

Package: kwb.en13508.2 (via r-universe)

August 26, 2024

Title Read and Write CCTV Inspection Data According to Norm EN13508-2

Version 0.3.0

Description functions to read and write CCTV inspections according to EN13508-2.

License MIT + file LICENSE

URL <https://github.com/kwb-r/kwb.en13508.2>

BugReports <https://github.com/kwb-r/kwb.en13508.2/issues>

Imports digest, kwb.utils (>= 0.15.0)

Suggests testthat (>= 3.0.0), ggplot2

Remotes github::kwb-r/kwb.utils

Encoding UTF-8

RoxygenNote 7.2.3

Config/testthat/edition 3

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

RemoteUrl <https://github.com/KWB-R/kwb.en13508.2>

RemoteRef HEAD

RemoteSha b4f698160a6cddc7cf9df50023bb26da06c422b7

Contents

| | |
|--|---|
| euCodedFileHeader | 2 |
| extractObservationBlocks | 3 |
| extractObservationData | 3 |
| getCodes | 4 |
| getHeaderLinesFromHeaderInfo | 5 |
| getLineDamageInfo | 5 |
| mergeInspectionData | 6 |
| numberOfInspections | 6 |
| plotObservations | 7 |
| readAndMergeEuCodedFiles | 7 |

| | |
|---------------------------------|----|
| readEuCodedFile | 8 |
| readEuCodedFiles | 9 |
| readPackageFile | 10 |
| setGlobalInspectionID | 11 |
| toEuFormat | 11 |
| toEuFormat_v1 | 12 |
| toEuFormat_v2 | 12 |
| valuesToCsv | 13 |
| writeEuCodedFile | 14 |
| writeEuCodedFiles | 14 |

Index **16**

| | |
|-------------------|---|
| euCodedFileHeader | <i>Generate List With EU Header Information</i> |
|-------------------|---|

Description

Generate List With EU Header Information

Usage

```
euCodedFileHeader(
  separator = ";",
  decimal = ".",
  quote = "\"",
  encoding = "ISO-8859-1",
  language = "en",
  year = 2010L
)
```

Arguments

| | |
|-----------|-----------------------|
| separator | default: ";" |
| decimal | default: "." |
| quote | default: "" |
| encoding | default: "ISO-8859-1" |
| language | default: "en" |
| year | default: 2010 |

Value

list with elements separator, decimal, quote, encoding, language, year

`extractObservationBlocks`*Extract Lines Between #C-Header and #Z End Tag*

Description

Extract Lines Between #C-Header and #Z End Tag

Usage

```
extractObservationBlocks(euLines, headerInfo, uniqueKey)
```

Arguments

| | |
|-------------------------|--|
| <code>euLines</code> | text lines read from EN13508.2-coded file |
| <code>headerInfo</code> | data frame with information about header lines |
| <code>uniqueKey</code> | identifier of C-header row, as given in <code>headerInfo\$uniqueKey</code> |

Value

list of vectors of character representing the "body" lines below the #C-headers of type specified in `uniqueKey`

`extractObservationData`*Extract Observations from EN13508.2-coded file*

Description

Extract Observations from EN13508.2-coded file

Usage

```
extractObservationData(  
  euLines,  
  headerInfo,  
  header.info,  
  file = "",  
  as.text = FALSE  
)
```

Arguments

| | |
|-------------|---|
| euLines | text lines read from EN13508.2-coded file |
| headerInfo | data frame with information about header lines |
| header.info | list as returned by <code>kwb.en13508.2::getFileHeaderFromEuLines</code> |
| file | optional. Name of the file from which euLines were read. |
| as.text | whether or not to keep columns in their original (character) type. The default is FALSE, i.e. columns that are expected to contain numeric values are converted to numeric, respecting the decimal separator that is given in header.info |

Value

data frame with columns A, B, C, ... as defined in EN13508.2 and a column `inspno` referring to the inspection number.

getCodes

Get EU Codes and Their Meaning

Description

Get a data frame containing EU codes and their meaning in different languages

Usage

```
getCodes(table = NULL, fields = NULL)
```

Arguments

| | |
|--------|--|
| table | name or vector of names of tables in the EU norm for which to get field information. Use <code>unique(getCodes())\$Table</code> to get the possible table names. |
| fields | set to a vector of field (column) names to restrict the columns returned |

Value

data frame

`getHeaderLinesFromHeaderInfo`*Get Header Lines From Header Info*

Description

Get Header Lines From Header Info

Usage

```
getHeaderLinesFromHeaderInfo(header.info)
```

Arguments

`header.info` list with elements encoding, language, separator, decimal, quote, year

