Package ‘redland’

February 24, 2024

Version 1.0.17-18

Title RDF Library Bindings in R

Date 2024-02-23

VignetteBuilder knitr

Description Provides methods to parse, query and serialize information stored in the Resource Description Framework (RDF). RDF is described at <https://www.w3.org/TR/rdf-primer/>.

This package supports RDF by implementing an R interface to the Redland RDF C library, described at <https://librdf.org/docs/api/index.html>. In brief, RDF provides a structured graph consisting of Statements composed of Subject, Predicate, and Object Nodes.

Depends R (>= 3.1.1), methods

Imports roxygen2

Suggests spelling, knitr, testthat, rmarkdown, stringi

SystemRequirements Mac OSX: redland (>= 1.0.14) ; Linux: librdf0 (>= 1.0.14), librdf0-dev (>= 1.0.14)

Collate 'redland.R' 'World.R' 'Node.R' 'Statement.R' 'Storage.R'
'Model.R' 'Parser.R' 'Query.R' 'QueryResults.R' 'Serializer.R'
'mergeNamespace_roclet.R' 'redland-package.R' 'util.R'

License Apache License 2.0

Copyright See file (inst/)COPYRIGHTS.

BugReports https://github.com/ropensci/redland-bindings/issues

RoxygenNote 7.2.3

https://github.com/ropensci/redland-bindings/tree/master/R

Encoding UTF-8

Language en-US

NeedsCompilation yes
Author Matthew B. Jones [aut, cre] (<https://orcid.org/0000-0003-0077-4738>),
Peter Slaughter [aut] (<https://orcid.org/0000-0002-2192-403X>),
Jeroen Ooms [aut] (<https://orcid.org/0000-0002-4035-0289>),
Carl Boettiger [aut] (<https://orcid.org/0000-0002-1642-628X>),
Scott Chamberlain [ctb] (<https://orcid.org/0000-0003-1444-9135>),
David Beckett [cph],
University of Bristol [cph],
Regents of the University of California [cph]
Maintainer Matthew B. Jones <jones@nceas.ucsb.edu>
Repository CRAN
Date/Publication 2024-02-24 01:10:02 UTC

R topics documented:

addStatement .................................................. 7
executeQuery .................................................... 8
freeModel ......................................................... 8
freeParser ......................................................... 9
freeQuery ......................................................... 10
freeQueryResults ................................................ 11
freeSerializer ................................................... 11
freeStatement ................................................... 12
freeStorage ...................................................... 13
freeWorld ......................................................... 14
getBlankNodeId .................................................. 14
getNodeType ...................................................... 15
getNodeValue .................................................... 16
getQueryResultLimit ............................................. 16
getResults ....................................................... 17
getTermType ...................................................... 18
initialize,Model-method ....................................... 19
initialize,Node-method ......................................... 19
initialize,Parser-method ....................................... 20
initialize,Query-method ....................................... 21
initialize,QueryResults-method ............................... 22
initialize,Serializer-method ................................... 22
initialize,Statement-method ................................... 23
initialize,Storage-method ...................................... 24
initialize,World-method ....................................... 25
is.null.externalptr .............................................. 25
length,SWIGArray-method ...................................... 26
librdf_copyright_string ......................................... 26
librdf_copyright_string_get .................................... 27
librdf_digest_final ............................................. 27
librdf_digest_init .............................................. 28
librdf_digest_to_string ....................................... 29
librdf_digest_update .......................................... 29
librdf_digest_update_string ........................................... 30
librdf_free_digest ..................................................... 31
librdf_free_hash ....................................................... 31
librdf_free_iterator ................................................... 32
librdf_free_model ....................................................... 33
librdf_free_node ....................................................... 33
librdf_free_parser ..................................................... 34
librdf_free_query ...................................................... 35
librdf_free_query_results ............................................. 35
librdf_free_serializer ............................................... 36
librdf_free_statement .................................................. 37
librdf_free_storage .................................................... 37
librdf_free_stream ..................................................... 38
librdf_free_uri ........................................................ 39
librdf_free_world ....................................................... 39
librdf_hash_to_string ................................................. 40
librdf_internal_test_error ........................................... 41
librdf_internal_test_warning ....................................... 41
librdf_iterator_end .................................................... 42
librdf_iterator_get_context ........................................... 43
librdf_iterator_get_object .......................................... 43
librdf_iterator_next ................................................... 44
librdf_log_message_code ............................................. 45
librdf_log_message_facility .......................................... 45
librdf_log_message_level ............................................ 46
librdf_log_message_locator ......................................... 47
librdf_log_message_message ......................................... 47
librdf_model_add ....................................................... 48
librdf_model_add_statement ....................................... 49
librdf_model_add_statements ....................................... 50
librdf_model_add_string_literal_statement ................. 50
librdf_model_add_typed_literal_statement ...................... 51
librdf_model_as_stream ............................................. 52
librdf_model_contains_context .................................... 53
librdf_model_contains_statement ................................ 54
librdf_model_context_add_statement ......................... 55
librdf_model_context_add_statements ....................... 56
librdf_model_context_as_stream .................................. 57
librdf_model_context_remove_statement ..................... 57
librdf_model_context_remove_statements .................... 58
librdf_model_find_statements .................................... 59
librdf_model_find_statements_in_context .................... 60
librdf_model_get_arc ............................................... 60
librdf_model_get_arcs .............................................. 61
librdf_model_get_arcs_in .......................................... 62
librdf_model_get_arcs_out ...................................... 63
librdf_model_get_contexts ........................................ 63
librdf_model_get_feature ......................................... 64
librdf_model_get_source .................................................. 65
librdf_model_get_sources .................................................. 65
librdf_model_get_target .................................................... 66
librdf_model_get_targets ................................................... 67
librdf_model_has_arc_in .................................................... 68
librdf_model_has_arc_out .................................................. 69
librdf_model_load .......................................................... 70
librdf_model_query_execute ................................................ 71
librdf_model_remove_statement ......................................... 71
librdf_model_set_feature .................................................. 72
librdf_model_size .......................................................... 73
librdf_model_sync .......................................................... 74
librdf_model_to_string ..................................................... 74
librdf_model_transaction_commit ....................................... 75
librdf_model_transaction_rollback ..................................... 76
librdf_model_transaction_start ......................................... 77
librdf_new_digest .......................................................... 77
librdf_new_hash ............................................................. 78
librdf_new_hash_from_array_of_strings ................................ 79
librdf_new_hash_from_string ............................................ 79
librdf_new_model ........................................................... 80
librdf_new_model_from_model ............................................ 81
librdf_new_model_with_options ......................................... 82
librdf_new_node ............................................................. 82
librdf_new_node_from_blank_identifier ................................ 83
librdf_new_node_from_literal ............................................. 84
librdf_new_node_from_normalised_uri_string ........................ 85
librdf_new_node_from_typed_literal .................................... 86
librdf_new_node_from_uri .................................................. 87
librdf_new_node_from_uri_local_name ................................ 88
librdf_new_node_from_uri_string ....................................... 88
librdf_new_parser .......................................................... 89
librdf_new_query ........................................................... 90
librdf_new_query_from_query ............................................. 91
librdf_new_serializer ...................................................... 91
librdf_new_statement ....................................................... 92
librdf_new_statement_from_nodes ....................................... 93
librdf_new_statement_from_statement ................................. 94
librdf_new_storage ........................................................ 94
librdf_new_storage_from_storage ...................................... 95
librdf_new_uri .............................................................. 96
librdf_new_uri_from_filename .......................................... 97
librdf_new_uri_from_uri ................................................... 97
librdf_new_world .......................................................... 98
librdf_node_equals ........................................................ 99
librdf_node_get_blank_identifier ..................................... 99
librdf_node_get_literal_value .......................................... 100
librdf_node_get_literal_value_as_latin1 .................................................. 101  
librdf_node_get_literal_value_datatype_uri ............................................. 101  
librdf_node_get_literal_value_is_wf_xml .............................................. 102  
librdf_node_get_literal_value_language ............................................... 103  
librdf_node_get_li_ordinal ....................................................................... 103  
librdf_node_get_type .............................................................................. 104  
librdf_node_get_uri .................................................................................. 105  
librdf_node_is_blank ................................................................................. 105  
librdf_node_is_literal ................................................................................ 106  
librdf_node_is_resource ............................................................................ 107  
librdf_parser_check_name .......................................................................... 107  
librdf_parser_get_accept_header .............................................................. 108  
librdf_parser_get_feature ......................................................................... 109  
librdf_parser_get_namespaces_seen_count .............................................. 109  
librdf_parser_get_namespaces_seen_prefix ............................................. 110  
librdf_parser_get_namespaces_seen_uri .................................................... 111  
librdf_parser_guess_name2 ....................................................................... 111  
librdf_parser_parse_as_stream .................................................................. 112  
librdf_parser_parse_counted_string_as_stream ......................................... 113  
librdf_parser_parse_counted_string_into_model ....................................... 114  
librdf_parser_parse_into_model ................................................................ 115  
librdfParser_parse_string_as_stream .......................................................... 116  
librdf_parser_parse_string_into_model ...................................................... 116  
librdf_parser_set_feature ......................................................................... 117  
librdf_query_execute .................................................................................. 118  
librdf_query_get_limit .............................................................................. 119  
librdf_query_get_offset ............................................................................. 120  
librdf_query_results_as_stream .................................................................. 120  
librdf_query_results_finished .................................................................. 121  
librdf_query_results_get_bindings_count ............................................... 122  
librdf_query_results_get_binding_name .................................................... 122  
librdf_query_results_get_binding_value ................................................. 123  
librdf_query_results_get_binding_value_by_name .................................... 124  
librdf_query_results_get_boolean .............................................................. 124  
librdf_query_results_get_count .................................................................. 125  
librdf_query_results_is_bindings ............................................................... 126  
librdf_query_results_is_boolean .................................................................. 126  
librdf_query_results_is_graph .................................................................... 127  
librdf_query_results_is_syntax ................................................................... 128  
librdf_query_results_next .......................................................................... 128  
librdf_query_results_to_file2 ................................................................... 129  
librdf_query_results_to_string2 ............................................................... 130  
librdf_query_set_limit ................................................................................ 131  
librdf_query_set_offset ............................................................................. 132  
librdf_serializer_check_name .................................................................... 132  
librdf_serializer_get_feature .................................................................... 133  
librdf_serializer_serialize_model_to_file .............................................. 134  
librdf_serializer_serialize_model_to_string ............................................ 135
topics documented:

librdf_serializer_serialize_stream_to_file ................................ 135
librdf_serializer_serialize_stream_to_string ............................. 136
librdf_serializer_set_feature ............................................. 137
librdf_serializer_set_namespace ........................................... 138
librdf_short_copyright_string ............................................. 139
librdf_short_copyright_string_get ....................................... 139
librdf_statement_equals .................................................. 140
librdf_statement_get_object .............................................. 141
librdf_statement_get_predicate .......................................... 141
librdf_statement_get_subject ............................................. 142
librdf_statement_is_complete ............................................. 143
librdf_statement_match ................................................... 143
librdf_statement_set_object .............................................. 144
librdf_statement_set_predicate .......................................... 145
librdf_statement_set_subject ............................................. 146
librdf_stream_end .......................................................... 146
librdf_stream_get_object .................................................. 147
librdf_stream_next ......................................................... 148
librdf_uri_compare ........................................................ 148
librdf_uri_equals .......................................................... 149
librdf_uri_to_string ....................................................... 150
librdf_version_decimal ..................................................... 150
librdf_version_decimal_get ................................................ 151
librdf_version_major ....................................................... 152
librdf_version_major_get .................................................. 152
librdf_version_minor ....................................................... 153
librdf_version_minor_get .................................................. 154
librdf_version_release ..................................................... 154
librdf_version_release_get ................................................ 155
librdf_version_string ..................................................... 156
librdf_version_string_get ................................................ 156
librdf_world_get_feature ................................................ 137
librdf_world_open ........................................................ 158
librdf_world_set_feature ................................................ 158
librdf_world_set_logger .................................................. 159
mergeNamespace_roclet ...................................................... 160
Model-class ................................................................. 161
Node-class ................................................................. 161
parseFileIntoModel ......................................................... 162
Parser-class ................................................................. 163
Query-class ................................................................. 164
QueryResults-class ......................................................... 165
raptor_locator_byte ........................................................ 165
raptor_locator_column ...................................................... 166
raptor_locator_file ........................................................ 167
raptor_locator_line ........................................................ 167
raptor_locator_uri .......................................................... 168
raptor_version_decimal ..................................................... 169
addStatement

Add a Statement object to the Model

Description

Add a Statement object to the Model

Usage

addStatement(.Object, statement)

## S4 method for signature 'Model,Statement'
addStatement(.Object, statement)
Arguments

,Object a Model object
statement the Statement that will be added

Examples

world <- new("World")
storage <- new("Storage", world, "hashes", name="", options="hash-type='memory'")
model <- new("Model", world, storage, options="")

executeQuery

Description

The initialize query is executed and the result is returned as a QueryResult object

Usage

executeQuery(.Object, model)

## S4 method for signature 'Query'
executeQuery(.Object, model)

Arguments

,Object a Query object
model a Model object containing the statements to query

Value

a QueryResults object

freeModel

Free memory used by a librdf model.

Description

Free memory used by a librdf model.

Usage

freeModel(.Object)

## S4 method for signature 'Model'
freeModel(.Object)
Arguments

/Object a Model object

Details

After this method is called, the Model object is no longer usable and should be deleted "rm(model)" and a new object created.

Examples

world <- new("World")
storage <- new("Storage", world, "hashes", name="", options="hash-type='memory'")
model <- new("Model", world, storage, options="")
# At this point, some operations would be performed with the model.
# See '?redland' for a complete example.
# When the Model object is no longer needed, the resources it has allocated can be freed.
freeModel(model)
rm(model)

freeParser  Free memory used by a librdf parser

Description

Free memory used by a librdf parser

Usage

freeParser(.Object)

## S4 method for signature 'Parser'
freeParser(.Object)

Arguments

/Object a Node object

Details

After freeNode is called, the Node object is no longer usable and should be deleted "rm(nodeName)" and a new object created.
Examples

```r
world <- new("World")
storage <- new("Storage", world, "hashes", name="", options="hash-type='memory'")
model <- new("Model", world, storage, options="")
parser <- new("Parser", world)
filePath <- system.file("extdata/example.rdf", package="redland")
parseFileIntoModel(parser, world, filePath, model)
# At this point, some operations would be performed with the Model that has been populated
# with the parser.
# See '?redland' for a complete example.
# When the parser object is no longer needed, the resources it had allocated can be freed.
freeParser(parser)
rm(parser)
```

---

### freeQuery

*Free memory used by a librdf query*

**Description**

Free memory used by a librdf query

**Usage**

```r
freeQuery(.Object)
```

```r
## S4 method for signature 'Query'
freeQuery(.Object)
```

**Arguments**

- `.Object` a `Query` object

**Details**

After this method is called, the `Query` object is no longer usable and should be deleted `"rm(query)"` and a new object created.

**Examples**

```r
world <- new("World")
storage <- new("Storage", world, "hashes", name="", options="hash-type='memory'")
model <- new("Model", world, storage, options="")
stmt <- new("Statement", world=world,
  subject="https://orcid.org/0000-0002-2192-403X",
  predicate="http://www.w3.org/ns/prov#Agent",
  object="slaughter",
  objectType="literal", datatype_uri="http://www.w3.org/2001/XMLSchema#string")
status <- addStatement(model, stmt)
queryString <- paste("PREFIX orcid: <https://orcid.org/>,
```

```r
```

```r
```
"PREFIX dataone: <https://cn.dataone.org/cn/v1/resolve/>",
"PREFIX prov: <http://www.w3.org/ns/prov#>",
"SELECT ?a ?c WHERE { ?a prov:Agent ?c . }", sep=" ")

query <- new("Query", world, queryString, base_uri=NULL,
query_language="sparql", query_uri=NULL)
# Return all results as a string
results <- getResults(query, model, "rdfxml")

# When the query object is no longer needed, the resources it had allocated can be freed.
freeQuery(query)
rm(query)

---

**freeQueryResults**

*Free memory used by a librdf query results*

**Description**

After this method is called, the QueryResults object is no longer usable and should be deleted with "rm(query)".

**Usage**

freeQueryResults(.Object)

### S4 method for signature 'QueryResults'

freeQueryResults(.Object)

**Arguments**

/Object/ a QueryResults object

---

**freeSerializer**

*Free memory used by a librdf serializer.*

**Description**

Free memory used by a librdf serializer.

**Usage**

freeSerializer(.Object)

### S4 method for signature 'Serializer'

freeSerializer(.Object)
Arguments

,Object a Serializer object

Examples

world <- new("World")
storage <- new("Storage", world, "hashes", name="", options="hash-type='memory'")
model <- new("Model", world, storage, options="")
filePath <- system.file("extdata/example.rdf", package="redland")
parser <- new("Parser", world)
parseFileIntoModel(parser, world, filePath, model)
# Creat the default "rdfxml" serizlizer
serializer <- new("Serializer", world)
# At this point, some operations would be performed with the Serializer object.
# See '?Serializer' for a complete example.
# When the serializer object is no longer needed, the resources it had allocated can be freed.
freeSerializer(serializer)
rm(serializer)

freeStatement Free memory used by a librdf statement

Description

Free memory used by a librdf statement

Usage

freeStatement(.Object)

## S4 method for signature 'Statement'
freeStatement(.Object)

Arguments

.Object a Statement object

Details

After this method is called, the Statement object is no longer usable and should be deleted "rm(statement)" and a new object created. This method frees all resources for the statement, as well as each node in the statement.
freeStorage

Examples

world <- new("World")
stmt <- new("Statement", world, subject="http://www.example.com/myevent",
          predicate="http://example.com/occurredAt",
          object="Tue Feb 17 14:05:13 PST 2015")

# At this point, some operations would be performed with the Statement.
# See '?redland' for a complete example.
# When the Statement object is no longer needed, the resources it had allocated can be freed.
freeStatement(stmt)
rm(stmt)

freeStorage <- function(.Object) {
  # Free memory used by a librdf storage object
  # After this method is called, the Storage object is no longer usable and should be deleted
  # rm(storage)
  # and a new object created.

  # Usage
  # freeStorage(.Object)

  # S4 method for signature 'Storage'
  .Object
}

Arguments

.Object a Storage object to free memory for

Examples

world <- new("World")
storage <- new("Storage", world, "hashes", name="", options="hash-type='memory'")

# At this point we would perform some operations using the storage object.
# See '?redland' for a complete example.
# When the Storage object is no longer needed, the resources it had allocated can be freed.
status <- freeStorage(storage)
rm(storage)
freeWorld

Free memory used by a librdf world object

**Description**

Free memory used by a librdf world object

**Usage**

```r
freeWorld(.Object)
```

```r
## S4 method for signature 'World'
freeWorld(.Object)
```

**Arguments**

- `.Object` a World object

**Details**

After this method is called, the World object is no longer usable and should be deleted "rm(world)" and a new object created.

**Examples**

```r
world <- new("World")
# At this point we would perform some operations using the world object.
# When the world object is no longer needed, we can free the resources it has allocated.
result <- freeWorld(world)
rm(world)
```

---

getBlankNodeId

Get the blank identifier that has been assigned for a specified Node object

**Description**

Get the blank identifier that has been assigned for a specified Node object

**Usage**

```r
getBlankNodeId(.Object)
```

```r
## S4 method for signature 'Node'
getBlankNodeId(.Object)
```
getNodeType

Arguments

/Object/ a Node object

Details

When a Node object is initialized with no value specified, i.e. `node <- Node()`, a blank node is created and a locally unique identifier is generated by librdf. This method retrieves this identifier and returns it to the caller.

