Package 'XBRL'

October 12, 2022

Type Package

Title Extraction of Business Financial Information from 'XBRL' Documents

Version 0.99.19.1

Date 2021-10-18

Author Roberto Bertolusso

Maintainer Roberto Bertolusso <rbertolusso@rice.edu>

Description Functions to extract business financial information from an Extensible Business Reporting Language ('XBRL') instance file and the associated collection of files that defines its 'Discoverable' Taxonomy Set ('DTS').

License GPL (>= 2)

Imports utils, Rcpp (>= 0.10.4)

SystemRequirements libxml2 (>= 2.9.1)

LinkingTo Rcpp

NeedsCompilation yes

Repository CRAN

Date/Publication 2022-03-28 17:03:52 UTC

R topics documented:

XBRL-package *Extraction of business financial information from XBRL documents.*

Description

Functions to extract business financial information from an Extensible Business Reporting Language (XBRL) instance file and the associated collection of files that defines its Discoverable Taxonomy Set (DTS), usually disseminated across different remote locations.

XBRL documents employ many technologies defined by XML standards, such as XML Schema, Namespaces, XLink, and XPath, which make the extraction of information involved. The functions provided by this package address this complexity, returning data frames containing the enclosed information readily available to perform analyses.

XBRL has been successfully tested in analyzing 10-K and 10-Q submissions from USA filing companies that are publicly available at the EDGAR system of the Securities and Exchange Commission (SEC). It has been reported to work with inline XBRL documents.

As published taxonomy files (such as "http://xbrl.fasb.org/us-gaap/2013/elts/us-gaap-2013-01-31.xsd") are immutable and are used by most filers, XBRL offers the option of downloading them only the first time they are referred, keeping a local cache copy that can be used from then on. This speeds-up the process, specially on slow connections.

XBRL is not and does not aim to be (at least yet) a validating XBRL processor.

XBRL is still a work in progress. As such, functions and produced data frames may change structure in future versions until stabilization.

Details

Package:	XBRL
Type:	Package
Version:	0.99.11
Date:	2014-01-01
License:	GPL (>=2)

The package offers 3 levels of access:

1) A function that "does it all" (see xbrlDoAll).

2) A "mutable state" function that exposes "methods" to be called individually (see XBRL).

3) Individual specialized functions in C++, glued to a corresponding R calling function (see xbrlParse).

See examples of use below.

Author(s)

Roberto Bertolusso and Marek Kimmel

Maintainer: Roberto Bertolusso <rbertolusso@rice.edu>

References

http://www.xbrl.org

See Also

xbrlDoAll, XBRL, xbrlParse

Examples

```
## Not run:
## Setting stringsAsFactors = FALSE is highly recommended
## to avoid data frames to create factors from character vectors.
options(stringsAsFactors = FALSE)

## Load the library
library(XBRL)

## XBRL instance file to be analyzed, accessed
## directly from SEC website:
inst <- "https://www.sec.gov/Archives/edgar/data/21344/000002134413000050/ko-20130927.xml"

## Level 1: Function that does all work and returns
## a list of data frames with extracted information:
xbrl.vars <- xbrlDoAll(inst, verbose=TRUE)

## Level 2: Using the XBRL() "mutable state" function:
xbrl <- XBRL()
xbrl$setCacheDir("XBRLcache")
xbrl$openInstance(inst)
## Perform a discovery of the taxonomy:
xbrl$processSchema(xbrl$getSchemaName())
## Process instance file:
xbrl$processContexts()
xbrl$processUnits()
xbrl$processFacts()
xbrl$processFootnotes()
xbrl$closeInstance()
xbrl.vars <- xbrl$getResults()

## Level 3: Using specialized functions that call C++ code directly:
## Parse the instance (doc is an pointer to external memory that needs to be freed):
## NOTE: in this case, inst needs to be a local file, or accessible
##       as http (not https).
inst <- "ko-20130927.xml"
doc <- xbrlParse(inst)
## Get a data frame with facts:
fct <- xbrlProcessFacts(doc)
## Get a data frame with contexts:
cts <- xbrlProcessContexts(doc)
## Get a data frame with units:
unt <- xbrlProcessUnits(doc)
## Free the external memory used:
```

```
xbrlFree(doc)

## End(Not run)
```

