Package 'epanet2toolkit'

November 6, 2023

Type Package

Title Call 'EPANET' Functions to Simulate Pipe Networks

Version 1.0.4

Date 2023-11-05

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 7.2.3

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 2023-11-06 08:10:11 UTC

R topics documented:

ENaddcontrol *Add a new simple control*

Description

Add a new simple control

Usage

```
ENaddcontrol(type, linkIndex, setting, nodeIndex, level)
```

Arguments

type	the type of control to add (see details)
linkIndex	the index of a link to control (starting from 1)
setting	control setting applied to the link
nodeIndex	index of the node used to control the link (0 for EN_TIMER and EN_TIMEOFDAY controls).
level	action level (tank level, junction pressure, or time in seconds) that triggers the control.

Value

index index of the new control.

ENaddcurve *Adds a new data curve to a project.*

Description

Adds a new data curve to a project.

Usage

```
ENaddcurve(id)
```

Arguments

id	The ID name of the curve to be added.

Details

The new curve contains a single data point (1.0, 1.0).

Value

null invisibly

ENadddemand *Appends a new demand to a junction node demands list.*

Description

Appends a new demand to a junction node demands list.

Usage

```
ENadddemand(nodeindex, base_demand, demand_pattern, demand_name)
```

Arguments

nodeindex	the index of a node (starting from 1).
base_demand	the demand's base value.
demand_pattern	the name of a time pattern used by the demand
demand_name	the name of the demand's category

ENaddlink *Add a link to the network*

Description

Add a link to the network

Usage

```
ENaddlink(id, type, from_node, to_node)
```

Arguments

id	name of new link
type	of new link, see details
from_node	id of source node for this link
to_node	id of target node for this link

Details

A new pipe is assigned a diameter of 10 inches (254 mm) and a length of 330 feet (~ 100 meters). Its roughness coefficient depends on the head loss formula in effect as follows: - Hazen-Williams formula: 130 - Darcy-Weisbach formula: 0.5 millifeet (0.15 mm) - Chezy-Manning formula: 0.01

All other pipe properties are set to 0.

A new pump has a status of EN_OPEN, a speed setting of 1, and has no pump curve or power rating assigned to it.

A new valve has a diameter of 10 inches (254 mm) and all other properties set to 0.

Type must be one of: EN_CVPIPE, EN_PIPE, EN_PUMP, EN_PRV, EN_PSV, EN_PBV, EN_FCV, EN_TCV, EN_GPV

Value

index of new link

ENaddnode *Adds a new node*

Description

Adds a new node

Usage

```
ENaddnode(nodeid, nodetype)
```

Arguments

nodeid	name of the node to be added
nodetype	the type of node being added. One of: EN_JUNCTION, EN_RESERVOIR, EN_TANK

Details

When a new node is created all of its properties are set to 0.

Value

index the index of the newly added node

ENaddpattern *Add a new time pattern*

Description

Add a new time pattern

Usage

```
ENaddpattern(patternid)
```

Arguments

patternid	the ID name of the pattern to add.

Details

The new pattern contains a single time period whose factor is 1.0.

Value

invisible NULL

ENaddrule *Adds a new rule-based control to a project*

Description

Adds a new rule-based control to a project

Usage

```
ENaddrule(rule)
```

Arguments

rule	text of the rule following the format used in an EPANET input file.

Value

null

ENclearreport *Clears the contents of a project's report file.*

Description

Clears the contents of a project's report file.

Usage

```
ENclearreport()
```

Value

Returns NULL invisibly; called for side effect

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

ENcopyreport *Copies the current contents of a project's report file to another file*

Description

Copies the current contents of a project's report file to another file

Usage

```
ENcopyreport(rptFile)
```

Arguments

rptFile	destination file

Details

This function allows toolkit clients to retrieve the contents of a project's report file while the project is still open.

