Package 'sf2010r'

March 29, 2023

Type Package

Title A Tool to Read StanForD2010 Forest Machine Reports

Version 0.1.1

Maintainer Helmer Belbo <helmer.belbo@nibio.no>

Description The main functions takes file names as arguments, and will read these files and return a list of tables containing most of the content of the xml document.

License MIT + file LICENSE

Encoding UTF-8

Imports data.table, digest, dplyr, lubridate, magrittr, plyr, purrr, rlang, stringr, tibble, tidyr, tidyselect, xml2

RoxygenNote 7.2.3

NeedsCompilation no

Author Helmer Belbo [aut, cre],
Wilson Lara Henao [aut, ctb]

R topics documented:

`fTZ` *Get Time zone from the StanForD2010 countrycode variables.*

Description

Get Time zone from the StanForD2010 countrycode variables.

Usage

```
fTZ(CountryCode = "")
```

Arguments

`CountryCode`	codes according to ISO3166-1 standard

Value

Time zone as Country/City

Examples

```
fTZ("752")
```

`getCombined.mwt` *Get Combined Machine Work Time for all and within a SF2010 .hpr file*

Description

Get Combined Machine Work Time for all and within a SF2010 .hpr file

Usage

```
getCombined.mwt(doc)
```

Arguments

`doc`	a StanFord2010 .mom xml-document

Value

a tibble

Examples

```
pth <- system.file(package = "sf2010r")
momf <- list.files(pth,".mom",ignore.case=TRUE,recursive=TRUE,full.names=TRUE)
momf_cmwt <- momf[which(stringr::str_detect(string=momf,pattern="combined_mwt"))]
doc <- xml2::read_xml(momf_cmwt[2])
getCombined.mwt(doc)
```

`getDelivery` *Delivery data from one delivery definition node*

Description

Delivery data from one delivery definition node

Usage

```
getDelivery(x)
```

Arguments

`x`	is a node tree for one delivery definition

Examples

```
pth <- system.file(package = "sf2010r")
fprfiles <- list.files(pth,".fpr",ignore.case=TRUE,recursive=TRUE,full.names=TRUE)
doc <- xml2::read_xml(fprfiles[1])
nodelist <- xml2::xml_find_all(doc, ".//d1:DeliveryDefinition")
getDelivery(nodelist[[1]]) %>% dplyr::glimpse()
plyr::ldply(nodelist[1], getDelivery)
```

getLocation *Location data from one Location node*

Description

Location data from one Location node

Usage

```
getLocation(x)
```

Arguments

x	is a node tree for one location

Examples

```
pth <- system.file(package = "sf2010r")
fprfiles <- list.files(pth,".fpr",ignore.case=TRUE,recursive=TRUE,full.names= TRUE)
doc <- xml2::read_xml(fprfiles[1])
locationlist <- xml2::xml_find_all(doc, ".//d1:LocationDefinition")
getLocation(locationlist[[1]]) %>% dplyr::glimpse()
plyr::ldply(locationlist[1], getLocation)
```

getLogs *Get logsdata for all stems within a SF2010 .hpr file*

Description

Get logsdata for all stems within a SF2010 .hpr file

Usage

```
getLogs(doc)
```

Arguments

doc	a StanFord2010 .hpr xml-document

Value

a tibble

Examples

```
pth <- system.file(package = "sf2010r")
hprfiles <- list.files(pth,".hpr",recursive=TRUE,full.names=TRUE)
doc <- xml2::read_xml(hprfiles[1])
getLogs(doc)
```

`getMachineReportHeader` *Organize the header data of the StanForD2010 report into a tibble*

Description

Organize the header data of the StanForD2010 report into a tibble

Usage

```
getMachineReportHeader(doc)
```

Arguments

`doc`	is an xml document

Value

a tibble

Examples

```
pth <- system.file(package = "sf2010r")
hprfiles <-  list.files(pth,".hpr",recursive=TRUE,full.names=TRUE)
doc <- xml2::read_xml(hprfiles[1])
getMachineReportHeader(doc)
```

getMom.all *mom-file reader function*

Description

mom-file reader function

Usage

```
getMom.all(momfile)
```

Arguments

momfile	filename and path of the mom file to read

Value

A list of data.frames: machinereport_meta, operators, objects, tracking, CombProdDat = a table listing production data (stems, volumes, etc), CombinedMachineWorkTime = a table listing work time, OperatorWorkTime = a table listing work time for each operator, IndividualMachineWorkTime