`getLineDamageInfo`*Get Information on Line Damages*

Description

Get Information on Line Damages

Usage

```
getLineDamageInfo(observations, dbg = TRUE)
```

Arguments

`observations` data frame with observations. Required columns: I (= horizontal or vertical position), J (= code for line damage), inspno (number of inspection to which the observation belongs)

`dbg` if TRUE, debug messages are shown

Value

data frame with columns `ino` (inspection number), `ldno` (line damage number), `beg.at`, `end.at`, `beg.x` (position of line damage begin), `end.x` (position of line damage end), `length` (length of line damage)

mergeInspectionData *Merge Inspection Data*

Description

Merge inspections and observations provided in a list

Usage

```
mergeInspectionData(x, warn = FALSE, naToEmpty = TRUE)
```

Arguments

| | |
|-----------|---|
| x | list of elements each of which represents inspection data read from an EN13508.2-encoded file by means of readEuCodedFile . |
| warn | logical indicating whether to warn about different header information. By default, warnings are not shown. |
| naToEmpty | logical indicating whether or not to replace NA with an empty string constant "" in all columns of type character. The default is TRUE. |

Value

list with elements header.info, inspections, observations.

numberOfInspections *Number of Inspections*

Description

Get number of inspections from list of inspection data

Usage

```
numberOfInspections(x)
```

Arguments

| | |
|---|---|
| x | list of inspection data elements each of which was read from an EN 13508-2-coded file by means of readEuCodedFile |
|---|---|

Value

vector of integer representing the number of inspections in each element of inspectionDataList

plotObservations *Plot Observations per Pipe*

Description

Plot Observations per Pipe

Usage

```
plotObservations(survey, to_pdf = TRUE, matrix_dim = c(7, 1))
```

Arguments

| | |
|------------|---|
| survey | list with elements inspections and observations as e.g. returned by <code>kwb.en13508.2:::readEuCodedFiles</code> |
| to_pdf | if TRUE (default) the output goes into a temporary PDF file |
| matrix_dim | vector of two specifying the number of rows and columns, respectively of the matrix in which to arrange the single plots. |

Examples

```
# Install packages if not yet installed
## Not run:
install.packages("ggplot2")
devtools::install_github("guiastrennec/ggplus")

## End(Not run)

# Get example data that are contained in the package
survey <- kwb.en13508.2:::getExampleData()

# Create one plot per inspection in "survey"
kwb.en13508.2::plotObservations(survey, to_pdf = FALSE)
```

readAndMergeEuCodedFiles

Read and Merge Files in EN13508.2-Format

Description

Read files in EN13508.2-format using [readEuCodedFiles](#) and merge them by means of [mergeInspectionData](#)

Usage

```
readAndMergeEuCodedFiles(
  input.files,
  dbg = FALSE,
  name.convention = "norm",
  ...,
  add.inspid = FALSE,
  project = NULL,
  default.time = "22:22",
  error.file = NULL
)
```

Arguments

| | |
|------------------------------|--|
| <code>input.files</code> | full path to text file containing CCTV inspection results in the format described in DIN EN 13508-2 |
| <code>dbg</code> | if TRUE debug messages are shown |
| <code>name.convention</code> | passed to readEuCodedFiles |
| <code>...</code> | further arguments passed to readEuCodedFiles |
| <code>add.inspid</code> | if TRUE (the default is FALSE) a globally unique inspection ID (inspid) is added to the data frames in elements "inspections" and "observations" of the returned list. |
| <code>project</code> | name of project to which the data are related, such as: "Lausanne" |
| <code>default.time</code> | passed to setGlobalInspectionID |
| <code>error.file</code> | optional. Path to error file, passed to setGlobalInspectionID . |

| | |
|------------------------------|--|
| <code>readEuCodedFile</code> | <i>Read CCTV Inspection Data in EN13508-2 Format</i> |
|------------------------------|--|

Description

Read CCTV inspection data from file coded according to EN13508-2

Usage

```
readEuCodedFile(
  input.file,
  encoding = "unknown",
  file.encoding = NULL,
  read.inspections = TRUE,
  name.convention = c("norm", "camel", "snake")[1L],
  simple.algorithm = TRUE,
  warn = TRUE,
```



```

    dbg = TRUE,
    check.encoding = TRUE,
    ...
)

