

# Rredland

April 19, 2009

---

`cleanXSDT`      *remove XSD data type information from strings*

---

## Description

remove XSD data type information from strings

## Usage

```
cleanXSDT(x)
```

## Arguments

`x`                  a string

## Details

uses `gsub` to remove XSD data type information

## Author(s)

Vince Carey <stvjc@channing.harvard.edu>

---

`getOWLProperties`      *various functions to extract owl/RDF model elements*

---

## Description

various functions to extract owl/RDF model elements

## Usage

```
getOWLProperties(redlMod, ns = "http://www.w3.org/2002/07/owl", strip = TRUE)
```

**Arguments**

<code>redlMod</code>	instance of <code>redlModel</code> class
<code>ns</code>	ontology namespace
<code>strip</code>	logical: remove namespace prefixes?

**Author(s)**

VJ Carey <stvjc@channing.harvard.edu>

**References**

[librdf.org](http://librdf.org)

**See Also**

~~objects to See Also as [help](#), ~~~

**Examples**

```
example(makeRedlURI)
ff = readRDF(uu)
getOWLProperties(ff)
```

---

gordb

*demonstration triple tables for RDF*

---

**Description**

demonstration triple tables for RDF

**Usage**

```
data(gordb)
```

**Details**

deserializes an RDF representation of part of GO or of intact

**Value**

data.frames

**Author(s)**

Vince Carey <stvjc@channing.harvard.edu>

**Examples**

```
data(gordb)
gordb[1:3, ]
```

---

makeRedlURI	<i>use the redland librdf system to define a URI object for further processing</i>
-------------	--

---

**Description**

use the redland librdf system to define a URI object for further processing

**Usage**

```
makeRedlURI(uri, w)
```

**Arguments**

uri	a character string defining a URI
w	a <code>redlWorld</code> instance

**Details**

executes C code to create a Redland URI object and returns an instance of the `redlURI` S4 class defined in the `Rredland` package, which includes an external pointer to the Redland object

**Value**

an instance of `redlURI`

**Author(s)**

Vince Carey <stvjc@channing.harvard.edu>

**Examples**

```
fi <- system.file("RDF/gopart.rdf", package="Rredland")
uu <- makeRedlURI(paste("file:", fi, sep=""))
uu
```

---

readRDF	<i>read an RDF document identified by URI (or deserialize a redland Berkeley DB representation) into a librdf model object</i>
---------	--

---

**Description**

read an RDF document identified by URI (or deserialize a redland Berkeley DB representation) into a librdf model object

**Usage**

```
readRDF(uri, storageType=c("internal", "bdb")[1], storageName="test", world=..Gr
restoreBDB(storageName, world=..GredlWorld, stoHash="hash-type='bdb',dir='.')
```

**Arguments**

<code>uri</code>	a <code>redlURI</code> instance, or a string encoding a URI
<code>storageType</code>	character string, with value "internal" or "bdb"
<code>storageName</code>	basename of file to store the hashes, if <code>storageType</code> is "bdb"
<code>world</code>	<code>librdf world</code> ( <code>redlWorld</code> class instance)
<code>stoHash</code>	a <code>librdf hash</code> specification of parameters to the new storage request; non-default values for advanced users only.

**Author(s)**

Vince Carey <stvjc@channing.harvard.edu>

**Examples**

```
# use character string URI
ii = readRDF(paste("file:", system.file("RDF/gopart.rdf", package="Rredland"),
                sep=""))
ii
freeRedl(ii)
    # make a URI for a fragment from GO distributed with the package
example(makeRedlURI)
    # read from it with defaults
mm = readRDF(uu)
mm
    # excerpt after transformation to data.frame
as(mm, "data.frame")[1:3,]
    # now we will do some disk operations with BDB
curd = getwd()
tt = tempdir()
    # change dir
setwd(tt)
    # read contents of previous URI, but use external storage
hh = readRDF(uu, storageType="bdb", storageName="gopart")
    # see the created files; note that they are not
    # populated until the storage/model is freed
dir()
    # free the model, so the BDB hashes are populated
freeRedl(hh)
hh
    # now restore the hashes and create a redlModel
ff = restoreBDB("gopart")
ff
    # cleanup
unlink("gopart-so2p.db")
unlink("gopart-po2s.db")
unlink("gopart-sp2o.db")
setwd(curd)
cat(paste("to clean up completely, execute unlink(\"",
        tt, "\", recursive=TRUE) in R, if it looks safe to do so.\n", sep=""))
```

---

redlModel-class      *Class "redlModel" represents librdf model in Redland RDF library*

---

### Description

represents librdf model in Redland RDF library

### Objects from the Class

Objects can be created by calls of the form `new("redlModel", ...)`. They encapsulate the reference to the librdf model object in Redland RDF library.