Examples

```
pth <- system.file(package = "sf2010r")
momfiles <- list.files(pth,".mom$",recursive=TRUE,ignore.case=TRUE,full.names= TRUE)
momtest1 <- getMom.all(momfiles[1])
```

getMom.cmwt.data *Combined machine work time data from one combined machine work time node*

Description

Combined machine work time data from one combined machine work time node

Usage

```
getMom.cmwt.data(x)
```

Arguments

x	is a node tree for one Combined machine work time entry

Examples

```
pth <- system.file(package = "sf2010r")
momf <- list.files(pth,".mom",ignore.case=TRUE,recursive=TRUE,full.names= TRUE)
momf_cmwt <- momf[which(stringr::str_detect(string=momf,pattern="combined_mwt"))]
doc <- xml2::read_xml(momf_cmwt[2])#'
cmwtlist <- xml2::xml_find_all(doc, ".//d1:CombinedMachineWorkTime")
getMom.cmwt.data(cmwtlist[[1]]) %>% dplyr::glimpse()
plyr::ldply(cmwtlist[1:2], getMom.cmwt.data)
```

getMom.imwt.activity *Indivdual machine work time activity data from one indivdual machine work time node*

Description

Indivdual machine work time activity data from one indivdual machine work time node

Indivdual machine work time activity data from one indivdual machine work time node

Usage

```
getMom.imwt.activity(x)

getMom.imwt.activity(x)
```

Arguments

x	is a node tree for one Indivdual machine work time entry

Examples

```
pth. <- system.file(package = "sf2010r")
momfiles <- list.files(pth.,".mom",ignore.case=TRUE,recursive=TRUE,full.names=TRUE)
momfiles_imwt <- momfiles[which(stringr::str_detect(momfiles,pattern="individual_mwt"))]
doc <- xml2::read_xml(momfiles_imwt[2])
imwtlist <- xml2::xml_find_all(doc, ".//d1:IndividualMachineWorkTime")
getMom.imwt.activity(imwtlist[[41]]) %>% dplyr::glimpse()
 plyr::ldply(imwtlist[25:42], getMom.imwt.activity)
pth. <- system.file(package = "sf2010r")
momfiles <- list.files(pth.,".mom",ignore.case=TRUE,recursive=TRUE,full.names=TRUE)
momfiles_imwt <- momfiles[which(stringr::str_detect(string=momfiles,pattern="individual_mwt"))]
doc <- xml2::read_xml(momfiles_imwt[2])
imwtlist <- xml2::xml_find_all(doc, ".//d1:IndividualMachineWorkTime")
getMom.imwt.activity(imwtlist[[41]]) %>% dplyr::glimpse()
 plyr::ldply(imwtlist[25:42], getMom.imwt.activity)
```

getMom.imwt.production *Indivdual machine work time production data from one individual machine work time node*

Description

Indivdual machine work time production data from one individual machine work time node

Indivdual machine work time production data from one individual machine work time node

Usage

```
getMom.imwt.production(x)

getMom.imwt.production(x)
```

Arguments

`x`	is a node tree for one Indivdual machine work time entry

Examples

```
pth <- system.file(package = "sf2010r")
momfiles <- list.files(pth,".mom",ignore.case=TRUE,recursive=TRUE,full.names=TRUE)
momfiles_imwt <- momfiles[which(stringr::str_detect(string = momfiles, pattern = "individual_mwt"))]
doc <- xml2::read_xml(momfiles_imwt[2])
imwtlist <- xml2::xml_find_all(doc, ".//d1:IndividualMachineWorkTime")
getMom.imwt.production(imwtlist[[3]]) %>% dplyr::glimpse()
plyr::ldply(imwtlist[92:94], getMom.imwt.production)
pth. <- system.file(package = "sf2010r")
momfiles <- list.files(pth.,".mom",ignore.case=TRUE,recursive=TRUE,full.names=TRUE)
momfiles_imwt <- momfiles[which(stringr::str_detect(string=momfiles,pattern="individual_mwt"))]
doc <- xml2::read_xml(momfiles_imwt[2])
imwtlist <- xml2::xml_find_all(doc, ".//d1:IndividualMachineWorkTime")
getMom.imwt.production(imwtlist[[3]]) %>% dplyr::glimpse()
plyr::ldply(imwtlist[92:94], getMom.imwt.production)
```

getMTPlogs *Fetch the multi tree processed logs belonging to one multi-stem entry in hpr*

