Package: kwb.logger (via r-universe)

September 12, 2024

Title Functions to read measurement data from logger files

Version 0.3.0

Description Functions to read measurement data from logger files.

License MIT + file LICENSE

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

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

Imports kwb.datetime, kwb.utils, qmrparser, XML

Suggests testthat

Remotes github::kwb-r/kwb.datetime, github::kwb-r/kwb.utils

Encoding UTF-8

RoxygenNote 7.2.0

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

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

RemoteRef HEAD

RemoteSha 258c99b2cbd17187a348b39d2664aa9a8f701f24

Contents

`.getMetadata` *Get Meta Data from File Lines*

Description

Lines between lines containing "Outer Diameter" and "Txducers Spacing", respectively, are extrated.

Usage

```
.getMetadata(
  mylines,
  beginPattern = "Outer Diameter",
  endPattern = "Txducers Spacing",
  transpose = TRUE
)
```

Arguments

`mylines`	vector of character, each representing a line of a text file
`beginPattern`	pattern matching the line in which meta data block begins
`endPattern`	pattern matching the line in which meta data block ends
`transpose`	(default: TRUE)

Value

meta data read from *mylines* in the form of a data frame with one row containing the meta data values in the different columns (transpose == FALSE) or a data frame with columns *names* and *values* (transpose == TRUE)

.getSampleTimes *Read Sample Times from Automatic Sampler File*

Description

Read Sample Times from Automatic Sampler File

Usage

```
.getSampleTimes(
  filepath,
  blockbegin,
  sep = ";",
  dec = ",",
  captionPattern = "Datum;Uhrzeit",
  warn = TRUE
)
```

Arguments

filepath	full path to file generated by automatic sample
blockbegin	blockbegin
sep	field separator (default: ";")
dec	decimal separator (default: ",")
captionPattern	pattern matching the caption line. Default: "Datum;Uhrzeit"
warn	should warnings be printed (default: TRUE)

.getValidTimes *Read Times from Given Lines*

Description

Read Times from Given Lines

Usage

```
.getValidTimes(mylines, timeformat = "%y-%m-%d %H:%M:%S")
```

Arguments

mylines	vector of character, each representing a line of a text file
timeformat	time format to be looked for in *mylines*. Default: "%y-%m-%d %H:%M:%S"

Value

list with elements *tstamps* (timestamps as text in ISO format) and *tstampIndex* (contains for each text line the index in *tstamps* at which the timestamp corresponding to the line is found.

`extdataFile` *Path to File in Installed Package*

Description

Path to File in Installed Package

Usage

```
extdataFile(..., must_exist = TRUE)
```

Arguments

`...`	parts of the file path to be passed to `system.file`
`must_exist`	if `TRUE` (the default) and the specified file does not exist, the program stops with an error message

Value

path to file in the package installation folder in the R library or "" if the path does not exist

path to the specified file

Examples

```
# List the files provided in the "extdata" folder of kwb.logger
dir(extdataFile())
```

`getExampleTimestamps` *Timestamp Strings from Example Looger Files*

Description

Timestamp Strings from Example Looger Files

Usage

```
getExampleTimestamps()
```

readLogger_Buehler_4010_Flaschenbericht

Read Logger File from Buehler 4010 (Flaschenbericht)

Description

Read Logger File from Buehler 4010 (Flaschenbericht)

Usage

```
readLogger_Buehler_4010_Flaschenbericht(
  samplerFile,
  headerPattern = paste0("Flasche\\s+Datum\\s+Zeit\\s+F.llzeit\\s+PN Soll",
    "\\s+PN Ist\\s+P/E"),
  endPattern = "---",
  tz = "Etc/GMT+1"
)
```

Arguments

samplerFile	full path to file generated by auto sampler
headerPattern	pattern matching the header row
endPattern	pattern matching the end of the table
tz	time zone. Default: "Etc/GMT+1

readLogger_Buehler_4010_Infospeicher

Read Logger File from Buehler 4010 (Infospeicher)

Description

Read Logger File from Buehler 4010 (Infospeicher)

Usage

```
readLogger_Buehler_4010_Infospeicher(samplerFile, unprocessed = TRUE, ...)
```