```

Arguments

| | |
|-------------------------------|--|
| <code>input.file</code> | full path to text file containing CCTV inspection results in the format described in DIN EN 13508-2 |
| <code>encoding</code> | default: "unknown", passed to readLines , see there. |
| <code>file.encoding</code> | Encoding to be assumed for the <code>input.file</code> . The default is NULL in which case the name of the encoding is read from the #A1 field of the <code>input.file</code> . |
| <code>read.inspections</code> | if TRUE, general inspection data (in #B-blocks) are read, otherwise skipped (use if function fails) |
| <code>name.convention</code> | one of c("norm", "camel", "snake") specifying the set of names used in the returned tables. "norm": as specified in the norm EN13508.2, "camel": CamelCase, "snake": snake_case |
| <code>simple.algorithm</code> | if TRUE (default), a simple (and faster) algorithm is used to extract the general information about the inspections from the #B-headers. It requires that all #B-headers have the same number and order of fields. If FALSE, another algorithm being able to treat differing #B-header rows is used. |
| <code>warn</code> | if TRUE, warnings are shown (e.g. if not all #A-header fields were found) |
| <code>dbg</code> | if TRUE, debug messages are shown, else not |
| <code>check.encoding</code> | logical indicating whether or not to check if the encoding string that is given in the #A1 header of the file is "known". The default is TRUE, i.e. the check is performed and an error is thrown if the encoding is not in the list of known encodings. |
| <code>...</code> | further arguments to be passed to <code>kwb.en13508.2::getObservationRecordsFromEuLines</code> |

Value

list with elements `header.info`, `inspections`, `observations`

`readEuCodedFiles` *Read Multiple CCTV Inspection Files*

Description

Read CCTV inspection data from multiple files coded according to EN13508-2

Usage

```
readEuCodedFiles(input.files, dbg = TRUE, append.file.names = TRUE, ...)
```

Arguments

| | |
|--------------------------------|---|
| <code>input.files</code> | vector of character paths to input files |
| <code>dbg</code> | if TRUE debug messages are shown |
| <code>append.file.names</code> | if TRUE (default) the filename will be provided in column <code>file</code> in the <code>inspections</code> element of the result list |
| <code>...</code> | arguments passed to <code>readEuCodedFile</code> , such as <code>read.inspections</code> , <code>simple.algorithm</code> , <code>warn</code> , see there. |

Value

list of sublists each of which is the result of a call to `readEuCodedFile`. The names of the list elements are constructed from the file names of the input files. Special characters in the file names are replaced with underscore. Names will get a preceding letter "x" if they start with a digit or with underscore. If files could not be read correctly, their indices are returned in attribute `which_failed`.

| | |
|------------------------------|--|
| <code>readPackageFile</code> | <i>Read CSV File from Package's extdata Folder</i> |
|------------------------------|--|

Description

Read CSV File from Package's extdata Folder

Usage

```
readPackageFile(file, ...)
```

Arguments

| | |
|-------------------|--|
| <code>file</code> | file name (without path) |
| <code>...</code> | additional arguments passed to <code>read.csv</code> |

Value

data frame representing the content of `file`

setGlobalInspectionID *Set Global Inspection ID*

Description

Convert inspections numbers (inspno) 1,2,3,... to globally unique identifiers (inspid), such as "e4b48d86"

Usage

```
setGlobalInspectionID(
  inspection.data,
  project = NULL,
  default.time = "22:22",
  name.convention = "norm",
  error.file = NULL
)
```

Arguments

| | |
|-----------------|---|
| inspection.data | list with elements header.info, inspections, observations |
| project | name of project to which the data are related, such as: "Lausanne" |
| default.time | default time string to use if column <inspection-time> is not available. Default: "22:22". A random number will be generated for the seconds, just to increase the chance that setting the time is enough to generate a unique key. |
| name.convention | one of c("norm", "camel", "snake") |
| error.file | optional. Path to file to which duplicates are written (if any). Default: "setGlobalInspectionID_duplicates.txt" |

Value

list with the same elements as in inspection.data but with columns inspid being added to the data frames "inspections" and "observations"

| | |
|------------|---|
| toEuFormat | <i>Generate Lines in EU Export Format</i> |
|------------|---|

Description

Generate Lines in EU Export Format

Usage

```
toEuFormat(inspection.data, version = 3L, ..., dbg = TRUE)
```

Arguments

| | |
|-----------------|--|
| inspection.data | inspection data as retrieved by e.g. readEuCodedFile |
| version | version of implementation. One of c(1, 2, 3) |
| ... | passed to toEuFormat_v2 |
| dbg | whether or not to show debug messages |

| | |
|---------------|--|
| toEuFormat_v1 | <i>Generate Lines in EU Export Format (v1)</i> |
|---------------|--|