Description

Fetch the multi tree processed logs belonging to one multi-stem entry in hpr

Usage

```
getMTPlogs(x)
```

Arguments

x	a StanFord2010 .hpr stem node tree

Value

a tibble with all logs belonging to one stem

Examples

```
pth <- system.file(package = "sf2010r")
hprfiles <- list.files(pth,".hpr",recursive=TRUE,full.names=TRUE)
doc <- xml2::read_xml(hprfiles[1])
stemlist <- xml2::xml_find_all(doc, ".//d1:Stem")
pcat <- ".//d1:ProcessingCategory"
wtch <- which(xml2::xml_text(xml2::xml_find_all(stemlist,pcat))=="MultiTreeProcessing")
getMTPlogs(stemlist[1])
if(length(wtch) > 0) { getMTPlogs(stemlist[wtch[1]])}
```

`getObjectDefinition` *Extracting all data defining one cut object from one ObjectDefinition xml-tree*

Description

Extracting all data defining one cut object from one ObjectDefinition xml-tree

Usage

```
getObjectDefinition(x)
```

Arguments

x	the object definition XML-tree

Value

a tibble, number of rows = number of sub objects

Examples

```
pth <- system.file(package = "sf2010r")
hprfiles <-  list.files(pth,".hpr",recursive=TRUE,full.names=TRUE)
doc <- xml2::read_xml(hprfiles[1])
Objects_nodes <- xml2::xml_find_all(doc, "//d1:ObjectDefinition")
getObjectDefinition(Objects_nodes[1])
plyr::ldply(Objects_nodes, getObjectDefinition)
```

getObjects *Extracting all cut object definitions from doc*

Description

Extracting all cut object definitions from doc

Usage

```
getObjects(doc)
```

Arguments

doc	should be an Stanford2010 xml document

Value

a tibble, number of rows = number of objects and sub objects in doc

Examples

```
pth <- system.file(package = "sf2010r")
hprfiles <- list.files(pth,".hpr",recursive=TRUE,full.names=TRUE)
doc <- xml2::read_xml(hprfiles[1])
getObjects(doc)
```

getOperators *Get the operator definitions from a StanFord2010 xml document*

Description

Get the operator definitions from a StanFord2010 xml document

Usage

```
getOperators(doc)
```

Arguments

doc	the xml document

Value

a tibble, one row for each operator

Examples

```
pth <- system.file(package = "sf2010r")
hprfiles <-  list.files(pth,".hpr",recursive=TRUE,full.names=TRUE)
doc <- xml2::read_xml(hprfiles[1])
getOperators(doc)
```

`getProductDef` *Product def from product definition nodetree*

Description

Product def from product definition nodetree

Usage

```
getProductDef(x)
```

Arguments

`x`	is a node tree of ProductDefinition

Examples

```
pth <- system.file(package = "sf2010r")
hprfiles <-  list.files(pth,".hpr",recursive=TRUE,full.names= TRUE)
doc <- xml2::read_xml(hprfiles[1])
ProductsList <- xml2::xml_find_all(doc, ".//d1:ProductDefinition" )
getProductDef(ProductsList[[1]]) %>% dplyr::glimpse()
plyr::ldply(ProductsList, getProductDef )
```

`getProductDefs` *Get all product definitions*

Description

Get all product definitions

Usage

```
getProductDefs(doc)
```

Arguments

`doc`	a StanFord2010 xml document

Value

a list

Examples

```
pth <- system.file(package = "sf2010r")
hprfiles <-  list.files(pth,".hpr",recursive=TRUE,full.names= TRUE)
doc <- xml2::read_xml(hprfiles[1])
getProductDefs(doc)
doc <- xml2::read_xml(hprfiles[2])
getProductDefs(doc)
```

