net1.rpt","Net1.bin")
ENsolveH()
ENsetqualtype("EN_CHEM", "Chlorine", "mg/L", "")
ENsolveQ()
ENclose()
# clean-up the created files
file.remove("Net1.rpt")
file.remove("Net1.bin")
```

ENstepQ *Advances WQ simulation one water quality time step.*

Description

Advances WQ simulation one water quality time step.

Usage

```
ENstepQ()
```

Value

time remaining in the overall simulation

epanet2toolkit *epanet2toolkit*

Description

Package for using EPANET 2 from R. Run a full EPANET simulation using ENepanet or build a custom simulation starting with toolkit functions like ENopen.

Author(s)

Ernesto Arandia & Bradley J. Eck

Index

ENclose, 3
ENcloseH, 4
ENcloseQ, 4
ENepanet, 5, *44*
ENgetcontrol, 5
ENgetcoord, 6
ENgetcount, 7
ENgetflowunits, 8, *13*
ENgetlinkid, 9, *10*
ENgetlinkindex, *9*, 9, *11, 12*
ENgetlinknodes, 10
ENgetlinktype, 11
ENgetlinkvalue, 12
ENgetnodeid, 13
ENgetnodeindex, 14
ENgetnodetype, 15
ENgetnodevalue, 16
ENgetoption, 17
ENgetpatternid, 18
ENgetpatternindex, 19
ENgetpatternlen, 19
ENgetpatternvalue, 20
ENgetqualinfo, 21
ENgetqualtype, 21
ENgettimeparam, 22
ENgetversion, 23
ENinitH, 24
ENinitQ, 25
ENnextH, 26
ENnextQ, 27
ENopen, *3*, 28, *44*
ENopenH, 29
ENopenQ, 30
ENreport, 30
ENrunH, 31
ENrunQ, 32
ENsaveH, 32
ENsaveinpfile, 33
ENsetcontrol, *6*, 33
ENsetcoord, 35
ENsetlinkvalue, 36
ENsetnodevalue, 37
ENsetoption, 38
ENsetpattern, 39
ENsetpatternvalue, 40
ENsetqualtype, 40
ENsettimeparam, 41
ENsolveH, 42
ENsolveQ, 43
ENstepQ, 44
epanet2toolkit, 44