Value

Returns NULL invisibly; called for side effect

Returns NULL invisibly; called for side effect

ENdeletecontrol *Deletes an existing simple control*

Description

Deletes an existing simple control

Usage

```
ENdeletecontrol(controlIndex)
```

Arguments

controlIndex	the index of the control to delete (starting from 1).

Value

null invisibly

ENdeletecurve *Deletes a data curve from a project*

Description

Deletes a data curve from a project

Usage

```
ENdeletecurve(index)
```

Arguments

index	the data curve's index (starting from 1).

Value

null invisibly

ENdeletedemand *Delete a demand from a junction node*

Description

Delete a demand from a junction node

Usage

```
ENdeletedemand(nodeindex, demandindex)
```

Arguments

nodeindex	the index of a node (starting from 1).
demandindex	the position of the demand in the node's demands list (starting from 1).

ENdeletelink *Delete a link from the project.*

Description

Delete a link from the project.

Usage

```
ENdeletelink(index, action = "EN_UNCONDITIONAL")
```

Arguments

index	the index of the link to be deleted.
action	The action taken if any control contains the link.

Details

If actionCode is EN_UNCONDITIONAL then the link and all simple and rule-based controls that contain it are deleted. If set to EN_CONDITIONAL then the link is not deleted if it appears in any control and error 261 is returned.

ENdeletenode *Deletes a node*

Description

Deletes a node

Usage

```
ENdeletenode(nodeindex, actionCode)
```

Arguments

nodeindex	the index of the node to be deleted.
actionCode	the action taken if any control contains the node and its links: EN_UNCONDITIONAL or EN_CONDITIONAL.

Details

If ‘actionCode‘ is EN_UNCONDITIONAL then the node, its incident links and all simple and rule-based controls that contain them are deleted. If set to EN_CONDITIONAL then the node is not deleted if it or its incident links appear in any controls and error code 261 is returned.

ENdeletepattern *Delete a new time pattern*

Description

Delete a new time pattern

Usage

```
ENdeletepattern(index)
```

Arguments

index	of the pattern to delete

Value

invisible NULL

ENdeleterule *Deletes an existing rule-based control*

Description

Deletes an existing rule-based control

Usage

```
ENdeleterule(index)
```

Arguments

index	the index of the rule to be deleted (starting from 1).

Value

null

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")
```

ENgetaveragepatternvalue *Get average of all time factors in a pattern*

Description

Get average of all time factors in a pattern

Usage

```
ENgetaveragepatternvalue(index)
```

Arguments

index	a time pattern index (starting from 1).

Value

the average value

ENgetbasedemand *Gets the base demand for one of a node's demand categories.*

Description

Gets the base demand for one of a node's demand categories.

Usage

```
ENgetbasedemand(nodeindex, demand_index = 1)
```

Arguments

nodeindex	a node's index (starting from 1).
demand_index	the index of a demand category for the node (starting from 1).

Value

the category's base demand.

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
EN_RULECOUNT	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()
```

ENgetcurveid *Retrieves the ID name of a curve given its index.*

Description

Retrieves the ID name of a curve given its index.

Usage

```
ENgetcurveid(index)
```

Arguments

index	a curve's index (starting from 1).

Value

the curve's ID name.

ENgetcurveindex *Retrieves the index of a curve given its ID name.*

Description

Retrieves the index of a curve given its ID name.

Usage

```
ENgetcurveindex(id)
```

Arguments

id	the ID name of a curve.

Value

The curve's index (starting from 1).

ENgetcurvelen *Retrieves the number of points in a curve.*

Description

Retrieves the number of points in a curve.

Usage

```
ENgetcurvelen(index)
```

Arguments

index	a curve's index (starting from 1).

Value

The number of data points assigned to the curve.

ENgetcurvetype *Retrieves a curve's type*

Description

Retrieves a curve's type

Usage

```
ENgetcurvetype(index)
```

Arguments

index	a curve's index (starting from 1).

Value

the curve's type

ENgetcurvevalue *Retrieves the value of a single data point for a curve.*

Description

Retrieves the value of a single data point for a curve.