getProductMatrixes *Get all price matrixes*

Description

Get all price matrixes

Usage

```
getProductMatrixes(doc)
```

Arguments

doc	an StanFord2010 xml document having product definitions

Value

A list

Examples

```
pth <- system.file(package = "sf2010r")
hprfiles <-  list.files(pth,".hpr",recursive=TRUE,full.names= TRUE)
doc <- xml2::read_xml(hprfiles[1])
pms <- getProductMatrixes(doc)
```

getProductMatrixItems *Get the Product Matrix items*

Description

Get the Product Matrix items

Usage

```
getProductMatrixItems(x)
```

Arguments

x	is a Product definition tree from StanFord2010 xml document

Value

A tibble, or NULL if no products are defined

Examples

```
pth <- system.file(package = "sf2010r")
hprfiles <-  list.files(pth,".hpr",recursive=TRUE,full.names=TRUE)
doc <- xml2::read_xml(hprfiles[1])
Productslist <- xml2::xml_find_all(doc, ".//d1:ProductDefinition" )
getProductMatrixItems(Productslist[[1]])
```

getSpeciesGroupDef *Species group def from one SpeciesGroupDefinition node*

Description

Species group def from one SpeciesGroupDefinition node

Usage

```
getSpeciesGroupDef(x)
```

Arguments

x	is a node tree for one SpeciesGrouDefinition

Examples

```
pth <- system.file(package = "sf2010r")
hprfiles <-  list.files(pth,".hpr",recursive=TRUE,full.names=TRUE)
doc <- xml2::read_xml(hprfiles[1])
SpeciesList <- xml2::xml_find_all(doc, ".//d1:SpeciesGroupDefinition" )
getSpeciesGroupDef(SpeciesList[[1]])
species <- plyr::ldply(SpeciesList, getSpeciesGroupDef)
```

---

getSpeciesGroupDefinitions *Get all species definitions within a SF2010 doc*

Description

Get all species definitions within a SF2010 doc

Usage

```
getSpeciesGroupDefinitions(doc)
```

Arguments

doc	a StanFord2010 xml document

Value

a tibble

Examples

```
pth <- system.file(package = "sf2010r")
hprfiles <-  list.files(pth,".hpr",recursive=TRUE,full.names=TRUE)
doc <- xml2::read_xml(hprfiles[1])
getSpeciesGroupDefinitions(doc)
```

---

getStemdata *Stem data from one Stem node*

Description

Stem data from one Stem node

Usage

```
getStemdata(x)
```

Arguments

x	is a node tree for one stem

Examples

```
pth <- system.file(package = "sf2010r")
hprfiles <- list.files(pth,".hpr",recursive=TRUE,full.names=TRUE)
doc <- xml2::read_xml(hprfiles[3])
stemlist <- xml2::xml_find_all(doc, ".//d1:Stem")
getStemdata(stemlist[[1]]) %>% dplyr::glimpse()
plyr::ldply(stemlist[1:10], getStemdata)
```

getStemGrades *Fetch the stem grades for each stem in hpr*

Description

Fetch the stem grades for each stem in hpr

Usage

```
getStemGrades(x)
```

Arguments

x	a StanFord2010 stem nodetree

Details

NB: One stem section might have several grades simultaneously

Value

a tibble.

Examples

```
pth <- system.file(package = "sf2010r")
hprfiles <- list.files(pth,".hpr",recursive = TRUE, full.names= TRUE)
doc <- xml2::read_xml(hprfiles[1])
stemlist <- xml2::xml_find_all(doc, ".//d1:Stem")
getStemGrades(stemlist[[1]])
plyr::ldply(stemlist[1:3], getStemGrades)
```

getStems *Get stemdata for all stems within a SF2010 .hpr file*

Description

Get stemdata for all stems within a SF2010 .hpr file

Usage

```
getStems(doc)
```

Arguments

doc	a StanFord2010 .hpr xml-document

Value

a tibble

Examples

```
pth <- system.file(package = "sf2010r")
hprfiles <- list.files(pth,".hpr",recursive=TRUE,full.names= TRUE)
doc <- xml2::read_xml(hprfiles[3])
getStems(doc)
```

getStemsAndLogs *Fetch all stems and all logs in hpr*

Description

the function returns a list with five data frames; stems is all the stems stplogs is all single tree processed logs mtplogs is all multi tree processed logs stemgrades is the grades of all stems stemdias is the diameter vector for the stem.