Arguments

samplerFile	full path to file generated by auto sampler Buehler 4010 (Infospeicher)
unprocessed	if TRUE, the information is returned "raw", i.e. without any aggregation
...	arguments passed to `.read_Infospeicher.raw`, e.g. *tz*, see there

readLogger_FLEXIM_F601 *Read Logger File from FLEXIM F601*

Description

Read Logger File from FLEXIM F601

Usage

```
readLogger_FLEXIM_F601(
  filename,
  sep = "\t",
  dec = ",",
  timeformat = "%d.%m.%Y %H:%M:%S",
  headerPattern = "Uhrzeit\tDruck",
  date_yyyymmdd = substr(basename(filename), 1, 8),
  dbg = TRUE
)
```

Arguments

filename	full path to logger file
sep	column separator
dec	decimal character
timeformat	time format string
headerPattern	pattern matching the header line
date_yyyymmdd	datestring, by default taken from the filename
dbg	if TRUE, debug messages are shown

References

http://www.flexim.com/files/tsfluxus_f601v1-5-1de_leu.pdf

Examples

```
# Set path to example file (contained in this package)
file <- extdataFile("FLEXIM/example_FLEXIM_F601_short.txt")

# Let's have a look on the file structure
writeLines(readLines(file))

## Not run:
# Now read the file
x <- readLogger_FLEXIM_F601(file)
```

```
# Show the first lines
head(x)

# Get the meta data and show its structure
str(kwb.utils::getAttribute(x, "metadata"))

## End(Not run)
```

`readLogger_FLUKE_1730` *Read Logger File from FLUKE_1730*

Description

Read Logger File from FLUKE_1730

Usage

```
readLogger_FLUKE_1730(filepath, sep = ";", dec = ".")
```

Arguments

`filepath`	full path to logger file
`sep`	column separator
`dec`	decimal character

References

http://assets.fluke.com/manuals/1730____umeng0000.pdf

Examples

```
## Not run:
# set path to example file (contained in this package)
file <- extdataFile("FLUKE/example_FLUKE_1730_1.txt")

# read the file
x <- readLogger_FLUKE_1730(file)

# examine the list structure of the result
str(x)

## End(Not run)
```

readLogger_InSituInc_Aquatroll
Read File From InSituInc Aquatroll

Description

Read File From InSituInc Aquatroll

Usage

```
readLogger_InSituInc_Aquatroll(
  csv,
  headerPattern = NULL,
  timestampFormat = c("%d.%m.%Y %H:%M:%S", "%d.%m.%Y%H:%M:%S"),
  tz = "Etc/GMT+1",
  maxRowToLookForHeader = 700,
  model = "",
  dbg = FALSE,
  fileEncoding = "latin1"
)
```

Arguments

csv	full path to logger file
headerPattern	pattern matching the header of the table. See source code for the default.
timestampFormat	vector of possible timestamp formats. See help for the default.
tz	time zone. default: "Etc/GMT+1"
maxRowToLookForHeader	number of first rows in the file to read in advance to look for the header line. Default: 700
model	model of the Aquatroll (either "" or "600")
dbg	if TRUE, debug messages are shown
fileEncoding	encoding of the input file. Default: "latin1"

Value

if model = "600" data frame with columns DateTime, Seconds, Temp, ActCond, SpecCond, TDS, Resis, WaterD, DOC, DOP, Tur, Sal otherwise data frame with columns DateTime, WaterTemp.C, SpecCond.uS

Examples

```
# set path to example file (contained in this package)
file_1 <- extdataFile("InSituInc/example_InSituInc_Aquatroll.csv")
file_2 <- extdataFile("InSituInc/example_InSituInc_Aquatroll_600.csv")

# let's have a look on the file structure (read first 75 lines)
writeLines(kwb.utils::readLinesWithEncoding(file_1, 75, fileEncoding = "latin1"))

# now read the file
x <- readLogger_InSituInc_Aquatroll(csv = file_1)

# show the first lines
head(x)

# Read a file of Aquatroll, model "600"
x600 <- readLogger_InSituInc_Aquatroll(csv = file_2, model = "600")

# show the first lines
head(x600)

# show the structure
str(x600)
```

---

readLogger_NIVUS_PCM4 *Read Logger File from NIVUS PCM4*

Description

Read Logger File from NIVUS PCM4

Usage

```
readLogger_NIVUS_PCM4(csv, completenessRequiredFor = c("DateTime", "H", "v"))
```

Arguments

csv	full path to CSV file
completenessRequiredFor	character string of column names that are required not to be empty

References

http://www.nivus.de/ximages/1397007_p4ba02en.pdf

Examples

```
## Not run:
# set path to example file (contained in this package)
file <- extdataFile("NIVUS/example_NIVUS_PCM4.TXT")

# read the file
x <- readLogger_NIVUS_PCM4(file)

# examine the list structure of the result
str(x)

## End(Not run)
```