Value

a blank node identifier

Examples

```r
world <- new("World")
# a blank node is created with a unique identifier generated by librdf
node <- new("Node", world, blank=NULL)
nodeId <- getBlankNodeId(node)
```

getNodeType

Determine the node type and return as a string

Description

A Node has a type that is assigned at initialization and can have one of the following values: 'resource', 'literal', 'blank' and 'unknown'.

Usage

```r
getNodeType(.Object)
```

Arguments

/Object/ a Node object

Value

a character vector containing the Node type

Examples

```r
world <- new("World")
node <- new("Node", world, uri="http://www.example.com")
nodetype <- getNodeType(node)
```
getNodeValue

Get the value of the node as a string

Description
Get the value of the node as a string

Usage
getNodeValue(.Object)

## S4 method for signature 'Node'
getNodeValue(.Object)

Arguments
.OBJECT a Node object

Details
The value of the node is returned as a string. If the node type is 'blank', then the blank node identifier is returned. If the node type is 'literal', then the literal value is returned with the form "string@language, e.g. "¡Hola, amigo! ¿Cómo estás?"@es". If the node type is 'uri' then the value is returned as a string.

Value
a string representation of the Node's value

Examples
world <- new("World")
node <- new("Node", world, literal="¡Hola, amigo! ¿Cómo estás?", language="es")
value <- getNodeValue(node)

getQueryResultLimit

Get the query result limit

Description
Get the query result limit

Usage
getQueryResultLimit(.Object)

## S4 method for signature 'Query'
getQueryResultLimit(.Object)
Arguments

.\Object \n.a Query object

Value

the query result limit. If a limit is set then the value will be $\geq 0$. If the value is $< 0$, no limit is set.

### S4 method for signature 'Query'
```r
getResults(.Object, model, formatName = "rdfxml")
```

Arguments

.\Object \n.a Query object

model \n.a Model object

... \nadditional parameters

formatName \na string specifying the RDF format name. Currently the supported formats are "rdfxml", "turtle", "json", "csv"

Details

After this method is called, the Query object is no longer usable and should be deleted "rm(query)"

Examples

```r
world <- new("World")
storage <- new("Storage", world, "hashes", name="", options="hash-type='memory'")
model <- new("Model", world, storage, options="")
stmt <- new("Statement", world=world,
subject="https://orcid.org/0000-0002-2192-403X",
predicate="http://www.w3.org/ns/prov#Agent",
object="slaughter",
objectType="literal", datatype_uri="http://www.w3.org/2001/XMLSchema#string")
#objectType="literal", language="en")
status <- addStatement(model, stmt)
queryString <- paste("PREFIX orcid: <https://orcid.org/>,
```
"PREFIX dataone: <https://cn.dataone.org/cn/v1/resolve/>",
"PREFIX prov: <http://www.w3.org/ns/prov#>",
"SELECT ?a ?c WHERE { ?a prov:Agent ?c . }", sep=" ")
query <- new("Query", world, queryString, base_uri=NULL, query_language="sparql", query_uri=NULL)
# Return all results as a string
results <- getResults(query, model, "rdfxml")
results <- getResults(query, model, "turtle")
results <- getResults(query, model, "json")

# When the query object is no longer needed, the resources it had allocated can be freed.
freeQuery(query)
rm(query)

---

### getTermType

**Return the redland node type for the specified RDF term in a statement**

**Description**

After a `Statement` object has been created, this method can be used to determine the RDF type ("uri", "literal", "blank") that has been assigned to the specified RDF term, i.e. "subject", "predicate", "object".

**Usage**

```r
getTermType(.Object, term)
```

```r
## S4 method for signature 'Statement, character'
gTermType(.Object, term)
```

**Arguments**

- `.Object` a `Statement` object
- `term` the RDF term for which the type will be returned

**Examples**

```r
world <- new("World")
subject <- new("Node", blank="_:myid1", world)
object <- new("Node", literal="thing", world)
stmt <- new("Statement", world, subject, predicate, object, world)
termType <- getTermType(stmt, "predicate")
```
## initialize,Model-method

Constructor for a Model object.

### Description

Constructor for a Model object.

### Usage

```r
## S4 method for signature 'Model'
initialize(.Object, world, storage, options)
```

### Arguments

- `.Object`: a Node object
- `world`: a World object
- `storage`: a Storage object
- `options`: extra options for model initialization

### Value

the World object

## initialize,Node-method

Initialize a Node object.

### Description

A Node has an associated type corresponding to the RDF component that it is representing. The list of possible types is "resource", "literal" or "blank".

### Usage

```r
## S4 method for signature 'Node'
initialize(.Object, world, literal, uri, blank, datatype_uri, language)
```
**Arguments**

- **.Object**: the Node object to be initialized
- **world**: a World object
- **literal**: a literal character value to be assigned to the node
- **uri**: a uri character value to be assigned to the node
- **blank**: a blank node identifier to be assigned to the node
- **datatype_uri**: a uri used to specify the datatype of a literal node, i.e. "http://www.w3.org/2001/XMLSchema#string"
- **language**: a character value specifying the RDF language tag (excluding the "@" symbol), i.e. "fr"

**Details**

The `url='` and `literal=` arguments determine which type of Node is created. The Node type affects how the Node is processed in serialization, for example a Node created with `node1 <- new("Node", literal="http://www.example.com")` is processed differently than a Node created with `node1 <- new("Node", url="http://www.example.com")`, with the former being processed as an RDF literal and the latter processed as an RDF resource.

**Value**

the Node object

**Note**

Refer to https://www.w3.org/TR/rdf11-concepts information on language tags.

---

**Description**

A Parser object is initialized for a specific RDF serialization.

**Usage**

```r
## S4 method for signature 'Parser'
initialize(
  .Object, 
  world, 
  name = "rdfxml", 
  mimeType = "application/rdf+xml", 
  typeUri = as.character(NA)
)
```
Arguments

- `Object` (the Parser object)
- `world` (a World object)
- `name` (name of the parser factory to use)
- `mimeType` (a mime type of the syntax of the model)
- `typeUri` (a URI for the syntax of the model)

Details

The serialization format that are supported by

Value

- the Parser object

initialize, Query-method

*Initialize the Query object.*

Description

Initialize the Query object.

Usage

```r
## S4 method for signature 'Query'
initialize(
  .Object, 
  world, 
  querystring, 
  base_uri = NULL, 
  query_language = "sparql", 
  query_uri = NULL
)
```

Arguments

- `Object` (the Query object)
- `world` (a World object)
- `querystring` (a query string for the language specified in `query_language`)
- `base_uri` (a URI to prepend to relative URI in the document)
- `query_language` (the query language to execute the querystring with)
- `query_uri` (a URI to prepend to terms in the query)

Value

- the Query object
**initialize,QueryResults-method**

*Initialize the QueryResults object.*

**Description**

The QueryResults object is initialized with the librdf query result from return value of `Query.execute()`.

**Usage**

```r
## S4 method for signature 'QueryResults'
initialize(.Object, results)
```

**Arguments**

- `.Object`: the QueryResults object.
- `results`: a librdf query result

**Details**

A QueryResults object is returned by the `Query.executeQuery()` method, so typically a user does not initialize a QueryResult object by calling `new("QueryResult", ...)`

**Value**

the QueryResults object

---

**initialize,Serializer-method**

*Construct a Serializer object.*

**Description**

Construct a Serializer object.

**Usage**

```r
## S4 method for signature 'Serializer'
initialize(
    .Object,
    world,
    name = "rdfxml",
    mimeType = "application/rdf+xml",
    typeUri = as.character(NA)
)
```
### initialize,Statement-method

**Arguments**

- `.Object` the Serializer object
- `world` a World object
- `name` name of a previously created serializer factory to use
- `mimeType` a mime type of the syntax of the model
- `typeUri` a URI for the syntax of the model

**Value**

the Serializer object

---

Construct a Statement object.

---

### Description

Construct a Statement object.

### Usage

```r
## S4 method for signature 'Statement'
initialize(
  .Object, world, subject, predicate, object, 
  subjectType = as.character(NA), 
  objectType = as.character(NA), 
  datatype_uri = as.character(NA), 
  language = as.character(NA)
)
```

### Arguments

- `.Object` the Statement object
- `world` a World object
- `subject` a Node object
- `predicate` a Node object
- `object` a Node object
- `subjectType` the Node type of the subject, i.e. "blank", "uri"
- `objectType` the Node type of the object, i.e. "blank", "uri", "literal"
- `datatype_uri` the datatype URI to associate with an object literal value
- `language` a character value specifying the RDF language tag for an object literal value (excluding the "@" symbol), i.e. "fr"
### initialize,Storage-method

**Initialize a Storage object**

**Description**

Initialize a Storage object

**Usage**

```r
## S4 method for signature 'Storage'
initialize(
  .Object,
  world,
  type = "hashes",
  name = "",
  options = "hash-type='memory'"
)
```

**Arguments**

- `.Object` the Storage object
- `world` the World object
- `type` the Redland storage type
- `name` storage instance name
- `options` storage options

**Value**

the Storage object

**Examples**

```r
world <- new("World")
storage <- new("Storage", world, "hashes", name="", options="hash-type='memory'")
```
**initialize, World-method**

*Initialize the World object.*

---

**Description**

Initialize the World object.

**Usage**

```r
## S4 method for signature 'World'
initialize(.Object)
```

**Arguments**

- `.Object` the World object

**Value**

the World object

---

**is.null.externalptr**

*Determine whether an externalptr object is NULL.*

---

**Description**

The pointer is treated as an externalptr and checked via a call in C to see if it is NULL.

**Usage**

```r
is.null.externalptr(pointer)
```

**Arguments**

- `pointer` externalptr to be checked for NULL value

**Value**

logical TRUE if the pointer is NULL, otherwise FALSE
length(SWIGArray)-method

Return length of a SWIGArray

Description
Return length of a SWIGArray

Usage
## S4 method for signature 'SWIGArray'
length(x)

Arguments

  x  the SWIGArray

librdf_copyright_string

Copyright string (multiple lines).

Description
Copyright string (multiple lines).

Usage
librdf_copyright_string (.copy)

Arguments

  .copy  NA

Value
character

References
https://librdf.org/docs/

See Also
This R function is a wrapper function that directly calls the the Redland RDF C libraries. For more information about Redland RDF, view the online documentation indicated in the ‘References’ section.
librdf_copyright_string_get

Return Redland RDF copyright string

Description
Return the Redland RDF copyright

Usage
librdf_copyright_string_get (.copy)

Arguments
.copy logical

Value
character

References
https://librdf.org/docs/

See Also
This R function is a wrapper function that directly calls the the Redland RDF C libraries. For more information about Redland RDF, view the online documentation indicated in the 'References' section.

librdf_digest_final
Finish the digesting of data.

Description
Finish the digesting of data.

Usage
librdf_digest_final ( digest )

Arguments
digest the digest ("_p_librdf_digest_s")
librdf_digest_init

Value

void

References

https://librdf.org/docs/

See Also

This R function is a wrapper function that directly calls the the Redland RDF C libraries. For more information about Redland RDF, view the online documentation indicated in the 'References' section.

---

librdf_digest_init  (Re)initialise the librdf_digest object.

Description

(Re)initialise the librdf_digest object.

Usage

librdf_digest_init ( digest )

Arguments

digest the digest ("_p_librdf_digest_s")

Value

void

References

https://librdf.org/docs/

See Also

This R function is a wrapper function that directly calls the the Redland RDF C libraries. For more information about Redland RDF, view the online documentation indicated in the 'References' section.
librdf_digest_to_string

Get a string representation of the digest object.

Description

Get a string representation of the digest object.

Usage

librdf_digest_to_string ( digest )

Arguments

digest the digest ("_p_librdf_digest_s")

Value

character

References

https://librdf.org/docs/

See Also

This R function is a wrapper function that directly calls the Redland RDF C libraries. For more information about Redland RDF, view the online documentation indicated in the 'References' section.

librdf_digest_update

Add more data to the librdf_digest object.

Description

Add more data to the librdf_digest object.

Usage

librdf_digest_update ( digest, buf, length )
librdf_digest_update_string

**Description**

Add a string to the librdf_digest object.

**Usage**

```
librdf_digest_update_string ( digest, string )
```

**Arguments**

- `digest` the digest ("_p_librdf_digest_s")
- `string` string to add ("character")

**Value**

`void`

**References**

[https://librdf.org/docs/](https://librdf.org/docs/)

This R function is a wrapper function that directly calls the Redland RDF C libraries. For more information about Redland RDF, view the online documentation indicated in the 'References' section.
See Also

This R function is a wrapper function that directly calls the the Redland RDF C libraries. For more information about Redland RDF, view the online documentation indicated in the 'References' section.

librdf_free_digest  Destructor - destroy a librdf_digest object.

Description

Destructor - destroy a librdf_digest object.

Usage

librdf_free_digest ( digest )

Arguments

digest the digest ("_p_librdf_digest_s")

Value

void

References

https://librdf.org/docs/

See Also

This R function is a wrapper function that directly calls the the Redland RDF C libraries. For more information about Redland RDF, view the online documentation indicated in the 'References' section.

librdf_free_hash  Destructor - destroy a librdf_hash object.

Description

Destructor - destroy a librdf_hash object.

Usage

librdf_free_hash ( hash )
Arguments

hash hash object ("_p_librdf_hash_s")

Value

void

References

https://librdf.org/docs/

See Also

This R function is a wrapper function that directly calls the Redland RDF C libraries. For more information about Redland RDF, view the online documentation indicated in the 'References' section.

librdf_free_iterator Destructor - destroy a librdf_iterator object.

Description

Destructor - destroy a librdf_iterator object.

Usage

librdf_free_iterator ( s_arg1 )

Arguments

s_arg1 the librdf_iterator object ("_p_librdf_iterator_s")

Value

void

References

https://librdf.org/docs/

See Also

This R function is a wrapper function that directly calls the Redland RDF C libraries. For more information about Redland RDF, view the online documentation indicated in the 'References' section.
librdf_free_model  Destructor - Destroy a librdf_model object.

Description

Destructor - Destroy a librdf_model object.

Usage

librdf_free_model ( model )

Arguments

model librdf_model model to destroy ("_p_librdf_model_s")

Value

void

References

https://librdf.org/docs/

See Also

This R function is a wrapper function that directly calls the the Redland RDF C libraries. For more information about Redland RDF, view the online documentation indicated in the 'References' section.

librdf_free_node  Destructor - destroy an librdf_node object.

Description

Destructor - destroy an librdf_node object.

Usage

librdf_free_node ( r )

Arguments

r librdf_node object ("_p_librdf_node_s")

Value

void
References

https://librdf.org/docs/

See Also

This R function is a wrapper function that directly calls the the Redland RDF C libraries. For more information about Redland RDF, view the online documentation indicated in the 'References' section.

librdf_free_parser

Destructor - destroys a librdf_parser object.

Description

Destructor - destroys a librdf_parser object.

Usage

librdf_free_parser ( parser )

Arguments

parser the parser (_p_librdf_parser_s)

Value

void

References

https://librdf.org/docs/

See Also

This R function is a wrapper function that directly calls the the Redland RDF C libraries. For more information about Redland RDF, view the online documentation indicated in the 'References' section.
### librdf_free_query

#### Description

Destructor - destroy a librdf_query object.

#### Usage

```r
librdf_free_query(query)
```

#### Arguments

- `query`  
  librdf_query object ("_p_librdf_query")

#### Value

void

#### References

https://librdf.org/docs/

#### See Also

This R function is a wrapper function that directly calls the the Redland RDF C libraries. For more information about Redland RDF, view the online documentation indicated in the 'References' section.

---

### librdf_free_query_results

#### Description

Destructor - destroy a librdf_query_results object.

#### Usage

```r
librdf_free_query_results(query_results)
```

#### Arguments

- `query_results`  
  librdf_query_results object ("_p_librdf_query_results")
librdf_free_serializer

Value

void

References

https://librdf.org/docs/

See Also

This R function is a wrapper function that directly calls the the Redland RDF C libraries. For more information about Redland RDF, view the online documentation indicated in the 'References' section.

---------------------------------------------------------------------

librdf_free_serializer

Destructor - destroys a librdf_serializer object.

---------------------------------------------------------------------

Description

Destructor - destroys a librdf_serializer object.

Usage

librdf_free_serializer ( serializer )

Arguments

serializer the serializer("_p_librdf_serializer_s")

Value

void

References

https://librdf.org/docs/

See Also

This R function is a wrapper function that directly calls the the Redland RDF C libraries. For more information about Redland RDF, view the online documentation indicated in the 'References' section.
librdf_free_statement  Destructor - destroy a librdf_statement.

Description
Destructor - destroy a librdf_statement.

Usage
librdf_free_statement ( statement )

Arguments
statement  librdf_statement object ("_p_librdf_statement_s")

Value
void

References
https://librdf.org/docs/

See Also
This R function is a wrapper function that directly calls the Redland RDF C libraries. For more information about Redland RDF, view the online documentation indicated in the 'References' section.

librdf_free_storage  Destructor - destroy a librdf_storage object.

Description
Destructor - destroy a librdf_storage object.

Usage
librdf_free_storage ( storage )

Arguments
storage  librdf_storage object ("_p_librdf_storage_s")

Value
void
References

https://librdf.org/docs/

See Also

This R function is a wrapper function that directly calls the Redland RDF C libraries. For more information about Redland RDF, view the online documentation indicated in the 'References' section.

librdf_free_stream

Destructor - destroy an libdf_stream object.

Description

Destructor - destroy an libdf_stream object.

Usage

librdf_free_stream ( stream )

Arguments

stream librdf_stream object ("_p_librdf_stream_s")

Value

void

References

https://librdf.org/docs/

See Also

This R function is a wrapper function that directly calls the Redland RDF C libraries. For more information about Redland RDF, view the online documentation indicated in the 'References' section.
librdf_free_uri

Destructor - destroy a librdf_uri object.

Description

Destructor - destroy a librdf_uri object.

Usage

librdf_free_uri ( uri )

Arguments

uri librdf_uri object ("_p_librdf_uri_s")

Value

void

References

https://librdf.org/docs/

See Also

This R function is a wrapper function that directly calls the Redland RDF C libraries. For more information about Redland RDF, view the online documentation indicated in the 'References' section.

librdf_free_world

Terminate the library and frees all allocated resources.

Description

Terminate the library and frees all allocated resources.

Usage

librdf_free_world ( world )

Arguments

world redland world object ("_p_librdf_world_s")

Value

void
References

https://librdf.org/docs/

See Also

This R function is a wrapper function that directly calls the Redland RDF C libraries. For more information about Redland RDF, view the online documentation indicated in the ‘References’ section.

librdf_hash_to_string

Format the hash as a string, suitable for parsing by librdf_hash_from_string.

Description

Format the hash as a string, suitable for parsing by librdf_hash_from_string.

Usage

librdf_hash_to_string(hash, filter)

Arguments

<table>
<thead>
<tr>
<th>Argument</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>hash</td>
<td>librdf_hash object (&quot;_p_librdf_hash_s&quot;)</td>
</tr>
<tr>
<td>filter</td>
<td>NULL terminated list of keys to ignore (&quot;_p_p_char&quot;)</td>
</tr>
</tbody>
</table>

Value

character

References

https://librdf.org/docs/

See Also

This R function is a wrapper function that directly calls the Redland RDF C libraries. For more information about Redland RDF, view the online documentation indicated in the ‘References’ section.
librdf_internal_test_error

For internal testing, not part of public API

Description

This function is for internal testing of the Redland software and is not part of the public API.