Usage

```
ENgetcurvevalue(curveIndex, pointIndex)
```

Arguments

curveIndex	a curve's index (starting from 1).
pointIndex	the index of a point on the curve (starting from 1).

Value

list with the point's x-value and y-value

ENgetdemandindex *Retrieves the index of a node's named demand category*

Description

Retrieves the index of a node's named demand category

Usage

```
ENgetdemandindex(nodeindex, demand_name)
```

Arguments

nodeindex	the index of a node (starting from 1).
demand_name	the name of the demand's category

Value

demand category index

ENgetdemandmodel *Get type of demand model in use and its parameters*

Description

Get type of demand model in use and its parameters

Usage

```
ENgetdemandmodel()
```

Value

named list with parameters of the demand model

ENgetdemandname *Retrieves the name of a node's demand category.*

Description

Retrieves the name of a node's demand category.

Usage

```
ENgetdemandname(nodeindex, demand_index = 1)
```

Arguments

nodeindex	a node's index (starting from 1).
demand_index	the index of one of the node's demand categories (starting from 1).

Value

The name of the selected category.

ENgetdemandpattern *Gets the base demand for one of a node's demand categories.*

Description

Gets the base demand for one of a node's demand categories.

Usage

```
ENgetdemandpattern(nodeindex, demand_index = 1)
```

Arguments

nodeindex	the node's index (starting from 1).
demand_index	the index of a demand category for the node (starting from 1).

Details

A returned pattern index of 0 indicates that no time pattern has been assigned to the demand category.

Value

the category's base demand.

Retrieves index of a time pattern assigned to one of a node's demand categories.

patIndex the index of the category's time pattern.

ENgetelseaction *Gets the properties of an ELSE action in a rule-based control.*

Description

Gets the properties of an ELSE action in a rule-based control.

Usage

```
ENgetelseaction(ruleIndex, actionIndex)
```

Arguments

ruleIndex	the rule's index (starting from 1).
actionIndex	the index of the ELSE action to retrieve (starting from 1).

Value

list with the following components:

linkIndex the index of the link in the action

status the status assigned to the link

setting the value assigned to the link's setting

ENgeterror *Returns the text of an error message generated by an error code*

Description

Returns the text of an error message generated by an error code

Usage

```
ENgeterror(errcode)
```

Arguments

errcode	an error code.

Value

error message

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()
```

ENgetheadcurveindex *Retrieves index of head curve used by a pump*

Description

Retrieves index of head curve used by a pump

Usage

```
ENgetheadcurveindex(linkindex)
```

Arguments

linkindex	index of the pump

Value

index of head curve

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()
```

ENgetnumdemands *Get number of demands for a junction node*

Description

Get number of demands for a junction node

Usage

```
ENgetnumdemands(nodeindex)
```

Arguments

nodeindex	the index of a node (starting from 1).

Value

number of demands

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()
```

ENgetpremise *Gets the properties of a premise in a rule-based control.*

Description

Gets the properties of a premise in a rule-based control.

Usage

```
ENgetpremise(ruleIndex, premiseIndex)
```

Arguments

ruleIndex	the rule's index (starting from 1).
premiseIndex	the position of the premise in the rule's list of premises (starting from 1).

Value

list with components:

logop the premise's logical operator (IF = 1, AND = 2, OR = 3)
object the type of object the premise refers to
objIndex the index of the object (e.g. the index of a tank)
variable the object's variable being compared
relop the premise's comparison operator
status the status that the object's status is compared to
value the value that the object's variable is compared to

ENgetpumptype *Retrieves type of head curve used by a pump*

Description

Retrieves type of head curve used by a pump

Usage

```
ENgetpumptype(linkindex)
```

Arguments

linkindex	index of the pump

Value

type of head curve

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")
```

ENgetresultindex *Retrieves the order in which a node or link appears in an output file.*

Description

Retrieves the order in which a node or link appears in an output file.

