

# Package ‘XML2R’

July 2, 2014

**Title** EasieR XML data collection

**Version** 0.0.6

**Author** Carson Sievert <sievert@iastate.edu>

**Maintainer** Carson Sievert <sievert@iastate.edu>

**Description** XML2R is a framework that reduces the effort required to transform XML content into number of tables while preserving parent to child relationships.

**License** GPL (>= 2)

**Depends** R (>= 2.15.1), XML

**Imports** RCurl, plyr

**Date** 2014-03-09

**URL** <http://cpsievert.github.com/XML2R>

**BugReports** <http://github.com/cpsievert/XML2R/issues>

**NeedsCompilation** no

**Repository** CRAN

**Date/Publication** 2014-03-10 00:29:38

## R topics documented:

add_key . . . . .	2
collapse_obs . . . . .	2
docsToNodes . . . . .	3
listsToObs . . . . .	3
nodesToList . . . . .	4
re_name . . . . .	4
urlsToDocs . . . . .	5
XML2Obs . . . . .	5
XML2R . . . . .	7
<b>Index</b>	<b>8</b>

---

add_key	<i>Add a key to connect parents to descendants</i>
---------	--

---

### Description

This function creates a mapping from parent observations to its descendants (which useful for merging/joining tables). Either an existing value in the parent observation can be recycled to its descendants or a new column will be created (if recycle is missing).

### Usage

```
add_key(obs, parent, recycle, key.name, quiet = FALSE)
```

### Arguments

obs	list. Should be the output from <a href="#">listsToObs</a> .
parent	character string. Should be present in the names of obs.
recycle	character string that matches a variable name among parent observations.
key.name	The desired column name of the newly generated key.
quiet	logical. Include message about the keys being generated?

### Value

A list of observations.

---

collapse_obs	<i>Collapse a list of observations into a list of tables.</i>
--------------	---

---

### Description

This function aggregates all observations with a similar name into a common table. Note that observations with a particular name don't need consistent variables (any missing information is filled with NAs).

### Usage

```
collapse_obs(obs)
```

### Arguments

obs	list of observations.
-----	-----------------------

### Value

Returns list with one element for each relevant XML node. Each element contains a matrix.

---

docsToNodes	<i>Parse XML Documents into XML Nodes</i>
-------------	---

---

**Description**

Essentially a recursive call to [getNodeSet](#).

**Usage**

```
docsToNodes(docs, xpath)
```

**Arguments**

docs	XML documents
xpath	xpath expression

---

listsToObs	<i>Flatten nested list into a list of observations</i>
------------	--

---

**Description**

This function flattens the nested list into a list of "observations" (that is, a list of matrices with one row). The names of the list that is returned reflects the XML ancestry of each observation.

**Usage**

```
listsToObs(l, urls, append.value = TRUE, as.equiv = TRUE, url.map = TRUE)
```

**Arguments**

l	list. Should be the output from <a href="#">nodesToList</a> .
urls	character vector the same length as l. Each element should map element of l to an XML file.
append.value	logical. Should the XML value be appended to the observation?
as.equiv	logical. Should observations from two different files (but the same ancestry) have the same name returned?
url.map	logical. If TRUE, the 'url_key' column will contain a condensed url identifier (for each observation) and full urls will be stored in the "url_map" element. If FALSE, the full urls are included (for each observation) as a 'url' column and no "url_map" is included.

**Value**

A list where each element reflects one "observation".

---

nodesToList	<i>Coerce XML Nodes into a list with both attributes and values</i>
-------------	---

---

**Description**

Essentially a recursive call to [xmlToList](#).

**Usage**

```
nodesToList(nodes)
```

**Arguments**

nodes	A collection of XML nodes. Should be the output from <a href="#">docsToNodes</a> .
-------	--

**Value**

A nested list with a structure that resembles the XML structure

---

re_name	<i>Rename rows of a list</i>
---------	------------------------------

---

**Description**

Sometimes, certain nodes in an XML ancestry may want to be neglected before any keys are created (see [add\\_key](#)) or observations are aggregated (see [collapse](#)). This function takes a list of "observations" (that is, a list of matrices with one row) and alters the names of that list. Note that any information lost from changing names is saved in a new column whose name is specified by `diff.name`.

**Usage**

```
re_name(obs, namez, equiv, diff.name = "diff_name", rename.as,
        quiet = FALSE)
```

**Arguments**

obs	list. Should be the output from <a href="#">XML2Obs</a> (or <a href="#">listsToObs</a> ).
namez	must be equivalent to <code>names(obs)</code> . Intended use is to avoid unnecessarily repeating that operation.
equiv	character vector with the appropriate (unique) names that should be regarded "equivalent".
diff.name	character string used for naming the variable that is appended to any observations whose name was overwritten. The value for this variable is the difference in from the original name and the overwritten name.
rename.as	character string to override naming of observations that are renamed.
quiet	logical. Include message about how observations are being renamed?