Usage

librdf_internal_test_error ( world )

Arguments

world  librdf_world object ("_p_librdf_world_s")

Value

void

References

https://librdf.org/docs/

See Also

This R function is a wrapper function that directly calls the the Redland RDF C libraries. For more information about Redland RDF, view the online documentation indicated in the 'References' section.

librdf_internal_test_warning

For internal testing, not part of public API

Description

This function is for internal testing of the Redland software and is not part of the public API.

Usage

librdf_internal_test_warning ( world )

Arguments

world  librdf_world ("_p_librdf_world_s")
librdf_iterator_end

Value

void

References

https://librdf.org/docs/

See Also

This R function is a wrapper function that directly calls the Redland RDF C libraries. For more information about Redland RDF, view the online documentation indicated in the 'References' section.

Description

Test if the iterator has finished.

Usage

librdf_iterator_end ( iterator, .copy )

Arguments

iterator the librdf_iterator object ("_p_librdf_iterator_s")
.copy NA

Value

integer

References

https://librdf.org/docs/

See Also

This R function is a wrapper function that directly calls the Redland RDF C libraries. For more information about Redland RDF, view the online documentation indicated in the 'References' section.
librdf_iterator_get_context

*Get the context of the current object on the iterator.*

**Description**

Get the context of the current object on the iterator.

**Usage**

```
librdf_iterator_get_context ( iterator )
```

**Arguments**

- **iterator**
  
  The librdf_iterator object ("_p_librdf_iterator_s")

**Value**

```
_p_librdf_node_s
```

**References**

[https://librdf.org/docs/](https://librdf.org/docs/)

**See Also**

This R function is a wrapper function that directly calls the Redland RDF C libraries. For more information about Redland RDF, view the online documentation indicated in the 'References' section.

---

librdf_iterator_get_object

*Get the current object from the iterator.*

**Description**

Get the current object from the iterator.

**Usage**

```
librdf_iterator_get_object ( iterator )
```

**Arguments**

- **iterator**
  
  The librdf_iterator object ("_p_librdf_iterator_s")
librdf_iterator_next

Value

_p_librdf_node_s

References

https://librdf.org/docs/

See Also

This R function is a wrapper function that directly calls the the Redland RDF C libraries. For more information about Redland RDF, view the online documentation indicated in the 'References' section.

librdf_iterator_next  Move to the next iterator element.

Description

Move to the next iterator element.

Usage

librdf_iterator_next ( iterator, .copy )

Arguments

iterator  the librdf_iterator object ("_p_librdf_iterator_s")
.copy     NA

Value

integer

References

https://librdf.org/docs/

See Also

This R function is a wrapper function that directly calls the the Redland RDF C libraries. For more information about Redland RDF, view the online documentation indicated in the 'References' section.
librdf_log_message_code

Retrieve error code from log message.

Description
Retrieve error code from log message.

Usage
librdf_log_message_code ( message, .copy )

Arguments
message log message ("_p_librdf_log_message")
.copy NA

Value
integer

References
https://librdf.org/docs/

See Also
This R function is a wrapper function that directly calls the the Redland RDF C libraries. For more information about Redland RDF, view the online documentation indicated in the 'References' section.

librdf_log_message_facility

Retrieve facility that generated the message.

Description
Retrieve facility that generated the message.

Usage
librdf_log_message_facility ( message, .copy )
Arguments

message  log message ("_p_librdf_log_message")
.copy    NA

Value

integer

References

https://librdf.org/docs/

See Also

This R function is a wrapper function that directly calls the Redland RDF C libraries. For more information about Redland RDF, view the online documentation indicated in the ‘References’ section.
**librdf_log_message_locator**

*Retrieve locator of log entry.*

**Description**

Retrieve locator of log entry.

**Usage**

```
librdf_log_message_locator ( message )
```

**Arguments**

- `message`: log message ("_p_librdf_log_message")

**Value**

"_p_raptor_locator"

**References**

[https://librdf.org/docs/](https://librdf.org/docs/)

**See Also**

This R function is a wrapper function that directly calls the the Redland RDF C libraries. For more information about Redland RDF, view the online documentation indicated in the 'References' section.

---

**librdf_log_message_message**

*Retrieve text message from log entry.*

**Description**

Retrieve text message from log entry.

**Usage**

```
librdf_log_message_message ( message )
```

**Arguments**

- `message`: log message ("_p_librdf_log_message")
librdf_model_add

Create and add a new statement about a resource to the model.

Description

Create and add a new statement about a resource to the model.

Usage

librdf_model_add ( model,
subject,
predicate,
object,
.copy )

Arguments

model model object ("_p_librdf_model_s")
subject librdf_node of subject ("_p_librdf_node_s")
predicate librdf_node of predicate ("_p_librdf_node_s")
object librdf_node of object (literal or resource) ("_p_librdf_node_s")
.copy NA

Value

integer

References

https://librdf.org/docs/
See Also
This R function is a wrapper function that directly calls the Redland RDF C libraries. For more information about Redland RDF, view the online documentation indicated in the 'References' section.

---

**librdf_model_add_statement**

*Add a statement to the model.*

---

**Description**
Add a statement to the model.

**Usage**

`librdf_model_add_statement ( model, statement, .copy )`

**Arguments**

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>model</td>
<td>model object (&quot;_p_librdf_model_s&quot;)</td>
</tr>
<tr>
<td>statement</td>
<td>statement object (&quot;_p_librdf_statement_s&quot;)</td>
</tr>
<tr>
<td>.copy</td>
<td>NA</td>
</tr>
</tbody>
</table>

**Value**
integer

**References**

[https://librdf.org/docs/](https://librdf.org/docs/)

**See Also**
This R function is a wrapper function that directly calls the Redland RDF C libraries. For more information about Redland RDF, view the online documentation indicated in the 'References' section.
librdf_model_add_statements

Description
Add a stream of statements to the model.

Usage
librdf_model_add_statements ( model,
statement_stream,
.copy )

Arguments
model model object("_p_librdf_model_s")
statement_stream stream of statements to use("_p_librdf_stream_s")
.copy NA

Value
integer

References
https://librdf.org/docs/

See Also
This R function is a wrapper function that directly calls the the Redland RDF C libraries. For
more information about Redland RDF, view the online documentation indicated in the 'References'
section.

librdf_model_add_string_literal_statement

Description
Create and add a new statement about a literal to the model.

Create and add a new statement about a literal to the model.
Usage

```
librdf_model_add_string_literal_statement ( model,
  subject,
  predicate,
  literal,
  inStrOrNull,
  is_wf_xm__xml,
  .copy )
```

Arguments

- **model**: model object ("_p_librdf_model_s")
- **subject**: librdf_node of subject ("_p_librdf_node_s")
- **predicate**: librdf_node of predicate ("_p_librdf_node_s")
- **literal**: string literal content ("character")
- **inStrOrNull**: language of literal ("character")
- **is_wf_xm__xml**: literal is XML ("integer")
- **.copy**: NA

Value

integer

References

https://librdf.org/docs/

See Also

This R function is a wrapper function that directly calls the Redland RDF C libraries. For more information about Redland RDF, view the online documentation indicated in the 'References' section.

**librdf_model_add_typed_literal_statement**

Create and add a new statement about a typed literal to the model.

Description

Create and add a new statement about a typed literal to the model.
librdf_model_add_typed_literal_statement (model, subject, predicate, string, inStrOrNull, inUriOrNull, .copy)

Arguments

- **model**: model object ("_p_librdf_model_s")
- **subject**: librdf_node of subject ("_p_librdf_node_s")
- **predicate**: librdf_node of predicate ("_p_librdf_node_s")
- **string**: string literal content ("character")
- **inStrOrNull**: language of literal ("character")
- **inUriOrNull**: datatype librdf_uri ("_p_librdf_uri_s")
- **.copy**: NA

Value

integer

References

https://librdf.org/docs/

See Also

This R function is a wrapper function that directly calls the Redland RDF C libraries. For more information about Redland RDF, view the online documentation indicated in the 'References' section.

librdf_model_as_stream

List the model contents as a stream of statements.

Description

List the model contents as a stream of statements.

Usage

librdf_model_as_stream (model)
librdf_model_contains_context

Check for a context in the model.

Description

Check for a context in the model.

Usage

librdf_model_contains_context ( model,
context,
.copy )

Arguments

model the model object ("_p_librdf_model_s")
context the context ("_p_librdf_node_s")
.copy NA

Value

integer

References

https://librdf.org/docs/
librdf_model_contains_statement

See Also

This R function is a wrapper function that directly calls the Redland RDF C libraries. For more information about Redland RDF, view the online documentation indicated in the 'References' section.

librdf_model_contains_statement

_Check for a statement in the model._

Description

Check for a statement in the model.

Usage

librdf_model_contains_statement ( model,
statement,
.copy )

Arguments

model the model object ("_p_librdf_model_s")
statement the statement ("_p_librdf_statement_s")
.copy NA

Value

integer

References

https://librdf.org/docs/

See Also

This R function is a wrapper function that directly calls the Redland RDF C libraries. For more information about Redland RDF, view the online documentation indicated in the 'References' section.
librdf_model_context_add_statement

Add a statement to a model with a context.

Description

Add a statement to a model with a context.

Usage

librdf_model_context_add_statement ( model,
context,
statement,
.copy )

Arguments

model librdf_model object ("_p_librdf_model_s")
context librdf_node context ("_p_librdf_node_s")
statement librdf_statement statement object ("_p_librdf_statement_s")
.copy NA

Value

integer

References

https://librdf.org/docs/

See Also

This R function is a wrapper function that directly calls the the Redland RDF C libraries. For more information about Redland RDF, view the online documentation indicated in the 'References' section.
librdf_model_context_add_statements

Add statements to a model with a context.

Description

Add statements to a model with a context.

Usage

librdf_model_context_add_statements ( model, context, stream, .copy )

Arguments

model       librdf_model object ("_p_librdf_model_s")
context     librdf_node context ("_p_librdf_node_s")
stream      librdf_stream stream object ("_p_librdf_stream_s")
.copy       NA

Value

integer

References

https://librdf.org/docs/

See Also

This R function is a wrapper function that directly calls the Redland RDF C libraries. For more information about Redland RDF, view the online documentation indicated in the 'References' section.
librdf_model_context_as_stream

List all statements in a model context.

Description

List all statements in a model context.

Usage

librdf_model_context_as_stream ( model, context )

Arguments

model librdf_model object (_p_librdf_model_s)  
context librdf_node context (_p_librdf_node_s)

Value

_p_librdf_stream_s

References

https://librdf.org/docs/

See Also

This R function is a wrapper function that directly calls the the Redland RDF C libraries. For more information about Redland RDF, view the online documentation indicated in the ‘References’ section.

librdf_model_context_remove_statement

Remove a statement from a model in a context.

Description

Remove a statement from a model in a context.

Usage

librdf_model_context_remove_statement ( model, context, statement, .copy )
Arguments

model: librdf_model object ("_p_librdf_model_s")
context: librdf_node context ("_p_librdf_node_s")
statement: librdf_statement statement ("_p_librdf_statement_s")
.copy: NA

Value

integer

References

https://librdf.org/docs/

See Also

This R function is a wrapper function that directly calls the the Redland RDF C libraries. For more information about Redland RDF, view the online documentation indicated in the ‘References’ section.

librdf_model_context_remove_statements

Remove statements from a model with the given context.

Description

Remove statements from a model with the given context.

Usage

librdf_model_context_remove_statements ( model, context, .copy )

Arguments

model: librdf_model object ("_p_librdf_model_s")
context: librdf_node context ("_p_librdf_node_s")
.copy: NA

Value

integer

References

https://librdf.org/docs/
See Also

This R function is a wrapper function that directly calls the Redland RDF C libraries. For more information about Redland RDF, view the online documentation indicated in the 'References' section.

---

**librdf_model_find_statements**

Find matching statements in the model.

**Description**

Find matching statements in the model.

**Usage**

librdf_model_find_statements ( model, statement )

**Arguments**

- **model**
  - the model object ("_p_librdf_model_s")
- **statement**
  - the partial statement to match ("_p_librdf_statement_s")

**Value**

_p_librdf_stream_s

**References**

https://librdf.org/docs/

**See Also**

This R function is a wrapper function that directly calls the Redland RDF C libraries. For more information about Redland RDF, view the online documentation indicated in the 'References' section.
librdf_model_find_statements_in_context

Search the model for matching statements in a given context.

Description

Search the model for matching statements in a given context.

Usage

librdf_model_find_statements_in_context ( model,
statement,
inNodeOrNull )

Arguments

model     librdf_model object ("_p_librdf_model_s")
statement librdf_statement partial statement to find ("_p_librdf_statement_s")
inNodeOrNull context librdf_node (or NULL) ("_p_librdf_node_s")

Value

_p_librdf_stream_s

References

https://librdf.org/docs/

See Also

This R function is a wrapper function that directly calls the the Redland RDF C libraries. For more information about Redland RDF, view the online documentation indicated in the ‘References’ section.

librdf_model_get_arc

Return one arc (predicate) of an arc in an RDF graph given source (subject) and target (object).

Description

Return one arc (predicate) of an arc in an RDF graph given source (subject) and target (object).

Usage

librdf_model_get_arc ( model,
source,
target )
librdf_model_get_arcs

Arguments

- model: librdf_model object ("_p_librdf_model_s")
- source: librdf_node source ("_p_librdf_node_s")
- target: librdf_node target ("_p_librdf_node_s")

Value

-_p_librdf_node_s

References

https://librdf.org/docs/

See Also

This R function is a wrapper function that directly calls the Redland RDF C libraries. For more information about Redland RDF, view the online documentation indicated in the ‘References’ section.

Description

Return the arcs (predicates) of an arc in an RDF graph given source (subject) and target (object).

Usage

librdf_model_get_arcs ( model, source, target )

Arguments

- model: librdf_model object ("_p_librdf_model_s")
- source: librdf_node source ("_p_librdf_node_s")
- target: librdf_node target ("_p_librdf_node_s")

Value

-_p_librdf_iterator_s

References

https://librdf.org/docs/
librdf_model_get_arcs_in

Return the properties pointing to the given resource.

Description

Return the properties pointing to the given resource.

Usage

librdf_model_get_arcs_in ( model, 
node )

Arguments

model librdf_model object ("_p_librdf_model_s")
node librdf_node resource node ("_p_librdf_node_s")

Value

_p_librdf_iterator_s

References

https://librdf.org/docs/

See Also

This R function is a wrapper function that directly calls the the Redland RDF C libraries. For more information about Redland RDF, view the online documentation indicated in the 'References' section.
**librdf_model_get_arcs_out**

*Return the properties pointing from the given resource.*

**Description**

Return the properties pointing from the given resource.

**Usage**

```r
librdf_model_get_arcs_out ( model, node )
```

**Arguments**

- `model` : librdf_model object ("_p_librdf_model_s")
- `node` : librdf_node resource node ("_p_librdf_node_s")

**Value**

 `_p_librdf_iterator_s`

**References**

https://librdf.org/docs/

**See Also**

This R function is a wrapper function that directly calls the the Redland RDF C libraries. For more information about Redland RDF, view the online documentation indicated in the 'References' section.

---

**librdf_model_get_contexts**

*Return the list of contexts in the graph.*

**Description**

Return the list of contexts in the graph.

**Usage**

```r
librdf_model_get_contexts ( model )
```
librdf_model_get_feature

Get the value of a graph feature.

**Description**

Get the value of a graph feature.

**Usage**

```
librdf_model_get_feature ( model, feature )
```

**Arguments**

- `model` : librdf_model object ("_p_librdf_model_s")
- `feature` : librdf_uri feature property ("_p_librdf_uri_s")

**Value**

"_p_librdf_node_s"

**References**

[https://librdf.org/docs/](https://librdf.org/docs/)

**See Also**

This R function is a wrapper function that directly calls the Redland RDF C libraries. For more information about Redland RDF, view the online documentation indicated in the 'References' section.
librdf_model_get_source

Return one source (subject) of arc in an RDF graph given arc (predicate) and target (object).

Description

Return one source (subject) of arc in an RDF graph given arc (predicate) and target (object).

Usage

librdf_model_get_source ( model,
arc,
target )

Arguments

model librdf_model object ("_p_librdf_model_s")
arc librdf_node arc ("_p_librdf_node_s")
target librdf_node target ("_p_librdf_node_s")

Value

_p_librdf_node_s

References

https://librdf.org/docs/

See Also

This R function is a wrapper function that directly calls the the Redland RDF C libraries. For more information about Redland RDF, view the online documentation indicated in the 'References' section.

librdf_model_get_sources

Return the sources (subjects) of arc in an RDF graph given arc (predicate) and target (object).

Description

Return the sources (subjects) of arc in an RDF graph given arc (predicate) and target (object).
librdf_model_get_target

Usage

librdf_model_get_sources ( model,
arc,
target )

Arguments

model      librdf_model object ("_p_librdf_model_s")
arc        librdf_node arc ("_p_librdf_node_s")
target     librdf_node target ("_p_librdf_node_s")

Value

_p_librdf_iterator_s

References

https://librdf.org/docs/

See Also

This R function is a wrapper function that directly calls the Redland RDF C libraries. For more information about Redland RDF, view the online documentation indicated in the "References" section.

---

librdf_model_get_target

Return one target (object) of an arc in an RDF graph given source (subject) and arc (predicate).

Description

Return one target (object) of an arc in an RDF graph given source (subject) and arc (predicate).

Usage

librdf_model_get_target ( model,
source,
arc )

Arguments

model      librdf_model object ("_p_librdf_model_s")
source     librdf_node source ("_p_librdf_node_s")
arc        librdf_node arc ("_p_librdf_node_s")
librdf_model_get_targets

Value

_p_librdf_node_s

References

https://librdf.org/docs/

See Also

This R function is a wrapper function that directly calls the Redland RDF C libraries. For more information about Redland RDF, view the online documentation indicated in the 'References' section.

librdf_model_get_targets

Return the targets (objects) of an arc in an RDF graph given source (subject) and arc (predicate).

Description

Return the targets (objects) of an arc in an RDF graph given source (subject) and arc (predicate).

Usage

librdf_model_get_targets ( model,
source,
arc )

Arguments

model librdf_model object ("_p_librdf_model_s")
source librdf_node source ("_p_librdf_node_s")
arc librdf_node arc ("_p_librdf_node_s")

Value

_p_librdf_iterator_s

References

https://librdf.org/docs/

See Also

This R function is a wrapper function that directly calls the Redland RDF C libraries. For more information about Redland RDF, view the online documentation indicated in the 'References' section.
librdf_model_has_arc_in

Check if a node has a given property pointing to it.

Description

Check if a node has a given property pointing to it.

Usage

librdf_model_has_arc_in ( model, node, property, .copy )

Arguments

model        librdf_model object ("_p_librdf_model_s")
node         librdf_node resource node ("_p_librdf_node_s")
property     librdf_node property node ("_p_librdf_node_s")
.copy        NA

Value

integer

References

https://librdf.org/docs/

See Also

This R function is a wrapper function that directly calls the the Redland RDF C libraries. For more information about Redland RDF, view the online documentation indicated in the 'References' section.
librdf_model_has_arc_out

Check if a node has a given property pointing from it.

Description

Check if a node has a given property pointing from it.