Usage

```
ENgetresultindex(type, index)
```

Arguments

type	a type of element (either EN_NODE or EN_LINK).
index	the element's current index (starting from 1).

Details

If the element does not appear in the file then its result index is 0.

This function can be used to correctly retrieve results from an EPANET binary output file after the order of nodes or links in a network's database has been changed due to editing operations.

Value

the order in which the element's results were written to file.

ENgetrule *Retrieves summary information about a rule-based control.*

Description

Retrieves summary information about a rule-based control.

Usage

```
ENgetrule(index)
```

Arguments

index	the rule's index (starting from 1).

Value

list with components: nPremises number of premises in the rule's IF section; nThenActions number of actions in the rule's THEN section; nElseActions number of actions in the rule's ELSE section; priority the rule's priority value.

ENgetruleID *Gets the ID name of a rule-based control given its index.*

Description

Gets the ID name of a rule-based control given its index.

Usage

```
ENgetruleID(index)
```

Arguments

index	the rule's index (starting from 1).

Value

rule's ID name.

ENgetstatistic *Analysis convergence statistics.*

Description

Analysis convergence statistics.

Usage

```
ENgetstatistic(stat)
```

Arguments

stat	one of the statistics tabulated below

Details

These statistics report the convergence criteria for the most current hydraulic analysis and the cumulative water quality mass balance error at the current simulation time.

EN_ITERATIONS	Number of hydraulic iterations taken.
EN_RELATIVEERROR	Sum of link flow changes / sum of link flows.
EN_MAXHEADERROR	Largest head loss error for links.
EN_MAXFLOWCHANGE	Largest flow change in links.
EN_MASSBALANCE	Cumulative water quality mass balance ratio.
EN_DEFICIENTNODES	Number of pressure deficient nodes.
EN_DEMANDREDUCTION	

Value

value of the stat

ENgetthenaction *Gets properties of THEN action in rule-based control*

Description

Gets properties of THEN action in rule-based control

Usage

```
ENgetthenaction(ruleIndex, actionIndex)
```

Arguments

ruleIndex	the rule's index (starting from 1).
actionIndex	the index of the THEN action to retrieve (starting from 1).

Value

list with components: * linkIndex the index of the link in the action (starting from 1) * status the status assigned to the link * setting the value assigned to the link's setting

---

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()
```

`ENgettitle` *ENgettitle*

Description

Retrieves the title lines of the project

Usage

```
ENgettitle()
```

Value

character vector of title lines

`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()
```

ENgetvertex *Get vertex coordinates*

Description

Get vertex coordinates

Usage

```
ENgetvertex(index, vertex)
```

Arguments

index	a link's index (starting from 1).
vertex	index of vertex for getting coords

Value

list with elements x and y

ENgetvertexcount *Get number of vertices for a link*

Description

Get number of vertices for a link

Usage

```
ENgetvertexcount(index)
```

Arguments

index	a link's index (starting from 1).

Value

number of indices

ENinit *ENinit*

Description

Initializes an empty EPANET network

Usage

```
ENinit(rptFile, outFile, unitsType, headLossType)
```

Arguments

rptFile	the name of a report file to be created (or "" if not needed).
outFile	the name of a binary output file to be created (or "" if not needed).
unitsType	the choice of flow units. One of: "EN_CFS", "EN_GPM", "EN_MGD", "EN_IMGD", "EN_AFD", "EN_LPS", "EN_LPM", "EN_MLD", "EN_CMH", "EN_CMD"
headLossType	the choice of head loss formula . One of: EN_HW, EN_DW, EN_CM

Details

This function should be called to create an empty EPANET project without an EPANET-formatted input file. If the project receives it's network data from an input file then there is no need to call this function; use ENopen instead.