**Value**

A list of "observations".

---

 urlsToDocs

---

*Parse XML Files into XML Documents*


---

**Description**

Essentially a recursive call to [xmlParse](#).

**Usage**

```
urlsToDocs(urls, async = TRUE, quiet = FALSE)
```

**Arguments**

urls	character vector or list of urls that point to an XML file (or anything readable by <a href="#">xmlParse</a> ).
async	logical. Allows for asynchronous download requests. This option is passed to the async option in the <code>RCurl::getURL</code> function.
quiet	logical. Print file name currently being parsed?

---

 XML2Obs

---

*Parse XML files into a list of "observations"*


---

**Description**

This function takes a collection of urls that point to XML files and coerces the relevant info into a list of observations. An "observation" is defined as a matrix with one row. An observation can also be thought of as a single instance of XML attributes (and value) for a particular level in the XML hierarchy. The names of the list reflect the XML node ancestry for which each observation was extracted from.

**Usage**

```
XML2Obs(urls, xpath, append.value = TRUE, as.equiv = TRUE,  
url.map = FALSE, async = FALSE, quiet = FALSE)
```

**Arguments**

<code>urls</code>	character vector or list of urls that point to an XML file (or anything readable by <a href="#">xmlParse</a> ).
<code>xpath</code>	XML XPath expression that is passed to <a href="#">getNodeSet</a> . If missing, the entire root and all descendents are captured and returned (ie, <code>tables = "/"</code> ).
<code>append.value</code>	logical. Should the XML value be appended for relevant observations?
<code>as.equiv</code>	logical. Should observations from two different files (but the same ancestry) have the same name returned?
<code>url.map</code>	logical. If TRUE, the 'url_key' column will contain a condensed url identifier (for each observation) and full urls will be stored in the "url_map" element. If FALSE, the full urls are included (for each observation) as a 'url' column and no "url_map" is included.
<code>async</code>	logical. Allows for asynchronous download requests. This option is passed to the <code>async</code> option in the <code>RCurl::getURL</code> function.
<code>quiet</code>	logical. Print file name currently being parsed?

**Details**

It's worth noting that a "url\_key" column is appended to each observation to help track the origin of each observation. The value of the "url\_key" column is not the actual file name, but a simplified identifier to avoid unnecessarily repeating long file names for each observation. For this reason, an addition element (named "url\_map") is added to the list of observations in case the actual file named want to be used.

**Value**

A list of "observations" and (possibly) the "url\_map" element.

**See Also**

[urlsToDocs](#), [docsToNodes](#), [nodesToList](#), [listsToObs](#)

**Examples**

```
## Not run:
urls <- c("http://gd2.mlb.com/components/game/mlb/year_2013/mobile/346180.xml",
         "http://gd2.mlb.com/components/game/mlb/year_2013/mobile/346188.xml")
obs <- XML2Obs(urls)
table(names(obs))

## End(Not run)
```

---

`XML2R`*Parse XML files into (a list of) matrices or data frame(s)*

---

**Description**

This function is an experimental wrapper around [XML2Obs](#). One should only use this function over [XML2Obs](#) if keys already exist in the XML data and ancestry doesn't need to be altered.

**Usage**

```
XML2R(urls, xpath, df = FALSE)
```

**Arguments**

<code>urls</code>	character vector or list of urls that point to an XML file (or anything readable by <a href="#">xmlParse</a> ).
<code>xpath</code>	XML XPath expression that is passed to <a href="#">getNodeSet</a> . If missing, the entire root and all descendents are captured and returned (ie, <code>tables = "/"</code> ).
<code>df</code>	logical. Should matrices be coerced into data frames?

**Value**

Returns list with one element for each relevant XML node. Each element contains a matrix by default.

**See Also**

[urlsToDocs](#), [docsToNodes](#), [nodesToList](#), [listsToObs](#)

**Examples**

```
## Not run:
urls2 <- c("http://gd2.mlb.com/components/game/mlb/year_2013/mobile/346180.xml",
          "http://gd2.mlb.com/components/game/mlb/year_2013/mobile/346188.xml")
dat3 <- XML2R(urls2)

cens <- "http://www.census.gov/developers/data/sf1.xml"
census <- XML2R(cens)

## End(Not run)
```

# Index

`add_key`, [2](#), [4](#)

`collapse`, [4](#)

`collapse_obs`, [2](#)

`docsToNodes`, [3](#), [4](#), [6](#), [7](#)

`getNodeSet`, [3](#), [6](#), [7](#)

`listsToObs`, [2](#), [3](#), [4](#), [6](#), [7](#)

`nodesToList`, [3](#), [4](#), [6](#), [7](#)

`re_name`, [4](#)

`urlsToDocs`, [5](#), [6](#), [7](#)

`XML2Obs`, [4](#), [5](#), [7](#)

`XML2R`, [7](#)

`xmlParse`, [5–7](#)

`xmlToList`, [4](#)