Usage

librdf_model_has_arc_out ( model, node, property, .copy )

Arguments

model librdf_model object ("_p_librdf_model_s")
node librdf_node resource node ("_p_librdf_node_s")
property librdf_node property node ("_p_librdf_node_s")
.copy NA

Value

integer

References

https://librdf.org/docs/

See Also

This R function is a wrapper function that directly calls the the Redland RDF C libraries. For more information about Redland RDF, view the online documentation indicated in the 'References' section.
**librdf_model_load**  
*Load content from a URI into the model.*

**Description**
Load content from a URI into the model.

**Usage**

```c
librdf_model_load ( model,
                    uri,
                    name,
                    mime_type,
                    type_uri,
                    .copy )
```

**Arguments**
- **model**: librdf_model object ("_p_librdf_model_s")
- **uri**: the URI to read the content ("_p_librdf_uri_s")
- **name**: the name of the parser (or NULL) ("character")
- **mime_type**: the MIME type of the syntax (NULL if not used) ("character")
- **type_uri**: URI identifying the syntax (NULL if not used) ("_p_librdf_uri_s")
- **.copy**: NA

**Value**
integer

**References**

https://librdf.org/docs/

**See Also**
This R function is a wrapper function that directly calls the the Redland RDF C libraries. For more information about Redland RDF, view the online documentation indicated in the ‘References’ section.
librdf_model_query_execute

Execute a query against the model.

Description

Execute a query against the model.

Usage

librdf_model_query_execute ( model, query )

Arguments

model librdf_model object ("_p_librdf_model_s")
query librdf_query object ("_p_librdf_query")

Value

_p_librdf_query_results

References

https://librdf.org/docs/

See Also

This R function is a wrapper function that directly calls the the Redland RDF C libraries. For more information about Redland RDF, view the online documentation indicated in the 'References' section.

librdf_model_remove_statement

Remove a known statement from the model.

Description

Remove a known statement from the model.

Usage

librdf_model_remove_statement ( model, statement, .copy )
librdf_model_set_feature

Set the value of a graph feature.

Description

Set the value of a graph feature.

Usage

librdf_model_set_feature ( model, feature, value, .copy )

Arguments

<table>
<thead>
<tr>
<th>Argument</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>model</td>
<td>librdf_model object (&quot;_p_librdf_model_s&quot;)</td>
</tr>
<tr>
<td>feature</td>
<td>librdf_uri feature property (&quot;_p_librdf_uri_s&quot;)</td>
</tr>
<tr>
<td>value</td>
<td>librdf_node feature property value (&quot;_p_librdf_node_s&quot;)</td>
</tr>
<tr>
<td>.copy</td>
<td>NA</td>
</tr>
</tbody>
</table>

Value

integer

References

https://librdf.org/docs/

See Also

This R function is a wrapper function that directly calls the Redland RDF C libraries. For more information about Redland RDF, view the online documentation indicated in the 'References' section.
librdf_model_size

References

https://librdf.org/docs/

See Also

This R function is a wrapper function that directly calls the Redland RDF C libraries. For more information about Redland RDF, view the online documentation indicated in the 'References' section.

librdf_model_size

Get the number of statements in the model.

Description

Get the number of statements in the model.

Usage

librdf_model_size ( model,
    .copy )

Arguments

model           librdf_model object ("_p_librdf_model_s")
    .copy          NA

Value

integer

References

https://librdf.org/docs/

See Also

This R function is a wrapper function that directly calls the Redland RDF C libraries. For more information about Redland RDF, view the online documentation indicated in the 'References' section.
librdf_model_to_string  
Write serialized model to a string.

Description
Write serialized model to a string.

Usage
librdf_model_to_string ( model, uri, name, mime_type, inUriOrNull )

librdf_model_sync  
Synchronise the model to the model implementation.

Description
Synchronise the model to the model implementation.

Usage
librdf_model_sync ( model )

Arguments
model  
librdf_model object ("_p_librdf_model_s")

Value
void

References
https://librdf.org/docs/

See Also
This R function is a wrapper function that directly calls the Redland RDF C libraries. For more information about Redland RDF, view the online documentation indicated in the 'References' section.
Arguments

- **model**: librdf_model object ("_p_librdf_model_s")
- **uri**: base URI to use in serializing (or NULL if not used) ("_p_librdf_uri_s")
- **name**: the name of the serializer (or NULL for default) ("character")
- **mime_type**: the MIME type of the syntax (NULL if not used) ("character")
- **inUriOrNull**: URI identifying the syntax (NULL if not used) ("_p_librdf_uri_s")

Value

- character

References

https://librdf.org/docs/

See Also

This R function is a wrapper function that directly calls the the Redland RDF C libraries. For more information about Redland RDF, view the online documentation indicated in the 'References' section.

---

Commit a transaction.

Description

Commit a transaction.

Usage

```r
librdf_model_transaction_commit ( model, .copy )
```

Arguments

- **model**: the model object ("_p_librdf_model_s")
- **.copy**: NA

Value

- integer

References

https://librdf.org/docs/
See Also

This R function is a wrapper function that directly calls the the Redland RDF C libraries. For more information about Redland RDF, view the online documentation indicated in the 'References' section.
librdf_model_transaction_start

Description

Start a transaction

Usage

librdf_model_transaction_start ( model, .copy )

Arguments

model the model object ("_p_librdf_model_s")
.copy NA

Value

integer

References

https://librdf.org/docs/

See Also

This R function is a wrapper function that directly calls the Redland RDF C libraries. For more information about Redland RDF, view the online documentation indicated in the 'References' section.

librdf_new_digest Constructor - create a new librdf_digest object.

Description

 Constructor - create a new librdf_digest object.

Usage

librdf_new_digest ( world, name )
librdf_new_hash

Arguments

world  redland world object ("_p_librdf_world_s")
name   the digest name to use to create this digest ("character")

Value

_p_librdf_digest_s

References

https://librdf.org/docs/

See Also

This R function is a wrapper function that directly calls the the Redland RDF C libraries. For more information about Redland RDF, view the online documentation indicated in the 'References' section.

Description

Constructor - create a new librdf_hash object.

Usage

librdf_new_hash ( world,
    name )

Arguments

world  redland world object ("_p_librdf_world_s")
name   factory name ("character")

Value

_p_librdf_hash_s

References

https://librdf.org/docs/

See Also

This R function is a wrapper function that directly calls the the Redland RDF C libraries. For more information about Redland RDF, view the online documentation indicated in the 'References' section.
**librdf_new_hash_from_array_of_strings**

Constructor - create a new librdf_hash object from an array of strings.

---

**Description**

Constructor - create a new librdf_hash object from an array of strings.

**Usage**

```c
librdf_new_hash_from_array_of_strings ( world, name, string )
```

**Arguments**

- **world**: redland world object ("_p_librdf_world_s")
- **name**: hash name ("character")
- **string**: address of the start of the array of char* pointers ("character")

**Value**

 `_p_librdf_hash_s`

**References**

[https://librdf.org/docs/](https://librdf.org/docs/)

**See Also**

This R function is a wrapper function that directly calls the the Redland RDF C libraries. For more information about Redland RDF, view the online documentation indicated in the 'References' section.

---

**librdf_new_hash_from_string**

Constructor - create a new librdf_hash object from a string.

---

**Description**

Constructor - create a new librdf_hash object from a string.
librdf_new_hash_from_string

Usage

librdf_new_hash_from_string ( world, name, string )

Arguments

world redland world object ("_p_librdf_world_s")
name hash name ("character")
string hash encoded as a string ("character")

Value

_p_librdf_hash_s

References

https://librdf.org/docs/

See Also

This R function is a wrapper function that directly calls the the Redland RDF C libraries. For more information about Redland RDF, view the online documentation indicated in the 'References' section.

librdf_new_model

Constructor - create a new storage librdf_model object.

Description

Constructor - create a new storage librdf_model object.

Usage

librdf_new_model ( world, storage, options_string )

Arguments

world redland world object ("_p_librdf_world_s")
storage librdf_storage to use ("_p_librdf_storage_s")
options_string options to initialise model ("character")

Value

_p_librdf_model_s
librdf_new_model_from_model

*Copy constructor - create a new librdf_model from an existing one.*

**Description**

Copy constructor - create a new librdf_model from an existing one.

**Usage**

```r
librdf_new_model_from_model ( model )
```

**Arguments**

- `model` the existing librdf_model ("_p_librdf_model_s")

**Value**

 `_p_librdf_model_s`

**References**

https://librdf.org/docs/

**See Also**

This R function is a wrapper function that directly calls the the Redland RDF C libraries. For more information about Redland RDF, view the online documentation indicated in the 'References' section.
librdf_new_model_with_options

Constructor - Create a new librdf_model with storage.

Description

Constructor - Create a new librdf_model with storage.

Usage

librdf_new_model_with_options ( world, storage, options )

Arguments

world       redland world object ("_p_librdf_world_s")
storage     librdf_storage storage to use ("_p_librdf_storage_s")
options     librdf_hash of options to use ("_p_librdf_hash_s")

Value

_p_librdf_model_s

References

https://librdf.org/docs/

See Also

This R function is a wrapper function that directly calls the the Redland RDF C libraries. For more information about Redland RDF, view the online documentation indicated in the 'References' section.

librdf_new_node

Constructor - create a new librdf_node object with a private identifier.

Description

Constructor - create a new librdf_node object with a private identifier.

Usage

librdf_new_node ( world )
`librdf_new_node_from_blank_identifier`  

**Arguments**  

- `world`  
  redland world object ("_p_librdf_world_s")  

**Value**  

- `_p_librdf_node_s`

**References**  

- [https://librdf.org/docs/](https://librdf.org/docs/)

**See Also**  

This R function is a wrapper function that directly calls the Redland RDF C libraries. For more information about Redland RDF, view the online documentation indicated in the 'References' section.

---

**Description**  

Constructor - create a new blank node `librdf_node` object from a blank node identifier.

**Usage**  

```r  
librdf_new_node_from_blank_identifier ( world,  
inStrOrNull )  
```

**Arguments**  

- `world`  
  redland world object ("_p_librdf_world_s")  
- `inStrOrNull`  
  UTF-8 encoded blank node identifier or NULL ("character")

**Value**  

- `_p_librdf_node_s`

**References**  

- [https://librdf.org/docs/](https://librdf.org/docs/)

**See Also**  

This R function is a wrapper function that directly calls the Redland RDF C libraries. For more information about Redland RDF, view the online documentation indicated in the 'References' section.
librdf_new_node_from_literal

Constructor - create a new literal librdf_node object.

Description

Constructor - create a new literal librdf_node object.

Usage

librdf_new_node_from_literal ( world,
    string,
    inStrOrNull,
    is_wf_xml )

Arguments

world         redland world object ("_p_librdf_world_s")
string        literal UTF-8 encoded string value ("character")
inStrOrNull   literal XML language (or NULL, empty string) ("character")
is_wf_xml     non 0 if literal is XML ("integer")

Value

_p_librdf_node_s

References

https://librdf.org/docs/

See Also

This R function is a wrapper function that directly calls the Redland RDF C libraries. For more information about Redland RDF, view the online documentation indicated in the 'References' section.
librdf_new_node_from_node

Copy constructor - create a new librdf_node object from an existing librdf_node object.

Description

Copy constructor - create a new librdf_node object from an existing librdf_node object.

Usage

librdf_new_node_from_node ( node )

Arguments

node librdf_node object to copy ("_p_librdf_node_s")

Value

_p_librdf_node_s

References

https://librdf.org/docs/

See Also

This R function is a wrapper function that directly calls the Redland RDF C libraries. For more information about Redland RDF, view the online documentation indicated in the 'References' section.

librdf_new_node_from_normalised_uri_string

Constructor - create a new librdf_node object from a UTF-8 encoded URI string normalised to a new base URI.

Description

Constructor - create a new librdf_node object from a UTF-8 encoded URI string normalised to a new base URI.

Usage

librdf_new_node_from_normalised_uri_string ( world, uri_string, source_uri, base_uri )
librdf\_new\_node\_from\_typed\_literal

Constructor - create a new typed literal librdf\_node object.

Description

Constructor - create a new typed literal librdf\_node object.

Usage

librdf\_new\_node\_from\_typed\_literal ( world, string, inStrOrNull, inUriOrNull )

Arguments

world redland world object ("\_p\_librdf\_world\_s"")
string literal UTF-8 encoded string value ("character")
inStrOrNull literal XML language (or NULL, empty string) ("character")
inUriOrNull URI of typed literal datatype or NULL ("\_p\_librdf\_uri\_s"")

Value

\_p\_librdf\_node\_s

References

https://librdf.org/docs/

See Also

This R function is a wrapper function that directly calls the the Redland RDF C libraries. For more information about Redland RDF, view the online documentation indicated in the 'References' section.
**librdf_new_node_from_uri**

*Constructor - create a new resource librdf_node object with a given URI.*

---

**Description**

Constructor - create a new resource librdf_node object with a given URI.

**Usage**

```r
librdf_new_node_from_uri ( world, uri )
```

**Arguments**

- `world`: redland world object ("_p_librdf_world_s")
- `uri`: librdf_uri object ("_p_librdf_uri_s")

**Value**

`_p_librdf_node_s`

**References**

https://librdf.org/docs/

**See Also**

This R function is a wrapper function that directly calls the Redland RDF C libraries. For more information about Redland RDF, view the online documentation indicated in the 'References' section.
Description

Constructor - create a new resource librdf_node object with a given URI and local name.

Usage

librdf_new_node_from_uri_local_name ( world, uri, local_name )

Arguments

world redland world object ("_p_librdf_world_s")
uri librdf_uri object ("_p_librdf_uri_s")
local_name local name to append to URI ("character")

Value

_p_librdf_node_s

References

https://librdf.org/docs/

See Also

This R function is a wrapper function that directly calls the the Redland RDF C libraries. For more information about Redland RDF, view the online documentation indicated in the 'References' section.
librdf_new_node_from_uri_string

Usage

librdf_new_node_from_uri_string ( world, string )

Arguments

world  redland world object ("p_librdf_world_s")
string  string representing a URI ("character")

Value

_p_librdf_node_s

References

https://librdf.org/docs/

See Also

This R function is a wrapper function that directly calls the the Redland RDF C libraries. For more information about Redland RDF, view the online documentation indicated in the 'References' section.

summary

librdf_new_parser  Constructor - create a new librdf_parser object.

Description

Constructor - create a new librdf_parser object.

Usage

librdf_new_parser ( world, name, mime_type, type_uri )

Arguments

world  redland world object ("p_librdf_world_s")
name  the parser factory name (or NULL or empty string if don’t care) ("character")
mime_type  the MIME type of the syntax (NULL if not used) ("character")
type_uri  URI of syntax (NULL if not used) ("p_librdf_uri_s")

Value

_p_librdf_parser_s
librdf_new_query

Constructor - create a new librdf_query object.

Description
Constructor - create a new librdf_query object.

Usage
librdf_new_query ( world,
    name,
    uri,
    query_string,
    base_uri )

Arguments
world redland world object ("_p_librdf_world_s")
name the name identifying the query language ("character")
uri the URI identifying the query language (or NULL) ("_p_librdf_uri_s")
query_string the query string ("character")
base_uri the base URI of the query string (or NULL) ("_p_librdf_uri_s")

Value
_p_librdf_query

References
https://librdf.org/docs/

See Also
This R function is a wrapper function that directly calls the the Redland RDF C libraries. For more information about Redland RDF, view the online documentation indicated in the 'References' section.
librdf_new_query_from_query

Copy constructor - create a new librdf_query object from an existing one

Description
Copy constructor - create a new librdf_query object from an existing one

Usage
librdf_new_query_from_query ( old_query )

Arguments
old_query the existing query librdf_query to use (_p_librdf_query)

Value
_p_librdf_query

References
https://librdf.org/docs/

See Also
This R function is a wrapper function that directly calls the the Redland RDF C libraries. For more information about Redland RDF, view the online documentation indicated in the 'References' section.

librdf_new_serializer

Constructor - create a new librdf_serializer object.

Description
Constructor - create a new librdf_serializer object.

Usage
librdf_new_serializer ( world, name, mime_type, type_uri )
Arguments

world     redland world object ("_p_librdf_world_s")
name      the serializer factory name (or NULL or empty string if don’t care) ("character")
mime_type the MIME type of the syntax (NULL if not used) ("character")
type_uri   URI of syntax (NULL if not used) ("_p_librdf_uri_s")

Value

_p_librdf_serializer_s

References

https://librdf.org/docs/

See Also

This R function is a wrapper function that directly calls the Redland RDF C libraries. For more information about Redland RDF, view the online documentation indicated in the ‘References’ section.
librdf_new_statement_from_nodes

Constructor - create a new librdf_statement from existing librdf_node objects.

Description

Constructor - create a new librdf_statement from existing librdf_node objects.

Usage

librdf_new_statement_from_nodes ( world, subject, predicate, object )

Arguments

world           redland world object ("_p_librdf_world_s")
subject         librdf_node ("_p_librdf_node_s")
predicate       librdf_node ("_p_librdf_node_s")
object          librdf_node ("_p_librdf_node_s")

Value

_p_librdf_statement_s

References

https://librdf.org/docs/

See Also

This R function is a wrapper function that directly calls the the Redland RDF C libraries. For more information about Redland RDF, view the online documentation indicated in the 'References' section.
librdf_new_statement_from_statement

Copy constructor - create a new librdf_statement from an existing librdf_statement. Creates a deep copy - changes to original statement nodes are not reflected in the copy.

Description

Copy constructor - create a new librdf_statement from an existing librdf_statement. Creates a deep copy - changes to original statement nodes are not reflected in the copy.

Usage

librdf_new_statement_from_statement ( statement )

Arguments

statement librdf_statement to copy ("_p_librdf_statement_s")

Value

_p_librdf_statement_s

References

https://librdf.org/docs/

See Also

This R function is a wrapper function that directly calls the the Redland RDF C libraries. For more information about Redland RDF, view the online documentation indicated in the 'References' section.

librdf_new_storage

Constructor - create a new librdf_storage object.

Description

Constructor - create a new librdf_storage object.

Usage

librdf_new_storage ( world, storage_name, name, options_string )
**Arguments**

- **world**
  - redland world object ("_p_librdf_world_s")
- **storage_name**
  - the storage factory name ("character")
- **name**
  - an identifier for the storage ("character")
- **options_string**
  - options to initialise storage ("character")

**Value**

- _p_librdf_storage_s

**References**

https://librdf.org/docs/

**See Also**

This R function is a wrapper function that directly calls the Redland RDF C libraries. For more information about Redland RDF, view the online documentation indicated in the 'References' section.

---

```r
librdf_new_storage_from_storage

  Copy constructor - create a new librdf_storage object from an existing one

```

**Description**

Copy constructor - create a new librdf_storage object from an existing one

**Usage**

```r
librdf_new_storage_from_storage ( old_storage )
```

**Arguments**

- **old_storage**
  - the existing storage librdf_storage to use ("_p_librdf_storage_s")

**Value**

- _p_librdf_storage_s

**References**

https://librdf.org/docs/
See Also

This R function is a wrapper function that directly calls the Redland RDF C libraries. For more information about Redland RDF, view the online documentation indicated in the 'References' section.

### Description

Constructor - create a new librdf_uri object from a URI string.

### Usage

```r
librdf_new_uri ( world, string )
```

### Arguments

- `world` redland world object ("_p_librdf_world_s")
- `string` URI in string form ("character")

### Value

`_p_librdf_uri_s`

### References

[https://librdf.org/docs/](https://librdf.org/docs/)

See Also

This R function is a wrapper function that directly calls the Redland RDF C libraries. For more information about Redland RDF, view the online documentation indicated in the 'References' section.
**librdf_new_uri_from_filename**

*Constructor - create a new librdf_uri object from a filename.*

**Description**

Constructor - create a new librdf_uri object from a filename.