Value

Returns NULL invisibly; called for side effect

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")
```

ENresetreport *Resets a project's report options to their default values.*

Description

Resets a project's report options to their default values.

Usage

```
ENresetreport()
```

Details

After calling this function the default reporting options are in effect. These are:

- no status report
- no energy report
- no nodes reported on
- no links reported on
- node variables reported to 2 decimal places
- link variables reported to 2 decimal places (3 for friction factor)
- node variables reported are elevation, head, pressure, and quality
- link variables reported are flow, velocity, and head loss.

Value

Returns NULL invisibly; called for side effect

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

`ENsavehydfile` *Saves temporary hydraulics file to disk*

Description

Saves temporary hydraulics file to disk

Usage

```
ENsavehydfile(hydfile)
```

Arguments

`hydfile`	the name of the file to be created.

Details

Use this function to save the current set of hydraulics results to a file, either for post-processing or to be used at a later time by calling `ENusehydfile`.

The hydraulics file contains nodal demands and heads and link flows, status, and settings for all hydraulic time steps, even intermediate ones.

Before calling this function hydraulic results must have been generated and saved by having called `ENsolveH` or the `ENinitH` - `ENrunH` - `ENnextH` sequence with the initflag argument of `ENinitH` set to EN_SAVE or EN_SAVE_AND_INIT.

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

ENsetbasedemand *Sets the base demand for one of a node's demand categories.*

Description

Sets the base demand for one of a node's demand categories.

Usage

```
ENsetbasedemand(nodeindex, demand_index = 1, base_demand)
```

Arguments

nodeindex	a node's index (starting from 1).
demand_index	the index of a demand category for the node (starting from 1).
base_demand	the category's base demand.

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()
```

ENsetcurveid *Changes the ID name of a data curve given its index.*

Description

Changes the ID name of a data curve given its index.

Usage

```
ENsetcurveid(index, id)
```

Arguments

index	a data curve index (starting from 1).
id	the data curve's new ID name.

Value

null invisibly

ENsetcurvevalue *Sets the value of a single data point for a curve.*

Description

Sets the value of a single data point for a curve.

Usage

```
ENsetcurvevalue(curveIndex, pointIndex, x, y)
```

Arguments

curveIndex	a curve's index (starting from 1).
pointIndex	the index of a point on the curve (starting from 1).
x	the point's new x-value.
y	the point's new y-value.

Value

null invisibly

`ENsetdemandmodel` *Sets the type of demand model to use and its parameters.*

Description

Sets the type of demand model to use and its parameters.

Usage

```
ENsetdemandmodel(model, pmin, preq, pexp)
```

Arguments

`model`	Type of demand model. EN_DDA for demand driven analysis or EN_PDA for pressure driven analysis
`pmin`	Pressure below which there is no demand
`preq`	Pressure required to deliver full demand
`pexp`	Pressure exponent in demand function

`ENsetdemandname` *Sets the name of a node's demand category.*

Description

Sets the name of a node's demand category.

Usage

```
ENsetdemandname(nodeindex, demand_index, demand_name)
```

Arguments

`nodeindex`	a node's index (starting from 1).
`demand_index`	the index of one of the node's demand categories (starting from 1).
`demand_name`	The name of the selected category. No more than 30 characters

ENsetdemandpattern — *Sets the index of a time pattern used for one of a node's demand categories.*

Description

Sets the index of a time pattern used for one of a node's demand categories.

Usage

```
ENsetdemandpattern(nodeindex, demand_index, pattern_index)
```

Arguments

nodeindex	a node's index (starting from 1).
demand_index	the index of one of the node's demand categories (starting from 1).
pattern_index	the index of the time pattern assigned to the category.

Details

Specifying a pattern index of 0 indicates that no time pattern is assigned to the demand category.

ENsetelseaction — *Set properties of an ELSE action in a rule-based control*

Description

Set properties of an ELSE action in a rule-based control

Usage

```
ENsetelseaction(ruleIndex, actionIndex, linkIndex, status, setting)
```

Arguments

ruleIndex	the rule's index (starting from 1).
actionIndex	the index of the ELSE action being modified (starting from 1).
linkIndex	the index of the link in the action (starting from 1).
status	the new status assigned to the link
setting	the new value assigned to the link's setting.

