

# Package ‘redland’

May 9, 2026

**Version** 1.0.17-19

**Title** RDF Library Bindings in R

**Date** 2025-12-09

**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** 7.3.1

**URL** <https://github.com/ropensci/redland-bindings/tree/master/R/redland>  
<https://github.com/ropensci/redland-bindings/tree/master/R>

**Encoding** UTF-8

**Language** en-US

**NeedsCompilation** yes

**Author** Matthew B. Jones [aut, cre] (ORCID: <https://orcid.org/0000-0003-0077-4738>),  
 Peter Slaughter [aut] (ORCID: <https://orcid.org/0000-0002-2192-403X>),  
 Jeroen Ooms [aut] (ORCID: <https://orcid.org/0000-0002-4035-0289>),  
 Carl Boettiger [aut] (ORCID: <https://orcid.org/0000-0002-1642-628X>),  
 Scott Chamberlain [ctb] (ORCID: <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** 2025-12-10 06:20:16 UTC

## Contents

addStatement	7
executeQuery	8
freeModel	9
freeParser	9
freeQuery	10
freeQueryResults	11
freeSerializer	12
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_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_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

<code>librdf_node_get_blank_identifier</code>	99
<code>librdf_node_get_literal_value</code>	100
<code>librdf_node_get_literal_value_as_latn1</code>	101
<code>librdf_node_get_literal_value_datatype_uri</code>	101
<code>librdf_node_get_literal_value_is_wf_xml</code>	102
<code>librdf_node_get_literal_value_language</code>	103
<code>librdf_node_get_li_ordinal</code>	103
<code>librdf_node_get_type</code>	104
<code>librdf_node_get_uri</code>	105
<code>librdf_node_is_blank</code>	105
<code>librdf_node_is_literal</code>	106
<code>librdf_node_is_resource</code>	107
<code>librdf_parser_check_name</code>	107
<code>librdf_parser_get_accept_header</code>	108
<code>librdf_parser_get_feature</code>	109
<code>librdf_parser_get_namespaces_seen_count</code>	109
<code>librdf_parser_get_namespaces_seen_prefix</code>	110
<code>librdf_parser_get_namespaces_seen_uri</code>	111
<code>librdf_parser_guess_name2</code>	111
<code>librdf_parser_parse_as_stream</code>	112
<code>librdf_parser_parse_counted_string_as_stream</code>	113
<code>librdf_parser_parse_counted_string_into_model</code>	114
<code>librdf_parser_parse_into_model</code>	115
<code>librdf_parser_parse_string_as_stream</code>	116
<code>librdf_parser_parse_string_into_model</code>	116
<code>librdf_parser_set_feature</code>	117
<code>librdf_query_execute</code>	118
<code>librdf_query_get_limit</code>	119
<code>librdf_query_get_offset</code>	120
<code>librdf_query_results_as_stream</code>	120
<code>librdf_query_results_finished</code>	121
<code>librdf_query_results_get_bindings_count</code>	122
<code>librdf_query_results_get_binding_name</code>	122
<code>librdf_query_results_get_binding_value</code>	123
<code>librdf_query_results_get_binding_value_by_name</code>	124
<code>librdf_query_results_get_boolean</code>	124
<code>librdf_query_results_get_count</code>	125
<code>librdf_query_results_is_bindings</code>	126
<code>librdf_query_results_is_boolean</code>	126
<code>librdf_query_results_is_graph</code>	127
<code>librdf_query_results_is_syntax</code>	128
<code>librdf_query_results_next</code>	128
<code>librdf_query_results_to_file2</code>	129
<code>librdf_query_results_to_string2</code>	130
<code>librdf_query_set_limit</code>	131
<code>librdf_query_set_offset</code>	132
<code>librdf_serializer_check_name</code>	132
<code>librdf_serializer_get_feature</code>	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 . . . . .	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 . . . . .	157
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
raptor_version_decimal_get . . . . .	169
raptor_version_major . . . . .	170
raptor_version_major_get . . . . .	171
raptor_version_minor . . . . .	171
raptor_version_minor_get . . . . .	172
raptor_version_release . . . . .	173
raptor_version_release_get . . . . .	173
raptor_version_string . . . . .	174
raptor_version_string_get . . . . .	175
rasqal_version_decimal . . . . .	175
rasqal_version_decimal_get . . . . .	176
rasqal_version_majior . . . . .	177
rasqal_version_majior_get . . . . .	177
rasqal_version_minor . . . . .	178
rasqal_version_minor_get . . . . .	179
rasqal_version_release . . . . .	179
rasqal_version_release_get . . . . .	180
rasqal_version_string . . . . .	181
rasqal_version_string_get . . . . .	181
redland . . . . .	182
roclet_output.roclet_mergeNamespace . . . . .	183
roclet_process.roclet_mergeNamespace . . . . .	184
Serializer-class . . . . .	185
serializeToCharacter . . . . .	185
serializeToFile . . . . .	186
setNameSpace . . . . .	187
setQueryResultLimit . . . . .	187
Statement-class . . . . .	188
Storage-class . . . . .	189
World-class . . . . .	189
writeResults . . . . .	190
[,ExternalReference-method . . . . .	191
[<-,ExternalReference-method . . . . .	192