**Usage**

```r
librdf_new_uri_from_filename ( world, filename )
```

**Arguments**

- `world` (Redland librdf_world object ("_p_librdf_world_s")
- `filename` (filename ("character")

**Value**

_p_librdf_uri_s

**References**

[https://librdf.org/docs/](https://librdf.org/docs/)

**See Also**

This R function is a wrapper function that directly calls the Redland RDF C libraries. For more information about Redland RDF, view the online documentation indicated in the 'References' section.

---

**librdf_new_uri_from_uri**

*Copy constructor - create a new librdf_uri object from an existing librdf_uri object.*

**Description**

Copy constructor - create a new librdf_uri object from an existing librdf_uri object.

**Usage**

```r
librdf_new_uri_from_uri ( uri )
```
Arguments

uri
librdf_uri object ("_p_librdf_uri_s")

Value

_p_librdf_uri_s

References

https://librdf.org/docs/

See Also

This R function is a wrapper function that directly calls the Redland RDF C libraries. For more information about Redland RDF, view the online documentation indicated in the 'References' section.
**librdf_node_equals**

*Compare two librdf_node objects for equality.*

**Description**

Compare two librdf_node objects for equality.

**Usage**

```r
librdf_node_equals ( first_node, second_node, .copy )
```

**Arguments**

- `first_node`: first librdf_node node ("_p_librdf_node_s")
- `second_node`: second librdf_node node ("_p_librdf_node_s")
- `.copy`: NA

**Value**

integer

**References**

https://librdf.org/docs/

**See Also**

This R function is a wrapper function that directly calls the Redland RDF C libraries. For more information about Redland RDF, view the online documentation indicated in the 'References' section.

**librdf_node_get_blank_identifier**

*Get the blank node identifier as a UTF-8 encoded string.*

**Description**

Get the blank node identifier as a UTF-8 encoded string.

**Usage**

```r
librdf_node_get_blank_identifier ( node )
```
Arguments

node the node object ("_p_librdf_node_s")

Value

character

References

https://librdf.org/docs/

See Also

This R function is a wrapper function that directly calls the the Redland RDF C libraries. For more information about Redland RDF, view the online documentation indicated in the 'References' section.

---

librdf_node_get_literal_value

Get the literal value of the node as a UTF-8 encoded string.

Description

Get the literal value of the node as a UTF-8 encoded string.

Usage

librdf_node_get_literal_value ( node )

Arguments

node the node object ("_p_librdf_node_s")

Value

character

References

https://librdf.org/docs/

See Also

This R function is a wrapper function that directly calls the the Redland RDF C libraries. For more information about Redland RDF, view the online documentation indicated in the 'References' section.
librdf_node_get_literal_value_as_latin1

Get the string literal value of the node as ISO Latin-1.

Description
Get the string literal value of the node as ISO Latin-1.

Usage
librdf_node_get_literal_value_as_latin1 ( node )

Arguments
node the node object ("_p_librdf_node_s")

Value
character

References
https://librdf.org/docs/

See Also
This R function is a wrapper function that directly calls the the Redland RDF C libraries. For more information about Redland RDF, view the online documentation indicated in the 'References' section.

librdf_node_get_literal_value_datatype_uri

Get the typed literal datatype URI of the literal node.

Description
Get the typed literal datatype URI of the literal node.

Usage
librdf_node_get_literal_value_datatype_uri ( node )

Arguments
node the node object ("_p_librdf_node_s")
librdf_node_get_literal_value_is_wf_xml

Get the XML well-formness property of the node.

Description

Get the XML well-formness property of the node.

Usage

librdf_node_get_literal_value_is_wf_xml ( node, .copy )

Arguments

node the node object ("_p_librdf_node_s") .copy NA

Value

integer

References

https://librdf.org/docs/

See Also

This R function is a wrapper function that directly calls the the Redland RDF C libraries. For more information about Redland RDF, view the online documentation indicated in the 'References' section.
librdf_node_get_literal_value_language

Get the XML language of the node.

Description
Get the XML language of the node.

Usage
librdf_node_get_literal_value_language ( node )

Arguments
node the node object ("_p_librdf_node_s")

Value
character

References
https://librdf.org/docs/

See Also
This R function is a wrapper function that directly calls the the Redland RDF C libraries. For more information about Redland RDF, view the online documentation indicated in the 'References' section.

librdf_node_get_li_ordinal

Get the node li object ordinal value.

Description
Get the node li object ordinal value.

Usage
librdf_node_get_li_ordinal ( node, .copy )

Arguments
node the node object ("_p_librdf_node_s")
.copy NA
librdf_node_get_type

Value

integer

References

https://librdf.org/docs/

See Also

This R function is a wrapper function that directly calls the Redland RDF C libraries. For more information about Redland RDF, view the online documentation indicated in the 'References' section.

Usage

librdf_node_get_type ( node,
  .copy )

Arguments

node the node object ("_p_librdf_node_s")
  .copy NA

Value

integer

References

https://librdf.org/docs/

See Also

This R function is a wrapper function that directly calls the Redland RDF C libraries. For more information about Redland RDF, view the online documentation indicated in the 'References' section.
librdf_node_get_uri

Get the URI for a node object.

Description

Get the URI for a node object.

Usage

librdf_node_get_uri ( node )

Arguments

node the node object ("_p_librdf_node_s")

Value

_p_librdf_uri_s

References

https://librdf.org/docs/

See Also

This R function is a wrapper function that directly calls the the Redland RDF C libraries. For more information about Redland RDF, view the online documentation indicated in the 'References' section.

librdf_node_is_blank

Check node is a blank nodeID.

Description

Check node is a blank nodeID.

Usage

librdf_node_is_blank ( node, .copy )

Arguments

node the node object ("_p_librdf_node_s")
.copy NA
librdf_node_is_literal

Check node is a literal.

Description

Check node is a literal.

Usage

librdf_node_is_literal ( node, .copy )

Arguments

node the node object ("_p_librdf_node_s")
.copy NA

Value

integer

References

https://librdf.org/docs/

See Also

This R function is a wrapper function that directly calls the the Redland RDF C libraries. For more information about Redland RDF, view the online documentation indicated in the 'References' section.
librdf_node_is_resource

Check node is a resource.

Description
Check node is a resource.

Usage
librdf_node_is_resource ( node, .copy )

Arguments
node the node object ("_p_librdf_node_s")
.copy NA

Value
integer

References
https://librdf.org/docs/

See Also
This R function is a wrapper function that directly calls the the Redland RDF C libraries. For more information about Redland RDF, view the online documentation indicated in the 'References' section.

librdf_parser_check_name

Check if a parser name is known

Description
Check if a parser name is known

Usage
librdf_parser_check_name ( world, name, .copy )
librdf_parser_get_accept_header

Get an HTTP Accept value for the parser.

Description
Get an HTTP Accept value for the parser.

Usage
librdf_parser_get_accept_header ( parser )

Arguments
parser parser ("_p_librdf_parser_s")

Value
character

References
https://librdf.org/docs/

See Also
This R function is a wrapper function that directly calls the the Redland RDF C libraries. For more information about Redland RDF, view the online documentation indicated in the 'References' section.
librdf_parser_get_feature

Get the value of a parser feature.

Description

Get the value of a parser feature.

Usage

librdf_parser_get_feature ( parser,
feature )

Arguments

parser librdf_parser object ("_p_librdf_parser_s")
feature librdf_Uuri feature property ("_p_librdf_uri_s")

Value

_p_librdf_node_s

References

https://librdf.org/docs/

See Also

This R function is a wrapper function that directly calls the Redland RDF C libraries. For more information about Redland RDF, view the online documentation indicated in the 'References' section.

librdf_parser_get_namespaces_seen_count

Get the number of namespaces seen during parsing

Description

Get the number of namespaces seen during parsing

Usage

librdf_parser_get_namespaces_seen_count ( parser,
copy )
librdf_parser_get_namespaces_seen_prefix

Arguments

parser  librdf_parser object ("_p_librdf_parser_s")
.offset  index into list of namespaces ("integer")

Value

character

References

https://librdf.org/docs/

See Also

This R function is a wrapper function that directly calls the the Redland RDF C libraries. For more information about Redland RDF, view the online documentation indicated in the ‘References’ section.

librdf_parser_get_namespaces_seen_prefix

*Get the prefix of namespaces seen during parsing*

Description

Get the prefix of namespaces seen during parsing

Usage

librdf_parser_get_namespaces_seen_prefix ( parser,
.offset )

Arguments

parser  librdf_parser object ("_p_librdf_parser_s")
.offset  index into list of namespaces ("integer")

Value

character

References

https://librdf.org/docs/

See Also

This R function is a wrapper function that directly calls the the Redland RDF C libraries. For more information about Redland RDF, view the online documentation indicated in the ’References’ section.
librdf_parser_get_namespaces_seen_uri

Get the uri of namespaces seen during parsing

Description
Get the uri of namespaces seen during parsing

Usage
librdf_parser_get_namespaces_seen_uri ( parser, offset )

Arguments
parser
librdf_parser object ("_p_librdf_parser_s")
offset
index into list of namespaces ("integer")

Value
_p_librdf_uri_s

References
https://librdf.org/docs/

See Also
This R function is a wrapper function that directly calls the Redland RDF C libraries. For more information about Redland RDF, view the online documentation indicated in the 'References' section.

librdf_parser_guess_name2

Get a parser name for content with type or identifier

Description
Get a parser name for content with type or identifier

Usage
librdf_parser_guess_name2 ( world, mime_type, buffer, identifier )
librdf_parser_parse_as_stream

Parse a URI to a librdf_stream of statements.

Description
Parse a URI to a librdf_stream of statements.

Usage
librdf_parser_parse_as_stream ( parser, uri, inUriorNull )

Arguments
parser the parser (_p_librdf_parser_s)
uri the URI to read (_p_librdf_uri_s)
inUriorNull the base URI to use or NULL (_p_librdf_uri_s)

Value
_p_librdf_stream_s

References
https://librdf.org/docs/
librdf_parser_parse_counted_string_as_stream

Parse a counted string of content to a librdf_stream of statements.

Description

Parse a counted string of content to a librdf_stream of statements.

Usage

librdf_parser_parse_counted_string_as_stream ( parser, string, length, base_uri )

Arguments

parser the parser ("_p_librdf_parser_s")
string the string to parse ("character")
length length of the string content (must be >0) ("integer")
base_uri the base URI to use or NULL ("_p_librdf_uri_s")

Value

_p_librdf_stream_s

References

https://librdf.org/docs/

See Also

This R function is a wrapper function that directly calls the the Redland RDF C libraries. For more information about Redland RDF, view the online documentation indicated in the 'References' section.
librdf_parser_parse_counted_string_into_model

Parse a counted string of content into an librdf_model.

Description

Parse a counted string of content into an librdf_model.

Usage

    librdf_parser_parse_counted_string_into_model ( parser,  
        string,  
        length,  
        base_uri,  
        model,  
        .copy )

Arguments

    parser            the parser ("_p_librdf_parser_s")
    string            the content to parse ("character")
    length            length of content (must be >0) ("integer")
    base_uri          the base URI to use or NULL ("_p_librdf_uri_s")
    model             the model to use ("_p_librdf_model_s")
    .copy             NA

Value

    integer

References

    https://librdf.org/docs/

See Also

    This R function is a wrapper function that directly calls the the Redland RDF C libraries. For  
    more information about Redland RDF, view the online documentation indicated in the 'References'  
    section.
**librdf_parser_parse_into_model**

*Parse a URI of content into an librdf_model.*

**Description**

Parse a URI of content into an librdf_model.

**Usage**

```r
librdf_parser_parse_into_model ( parser, uri, inUriOrNull, model, .copy )
```

**Arguments**

- `parser`: the parser ("_p_librdf_parser_s")
- `uri`: the URI to read the content ("_p_librdf_uri_s")
- `inUriOrNull`: the base URI to use or NULL ("_p_librdf_uri_s")
- `model`: the model to use ("_p_librdf_model_s")
- `.copy`: NA

**Value**

integer

**References**

[https://librdf.org/docs/](https://librdf.org/docs/)

**See Also**

This R function is a wrapper function that directly calls the Redland RDF C libraries. For more information about Redland RDF, view the online documentation indicated in the 'References' section.
llibrdf_parser_parse_string_as_stream

Parse a string of content to a librdf_stream of statements.

**Description**

Parse a string of content to a librdf_stream of statements.

**Usage**

```c
llibrdf_parser_parse_string_as_stream ( parser, 
string, 
base_uri )
```

**Arguments**

- **parser**: the parser ("_p_librdf_parser_s")
- **string**: the string to parse ("character")
- **base_uri**: the base URI to use or NULL ("_p_librdf_uri_s")

**Value**

`_p_librdf_stream_s`

**References**

[https://librdf.org/docs/](https://librdf.org/docs/)

**See Also**

This R function is a wrapper function that directly calls the the Redland RDF C libraries. For more information about Redland RDF, view the online documentation indicated in the 'References' section.

---

llibrdf_parser_parse_string_into_model

Parse a string of content into an librdf_model.

**Description**

Parse a string of content into an librdf_model.

---
librdf_parser_set_feature

Usage

librdf_parser_parse_string_into_model ( parser, string, base_uri, model, .copy )

Arguments

parser     the parser ("_p_librdf_parser_s")
string     the content to parse ("character")
base_uri   the base URI to use or NULL ("_p_librdf_uri_s")
model      the model to use ("_p_librdf_model_s")
.copy      NA

Value

integer

References

https://librdf.org/docs/

See Also

This R function is a wrapper function that directly calls the Redland RDF C libraries. For more information about Redland RDF, view the online documentation indicated in the 'References' section.
librdf_query_execute

Arguments

- parser: librdf_parser object ("_p_librdf_parser_s")
- feature: librdf_uri feature property ("_p_librdf_uri_s")
- value: librdf_node feature property value ("_p_librdf_node_s")
- .copy: NA

Value

- integer

References

https://librdf.org/docs/

See Also

This R function is a wrapper function that directly calls the the Redland RDF C libraries. For more information about Redland RDF, view the online documentation indicated in the 'References' section.

---

librdf_query_execute Run the query on a model.

Description

Run the query on a model.

Usage

librdf_query_execute ( query, model )

Arguments

- query: librdf_query object ("_p_librdf_query")
- model: model to operate query on ("_p_librdf_model_s")

Value

- _p_librdf_query_results

References

https://librdf.org/docs/
See Also

This R function is a wrapper function that directly calls the Redland RDF C libraries. For more information about Redland RDF, view the online documentation indicated in the 'References' section.

---

**librdf_query_get_limit**

Get the query-specified limit on results.

### Description

Get the query-specified limit on results.

### Usage

```r
librdf_query_get_limit ( query,
 .copy )
```

### Arguments

- `query` : librdf_query query object ("_p_librdf_query")
- `.copy` : NA

### Value

integer

### References

[https://librdf.org/docs/](https://librdf.org/docs/)

### See Also

This R function is a wrapper function that directly calls the Redland RDF C libraries. For more information about Redland RDF, view the online documentation indicated in the 'References' section.
librdf_query_get_offset

*Get the query-specified offset on results.*

**Description**

Get the query-specified offset on results.

**Usage**

```r
librdf_query_get_offset ( query, .copy )
```

**Arguments**

- `query`: librdf_query query object ("_p_librdf_query")
- `.copy`: NA

**Value**

integer

**References**

[https://librdf.org/docs/](https://librdf.org/docs/)

**See Also**

This R function is a wrapper function that directly calls the Redland RDF C libraries. For more information about Redland RDF, view the online documentation indicated in the 'References' section.

---

librdf_query_results_as_stream

*Get a query result as an RDF graph in librdf_stream form*

**Description**

Get a query result as an RDF graph in librdf_stream form

**Usage**

```r
librdf_query_results_as_stream ( query_results )
```
Arguments

query_results  librdf_query_results query_results ("_p_librdf_query_results")

Value

_p_librdf_stream_s

References

https://librdf.org/docs/

See Also

This R function is a wrapper function that directly calls the the Redland RDF C libraries. For more information about Redland RDF, view the online documentation indicated in the 'References' section.
librdf_query_results_get_bindings_count

Get the number of bound variables in the result.

Description

Get the number of bound variables in the result.

Usage

librdf_query_results_get_bindings_count ( query_results,
 .copy )

Arguments

query_results  librdf_query_results query results ("_p_librdf_query_results")
.copy            NA

Value

integer

References

https://librdf.org/docs/

See Also

This R function is a wrapper function that directly calls the the Redland RDF C libraries. For more information about Redland RDF, view the online documentation indicated in the 'References' section.

librdf_query_results_get_binding_name

Get binding name for the current result.

Description

Get binding name for the current result.

Usage

librdf_query_results_get_binding_name ( query_results,
 offset )
Arguments

query_results  librdf_query_results query results ("_p_librdf_query_results")
offset          offset of binding name into array of known names ("integer")

Value

character

References

https://librdf.org/docs/

See Also

This R function is a wrapper function that directly calls the Redland RDF C libraries. For
more information about Redland RDF, view the online documentation indicated in the 'References'
section.

librdf_query_results_get_binding_value_get one binding value for the current result.

Description

Get one binding value for the current result.

Usage

librdf_query_results_get_binding_value ( query_results,
offset )

Arguments

query_results  librdf_query_results query results ("_p_librdf_query_results")
offset          offset of binding name into array of known names ("integer")

Value

_p_librdf_node_s

References

https://librdf.org/docs/

See Also

This R function is a wrapper function that directly calls the Redland RDF C libraries. For
more information about Redland RDF, view the online documentation indicated in the 'References'
section.
librdf_query_results_get_boolean

Get boolean query result.

Description

Get boolean query result.

Usage

    librdf_query_results_get_boolean ( query_results, .copy )

Arguments

    query_results  librdf_query_results query results (".p_librdf_query_results")
    name           variable name ("character")

Value

    _p_librdf_node_s

References

    https://librdf.org/docs/

See Also

    This R function is a wrapper function that directly calls the the Redland RDF C libraries. For more information about Redland RDF, view the online documentation indicated in the 'References' section.
librdf_query_results_get_count

Arguments

query_results  librdf_query_results query_results ("_p_librdf_query_results")
.copy       NA

Value

integer

References

https://librdf.org/docs/

See Also

This R function is a wrapper function that directly calls the the Redland RDF C libraries. For more information about Redland RDF, view the online documentation indicated in the ‘References’ section.

librdf_query_results_get_count

Get number of bindings so far.

Description

Get number of bindings so far.

Usage

librdf_query_results_get_count ( query_results, 
.copy )

Arguments

query_results  librdf_query_results query results ("_p_librdf_query_results")
.copy       NA

Value

integer

References

https://librdf.org/docs/

See Also

This R function is a wrapper function that directly calls the the Redland RDF C libraries. For more information about Redland RDF, view the online documentation indicated in the ‘References’ section.
librdf_query_results_is_bindings

Test if librdf_query_results is variable bindings format.

Description

Test if librdf_query_results is variable bindings format.

Usage

librdf_query_results_is_bindings ( query_results, .copy )

Arguments

query_results  librdf_query_results object("_p_librdf_query_results")
.copy  NA

Value

integer

References

https://librdf.org/docs/

See Also

This R function is a wrapper function that directly calls the the Redland RDF C libraries. For more information about Redland RDF, view the online documentation indicated in the 'References' section.

librdf_query_results_is_boolean

Test if librdf_query_results is boolean format.