Value

null invisibly

ENsetflowunits *Sets flow units.*

Description

Sets flow units.

Usage

```
ENsetflowunits(units)
```

Arguments

units	the choice of flow units. One of: "EN_CFS", "EN_GPM", "EN_MGD", "EN_IMGD", "EN_AFD", "EN_LPS", "EN_LPM", "EN_MLD", "EN_CMH", "EN_CMD"

Details

Flow units in liters or cubic meters implies that SI metric units are used for all other quantities in addition to flow. Otherwise US Customary units are employed.

Value

null invisibly

ENsetheadcurveindex *Sets index of head curve used by a pump*

Description

Sets index of head curve used by a pump

Usage

```
ENsetheadcurveindex(linkindex, curveindex)
```

Arguments

linkindex	index of the pump
curveindex	index of head curve to assign

Value

null invisibly

ENsetjuncdata *Sets properties for a junction*

Description

Sets properties for a junction

Usage

```
ENsetjuncdata(nodeindex, elevation, demand, demand_pattern = "")
```

Arguments

nodeindex	a junction node's index (starting from 1).
elevation	the value of the junction's elevation.
demand	the value of the junction's primary base demand.
demand_pattern	the ID name of the demand's time pattern ("" for no pattern)

Details

These properties have units that depend on the units used for flow rate.

ENsetlinkid *Change the ID of a link*

Description

Change the ID of a link

Usage

```
ENsetlinkid(index, newid)
```

Arguments

index	of the target link
newid	new name for the link (no more than 30 characters)

ENsetlinknodes *Set the indexes of a link's start- and end-nodes*

Description

Set the indexes of a link's start- and end-nodes

Usage

```
ENsetlinknodes(index, node1_index, node2_index)
```

Arguments

index	a link's index (starting from 1).
node1_index	The index of the link's start node (starting from 1).
node2_index	The index of the link's end node (starting from 1).

ENsetlinktype *Change a link's type*

Description

Change a link's type

Usage

```
ENsetlinktype(index, type, action = "EN_UNCONDITIONAL")
```

Arguments

index	of link before type change
type	the new type to change the link to (see details)
action	the action taken if any controls contain the link (see details)

Details

Link type is one of: EN_CVPIPE, EN_PIPE, EN_PUMP, EN_PRV, EN_PSV, EN_PBV, EN_FCV, EN_TCV, EN_GPV

If actionCode is EN_UNCONDITIONAL then all simple and rule-based controls that contain the link are deleted when the link's type is changed. If set to EN_CONDITIONAL then the type change is cancelled if the link appears in any control and error 261 is returned.

Value

link index after the type change

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()
```

ENsetnodeid *Changes the ID name of a node*

Description

Changes the ID name of a node

Usage

```
ENsetnodeid(nodeindex, newid)
```

Arguments

nodeindex	index of the node
newid	new ID name of the node

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()
```

ENsetpatternid *Change the ID name of a time pattern given its index.*

Description

Change the ID name of a time pattern given its index.

Usage

```
ENsetpatternid(index, id)
```

Arguments

index	a time pattern index (starting from 1).
id	the time pattern's new ID name.

Value

NULL invisibly

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()
```

ENsetpremise *Sets the properties of a premise in a rule-based control.*

Description

Sets the properties of a premise in a rule-based control.

Usage

```
ENsetpremise(
  ruleIndex,
  premiseIndex,
  logop,
  object,
  objIndex,
  variable,
  relop,
  status,
  value
)
```

Arguments

ruleIndex	the rule's index (starting from 1).
premiseIndex	the position of the premise in the rule's list of premises.
logop	the premise's logical operator (IF = 1, AND = 2, OR = 3).
object	the type of object the premise refers to

objIndex	the index of the object (e.g. the index of a tank)
variable	the object's variable being compared
relop	the premise's comparison operator
status	the status that the object's status is compared to
value	the value that the object's variable is compared to.

Value

null

ENsetpremiseindex *Sets the index of an object in a premise of a rule-based control*

Description

Sets the index of an object in a premise of a rule-based control

Usage