`Low-level functions` *Set of low-level functions to parse an XBRL file and extract information from it.*

Description

`xbrlParse` performs the parsing of the document. It returns a pointer to external memory used by the low-level functions. `xbrlFree` releases the external memory used.

`xbrlGetSchemaName` is used on an instance document to extract the name of the associated schema. `xbrlGetLinkbaseNames` and `xbrlGetImportNames` are used on a taxonomy document to extract the names of linked documents during the process of discovery.

`xbrlProcessXXX` functions are used to extract information from taxonomy and instance files in the form of data frames.

Usage

```
## Use on any XBRL file
xbrlParse(file)
xbrlFree(doc)

## Use on an instance file
xbrlGetSchemaName(doc)

## Use on a taxonomy file
xbrlGetLinkbaseNames(doc)
xbrlGetImportNames(doc)

xbrlProcessElements(doc)
xbrlProcessLabels(doc)
xbrlProcessArcs(doc, arcType)

## Use on a schema file
xbrlProcessRoles(doc)

## Use on an instance file
xbrlProcessContexts(doc)
xbrlProcessFacts(doc)
xbrlProcessUnits(doc)
xbrlProcessFootnotes(doc)
```

Arguments

`file`	the name of the XBRL document. It can be a URL or local file name
`doc`	pointer to external memory, returned by `xbrlParse`
`arcType`	Either "presentation", "calculation", or "definition"

Value

`xbrlParse` returns an external pointer.

`xbrlFree` returns NULL.

`xbrlGetSchemaName`, `xbrlGetLinkbaseNames` and `xbrlGetImportNames` return a character vector

`xbrlProcessXXX` returns a data.frame

Author(s)

Roberto Bertolusso and Marek Kimmel

See Also

`xbrlDoAll`

Examples

```
## Not run:
## Setting stringsAsFactors = FALSE is highly recommended
## to avoid data frames to create factors from character vectors.
options(stringsAsFactors = FALSE)

## XBRL instance file to be analyzed.
## NOTE: in this case, inst needs to be a local file, or accessible
##       as http (not https).
inst <- "ko-20130927.xml"
## Parse the instance (doc is an pointer to
## external memory that needs to be freed after use):
doc <- xbrlParse(inst)
## Get a data frame with facts:
fct <- xbrlProcessFacts(doc)
## Get a data frame with contexts:
cts <- xbrlProcessContexts(doc)
## Get a data frame with units:
unt <- xbrlProcessUnits(doc)
sch <- xbrlGetSchemaName(doc)
## Free the external memory used:
xbrlFree(doc)

dname <- dirname(inst)
## Parse the schema file:
docS <- xbrlParse(paste0(dname,"/",sch))
## Get roles:
rls <- xbrlProcessRoles(docS)
head(rls)
```

```
## Free the external memory used:
xbrlFree(docS)

## End(Not run)
```

XBRL *"Mutable state" function that exposes "methods" that analyze an XBRL instance an its associated DTS.*

Description

XBRL is a "mutable state" function that offers a object-oriented programming-like interface, by exposing "methods" that perform accions on XBRL files. It keeps a list of data frames that are populated by the "methods" by calls to the low-level functions (see xbrlParse).

The approach used is derived from an example included in *10.7 Scope* of *An Introduction to R*. The employed terminology may be inaccurate, hence the quotations. XBRL "methods" are called by xbrlDoAll. See examples of use below.