### Slots

**ref:** Object of class "externalptr" pointer to malloc'd model space.  
**storageType:** Object of class "character" can be "bdb" or "internal"  
**stateEnv:** Object of class "environment" used to indicate whether model is open or not  
**world:** Object of class "redlWorld", the world object in which the model or URI was constructed  
**URIstring:** (for redlURI: string in use as URI

### Methods

**coerce** signature(from = "redlModel", to = "data.frame"): simple transformation to 3-column dataframe (subject, predicate, object)  
**coerce** signature(from = "redlModel", to = "graphNEL"): ...  
**freeRedl** signature(x = "redlModel"): call the librdf close method; kills the model.  
**getStatus** signature(x = "redlModel"): determine if a model object is open from the perspective of R  
**ref** signature(x = "redlModel"): extract the externalptr  
**setStatus** signature(x = "redlModel"): set a status flag in the R container for the model object  
**show** signature(object = "redlModel"): simple report  
**size** signature(x = "redlModel"): tell the number of statements in the model

### Author(s)

VJ Carey <stvjc@channing.harvard.edu>

### Examples

```
example(makeRedlURI)
x = readRDF( uu )
ref(x)
size(x)
getStatus(x)
freeRedl(x)
getStatus(x)
x
```

---

redlWorld-class      *Class "redlWorld" for representing RDF worlds using Redland librdf*

---

### Description

represents RDF worlds using Redland librdf

### Objects from the Class

Objects can be created by calls of the form `new("redlWorld", ...)`. In general only one open world should exist in any session.

### Slots

**ref:** Object of class "externalptr", pointer to malloc'd memory for the librdf\_world instance

**stateEnv:** Object of class "environment", holds information on status of world instance

### Methods

**freeRedl** signature(x = "redlWorld"): execute free and close methods of librdf

**getStatus** signature(x = "redlWorld"): obtain the status string from the R instance

**makeRedlURI** signature(uri = "character", w = "redlWorld"): create a URI reference in the current world

**setStatus** signature(x = "redlWorld"): set the status field with a string. Use any value other than 'open' to close the world from the perspective of R.

**show** signature(object = "redlWorld"): print simple report.

### Author(s)

VJ Carey <stvjc@channing.harvard.edu>

### References

[librdf.org](http://librdf.org)

### Examples

```
nw = openRedlWorld(.force=TRUE)
nw
```

# Index

## \*Topic classes

redlModel-class, 5  
redlWorld-class, 6

## \*Topic models

cleanXSDT, 1  
getOWLProperties, 1  
gordb, 2  
makeRedlURI, 3  
readRDF, 3

cleanXSDT, 1  
coerce, redlModel, data.frame-method  
(redlModel-class), 5  
coerce, redlModel, graphNEL-method  
(redlModel-class), 5

EMAPdf (gordb), 2

freeRedl (redlModel-class), 5  
freeRedl, redlModel-method  
(redlModel-class), 5  
freeRedl, redlWorld-method  
(redlWorld-class), 6

getArcsWith (getOWLProperties), 1  
getClassElements  
(getOWLProperties), 1  
getClassGraph (getOWLProperties),  
1  
getDatatypeProperties  
(getOWLProperties), 1  
getObjectProperties  
(getOWLProperties), 1  
getOWLClasses (getOWLProperties),  
1  
getOWLProperties, 1  
getOWLSubclasses  
(getOWLProperties), 1  
getPropertiesWithDomain  
(getOWLProperties), 1  
getPropertyRange  
(getOWLProperties), 1  
getStatus (redlModel-class), 5  
getStatus, redlModel-method  
(redlModel-class), 5

getStatus, redlWorld-method  
(redlWorld-class), 6

gordb, 2

help, 2

makeRedlURI, 3  
makeRedlURI, character, missing-method  
(redlModel-class), 5  
makeRedlURI, character, redlWorld-method  
(redlWorld-class), 6

nodeFromURIString  
(redlModel-class), 5

openRedlWorld (redlWorld-class), 6

readRDF, 3  
redlModel-class, 5  
redlNode-class (redlModel-class),  
5  
redlURI-class (redlModel-class), 5  
redlWorld, 3  
redlWorld (redlWorld-class), 6  
redlWorld-class, 6  
ref (redlModel-class), 5  
ref, redlModel-method  
(redlModel-class), 5  
ref, redlNode-method  
(redlModel-class), 5  
ref, redlURI-method  
(redlModel-class), 5  
ref, redlWorld-method  
(redlWorld-class), 6  
restoreBDB (readRDF), 3

setStatus (redlModel-class), 5  
setStatus, redlModel-method  
(redlModel-class), 5  
setStatus, redlWorld-method  
(redlWorld-class), 6  
show, redlModel-method  
(redlModel-class), 5  
show, redlNode-method  
(redlModel-class), 5

show, redlURI-method  
    (*redlModel-class*), 5  
show, redlWorld-method  
    (*redlWorld-class*), 6  
size (*redlModel-class*), 5  
size, redlModel-method  
    (*redlModel-class*), 5  
  
world (*redlModel-class*), 5  
world, redlNode-method  
    (*redlModel-class*), 5