`readLogger_NIVUS_PCM4_2`

Read Logger File from NIVUS PCM4

Description

Read Logger File from NIVUS PCM4

Usage

```
readLogger_NIVUS_PCM4_2(
  filepath,
  headerRow = 9,
  sep = "\t",
  maxCols = 50,
  removeEmptyColumns = FALSE
)
```

Arguments

`filepath`	full path to logger file
`headerRow`	number of row containing the header row of the table
`sep`	column separator
`maxCols`	maximum number of columns
`removeEmptyColumns`	if `TRUE` empty columns are removed

References

http://www.nivus.de/ximages/1397007_p4ba02en.pdf

Examples

```
## Not run:
# set paths to example files (contained in this package)
files <- c(
  extdataFile("NIVUS/example_NIVUS_PCM4_ALT.TXT"),
  extdataFile("NIVUS/example_NIVUS_PCM4_NEU.TXT"),
  extdataFile("NIVUS/example_NIVUS_PCM4_STR.TXT")
)

# read the files
x1 <- readLogger_NIVUS_PCM4_2(files[1L])
x2 <- readLogger_NIVUS_PCM4_2(files[2L])
x3 <- readLogger_NIVUS_PCM4_2(files[3L])

# compare structures
str(x1)
str(x2)
str(x3)

# get metadata
(metadata <- kwb.utils::getAttribute(x1, "metadata"))

# show time adjusts
metadata$timeAdjust

## End(Not run)
```

---

readLogger_Ori_BasicEx1

Read Logger File from Ori BasicEx1

Description

Read Logger File from Ori BasicEx1

Usage

```
readLogger_Ori_BasicEx1(
  filepath,
  infotype = validInfoTypes(),
  blockbegin = "ORI BasicEx1",
  warn = TRUE,
  sep = ";",
  dec = ",",
  colnameDate = "Datum",
  colnameTime = "Uhrzeit",
  dateformat = .defaultTimeFormat("v5"),
  timeformat = .defaultTimeFormat("v1")
)
```

Arguments

filepath	full path to file generated by automatic sampler
infotype	one or more of the values returned by validInfoTypes
blockbegin	identification of "block begins"; Default: "ORI BasicEx1"
warn	if TRUE, warnings are generated if a block does not contain "Probe"
sep	column separator
dec	decimal character
colnameDate	name of date column
colnameTime	name of time column
dateformat	date format string
timeformat	time format string

References

http://www.origmbh.de/fileadmin/user_upload/pdf/basic_ex_1_mobil/ORI_Basic_Ex1_mobil_de.pdf

Examples

```
## Not run:
# set path to example file (contained in this package)
(file <- extdataFile("Ori/example_Ori_BasicEx1.csv"))

# read the "actions" from the file
readLogger_Ori_BasicEx1(file, infotype = "actions")

# read the sample times from the file
readLogger_Ori_BasicEx1(
  file, infotype = "times", blockbegin = "ORI BasicEx1 TU Berlin"
)

# read both at the same time
x <- readLogger_Ori_BasicEx1(
  file, blockbegin = "ORI BasicEx1 TU Berlin"
)

# examine the list structure of the result
str(x)

## End(Not run)
```

readLogger_Ori_MLog *Read Water Level File from Radar Probe*

Description

Read Water Level File from Radar Probe

Usage

```
readLogger_Ori_MLog(
  csv,
  sep = "\t",
  dec = ",",
  timeFormat = c(.defaultTimeFormat("v8"), .defaultTimeFormat("v7")),
  ...
)
```

Arguments

csv	full path to file generated by radar probe
sep	column separator. Default: tabulator character
dec	decimal character. Default: ","
timeFormat	time format string. Default: "%d.%m.%Y %H:%M:%S"
...	further arguments passed to readCsvInputFile, e.g. *stopOnMissingColumns*

References

http://www.origmbh.de/fileadmin/user_upload/produkte/sensoren/DA_MLOG_Vega_Radar_Puls_61_en.pdf

Examples

```
## Not run:
file <- extdataFile("Ori/example_Ori_MLog_1.csv")

# set separator different from default (tabulator) and do not complain
# about missing columns
x <- readLogger_Ori_MLog(filepath, sep = ";", stopOnMissingColumns = FALSE)

str(x)

## End(Not run)
```

`readLogger_PCE_PA8000` *Read Logger File from PCE PA8000*

Description

Read Logger File from PCE PA8000

Usage

```
readLogger_PCE_PA8000(
  filename,
  sep = "\t",
  dec = ",",
  timeformat = .defaultTimeFormat("v2"),
  headerPattern = "Position\tDate\tTime"
)
```