Usage

```
XBRL()
```

Arguments

	no arguments

Value

a list of "methods"

Author(s)

Roberto Bertolusso and Marek Kimmel

See Also

xbrlDoAll, xbrlParse

Examples

```
## Not run:
## Setting stringsAsFactors = FALSE is highly recommended
## to avoid data frames to create factors from character vectors.
options(stringsAsFactors = FALSE)

## XBRL instance file to be analyzed, accessed
## directly from SEC website:
inst <- "https://www.sec.gov/Archives/edgar/data/21344/000002134413000050/ko-20130927.xml"
```

```
xbrl <- XBRL()
xbrl$setCacheDir("XBRLcache")
xbrl$openInstance(inst)
## Perform a discovery of the taxonomy:
xbrl$processSchema(xbrl$getSchemaName())
## Process instance file:
xbrl$processContexts()
xbrl$processUnits()
xbrl$processFacts()
xbrl$processFootnotes()
xbrl$closeInstance()
xbrl.vars <- xbrl$getResults()

## End(Not run)
```

`xbrlDoAll` *Function to do all required work on an XBRL instance and its associated DTS.*

Description

`xbrlDoAll` performs all the steps necessary to extract a list of dataframes, starting from an XBRL instance file. `xbrlDoAll` offers the highest level of abstraction from the required steps. A medium-level access can be achieved by using `XBRL`. A low-level access can be achieved by using the functions related to `xbrlParse`.

Usage

```
xbrlDoAll(file.inst, cache.dir = "xbrl.Cache", prefix.out = NULL,
          verbose = FALSE, delete.cached.inst = TRUE)
```

Arguments

`file.inst`	the name of the instance file
`cache.dir`	the name of a local directory to be used as a cache of discovered taxonomies. if NULL, no cache is performed
`prefix.out`	if given, prefix used to create csv files with the content of the data frames produced
`verbose`	Set to TRUE to see output of the process of discovery and analysis if documents
`delete.cached.inst`	Set to TRUE to delete downloaded instance, schema and linkbase files

Value

a list of data frames with extracted information

Author(s)

Roberto Bertolusso and Marek Kimmel

See Also

XBRL, xbrlParse

Examples

```
## Not run:
## Setting stringsAsFactors = FALSE is highly recommended
## to avoid data frames to create factors from character vectors.
options(stringsAsFactors = FALSE)

## XBRL instance file to be analyzed, accessed
## directly from SEC website:
inst <- "https://www.sec.gov/Archives/edgar/data/21344/000002134413000050/ko-20130927.xml"

xbrl.vars <- xbrlDoAll(inst, cache.dir="XBRLcache", prefix.out="out", verbose=TRUE)

## End(Not run)
```

---

xbrlSECdev01 *"Mutable state" function that exposes "methods" that analyze a US SEC submission.*

Description

xbrlSECtmp is a "mutable state" function (see XBRL). It exposes the method showPresentationHierarchy. It is a work in progress.

Usage

```
xbrlSECdev01(xbrl.vars)
```

Arguments

xbrl.vars	list of variables returned by xbrlDoAll

Details

- showPresentationHierarchy(showLabels=TRUE, showFacts=FALSE, file="") Shows a tree of the presentations provided.

Value

a list of "methods"

Author(s)

Roberto Bertolusso and Marek Kimmel

See Also

xbrlDoAll, xbrlParse

Examples

```
## Not run:
## Setting stringsAsFactors = FALSE is highly recommended
## to avoid data frames to create factors from character vectors.
options(stringsAsFactors = FALSE)

## XBRL instance file to be analyzed, accessed
## directly from SEC website:
inst <- "https://www.sec.gov/Archives/edgar/data/21344/000002134413000050/ko-20130927.xml"

xbrl.vars <- xbrlDoAll(inst)
xbrl.sec <- xbrlSECdev01(xbrl.vars)
xbrl.sec$showPresentationHierarchy(showLabels=TRUE, showFacts=TRUE)

## End(Not run)
```

Index