Usage

```
getStemsAndLogs(doc)
```

Arguments

doc	a StanFord2010 .hpr document

Value

a list with five data frames.

Examples

```
pth <- system.file(package = "sf2010r")
hprfiles <- list.files(pth,".hpr",recursive=TRUE,full.names=TRUE)
doc <- xml2::read_xml(hprfiles[1])
stl <- getStemsAndLogs(doc)
doc <- xml2::read_xml(hprfiles[2])
stl <- getStemsAndLogs(doc)
doc <- xml2::read_xml(hprfiles[3])
stl <- getStemsAndLogs(doc)
```

`getStemTypeDefs` *get stem type definitions from one SpeciesGroupDefinition*

Description

get stem type definitions from one SpeciesGroupDefinition

Usage

```
getStemTypeDefs(x)
```

Arguments

x	is a node tree of SpeciesGrouDefinition

Examples

```
pth <- system.file(package = "sf2010r")
hprfiles <-  list.files(pth,".hpr",recursive=TRUE,full.names= TRUE)
doc <- xml2::read_xml(hprfiles[1])
SpeciesList <- xml2::xml_find_all(doc, ".//d1:SpeciesGroupDefinition" )
getStemTypeDefs(SpeciesList[[1]])
```

`getStemTypes` *Get all stem type definitions for all SpeciesGroupDefinitions*

Description

Get all stem type definitions for all SpeciesGroupDefinitions

Usage

```
getStemTypes(doc)
```

Arguments

`doc` a StanFord2010 xml document

Value

a list

Examples

```
pth <- system.file(package = "sf2010r")
hprfiles <-  list.files(pth,".hpr",recursive=TRUE,full.names=TRUE)
doc <- xml2::read_xml(hprfiles[1])
getStemTypes(doc)
```

`getSTPlogs` *Fetch the single tree processed logs from one stem node tree*

Description

Fetch the single tree processed logs from one stem node tree

Usage

```
getSTPlogs(x)
```

Arguments

`x` a StanFord2010 .hpr stem node tree

Value

a tibble with all logs belonging to one stem

Examples

```
pth <- system.file(package = "sf2010r")
hprfiles <- list.files(pth,".hpr",recursive=TRUE,full.names=TRUE)
doc <- xml2::read_xml(hprfiles[1])
stemlist <- xml2::xml_find_all(doc, ".//d1:Stem")
pcat <- ".//d1:ProcessingCategory"
wtch <-  which(xml2::xml_text(xml2::xml_find_all(stemlist,pcat))=="MultiTreeProcessing")
getSTPlogs(stemlist[1]) %>% dplyr::glimpse()
if(length(wtch) > 0) { getSTPlogs(stemlist[wtch[1]]) %>% dplyr::glimpse()}
```

`getSTP_diameters` *get SingleTreeProcessed tree's diametres*

Description

get SingleTreeProcessed tree's diametres

Usage

```
getSTP_diameters(doc)
```

Arguments

doc	a hpr document (xml)

Value

a tibble

Examples

```
pth <- system.file(package = "sf2010r")
hprfiles <- list.files(pth,".hpr",recursive=TRUE,full.names= TRUE)
doc <- xml2::read_xml(hprfiles[1])
getSTP_diameters(doc)
doc <- xml2::read_xml(hprfiles[2])
getSTP_diameters(doc)
doc <- xml2::read_xml(hprfiles[3])
getSTP_diameters(doc)
```

`getTracking.data` *Tracking data from mom-files#'*

Description

Tracking data from mom-files#'

Tracking data from mom-files#'

Usage

```
getTracking.data(doc)

getTracking.data(doc)
```

Arguments

doc	a StanFord2010 .mom xml-document

Examples

```
pth <- system.file(package = "sf2010r")
momfiles <- list.files(pth,".mom",ignore.case=TRUE,recursive=TRUE,full.names=TRUE)
doc <- xml2::read_xml(momfiles[3])#'
getTracking.data(doc) %>% dplyr::glimpse()
pth <- system.file(package = "sf2010r")
momfiles <- list.files(pth,".mom",ignore.case=TRUE,recursive=TRUE,full.names= TRUE)
doc <- xml2::read_xml(momfiles[3])#'
getTracking.data(doc) %>% dplyr::glimpse()
```