Description

Test if librdf_query_results is boolean format.

Usage

librdf_query_results_is_boolean ( query_results, .copy )
librdf_query_results_is_graph

Arguments

query_results  librdf_query_results object ("_p_librdf_query_results")
.copy          NA

Value

text

References

https://librdf.org/docs/

See Also

This R function is a wrapper function that directly calls the Redland RDF C libraries. For more information about Redland RDF, view the online documentation indicated in the 'References' section.

---

librdf_query_results_is_graph

Test if librdf_query_results is RDF graph format.

Description

Test if librdf_query_results is RDF graph format.

Usage

librdf_query_results_is_graph ( query_results,
.copy )

Arguments

query_results  librdf_query_results object ("_p_librdf_query_results")
.copy          NA

Value

text

References

https://librdf.org/docs/

See Also

This R function is a wrapper function that directly calls the Redland RDF C libraries. For more information about Redland RDF, view the online documentation indicated in the 'References' section.
**librdf_query_results_is_syntax**

*Test if librdf_query_results is a syntax.*

**Description**

Test if librdf_query_results is a syntax.

**Usage**

```
librdf_query_results_is_syntax ( query_results, .copy )
```

**Arguments**

- `query_results` : librdf_query_results object ("_p_librdf_query_results")
- `.copy` : NA

**Value**

integer

**References**

[https://librdf.org/docs/](https://librdf.org/docs/)

**See Also**

This R function is a wrapper function that directly calls the the Redland RDF C libraries. For more information about Redland RDF, view the online documentation indicated in the 'References' section.

**librdf_query_results_next**

*Move to the next result.*

**Description**

Move to the next result.

**Usage**

```
librdf_query_results_next ( query_results, .copy )
```
librdf_query_results_to_file2

Write a query results to a file.

Description

Write a query results to a file.

Usage

librdf_query_results_to_file2 ( query_results,
name,
mime_type,
format_uri,
base_uri,
.copy )

Arguments

query_results librdf_query_results object ("_p_librdf_query_results")
name filename to write to ("character")
mime_type mime type (or NULL) ("character")
format_uri URI of syntax to format to (or NULL) ("_p_librdf_uri_s")
base_uri Base URI of output formatted syntax (or NULL) ("_p_librdf_uri_s")
.copy NA
librdf_query_results_to_string2

Turn a query results into a string.

Description

Turn a query results into a string.

Usage

librdf_query_results_to_string2 ( query_results, name, mime_type, format_uri, base_uri )

Arguments

query_results  librdf_query_results object ("_p_librdf_query_results")
name  format name ("character")
mime_type  format mime type (or NULL) ("character")
format_uri  URI of syntax to format to (or NULL) ("_p_librdf_uri_s")
base_uri  Base URI of output formatted syntax (or NULL) ("_p_librdf_uri_s")

Value

character

References

https://librdf.org/docs/
See Also

This R function is a wrapper function that directly calls the Redland RDF C libraries. For
more information about Redland RDF, view the online documentation indicated in the 'References'
section.

librdf_query_set_limit

Set the query-specified limit on results.

Description

Set the query-specified limit on results.

Usage

librdf_query_set_limit ( query,
limit,
.copy )

Arguments

query       librdf_query query object ("_p_librdf_query")
limit       the limit on results, >=0 to set a limit, <0 to have no limit ("integer")
.copy       NA

Value

integer

References

https://librdf.org/docs/

See Also

This R function is a wrapper function that directly calls the Redland RDF C libraries. For
more information about Redland RDF, view the online documentation indicated in the 'References'
section.
librdf_query_set_offset

Set the query-specified offset on results.

Description

Set the query-specified offset on results.

Usage

librdf_query_set_offset ( query,
  offset,
  .copy )

Arguments

query        librdf_query query object ("_p_librdf_query")
offset       offset for results, >=0 to set an offset, <0 to have no offset ("integer")
.copy        NA

Value

integer

References

https://librdf.org/docs/

See Also

This R function is a wrapper function that directly calls the Redland RDF C libraries. For more information about Redland RDF, view the online documentation indicated in the 'References' section.

librdf_serializer_check_name

Check if a serializer name is known

Description

Check if a serializer name is known
librdf_serializer_get_feature

Usage

    librdf_serializer_check_name ( world, name, .copy )

Arguments

    world      redland world object ("_p_librdf_world_s")
    name       name of serializer ("character")
    .copy      NA

Value

    integer

References

    https://librdf.org/docs/

See Also

    This R function is a wrapper function that directly calls the the Redland RDF C libraries. For more information about Redland RDF, view the online documentation indicated in the 'References' section.

__________________________________________________________________________

librdf_serializer_get_feature

    Get the value of a serializer feature.

__________________________________________________________________________

Description

    Get the value of a serializer feature.

Usage

    librdf_serializer_get_feature ( serializer, feature )

Arguments

    serializer    serializer object ("_p_librdf_serializer_s")
    feature       URI of feature ("_p_librdf_uri_s")

Value

    _p_librdf_node_s
librdf_serializer_serialize_model_to_file

Write a serialized librdf_model to a file.

Description
Write a serialized librdf_model to a file.

Usage
librdf_serializer_serialize_model_to_file ( serializer, name, inUriOrNull, model, .copy )

Arguments
- serializer: the serializer ("_p_librdf_serializer_s")
- name: filename to serialize to ("character")
- inUriOrNull: the base URI to use (or NULL) ("_p_librdf_uri_s")
- model: the librdf_model model to use ("_p_librdf_model_s")
- .copy: NA

Value
integer

References
https://librdf.org/docs/

See Also
This R function is a wrapper function that directly calls the the Redland RDF C libraries. For more information about Redland RDF, view the online documentation indicated in the 'References' section.
librdf_serializer_serialize_model_to_string
Write a serialized librdf_model to a string. The returned string must be freed by the caller using librdf_free_memory().

Description
Write a serialized librdf_model to a string. The returned string must be freed by the caller using librdf_free_memory().

Usage
librdf_serializer_serialize_model_to_string ( serializer,
inUriOrNull,
model )

Arguments
serializer       the serializer ("_p_librdf_serializer_s")
inUriOrNull        the base URI to use (or NULL) ("_p_librdf_uri_s")
model             the librdf_model model to use ("_p_librdf_model_s")

Value
character

References
https://librdf.org/docs/

See Also
This R function is a wrapper function that directly calls the the Redland RDF C libraries. For more information about Redland RDF, view the online documentation indicated in the ’References’ section.
librdf_serializer_serialize_stream_to_string

Usage

librdf_serializer_serialize_stream_to_file ( serializer,
       name,
       base_uri,
       stream,
       .copy )

Arguments

  serializer   the serializer ("_p_librdf_serializer_s")
  name         filename to serialize to ("character")
  base_uri     the base URI to use (or NULL) ("_p_librdf_uri_s")
  stream       the librdf_stream stream to use ("_p_librdf_stream_s")
  .copy        NA

Value

  integer

References

  https://librdf.org/docs/

See Also

  This R function is a wrapper function that directly calls the Redland RDF C libraries. For more information about Redland RDF, view the online documentation indicated in the 'References' section.

librdf_serializer_serialize_stream_to_string

Write a librdf_stream to a string.

Description

Write a librdf_stream to a string.

Usage

librdf_serializer_serialize_stream_to_string ( serializer,
       base_uri,
       stream )
Arguments

serializer the serializer ("_p_librdf_serializer_s")
base_uri the base URI to use (or NULL) ("_p_librdf_uri_s")
stream the librdf_stream stream to use ("_p_librdf_stream_s")

Value

character

References

https://librdf.org/docs/

See Also

This R function is a wrapper function that directly calls the the Redland RDF C libraries. For more information about Redland RDF, view the online documentation indicated in the 'References' section.

librdf_serializer_set_feature

Set the value of a serializer feature.

Description

Set the value of a serializer feature.

Usage

librdf_serializer_set_feature ( serializer, feature, value, .copy )

Arguments

serializer serializer object ("_p_librdf_serializer_s")
feature URI of feature ("_p_librdf_uri_s")
value value to set ("_p_librdf_node_s")
.copy NA

Value

integer
librdf_serialzer_set_namespace

Set a namespace URI/prefix mapping.

Description

Set a namespace URI/prefix mapping.

Usage

librdf_serializer_set_namespace ( serializer, nspace, prefix, .copy )

Arguments

  serializer    serializer object ("_p_librdf_serializer_s")
  nspace        URI of namespace or NULL ("_p_librdf_uri_s")
  prefix        prefix to use or NULL ("character")
  .copy         NA

Value

  integer

References

  https://librdf.org/docs/

See Also

  This R function is a wrapper function that directly calls the the Redland RDF C libraries. For more information about Redland RDF, view the online documentation indicated in the 'References' section.
librdf_short_copyright_string

Description

Short copyright string (one line).

Usage

librdf_short_copyright_string (.copy )

Arguments

.copy NA

Value

character

References

https://librdf.org/docs/

See Also

This R function is a wrapper function that directly calls the the Redland RDF C libraries. For more information about Redland RDF, view the online documentation indicated in the 'References' section.

librdf_short_copyright_string_get

Return Redland librdf copyright string

Description

Return Redland librdf copyright string

Usage

librdf_short_copyright_string_get( .copy )

Arguments

.copy logical
Check if two statements are equal.

Description
Check if two statements are equal.

Usage
librdf_statement_equals ( statement1, statement2, .copy )

Arguments
statement1  first librdf_statement("_p_librdf_statement_s")
statement2  second librdf_statement("_p_librdf_statement_s")
.copy      NA

Value
integer

References
https://librdf.org/docs/

See Also
This R function is a wrapper function that directly calls the Redland RDF C libraries. For more information about Redland RDF, view the online documentation indicated in the 'References' section.
librdf_statement_get_object

Get the statement object.

Description
Get the statement object.

Usage
librdf_statement_get_object ( statement )

Arguments
    statement  librdf_statement object ("_p_librdf_statement_s")

Value
_p_librdf_node_s

References
https://librdf.org/docs/

See Also
This R function is a wrapper function that directly calls the Redland RDF C libraries. For more information about Redland RDF, view the online documentation indicated in the 'References' section.

librdf_statement_get_predicate

Get the statement predicate.

Description
Get the statement predicate.

Usage
librdf_statement_get_predicate ( statement )

Arguments
    statement  librdf_statement object ("_p_librdf_statement_s")
Function: `librdf_statement_get_subject`

Description

Get the statement subject.

Usage

```r
librdf_statement_get_subject ( statement )
```

Arguments

- `statement`: `librdf_statement` object ("_p_librdf_statement_s")

Value

- `_p_librdf_node_s`

References

- [https://librdf.org/docs/](https://librdf.org/docs/)

See Also

- This R function is a wrapper function that directly calls the Redland RDF C libraries. For more information about Redland RDF, view the online documentation indicated in the 'References' section.
librdf_statement_is_complete

Description
Check if statement is a complete and legal RDF triple.

Usage
librdf_statement_is_complete ( statement, .copy )

Arguments
- statement: librdf_statement object ("_p_librdf_statement_s")
- .copy: NA

Value
integer

References
https://librdf.org/docs/

See Also
This R function is a wrapper function that directly calls the Redland RDF C libraries. For more information about Redland RDF, view the online documentation indicated in the 'References' section.

librdf_statement_match

Description
Match a statement against a 'partial' statement.

Usage
librdf_statement_match ( statement, partial_statement, .copy )
librdf_statement_set_object

Arguments

<table>
<thead>
<tr>
<th>argument</th>
<th>description</th>
</tr>
</thead>
<tbody>
<tr>
<td>statement</td>
<td>librdf_statement object (&quot;_p_librdf_statement_s&quot;)</td>
</tr>
<tr>
<td>partial_statement</td>
<td>statement with possible empty parts (&quot;_p_librdf_statement_s&quot;)</td>
</tr>
<tr>
<td>.copy</td>
<td>NA</td>
</tr>
</tbody>
</table>

Value

integer

References

https://librdf.org/docs/

See Also

This R function is a wrapper function that directly calls the Redland RDF C libraries. For more information about Redland RDF, view the online documentation indicated in the 'References' section.
**librdf_statement_set_predicate**

*Set the statement predicate.*

**Description**

Set the statement predicate.

**Usage**

```r
librdf_statement_set_predicate ( statement, predicate )
```

**Arguments**

- **statement**: librdf_statement object ("_p_librdf_statement_s")
- **predicate**: librdf_node of predicate ("_p_librdf_node_s")

**Value**

`void`

**References**

https://librdf.org/docs/

**See Also**

This R function is a wrapper function that directly calls the Redland RDF C libraries. For more information about Redland RDF, view the online documentation indicated in the 'References' section.
librdf_statement_set_subject

Set the statement subject.

Description

Set the statement subject.

Usage

librdf_statement_set_subject ( statement, subject )

Arguments

statement : librdf_statement object ("_p_librdf_statement_s")
subject : librdf_node of subject ("_p_librdf_node_s")

Value

void

References

https://librdf.org/docs/

See Also

This R function is a wrapper function that directly calls the Redland RDF C libraries. For more information about Redland RDF, view the online documentation indicated in the 'References' section.

librdf_stream_end

Test if the stream has ended.

Description

Test if the stream has ended.

Usage

librdf_stream_end ( stream, .copy )
librdf_stream_get_object

Arguments

stream  librdf_stream object ("_p_librdf_stream_s")
.copy    NA

Value

integer

References

https://librdf.org/docs/

See Also

This R function is a wrapper function that directly calls the Redland RDF C libraries. For more information about Redland RDF, view the online documentation indicated in the 'References' section.

Description

Get the current librdf_statement in the stream.

Usage

librdf_stream_get_object ( stream )

Arguments

stream  librdf_stream object ("_p_librdf_stream_s")

Value

_p_librdf_statement_s

References

https://librdf.org/docs/

See Also

This R function is a wrapper function that directly calls the Redland RDF C libraries. For more information about Redland RDF, view the online documentation indicated in the 'References' section.
librdf_stream_next  
Move to the next librdf_statement in the stream.

Description

Move to the next librdf_statement in the stream.

Usage

librdf_stream_next ( stream, 
.copy )

Arguments

stream  
librdf_stream object ("_p_librdf_stream_s")
.copy  
NA

Value

integer

References

https://librdf.org/docs/

See Also

This R function is a wrapper function that directly calls the the Redland RDF C libraries. For more information about Redland RDF, view the online documentation indicated in the 'References' section.

librdf_uri_compare  
Compare two librdf_uri objects lexicographically.

Description

Compare two librdf_uri objects lexicographically.

Usage

librdf_uri_compare ( first_uri, 
.second_uri, 
.copy )
librdf_uri_equals

Arguments

- `first_uri`  
  librdf_uri object 1 or NULL ("_p_librdf_uri_s")
- `second_uri`  
  librdf_uri object 2 or NULL ("_p_librdf_uri_s")
- `.copy`  
  NA

Value

integer

References

https://librdf.org/docs/

See Also

This R function is a wrapper function that directly calls the the Redland RDF C libraries. For more information about Redland RDF, view the online documentation indicated in the 'References' section.

`librdf_uri_equals`  
*Compare two librdf_uri objects for equality.*

Description

Compare two librdf_uri objects for equality.

Usage

`librdf_uri_equals ( first_uri, second_uri, .copy )`

Arguments

- `first_uri`  
  librdf_uri object 1 ("_p_librdf_uri_s")
- `second_uri`  
  librdf_uri object 2 ("_p_librdf_uri_s")
- `.copy`  
  NA

Value

integer

References

https://librdf.org/docs/
librdf_version_decimal

Library full version as a decimal integer.

Description

Library full version as a decimal integer.

Usage

librdf_version_decimal ( .copy )

librdf_uri_to_string

Format the URI as a string.

Description

Format the URI as a string.

Usage

librdf_uri_to_string ( uri )

Arguments

uri

librdf_uri object ("_p_librdf_uri_s")

Value

character

References

https://librdf.org/docs/

See Also

This R function is a wrapper function that directly calls the Redland RDF C libraries. For more information about Redland RDF, view the online documentation indicated in the 'References' section.
Arguments

.copy NA

Value

integer

References

https://librdf.org/docs/

See Also

This R function is a wrapper function that directly calls the the Redland RDF C libraries. For more information about Redland RDF, view the online documentation indicated in the ‘References’ section.
**librdf_version_major**  
*Library major version number as a decimal integer.*

**Description**  
Library major version number as a decimal integer.

**Usage**  
`librdf_version_major ( .copy )`

**Arguments**  
- `.copy`  
  *NA*

**Value**  
*integer*

**References**  
[https://librdf.org/docs/](https://librdf.org/docs/)

**See Also**  
This R function is a wrapper function that directly calls the Redland RDF C libraries. For more information about Redland RDF, view the online documentation indicated in the 'References' section.

---

**librdf_version_major_get**  
*Return the Redland librdf major version number*

**Description**  
Return the Redland librdf major version number

**Usage**  
`librdf_version_major_get ( .copy )`

**Arguments**  
- `.copy`  
  *logical*
**librdf_version_minor**

**Value**

integer

**References**

https://librdf.org/docs/

**See Also**

This R function is a wrapper function that directly calls the Redland RDF C libraries. For more information about Redland RDF, view the online documentation indicated in the 'References' section.

---

**librdf_version_minor  Library minor version number as a decimal integer.**

**Description**

Library minor version number as a decimal integer.

**Usage**

librdf_version_minor (.copy)

**Arguments**

.copy NA

**Value**

integer

**References**

https://librdf.org/docs/

**See Also**

This R function is a wrapper function that directly calls the Redland RDF C libraries. For more information about Redland RDF, view the online documentation indicated in the 'References' section.
librdf_version_minor_get

*Return the Redland librdf minor version number*

**Description**

Return the Redland librdf minor version number

**Usage**

`librdf_version_minor_get (.copy)`

**Arguments**

- `.copy` logical

**Value**

integer

**References**

https://librdf.org/docs/

**See Also**

This R function is a wrapper function that directly calls the Redland RDF C libraries. For more information about Redland RDF, view the online documentation indicated in the 'References' section.

---

librdf_version_release

*Library release version number as a decimal integer.*

**Description**

Library release version number as a decimal integer.

**Usage**

`librdf_version_release (.copy)`

**Arguments**

- `.copy` NA
librdf_version_release_get

Description

Return the Redland librdf release version number

Usage

librdf_version_release_get(.copy)

Arguments

.copy logical

Value

integer

References

https://librdf.org/docs/

See Also

This R function is a wrapper function that directly calls the Redland RDF C libraries. For more information about Redland RDF, view the online documentation indicated in the 'References' section.
librdf_version_string_get

Description
Library full version as a string.

Usage
librdf_version_string ( .copy )

Arguments
.copy NA

Value
character

References
https://librdf.org/docs/

See Also
This R function is a wrapper function that directly calls the Redland RDF C libraries. For more information about Redland RDF, view the online documentation indicated in the 'References' section.

librdf_version_string_get
Return the Redland librdf version as a string.

Description
Return the Redland librdf version as a string.

Usage
librdf_version_string_get ( .copy )

Arguments
.copy logical
librdf_world_get_feature

Value

classacter

References

https://librdf.org/docs/

See Also

This R function is a wrapper function that directly calls the Redland RDF C libraries. For more information about Redland RDF, view the online documentation indicated in the 'References' section.
librdf_world_set_feature

**Description**

Set the value of a world feature.

**Usage**

```r
librdf_world_set_feature ( world, feature, value, .copy )
```

**librdf_world_open**

*Open a created redland world environment.*

**Description**

Open a created redland world environment.