Description

Generate lines in EU export format (version 1: slow)

Usage

```
toEuFormat_v1(header.info, inspections, observations, dbg = TRUE)
```

Arguments

| | |
|--------------|--|
| header.info | according to list element "header.info" of list returned by readEuCodedFile |
| inspections | according to list element "inspections" of list returned by readEuCodedFile |
| observations | according to list element "observations" of list returned by readEuCodedFile |
| dbg | whether or not to show debug messages |

| | |
|---------------|--|
| toEuFormat_v2 | <i>Generate Lines in EU Export Format (v2)</i> |
|---------------|--|

Description

Generate lines in EU export format (version 2: faster than version 1)

Usage

```
toEuFormat_v2(inspection.data, mycsv, ..., dbg = TRUE)
```

Arguments

| | |
|-----------------|--|
| inspection.data | inspection data as retrieved by e.g. readEuCodedFile |
| mycsv | logical. If TRUE "my" version of writing CSV is used (fast), otherwise CSV is written by means of write.table (slow) |
| ... | further arguments passed to dataFrameContentToTextLines |
| dbg | whether or not to show debug messages |

valuesToCsv

Values to CSV

Description

Values to CSV

Usage

```
valuesToCsv(
  x,
  dec = ".",
  sep = ",",
  na = "",
  qchar = "\"",
  qmethod = c("double", "escape")[1]
)
```

Arguments

| | |
|---------|--|
| x | vector of values representing a row in a CSV file |
| dec | decimal character |
| sep | field separating character |
| na | text to be used in case of NA values |
| qchar | quoting character to be used to surround text fields containing the field separator |
| qmethod | method used to indicate that a quoting character within a quoted text field is not the ending quote. Either "double" (double the quote character) or "escape" (backslash in front of the quoting character). |

writeEuCodedFile *Write Inspection Data to File in EU Format*

Description

Write Inspection Data to File in EU Format

Usage

```
writeEuCodedFile(
  inspection.data,
  output.file = NULL,
  version = 3L,
  dbg = TRUE,
  ...
)
```

Arguments

| | |
|-----------------|--|
| inspection.data | inspection data as retrieved by e.g. readEuCodedFile : list with elements header.info, inspections, observations. List element header.info is a list with at least the following elements: separator (column separator), decimal (decimal character "." or ","), quote (character used to quote texts in order to allow the separator sign to be used within the text). List element observations is a data.frame with required columns inspno (inspection number) |
| output.file | full path to output file |
| version | version of implementation. One of c(1, 2, 3) |
| dbg | if TRUE debug messages are shown |
| ... | passed to toEuFormat_v2 |

Value

if output.file is given, the path to the output file is returned, otherwise (output.file = NULL) the file content is returned as a vector of character representing the lines of the file.

writeEuCodedFiles *Write Inspection Data to Files in EU Format*

Description

Write inspection data to files in EU format with each file containing data of a fix number (default: 100) of inspections

Usage

```
writeEuCodedFiles(survey, file, blocksize = 100, dbg = TRUE)
```

Arguments

| | |
|-----------|--|
| survey | list with elements header.info, inspections, observations, just as required for parameter "inspection.data" in writeEuCodedFile . |
| file | Full path to output file. The file name must end with ".txt" which will be replaced with "_<i>_<j>.txt" with i and j being the number of the first and last inspection, respectively, contained in the file. |
| blocksize | number of inspections to be written to one file. Default: 100 |
| dbg | if TRUE (default) debug messages are shown. |

See Also

[writeEuCodedFile](#)

Index

euCodedFileHeader, [2](#)
extractObservationBlocks, [3](#)
extractObservationData, [3](#)

file, [10](#)

getCodes, [4](#)
getHeaderLinesFromHeaderInfo, [5](#)
getLineDamageInfo, [5](#)

mergeInspectionData, [6, 7](#)

numberOfInspections, [6](#)

plotObservations, [7](#)

readAndMergeEuCodedFiles, [7](#)
readEuCodedFile, [6, 8, 10, 12–14](#)
readEuCodedFiles, [7, 8, 9](#)
readLines, [9](#)
readPackageFile, [10](#)

setGlobalInspectionID, [8, 11](#)

toEuFormat, [11](#)
toEuFormat_v1, [12](#)
toEuFormat_v2, [12](#)

valuesToCsv, [13](#)

writeEuCodedFile, [14, 15](#)
writeEuCodedFiles, [14](#)