`get_Harv_data` *Recursive processing of other functions in the package*

Description

Recursive processing of other functions in the package

Usage

```
get_Harv_data(pth, funcs = c(mom = "getMom.all", hpr = "hprdata"))
```

Arguments

`pth`	set of file paths
`funcs`	names of other inpackage functions to iterate over

Examples

```
pth <- list.files(path =  system.file(package = "sf2010r"),
pattern = ".", recursive = TRUE, full.names= TRUE)
tmp <- get_Harv_data(pth[4:5])
```

`hprdata` *Hpr-file reader function*

Description

Hpr-file reader function

Usage

```
hprdata(hprfile)
```

Arguments

`hprfile`	filename and path of the hpr file to read

Value

A list of data.frames: stems, products, logs, machinereport_meta, operators, objects, stem_grades, pricematrixes, stemdiametervectors, and stemtypes.

Examples

```
hprfiles <- list.files(path =  system.file(package = "sf2010r"),
pattern = ".hpr", recursive = TRUE, full.names= TRUE)
hprtest1 <- hprdata(hprfiles[1])
hprtest2 <- hprdata(hprfiles[2])
hprtest3 <- hprdata(hprfiles[3])
```

`price_matr_entry_base_log_class`

finding price matrix entry for base logs

Description

For all productKeys: find the 49/20 log value

Usage

```
price_matr_entry_base_log_class(
  pricematrixes,
  base_lengthcm = 490,
  base_diamm = 200
)
```

Arguments

`pricematrixes`	is a data frame with all products diams lengths pricetags
`base_lengthcm`	length of baselog, normally 49 dm
`base_diamm`	diameter of baselog, normally 20 cm

Value

a tibble of all products' base logs and pricetags

Examples

```
pth <- system.file(package = "sf2010r")
hprfiles <-  list.files(pth,".hpr",recursive=TRUE,full.names=TRUE)
doc <- xml2::read_xml(hprfiles[1])
pricematrixes <- getProductMatrixes(doc)
price_matr_entry_base_log_class(pricematrixes)
```

sf2010r_example *Get path to Stanford2010 example files in the package*

Description

sf2010r comes bundled with a number of sample files in its ‘inst/extdata‘ directory. This function make them easy to access. The function is inspired by the readr_example in the readr package

Usage

```
sf2010r_example(fileending = NULL)
```

Arguments

fileending	one of the file endings in StanFord2010; "hpr", "hqc" etc

Examples

```
sf2010r_example()
sf2010r_example(fileending = "hpr")
```

xml_childs_dt *Make a data table of node children values and corresponding attribute values*

Description

This function takes a xml node and return a data.table dataframe, where the names are the child node names and the values are the node values. Attributes and attribute values are included.

Usage

```
xml_childs_dt(y)
```

Arguments

y	a list xml_node with some children nodes

Value

a data table

Examples

```
pth <- system.file(package = "sf2010r")
hprfiles <-  list.files(pth,".hpr",recursive=TRUE,full.names=TRUE)
doc <- xml2::read_xml(hprfiles[1])
x <- xml2::xml_children(doc)[2] # Get one node of doc, the first is normally a header
xml_childs_dt(x[[1]]) # The function returns a data.table
x <- xml2::xml_children(x)
xml_childs_dt(x[[27]])
 x <- xml2::xml_children(x[[27]])
xml_childs_dt(x)
x %>% purrr::map_dfr( ~ xml_childs_dt(.x))  # Converted to tibble
```

`xml_childs_nchr` *Make a named character vector of children values*

Description

This function takes a xml node and return a named character vector, where the names are the child node names and the values are the node values. No attributes or attribute values are included.

Usage

```
xml_childs_nchr(y)
```

Arguments

`y`	a list xml_node with some children nodes

Value

a named vector

Examples

```
pth <- system.file(package = "sf2010r")
hprfiles <-  list.files(pth,".hpr",recursive=TRUE,full.names=TRUE)
doc <- xml2::read_xml(hprfiles[1])
x <-  xml2::xml_children(doc)[1] # Get one node of doc, the first is normally a header
xml_childs_nchr(x[[1]]) # The function returns a named vector
x %>% purrr::map_dfr( ~ xml_childs_nchr(.x))  # Converted to tibble
```

Index