Arguments

`filename`	full path to logger file
`sep`	column separator
`dec`	decimal character
`timeformat`	time format string
`headerPattern`	pattern matching the table header row

References

http://www.industrial-needs.com/manual/manual-pce-pa8000.pdf

Examples

```
## Not run:
# set path to example file (contained in this package)
(file <- extdataFile("PCE/example_PCE_PA8000.txt"))

# read the file
x <- readLogger_PCE_PA8000(file)

# examine the structure of the result
str(x)

## End(Not run)
```

`readLogger_PCE_TDS100` *Read Logger File from PCE TDS100*

Description

Read logger file of flow meter TDS 100 by PCE Deutschland GmbH

Usage

```
readLogger_PCE_TDS100(txt, timeformat = .defaultTimeFormat("v3"))
```

Arguments

`txt`	full path to text file generated by logger
`timeformat`	Date and time format used in the file. Default: "%y-%m-%d %H:%M:%S"

Value

data frame with columns *tstamp*, *flow*, *flowunit*, *vel*, *velunit*, *UP*, *DN*, *Q*. Duplicate lines are removed. The data frame has an attribute *metadata* containing the meta data of the file, as returned by `.getMetadata`

References

http://www.warensortiment.de/bedienung/ba_durchflussmessgeraet-pce-tds100h_de-v-1-1.pdf

http://www.industrial-needs.com/manual/manual-pce-tds-100h.pdf

Examples

```
## Not run:
# set path to example file (contained in this package)
file_1 <- extdataFile("PCE/example_PCE_TDS100.log")
file_2 <- extdataFile("PCE/example_PCE_TDS100_noMeta.log")

# read a file containing metadata
x1 <- readLogger_PCE_TDS100(file_1) # warnings about duplicate timestamps

# read a file not containing metadata
x2 <- readLogger_PCE_TDS100(file_2) # warning about missing meta data

# examine the structures of the results
str(x1)
str(x2)

# get meta data from attribute "metadata"
kwb.utils::getAttribute(x1, "metadata")
```

```
## End(Not run)
```

`readLogger_SIGMA_SD900` *Read Logger File from SIGMA_SD900*

Description

Read Logger File from SIGMA_SD900

Usage

```
readLogger_SIGMA_SD900(
  filepath,
  successOnly = FALSE,
  sep = ",",
  dateformat = .defaultTimeFormat("v6")
)
```

Arguments

`filepath`	full path to input file
`successOnly`	if TRUE, only file entries with "SUCCESS" in column "-RESULT-" are returned. Default: FALSE
`sep`	column separator. Default: ","
`dateformat`	format of timestamp. Default: "%H:%M %m/%d/%Y"

References

http://www.hach.com/asset-get.download.jsa?id=7639983273

Examples

```
## Not run:
# set path to example file (contained in this package)
file <- extdataFile("SIGMA/example_SIGMA_SD900.csv")

# read the file
(samples <- readLogger_SIGMA_SD900(file))

# read only lines representing successful samples
(samplesOk <- readLogger_SIGMA_SD900(file, successOnly = TRUE))

# show metadata (given in attribute "metadata")
kwb.utils::getAttribute(samplesOk, "metadata")

## End(Not run)
```

readLogger_STS_DL70 *Read Logger File from STS DL70*

Description

Read Logger File from STS DL70

Usage

```
readLogger_STS_DL70(
  filepath,
  sep = "\t",
  dec = ".",
  dateformat = .defaultTimeFormat("v5"),
  timeformat = .defaultTimeFormat("v1")
)
```

Arguments

filepath	full path to logger file
sep	column separator
dec	decimal character
dateformat	date format string
timeformat	time format string

Value

data frame with attribute "metadata"

References

http://www.stssensoren.de/app/download/5648435717/Manual_DL70-PC-Software_de_DDB013B.pdf?t=1372317244

Examples

```
## Not run:
# set path to example file (contained in this package)
file <- extdataFile("STS/example_STS_DLN70_H.txt")

# read the file
x <- readLogger_STS_DL70(file)

# examine the structure of the result
str(x)

## End(Not run)
```

readLogger_VEGA_BAR54 *Read Logger File from VEGA BAR54*

Description

Read Logger File from VEGA BAR54

Usage

```
readLogger_VEGA_BAR54(
  filename,
  date_yyyymmdd = "",
  sep = "\t",
  dec = ",",
  timeformat = .defaultTimeFormat("v2"),
  headerPattern = "Uhrzeit\tDruck"
)
```