**Usage**

```r
librdf_world_open ( world )
```

**Arguments**

- `world` redland world object ("_p_librdf_world_s")

**Value**

`void`

**References**

https://librdf.org/docs/

**See Also**

This R function is a wrapper function that directly calls the the Redland RDF C libraries. For more information about Redland RDF, view the online documentation indicated in the 'References' section.
librdf_world_set_logger

Arguments

- world: librdf_world object ("_p_librdf_world_s")
- feature: librdf_uri feature property ("_p_librdf_uri_s")
- value: librdf_node feature property value ("_p_librdf_node_s")
- .copy: NA

Value

integer

References

https://librdf.org/docs/

See Also

This R function is a wrapper function that directly calls the Redland RDF C libraries. For more information about Redland RDF, view the online documentation indicated in the 'References' section.

---

librdf_world_set_logger

*Set the world log handling function.*

Description

Set the world log handling function.

Usage

```r
librdf_world_set_logger ( world, user_data, log_handler )
```

Arguments

- world: redland world object ("_p_librdf_world_s")
- user_data: user data to pass to function ("_p_void")
- log_handler: pointer to the function ("_p_librdf_log_func")

Value

void

References

https://librdf.org/docs/
mergeNamespace_roclet

A custom Roxygen roclet that adds Redland RDF functions to NAMESPACE file generated by Roxygen.

Description

The redland package uses the SWIG (Simplified Wrapper and Interface Generator) to create the bindings between the Redland RDF C/C++ libraries and R. SWIG creates a NAMESPACE file that contains the function names for the librdf wrapper that it creates, but as of swig 3.0.2 this NAMESPACE file is incorrect and will also be overwritten by Roxygen when 'roxygenize()' or 'devtools::document()' is called, as the wrapper R code doesn’t contain Roxygen export annotations used by Roxygen to build the namespace file. To allow for building a NAMESPACE file from all programs in the redland package, this roclet determines the set of wrapper R functions and adds these to the Roxygen generated NAMESPACE file that contains all names from the native R code in the redland package.

Usage

mergeNamespace_roclet(x, ...)

Arguments

x  a roclet
... additional parameters

Details

The following line must be present in the DESCRIPTION file for this roclet to be called automatically when 'roxygen2::roxygenize()' or 'devtools::document()' is called:

Roxygen: list(roclets = c("collate", "rd", "namespace", "mergeNamespace_roclet"))

The 'namespace' roclet must always run before the 'mergeNamespace' roclet.

Examples

## Not run:
roxygen2::roxygenize()
devtools::document()

## End(Not run)
Model-class

A Redland Model object

Description

A Model object is used to store the statements (triples) of an RDF model.

Details

A Model may be created manually by creating Statement and adding them to the Model using addStatement, or a Model may be read in from a previously saved file using parseFileIntoModel. Once a Model is created, it can be queried using Query.

Slots

librdf_model A redland model object

Methods

Model-initialize: Initialize a Model object
addStatement: Add a Statement object to the Model
freeModel: Free memory used by a librdf model object

See Also

View examples of creating models by viewing the 'redland_overview' vignette: 'vignette("redland_overview")'
redland: redland package

Examples

world <- new("World")
storage <- new("Storage", world, "hashes", name="", options="hash-type='memory'")
model <- new("Model", world, storage, options="")

Node-class

A Redland Node, used to store one node in an RDF triple statement.

Description

A Node represents a RDF Resource, Property, Literal or an RDF blank Node.

Slots

librdf_node A redland node object
Methods

**Node-initialize**: Initialize a Node object. **getNodeType**: Determine the node type and return as a string. **getNodeValue**: Determine the node type and return as a string. **getBlankNodeId**: Get the value of the node as a string.

See Also

redland: redland package

Examples

```r
world <- new("World")
# a blank node is created with a unique identifier generated by librdf
node <- new("Node", world)
# a blank node is created with a unique identifier generated by librdf
node <- new("Node", world, blank=NULL)
# a blank node is created with the user specified identifier, i.e. ":id1"
node <- new("Node", world, blank="someid")
# a node type of 'literal' is created
node <- new("Node", world, literal="A Node Value")
# a Node type of 'resource' is created
node <- new("Node", world, uri="http://www.example.com")
# Create a literal node, specifying a language encoding
node <- new("Node", world, literal="Gérard de La Martinière", language="fr")
```

---

**parseFileIntoModel**  
*Parse the contents of a file into a model*

Description

The contents of a the specified file are read and parsed into the initialized Parser object

Usage

```r
parseFileIntoModel(.Object, world, filePath, model, ...)
```

## S4 method for signature 'Parser,World,character,Model'

```r
parseFileIntoModel(.Object, world, filePath, model, baseUri = as.character(NA))
```

Arguments

- `.Object`  
a Parser object
- `world`  
a World object
- `filePath`  
a file that contains the RDF content
- `model`  
a Model object to parse the RDF content into
- `...`  
(Additional parameters)
- `baseUri`  
a base URI (i.e. XML base) to apply to the model
Details

The parser factory name specified during initialization determines how the content is parsed, for example, if 'rdfxml' was specified during parser initialization, then the parser expects RDF/XML content as specified in the W3C recommendation (http://www.w3.org/TR/REC-rdf-syntax)

Examples

```r
world <- new("World")
storage <- new("Storage", world, "hashes", name='', options="hash-type='memory'")
model <- new("Model", world, storage, options='')
# Create the default "rdfxml" parser
parser <- new("Parser", world)
filePath <- system.file("extdata/example.rdf", package="redland")
parseFileIntoModel(parser, world, filePath, model)
```

Parser-class

An RDF Parser object

Description

The Parser class provides methods to parse RDF content into a Redland RDF model.

Slots

librdf_parser A redland parser object

Methods

Parser-initialize: Initialize a Parser object. parseFileIntoModel: Parse the contents of a file into a model. freeParser: Free memory used by a librdf parser.

See Also

redland: redland package

Examples

```r
world <- new("World")
storage <- new("Storage", world, "hashes", name='', options="hash-type='memory'")
model <- new("Model", world, storage, options='')
# Create the default "rdfxml" parser
parser <- new("Parser", world)
filePath <- system.file("extdata/example.rdf", package="redland")
parseFileIntoModel(parser, world, filePath, model)
```
Description

The Query class is used to execute a query on a Model object using the default query language SPARQL. For more information, please refer to https://librdf.org/rasqal/ for details on supported query languages.

Details

A Query is executed using the executeQuery method, which returns a QueryResults object that can be iterated over the query solution sequence.

Slots

- librdf_query A redland query object
- librdf_world A redland world object

Methods

- Query Initialize: Initialize a Query object. executeQuery: Execute a query. setQueryResultLimit: Set limit on returned query results. getQueryResultLimit: Get the query result limit. getResults: Return all query results. writeResults: Write query results to a file. freeParser: Free memory used by a librdf query.

References

www.example.com

See Also

- redland: redland package

Examples

world <- new("World")
storage <- new("Storage", world, "hashes", name="", options="hash-type='memory'")
model <- new("Model", world, storage, options="")
stmt <- new("Statement", world=world,
  subject="https://cn.dataone.org/cn/v1/resolve/urn:uuid:274a0c5c-3082-4562-bbd3-2b1288768cac",
  predicate="http://www.w3.org/ns/prov#hadPlan",
  object="https://cn.dataone.org/cn/v1/resolve/urn:uuid:01305f45-f22b-40c8-8d27-00357d01e4a5")
ostatus <- addStatement(model, stmt)
stmt <- new("Statement", world=world,
  subject="https://orcid.org/0000-0002-2192-403X",
  predicate="http://www.w3.org/ns/prov#Agent",
  object="slaughter",
  objectType="literal",
...
status <- addStatement(model, stmt)
queryString <-
paste("PREFIX orcid: <https://orcid.org/>",
"PREFIX dataone: <https://cn.dataone.org/cn/v1/resolve/>",
"PREFIX prov: <http://www.w3.org/ns/prov#>",
"SELECT ?a ?c WHERE { ?a prov:Agent ?c . }", sep=" ")
query <- new("Query", world, queryString, base_uri=NULL, query_language="sparql", query_uri=NULL)
# Return all results as a string
results <- getResults(query, model, "rdfxml")

QueryResults-class  A Redland QueryResults object is used to inspect query results from a
Query object.

Description

The QueryResults object contains the RDF statements that were returned from a query on an RDF
model.

Slots

librdf_query_results  A redland query object

Methods

QueryResults-initialize: Initialize a QueryResults object. freeQueryResults: Free
memory used by a librdf query result.

See Also

redland: redland package

raptor_locator_byte  Get the locator byte offset from locator.

Description

Get the locator byte offset from locator

Usage

raptor_locator_byte ( locator, .copy )
raptor_locator_column

Arguments

locator raptor locator ("_p_raptor_locator")
.copy logical

Value

character

References

https://librdf.org/docs/

See Also

This R function is a wrapper function that directly calls the the Redland RDF C libraries. For more information about Redland RDF, view the online documentation indicated in the 'References' section.

raptor_locator_column  Get column number from locator

Description

Get column number from locator

Usage

raptor_locator_column ( locator, .copy )

Arguments

locator raptor locator ("_p_raptor_locator")
.copy logical

Value

integer

References

https://librdf.org/docs/

See Also

This R function is a wrapper function that directly calls the the Redland RDF C libraries. For more information about Redland RDF, view the online documentation indicated in the 'References' section.
raptor_locator_file  Get file name from locator.

Description
Get file name from locator.

Usage
raptor_locator_file ( locator )

Arguments
locator  raptor locator ("_p_raptor_locator")

Value
character

References
https://librdf.org/docs/

See Also
This R function is a wrapper function that directly calls the the Redland RDF C libraries. For more information about Redland RDF, view the online documentation indicated in the 'References' section.

raptor_locator_line  Get line number from locator.

Description
Get line number from locator.

Usage
raptor_locator_line ( locator, .copy )

Arguments
locator  raptor locator ("_p_raptor_locator")
.copy  logical
**raptor_locator_uri**

**Value**

integer

**References**

https://librdf.org/docs/

**See Also**

This R function is a wrapper function that directly calls the the Redland RDF C libraries. For more information about Redland RDF, view the online documentation indicated in the 'References' section.

---

**raptor_locator_uri**  Get URI from locator.

**Description**

Get URI from locator.

**Usage**

raptor_locator_uri ( locator )

**Arguments**

<table>
<thead>
<tr>
<th>Argument</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>locator</td>
<td>raptor locator (&quot;_p_raptor_locator&quot;)</td>
</tr>
</tbody>
</table>

**Value**

character

**References**

https://librdf.org/docs/

**See Also**

This R function is a wrapper function that directly calls the the Redland RDF C libraries. For more information about Redland RDF, view the online documentation indicated in the 'References' section.
**raptor_version_decimal**

*Raptor version as a decimal number*

### Description

Raptor version as a decimal number

### Usage

```
raptor_version_decimal ( .copy )
```

### Arguments

- `.copy` logical

### Value

integer

### References

[https://librdf.org/docs/](https://librdf.org/docs/)

### See Also

This R function is a wrapper function that directly calls the Redland RDF C libraries. For more information about Redland RDF, view the online documentation indicated in the 'References' section.

---

**raptor_version_decimal_get**

*Raptor version as a decimal number.*

### Description

Raptor version as a decimal number.

### Usage

```
raptor_version_decimal_get ( .copy )
```

### Arguments

- `.copy` logical
raptor_version_major

Value
integer

References
https://librdf.org/docs/

See Also
This R function is a wrapper function that directly calls the the Redland RDF C libraries. For more information about Redland RDF, view the online documentation indicated in the 'References' section.

raptor_version_major  Raptor library major version

Description
Raptor library major version.

Usage
raptor_version_major ( .copy )

Arguments
.copy     logical

Value
integer

References
https://librdf.org/docs/

See Also
This R function is a wrapper function that directly calls the the Redland RDF C libraries. For more information about Redland RDF, view the online documentation indicated in the 'References' section.
raptor_version_major_get

Get Raptor library major version

Description
Get Raptor library major version.

Usage
raptor_version_major_get (.copy)

Arguments
.copy logical

Value
integer

References
https://librdf.org/docs/

See Also
This R function is a wrapper function that directly calls the the Redland RDF C libraries. For more information about Redland RDF, view the online documentation indicated in the ‘References’ section.

raptor_version_minor

Raptor library minor version.

Description
Raptor library minor version.

Usage
raptor_version_minor (.copy)

Arguments
.copy logical
raptor_version_minor_get

Get Raptor library minor version.

Description
Get Raptor library minor version.

Usage
raptor_version_minor_get (.copy)

Arguments
.copy logical

Value
integer

References
https://librdf.org/docs/

See Also
This R function is a wrapper function that directly calls the Redland RDF C libraries. For more information about Redland RDF, view the online documentation indicated in the 'References' section.
### raptor_version_release

**Description**

Raptor library release.

**Usage**

```r
raptor_version_release(.copy)
```

**Arguments**

- `.copy` logical

**Value**

integer

**References**

https://librdf.org/docs/

**See Also**

This R function is a wrapper function that directly calls the Redland RDF C libraries. For more information about Redland RDF, view the online documentation indicated in the 'References' section.

### raptor_version_release_get

**Description**

Get Raptor library release.

**Usage**

```r
raptor_version_release_get(.copy)
```

**Arguments**

- `.copy` logical
raptor_version_string

Value

integer

References

https://librdf.org/docs/

See Also

This R function is a wrapper function that directly calls the Redland RDF C libraries. For more information about Redland RDF, view the online documentation indicated in the 'References' section.

raptor_version_string  Raptor library version string.

Description

Raptor library version string.

Usage

raptor_version_string ( .copy )

Arguments

.copy logical

Value

character

References

https://librdf.org/docs/

See Also

This R function is a wrapper function that directly calls the Redland RDF C libraries. For more information about Redland RDF, view the online documentation indicated in the 'References' section.
raptor_version_string_get

Get Raptor library version string.

**Description**

Get Raptor library version string.

**Usage**

```
raptor_version_string_get ( .copy )
```

**Arguments**

- `.copy` logical

**Value**

character

**References**

[https://librdf.org/docs/](https://librdf.org/docs/)

**See Also**

This R function is a wrapper function that directly calls the Redland RDF C libraries. For more information about Redland RDF, view the online documentation indicated in the 'References' section.

---

rasqal_version_decimal

*Rasqal version as a decimal number.*

**Description**

Rasqal version as a decimal number.

**Usage**

```
rasqal_version_decimal ( .copy )
```

**Arguments**

- `.copy` logical
rasqal_version_decimal_get

Description
Get the Rasqal version as a decimal number.

Usage
rasqal_version_decimal_get(.copy)

Arguments
.copy logical

Value
integer

References
https://librdf.org/docs/

See Also
This R function is a wrapper function that directly calls the the Redland RDF C libraries. For more information about Redland RDF, view the online documentation indicated in the 'References' section.
**rasqal_version_major**  
*Rasqal major version number.*

---

**Description**

Rasqal major version number.

**Usage**

```
rasqal_version_major ( .copy )
```

**Arguments**

| .copy     | logical |

**Value**

integer

**References**

https://librdf.org/docs/

**See Also**

This R function is a wrapper function that directly calls the Redland RDF C libraries. For more information about Redland RDF, view the online documentation indicated in the 'References' section.

---

**rasqal_version_major_get**

*Get Rasqal major version number.*

---

**Description**

Get Rasqal major version number.

**Usage**

```
rasqal_version_major_get ( .copy )
```

**Arguments**

| .copy     | logical |

---

Value
integer

References
https://librdf.org/docs/

See Also
This R function is a wrapper function that directly calls the the Redland RDF C libraries. For more information about Redland RDF, view the online documentation indicated in the 'References' section.

---

rasqal_version_minor  Rasqal minor version number.

Description
Rasqal minor version number.

Usage

rasqal_version_minor ( .copy )

Arguments

.copy  logical

Value
integer

References
https://librdf.org/docs/

See Also
This R function is a wrapper function that directly calls the the Redland RDF C libraries. For more information about Redland RDF, view the online documentation indicated in the 'References' section.
rasqal_version_minor_get

Get the Rasqal minor version number.

Description
Get the Rasqal minor version number.

Usage

rasqal_version_minor_get ( .copy )

Arguments

.copy logical

Value
integer

References

https://librdf.org/docs/

See Also
This R function is a wrapper function that directly calls the Redland RDF C libraries. For more information about Redland RDF, view the online documentation indicated in the 'References' section.

rasqal_version_release

Rasqal release version number.

Description
Rasqal release version number.

Usage

rasqal_version_release ( .copy )

Arguments

.copy logical
Value
integer

References
https://librdf.org/docs/

See Also
This R function is a wrapper function that directly calls the the Redland RDF C libraries. For more information about Redland RDF, view the online documentation indicated in the 'References' section.

rasqal_version_release_get

Description
Get the Rasqal release version number.

Usage
rasqal_version_release_get (.copy )

Arguments
.copy logical

Value
integer

References
https://librdf.org/docs/

See Also
This R function is a wrapper function that directly calls the the Redland RDF C libraries. For more information about Redland RDF, view the online documentation indicated in the 'References' section.
**rasqal_version_string**  
*Rasqal version as a string*

### Description
Rasqal version as a string.

### Usage
```r
rasqal_version_string (.copy)
```

### Arguments
- `.copy` logical

### Value
integer

### References
[https://librdf.org/docs/](https://librdf.org/docs/)

### See Also
This R function is a wrapper function that directly calls the Redland RDF C libraries. For more information about Redland RDF, view the online documentation indicated in the 'References' section.

---

**rasqal_version_string_get**  
*Get the Rasqal version as a string*

### Description
Get the Rasqal version as a string.

### Usage
```r
rasqal_version_string_get (.copy)
```

### Arguments
- `.copy` logical
redland

Value

integer

References

https://librdf.org/docs/

See Also

This R function is a wrapper function that directly calls the the Redland RDF C libraries. For more information about Redland RDF, view the online documentation indicated in the 'References' section.

redland

Create, query and write RDF graphs.

Description

The R package redland provides methods to create, query and write information stored in the Resource Description Framework (RDF). This package is implemented as R scripts that provide an R interface (aka "wrapper") to the Redland RDF C libraries. Documentation for the redland R package classes and functions are available from the standard R help facility, for example, 'help("Node-class")', '?getNodeType', etc.

An overview of the redland R package is available with the R command: 'vignette("redland_overview")'.

The Redland C library functions are described at https://librdf.org/docs/api/index.html.

An introduction to RDF can be found at https://www.w3.org/TR/rdf-primer/.