```
ENsetpremiseindex(ruleIndex, premiseIndex, objIndex)
```

Arguments

ruleIndex	the rule's index (starting from 1).
premiseIndex	the premise's index (starting from 1).
objIndex	the index of the premise's object (e.g. the index of a tank).

Value

null

ENsetpremisestatus *Sets the status being compared to in a premise of a rule-based control*

Description

Sets the status being compared to in a premise of a rule-based control

Usage

```
ENsetpremisestatus(ruleIndex, premiseIndex, status)
```

Arguments

ruleIndex	the rule's index (starting from 1).
premiseIndex	the premise's index (starting from 1).
status	the status that the premise's object status is compared to (see @ref EN_RuleStatus).

Value

null

ENsetpremisevalue *Sets the value in a premise of a rule-based control*

Description

Sets the value in a premise of a rule-based control

Usage

```
ENsetpremisevalue(ruleIndex, premiseIndex, value)
```

Arguments

ruleIndex	the rule's index (staring from 1).
premiseIndex	the premise's index (starting from 1).
value	The value that the premise's variable is compared to.

Value

null

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")
```

ENsetreport *Processes a reporting format command.*

Description

Processes a reporting format command.

Usage

```
ENsetreport(format)
```

Arguments

format	report formatting command: one line from the [REPORT] section of an inp file

---

ENsetrulepriority *Sets the priority of a rule-based control.*

Description

Sets the priority of a rule-based control.

Usage

```
ENsetrulepriority(index, priority)
```

Arguments

index	the rule's index (starting from 1).
priority	the priority value assigned to the rule.

Value

null

ENsetstatusreport *Sets the level of hydraulic status reporting.*

Description

Sets the level of hydraulic status reporting.

Usage

```
ENsetstatusreport(level)
```

Arguments

level	one of: EN_NO_REPORT, EN_NORMAL_REPORT, EN_FULL_REPORT

ENsettankdata *Sets properties for a tank*

Description

Sets properties for a tank

Usage

```
ENsettankdata(
  nodeindex,
  elevation,
  init_level,
  min_level,
  max_level,
  diameter,
  min_volume,
  volume_curve = ""
)
```

Arguments

nodeindex	tank's node index (starting from 1)
elevation	the tank's bottom elevation.
init_level	the initial water level in the tank.
min_level	the minimum water level for the tank.
max_level	the maximum water level for the tank.
diameter	the tank's diameter (0 if a volume curve is supplied).
min_volume	the volume of the tank at its minimum water level.
volume_curve	the name of the tank's volume curve ("" for no curve)

ENsetthenaction *Set properties of THEN action in a rule-based control*

Description

Set properties of THEN action in a rule-based control

Usage

```
ENsetthenaction(ruleIndex, actionIndex, linkIndex, status, setting)
```

Arguments

ruleIndex	the rule's index (starting from 1)
actionIndex	the index of the THEN action to modify (starting from 1)
linkIndex	the index of the link in the action
status	the new status assigned to the link
setting	the new value assigned to the link's setting

Value

null

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()
```

ENsetvertices *Set a link's vertices*

Description

Set a link's vertices

Usage

```
ENsetvertices(index, x, y)
```

Arguments

index	a link's index
x	numeric vector of x-coordinates
y	numeric vector of y-coordinates

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

ENusehydfile *Uses previously saved binary hydraulics file to supply a project's hydraulics.*

Description

Uses previously saved binary hydraulics file to supply a project's hydraulics.

Usage

```
ENusehydfile(hydfile)
```

Arguments

hydfile	name of file containing hydraulic results

Details

Call this function to re-use a set of hydraulic analysis results saved previously. This can save computational time if water quality analyses are being made under the same set of hydraulic conditions.

Do not call this function while the hydraulics solver is open.

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