Arguments

filename	full path to inoput file
date_yyyymmdd	day of measurements as text in format "yyyymmdd", e.g. "20140423" for April 23 of 2014
sep	column separator
dec	decimal character
timeformat	time format string
headerPattern	pattern matching the table header row

References

http://www.vega.com/downloads/PI/EN/37528-EN.PDF

See Also

readLogger_VEGA_BAR54_raw

Examples

```
## Not run:
# set path to example file (contained in this package)
(file <- extdataFile("VEGA/example_VEGA_BAR54.txt"))

# read the file
x <- readLogger_VEGA_BAR54(file)

# examine the structure of the result
str(x)
```

```
## End(Not run)
```

---

readLogger_VEGA_BAR54_raw *Read Logger File from VEGA BAR54 raw*

Description

Read Logger File from VEGA BAR54 raw

Usage

```
readLogger_VEGA_BAR54_raw(filepath, trimMetadata = TRUE)
```

Arguments

filepath	full path to logger file
trimMetadata	if TRUE, metadata of class "character" are trimmed (by using hsTrim)

See Also

readLogger_VEGA_BAR54

Examples

```
## Not run:
# set paths to example files (contained in this package)
file_1 <- extdataFile("VEGA/example_VEGA_BAR54_raw1.gnd")
file_2 <- extdataFile("VEGA/example_VEGA_BAR54_raw2.gnd")

# read the files
x1 <- readLogger_VEGA_BAR54_raw(file_1)
x2 <- readLogger_VEGA_BAR54_raw(file_2)

# get meta data
kwb.utils::getAttribute(x1, "metadata")
kwb.utils::getAttribute(x2, "metadata")

## End(Not run)
```

read_aquatroll_data *Read Aquatroll Data*

Description

Use readLogger_InSituInc_Aquatroll instead!

Usage

```
read_aquatroll_data(file)
```

Arguments

file	full path to logger file

read_logger_LT_EDGE_M100 *Read Logger File from LT EDGE M100*

Description

Read an input file of water level logger "LT EDGE M100"

Usage

```
read_logger_LT_EDGE_M100(
  file,
  timeFormat = .defaultTimeFormat("v4"),
  metaToColumns = FALSE,
  dbg = TRUE,
  ...
)
```

Arguments

file	full path to the file (either .xle or .csv)
timeFormat	time format in which the timestamps are given in the file. Default: "yyyy/mm/dd HH:MM:SS". It is assumed that the timestamps represent Berlin normal time = Berlin winter time (no daylight saving)
metaToColumns	if TRUE, metadata are written into the last columns of the returned data frame
dbg	if TRUE (default) debug messages (e.g. "Reading...") are shown
...	arguments passed to read_logger_LT_EDGE_M100_csv or read_logger_LT_EDGE_M100_xle

Value

data frame with columns *TimestampInFile* (character timestamp as it appeared in the file), *DateTimeUTC* (POSIXct timestamp in time zone "UTC"), *LocalDateTime* (POSIXct timestamp in time zone "Europe/Berlin"), *UTCOffset* (integer Offset in hours between UTC and local time), *Level* (water level in unit as given in the file header), *Temperature* (water temperature in unit as given in the file header)

---

read_logger_LT_EDGE_M100_csv

Read Logger File (.csv) from LT EDGE M100

Description

Read an input file of water level logger "LT EDGE M100" from .csv file

Usage

```
read_logger_LT_EDGE_M100_csv(file, country = "en", maxRowToLookForHeader = 15)
```

Arguments

file	full path to logger file
country	one of "en" (English format: column separator = ",", decimal sign = ".") or "de" (German format: column separator = ";", decimal sign = ",")
maxRowToLookForHeader	number of rows to look for column headers (default: 15)

---

read_logger_LT_EDGE_M100_xle

Read Logger File (.xle) from LT EDGE M100

Description

Read an input file of water level logger "LT EDGE M100" from .xle file

Usage

```
read_logger_LT_EDGE_M100_xle(
  file,
  timeFormat = .defaultTimeFormat("v4"),
  country = NA
)
```

Arguments

file	full path to logger file
timeFormat	time format string
country	one of "en" (English format: decimal sign = ".") or "de" (German format: decimal sign = ","). If country is NA (default) it is guessed from the file content

---

validInfoTypes *Valid Info Types*

Description

Valid Info Types

Usage

```
validInfoTypes()
```

Index