Details

The redland R package classes and the corresponding Redland C library types are shown in the following table:

<table>
<thead>
<tr>
<th>Concept</th>
<th>Redland C type</th>
<th>redland R class</th>
<th>Purpose</th>
</tr>
</thead>
<tbody>
<tr>
<td>Resource / Literal</td>
<td>librdf_node</td>
<td>Node</td>
<td>RDF Model &amp; Syntax nodes</td>
</tr>
<tr>
<td>Statement / Triple</td>
<td>librdf_statement</td>
<td>Statement</td>
<td>RDF Model &amp; Syntax arcs (statements, triples)</td>
</tr>
<tr>
<td>Model</td>
<td>librdf_model</td>
<td>Model</td>
<td>Set of Statements usually held in one Storage</td>
</tr>
<tr>
<td>Node</td>
<td>librdf_node</td>
<td>Node</td>
<td>The subject, predicate or object of a Statement</td>
</tr>
<tr>
<td>Storage</td>
<td>librdf_storage</td>
<td>Storage</td>
<td>Storage for Models either persistent or in-memory.</td>
</tr>
<tr>
<td>Parser</td>
<td>librdf_parser</td>
<td>Parser</td>
<td>Syntax parsers delivering Stream of Statements or writing to a Query of an Model delivering a QueryResults</td>
</tr>
<tr>
<td>Query</td>
<td>librdf_query</td>
<td>Query</td>
<td>Results of applying an Query to a Model giving either variable bindings or a Stream of Statements</td>
</tr>
<tr>
<td>QueryResults</td>
<td>librdf_query_results</td>
<td>QueryResults</td>
<td>Serializes a Model into a syntax such as RDF/XML</td>
</tr>
<tr>
<td>Serializer</td>
<td>librdf_serializer</td>
<td>Serializer</td>
<td>RDF wrapper class handling Redland startup/shutdown</td>
</tr>
<tr>
<td>World</td>
<td>librdf_world</td>
<td>World</td>
<td>RDF wrapper class handling Redland startup/shutdown</td>
</tr>
</tbody>
</table>
Note

In order to communicate with the Redland RDF C libraries, the redland R package uses an inter-
face layer that is created with the software package Simplified Wrapper and Interface Generator
(SWIG). The relationship between the redland R package and the Redland C libraries is:

User script -> redland R package -> SWIG R interface -> Redland C libraries -> RDF data

It is recommended that the redland package R classes be used to interact with RDF, as these higher
level classes take care of many of the the details of communicating with the Redland C libraries.
However, all of the lower level R interface functions generated by SWIG are made available by
the redland package. These interface functions usually have names beginning with 'librdf_,'
'raskal_' or 'raptor_' and are usually the same name as the underlying C library function. Doc-
umentation for the R SWIG interface functions can be found via R help i.e. '?librdf_iterator'.

Author(s)

Matthew B. Jones (NCEAS) and Peter Slaughter (NCEAS)

Examples

```r
# This example creates the necessary R objects to hold an RDF model and reads
# a file that contains RDF/XML statements. This model is then queried for
# and the query results inspected.
world <- new("World")
storage <- new("Storage", world, "hashes", name="", options="hash-type='memory'")
model <- new("Model", world, storage, options="")
filePath <- system.file("extdata/example.rdf", package="redland")
parser <- new("Parser", world)
parseFileIntoModel(parser, world, filePath, model)
queryString <- paste("PREFIX dc: <http://purl.org/dc/elements/1.1/> ",
"SELECT ?a ?c WHERE { ?a dc:description ?c . }", sep="")
query <- new("Query", world, queryString, base_uri=NULL,
query_language="sparql", query_uri=NULL)
results <- getResults(query, model, "rdfxml")

# When the query object is no longer needed, the resources it had allocated can be freed.
freeQuery(query)
rm(query)
```

Description

The `roclet_output` function handles output of the results from the `roc_process` function. This
function merges the NAMESPACE file created by the `namespace` roclet with the list of Redland
RDF functions determined by the `roc_process` function.
Usage

## S3 method for class 'roclet_mergeNamespace'
roclet_process(x, blocks, env, base_path)

Arguments

x the currently running roclet
blocks the documentation blocks
event the current env
base_path the top directory of the R package

Details

This function loads the Redland interface file and tests each loaded function to see if it should be exported via the NAMESPACE file.
Serializer-class

An RDF Serializer object.

Description

The Serializer class provides methods to convert a Model object to other forms, for example, write out a Model to a file.

Slots

librdf_serializer A redland statement object

Methods

Serializer-initialize: Initialize a Serializer object. setNameSpace: Set a namespace for the serializer. serializeToCharacter: Serialize a model to a character vector. serializeToFile: Serialize a model to a file. freeSerializer: Free memory used by a librdf serializer.

See Also

redland: redland package

Examples

```r
world <- new("World")
storage <- new("Storage", world, "hashes", name="", options="hash-type='memory'")
model <- new("Model", world, storage, options="")
filePath <- system.file("extdata/example.rdf", package="redland")
parser <- new("Parser", world)
parseFileIntoModel(parser, world, filePath, model)
# Create the default "rdfxml" serializer
serializer <- new("Serializer", world)
# Add a namespace definition to the serializer
status <- setNameSpace(serializer, world, namespace="http://purl.org/dc/elements/1.1/", prefix="dc")
rdf <- serializeToCharacter(serializer, world, model, baseUri="")
```

serializeToCharacter

Serialize a model to a character vector.

Description

Serialize a model to a character vector.

Usage

```r
serializeToCharacter(.Object, world, model, ...)  
```
Argumentsn

.Object. a Serializer object
world a World object
model a Model object
... Additional parameters
baseUri a URI to prepend to relative URIs in the document

Valuen

a character vector containing the serialized model

---

`serializeToFile` *Serialize a model to a file.*

**Description**

Serialize a model to a file.

**Usage**

```r
serializeToFile(.Object, world, model, filePath, ...)  
## S4 method for signature 'Serializer,World,Model,character'
serializeToFile(.Object, world, model, filePath, baseUri = as.character(NA))
```

**Arguments**

.Object. a Serializer object
world a World object
model a Model object
filePath a file path that the serialized model will be written to
... Additional parameters
baseUri a base URI to use for the serialization

**Value**

an integer containing the return status where non zero indicates an error occurred during serialization
**setNameSpace**

*Set a namespace for the serializer.*

**Description**

Set a namespace for the serializer.

**Usage**

```
setNameSpace(.Object, world, namespace, prefix)
```

```
## S4 method for signature 'Serializer,World,character,character'
setNameSpace(.Object, world, namespace, prefix)
```

**Arguments**

- `.Object`: a Serializer object
- `world`: a World object
- `namespace`: the namespace to add to the serializer
- `prefix`: the namespace prefix to associate with the namespace

**setQueryResultLimit**

*Set limit on returned query results*

**Description**

Set limit on returned query results.

**Usage**

```
setQueryResultLimit(.Object, limit)
```

```
## S4 method for signature 'Query'
setQueryResultLimit(.Object, limit)
```

**Arguments**

- `.Object`: a Query object
- `limit`: the result set limit. Specify a value \(\geq\) to have a limit, or a value \(< 0\) to have no limit.
Description

A Statement object is created using the provided subject, predicate and object.

Details

A Statement object can be created from Node objects that are provided for the subject, predicate and object. An alternative way to create a Statement object is to provide the subject, predicate and object as character values. If this later method is used, the character values will be evaluated to determine the appropriate RDF type for the subject and object. Note that the RDF type for the predicate will always be 'uri' (aka 'resource'). If the automatic determination of RDF types is not desired, then the subjectType and objectType parameters can be specified to explicitly set the RDF types.

Slots

librdf_statement A redland statement object

Methods

Statement-initialize: Initialize a Statement object. getTermType: Return the redland node type for the specified RDF term in a statement. freeStatement: Free memory used by a librdf statement.

See Also

redland: redland package

Examples

```r
world <- new("World")
# Create nodes manually and add to the statement
subject <- new("Node", blank="_:myid1", world)
object <- new("Node", literal="thing", world)
stmt <- new("Statement", world, subject, predicate, object)

# Create the statement specifying node values directly
stmt <- new("Statement", world, subject="http://www.example.com/myevent",
predicate="http://example.com/occurredAt",
object="Tue Feb 17 14:05:13 PST 2015")
stmt <- new("Statement", world, subject=NULL,
predicate="http://www.example.com/hasAddr",
object="http://www.nothing.com", objectType="literal")
stmt <- new("Statement", world, subject="http://www.example.com/BobSmith",
predicate="http://www.example.com/says",
```
Storage-class

A Redland Storage object

Description

A Redland Storage object

Slots

librdf_storage A redland storage object
type the storage type to create, i.e. "hashes", "mysql", "postgresql", ...

Methods

Storage-initialize: Initialize a Storage object freeStorage: Free memory used by a librdf storage object

See Also

redland: redland package

Examples

world <- new("World")
storage <- new("Storage", world, "hashes", name="", options="hash-type='memory'")

World-class

A Redland World object, used to initialize the Redland RDF library.

Description

A World object is the top level object in the Redland RDF library implementation, so it contains all other objects needed to process RDF Models.

Slots

librdf_world A redland world object

Methods

World-initialize: Initialize a World object freeWorld: Free memory used by a librdf world object
Write query results to a file.

## S4 method for signature 'Query'
writeResults(
  .Object,
  model,
  file,
  mimeType = "application/x-turtle",
  format_uri = NULL,
  base_uri = NULL
)

### Arguments

- `.Object` a Query object
- `model` a Model object
- `...` additional parameters
- `file` a string specifying the output file
- `mimeType` a string specifying the mimeType of the output file. Currently supported values are "application/x-turtle", "text/plain", "application/json", "text/html"
- `format_uri` (not currently used)
- `base_uri` (not currently used)

### Details

After this method is called, the Query object is no longer usable and should be deleted "rm(query)" and a new object created.
Examples

```r
world <- new("World")
storage <- new("Storage", world, "hashes", name="", options="hash-type='memory'")
model <- new("Model", world, storage, options="")
stmt <- new("Statement", world=world,
            subject="https://orcid.org/0000-0002-2192-403X",
            predicate="http://www.w3.org/ns/prov#Agent",
            object="slaughter",
            objectType="literal", datatype_uri="http://www.w3.org/2001/XMLSchema#string")
status <- addStatement(model, stmt)
queryString <- paste("PREFIX orcid: <https://orcid.org/>",
                     "PREFIX dataone: <https://cn.dataone.org/cn/v1/resolve/>",
                     "PREFIX prov: <http://www.w3.org/ns/prov#>",
                     "SELECT ?a ?c WHERE { ?a prov:Agent ?c . }", sep=" ")
query <- new("Query", world, queryString, base_uri=NULL, query_language="sparql", query_uri=NULL)
# Return all results as a string
tf <- tempfile()
writeResults(query, model, file=tf, mimeType="application/x-turtle")
# When the query object is no longer needed, the resources it had allocated can be freed.
freeQuery(query)
rm(query)
```

### [, method

Subset a list of ExternalReferences

#### Description

Subset a list of ExternalReferences

#### Usage

```r
## S4 method for signature 'ExternalReference'
x[i, j, ... , drop = TRUE]
```

#### Arguments

- `x` a list of ExternalReferences
- `i` row subscript
- `j` column subscript
- `...` additional arguments
- `drop` a logical
Description

Assign values in a list of ExternalReferences

Usage

```r
# S4 replacement method for signature 'ExternalReference'
x[i, j, ...] <- value
```

Arguments

- `x` : a list of ExternalReferences
- `i` : row subscript
- `j` : column subscript
- `...` : additional arguments
- `value` : a value to assign
Index

* classes
  Model-class, 161
  Node-class, 161
  Parser-class, 163
  Query-class, 164
  QueryResults-class, 165
  Serializer-class, 185
  Statement-class, 188
  Storage-class, 189
  World-class, 189

[] ExternalReference-method, 191
[<-, ExternalReference-method, 192

addStatement, 7, 161
addStatement, Model, Statement-method (addStatement), 7

executeQuery, 8, 164
executeQuery, Query-method (executeQuery), 8

freeModel, 8, 161
freeModel, Model-method (freeModel), 8
freeParser, 9, 163, 164
freeParser, Parser-method (freeParser), 9
freeQuery, 10
freeQuery, Query-method (freeQuery), 10
freeQueryResults, 11, 165
freeQueryResults, QueryResults-method (freeQueryResults), 11
freeSerializer, 11, 185
freeSerializer, Serializer-method (freeSerializer), 11
freeStatement, 12, 188
freeStatement, Statement-method (freeStatement), 12
freeStorage, 13, 189
freeStorage, Storage-method (freeStorage), 13
freeWorld, 14, 189

freeWorld, World-method (freeWorld), 14
getBlankNodeId, 14, 162
getBlankNodeId, Node-method (getBlankNodeId), 14
getNodeType, 15, 162
getNodeType, Node-method (getNodeType), 15
getNodeValue, 16, 162
getNodeValue, Node-method (getNodeValue), 16
getQueryResultLimit, 16, 164
getQueryResultLimit, Query-method (getQueryResultLimit), 16
getResults, 17, 164
getResults, Query-method (getResults), 17
getTermType, 18, 188
getTermType, Statement, character-method (getTermType), 18
initialize, Model-method, 19
initialize, Node-method, 19
initialize, Parser-method, 20
initialize, Query-method, 21
initialize, QueryResults-method, 22
initialize, Serializer-method, 22
initialize, Storage-method, 24
initialize, World-method, 25
is.null.externalptr, 25
length, SWIGArray-method, 26
librdf_copyright_string, 26
librdf_copyright_string_get, 27
librdf_digest_final, 27
librdf_digest_init, 28
librdf_digest_to_string, 29
librdf_digest_update, 29
librdf_digest_update_string, 30
librdf_free_digest, 31
librdf_free_hash, 31
librdf_free_iterator, 32
librdf_free_model, 33
librdf_free_node, 33
librdf_free_parser, 34
librdf_free_query, 35
librdf_free_query_results, 35
librdf_free_serializer, 36
librdf_free_statement, 37
librdf_free_storage, 37
librdf_free_stream, 38
librdf_free_uri, 39
librdf_free_world, 39
librdf_hash_to_string, 40
librdf_internal_test_error, 41
librdf_internal_test_warning, 41
librdf_iterator_end, 42
librdf_iterator_get_context, 43
librdf_iterator_get_object, 43
librdf_iterator_next, 44
librdf_log_message_code, 45
librdf_log_message_facility, 45
librdf_log_message_level, 46
librdf_log_message_locator, 47
librdf_log_message_message, 47
librdf_model_add, 48
librdf_model_add_statement, 49
librdf_model_add_statements, 50
librdf_model_add_string_literal_statement, 50
librdf_model_add_typed_literal_statement, 51
librdf_model_as_stream, 52
librdf_model_contains_context, 53
librdf_model_contains_statement, 54
librdf_model_context_add_statement, 55
librdf_model_context_add_statements, 56
librdf_model_context_as_stream, 57
librdf_model_context_remove_statement, 57
librdf_model_context_remove_statements, 58
librdf_model_find_statements, 59
librdf_model_find_statements_in_context, 60
librdf_model_get_arc, 60
librdf_model_get_arcs, 61
librdf_model_get_arcs_in, 62
librdf_model_get_arcs_out, 63
librdf_model_get_contexts, 63
librdf_model_get_feature, 64
librdf_model_get_source, 65
librdf_model_get_sources, 65
librdf_model_get_target, 66
librdf_model_get_targets, 67
librdf_model_has_arc_in, 68
librdf_model_has_arc_out, 69
librdf_model_load, 70
librdf_model_query_execute, 71
librdf_model_remove_statement, 71
librdf_model_set_feature, 72
librdf_model_size, 73
librdf_model_sync, 74
librdf_model_to_string, 74
librdf_model_transaction_commit, 75
librdf_model_transaction_rollback, 76
librdf_model_transaction_start, 77
librdf_new_digest, 77
librdf_new_hash, 78
librdf_new_hash_from_array_of_strings, 79
librdf_new_hash_from_string, 79
librdf_new_model, 80
librdf_new_model_from_model, 81
librdf_new_model_with_options, 82
librdf_new_node, 82
librdf_new_node_from_blank_identifier, 83
librdf_new_node_from_literal, 84
librdf_new_node_from_node, 85
librdf_new_node_from_normalised_uri_string, 85
librdf_new_node_from_typed_literal, 86
librdf_new_node_from_uri, 87
librdf_new_node_from_uri_local_name, 88
librdf_new_node_from_uri_string, 88
librdf_new_options, 89
librdf_new_parser, 90
librdf_new_query, 90
librdf_new_query_from_query, 91
librdf_new_serializer, 91
librdf_new_statement, 92
librdf_new_statement_from_nodes, 93
librdf_new_statement_from_statement, 94
librdf_new_storage, 94
librdf_new_storage_from_storage, 95
librdf_new_uri, 96
librdf_new_uri_from_filename, 97
librdf_new_uri_from_uri, 97
librdf_new_world, 98
librdf_node_equals, 99
librdf_node_get_blank_identifier, 99
librdf_node_get_li_ordinal, 103
librdf_node_get_literal_value, 100
librdf_node_get_literal_value_as_latin1, 101
librdf_node_get_literal_value_datatype_uri, 101
librdf_node_get_literal_value_is_wf_xml, 102
librdf_node_get_literal_value_language, 103
librdf_node_get_type, 104
librdf_node_get_uri, 105
librdf_node_is_blank, 105
librdf_node_is_literal, 106
librdf_node_is_resource, 107
librdf_node_is_resource_name, 107
librdf_parser_check_name, 107
librdf_parser_get_accept_header, 108
librdf_parser_get_feature, 109
librdf_parser_get_namespaces_seen_count, 109
librdf_parser_get_namespaces_seen_prefix, 109
librdf_parser_get_namespaces_seen_uri, 110
librdf_parser_guess_name2, 111
librdf_parser_parse_as_stream, 112
librdf_parser_parse_counted_string_as_stream, 113
librdf_parser_parse_counted_string_into_model, 114
librdf_parser_parse_into_model, 115
librdf_parser_parse_string_as_stream, 116
librdf_parser_parse_string_into_model, 116
librdf_parser_set_feature, 117
librdf_query_execute, 118
librdf_query_get_limit, 119
librdf_query_get_offset, 120
librdf_query_results_as_stream, 120
librdf_query_results_finished, 121
librdf_query_results_get_binding_name, 122
librdf_query_results_get_binding_value, 123
librdf_query_results_get_binding_value_by_name, 124
librdf_query_results_get_bindings_count, 122
librdf_query_results_get_boolean, 124
librdf_query_results_get_count, 125
librdf_query_results_is_bindings, 126
librdf_query_results_is_boolean, 126
librdf_query_results_is_graph, 127
librdf_query_results_is_syntax, 128
librdf_query_results_next, 128
librdf_query_results_to_file2, 129
librdf_query_results_to_string2, 130
librdf_query_set_limit, 131
librdf_query_set_offset, 132
librdf_serializer_check_name, 132
librdf_serializer_get_feature, 133
librdf_serializer_serialize_model_to_file, 134
librdf_serializer_serialize_model_to_string, 135
librdf_serializer_serialize_stream_to_file, 135
librdf_serializer_serialize_stream_to_string, 136
librdf_serializer_set_feature, 137
librdf_serializer_set_namespace, 138
librdf_short_copyright_string, 139
librdf_short_copyright_string_get, 139
librdf_statement_equals, 140
librdf_statement_get_object, 141
librdf_statement_get_predicate, 141
librdf_statement_get_subject, 141
librdf_statement_is_complete, 143
librdf_statement_is_equal, 143
librdf_statement_match, 143
librdf_statement_set_object, 144
librdf_statement_set_predicate, 144
librdf_statement_set_subject, 146
librdf_stream_end, 146
librdf_stream_get_object, 147
librdf_stream_next, 148
librdf_uri_compare, 148
librdf_uri_equals, 149
setQueryResultsLimit
  (setQueryResultLimit), 187
Statement, 161, 182
Statement (Statement-class), 188
Statement-class, 188
Statement-initialize
  (initialize,Statement-method), 23
Storage, 182
Storage (Storage-class), 189
Storage-class, 189
Storage-initialize
  (initialize,Storage-method), 24
World, 182
World (World-class), 189
World-class, 189
World-initialize
  (initialize,World-method), 25
writeResults, 164, 190
writeResults, Query-method
  (writeResults), 190