**Index****193**


---

addStatement	<i>Add a Statement object to the Model</i>
--------------	--

---

**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

*Execute a query*

---

**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	<i>Free memory used by a librdf model.</i>
-----------	--

---

**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	<i>Free memory used by a librdf parser</i>
------------	--

---

**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**

```
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**

```
freeQuery(.Object)

## 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**

```

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
results <- getResult(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)
# Create 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.

**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

*Free memory used by a librdf storage object*

---

**Description**

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'
freeStorage(.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	<i>Free memory used by a librdf world object</i>
-----------	--

---

**Description**

Free memory used by a librdf world object

**Usage**

```
freeWorld(.Object)

## 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**

```
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	<i>Get the blank identifier that has been assigned for a specified Node object</i>
----------------	--

---

**Description**

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

**Usage**

```
getBlankNodeId(.Object)

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

**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 in to the caller.

**Value**

a blank node identifier

**Examples**

```
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**

```
getNodeType(.Object)

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

**Arguments**

.Object            a Node object

**Value**

a character vector containing the Node type

**Examples**

```
world <- new("World")
node <- new("Node", world, uri="http://www.example.com")
nodeType <- getNodeTypes(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            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

---

getResults	<i>Return all query results</i>
------------	---------------------------------

---

**Description**

Return all query results

**Usage**

```
getResults(.Object, model, ...)

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

**Arguments**

.Object            a Query object  
 model             a Model object  
 ...                additional parameters  
 formatName        a 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)" 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="")
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

```

getTermType(.Object, term)

## S4 method for signature 'Statement,character'
getTermType(.Object, term)

```

### Arguments

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

### Examples

```

world <- new("World")
subject <- new("Node", blank="_:myid1", world)
predicate <- new("Node", uri="http://www.example.com/isa", 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**

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

**Arguments**

<code>.Object</code>	a Node object
<code>world</code>	a World object
<code>storage</code>	a Storage object
<code>options</code>	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**

```
## 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.

---

initialize,Parser-method

*Initialize a Parser object.*

---

**Description**

A Parser object is initialized for a specific RDF serialization.

**Usage**

```
## 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
contentType	a mime type of the syntax of the model
uri	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**

```
## 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**

```
## 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**

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

**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

---

initialize,Statement-method  
*Construct a Statement object.*

---

**Description**

Construct a Statement object.

**Usage**

```
## 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 a object literal value
language	a character value specifying the RDF language tag for an object literal value (excluding the "@" symbol), i.e. "fr"

**Value**

the Statement object

---

```
initialize,Storage-method  
Initialize a Storage object
```

---

**Description**

Initialize a Storage object

**Usage**

```
## 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**

```
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**

```
## 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**

```
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")

**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 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 )
```

**Arguments**

digest	the digest ("_p_librdf_digest_s")
buf	the data buffer ("character")
length	the length of the data ("integer")

**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\_update\_string

*Add a string to the librdf\_digest object.*

---

**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/>

**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 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 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      *Destructor - destroy a librdf\_query object.*

---

**Description**

Destructor - destroy a librdf\_query object.

**Usage**

```
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      *Destructor - destroy a librdf\_query\_results object.*

---

**Description**

Destructor - destroy a librdf\_query\_results object.

**Usage**

```
librdf_free_query_results ( query_results )
```

**Arguments**

query\_results      librdf\_query\_results object ("p\_librdf\_query\_results")

**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 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 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 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 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 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**

hash	librdf_hash object (" <code>_p_librdf_hash_s</code> ")
filter	NULL terminated list of keys to ignore (" <code>_p_p_char</code> ")

**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_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")

**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\_iterator\_end    *Test if the iterator has finished.*

---

**Description**

Test if the iterator has finished.

**Usage**

```
librdf_iterator_end ( iterator,  
                    .copy )
```

**Arguments**

iterator	the librdf_iterator object (" <code>_p_librdf_iterator_s</code> ")
.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_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/>

**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_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")

**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**

<code>iterator</code>	the <code>librdf_iterator</code> object (" <code>_p_librdf_iterator_s</code> ")
<code>.copy</code>	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 the Redland RDF C libraries. For more information about Redland RDF, view the online documentation indicated in the 'References' section.

---

librdf\_log\_message\_level

*Retrieve severity of log message.*

---

**Description**

Retrieve severity of log message.

**Usage**

```
librdf_log_message_level ( 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\_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/>

**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")

**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_model_add	<i>Create and add a new statement about a resource to the model.</i>
------------------	--

---

**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 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**

<code>model</code>	model object (" <code>_p_librdf_model_s</code> ")
<code>statement</code>	statement object (" <code>_p_librdf_statement_s</code> ")
<code>.copy</code>	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\_statements

*Add a stream of statements to the model.*

---

### 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

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

---

### Description

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_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 conten ("character")
inStrOrNull	language of literal ("character")
is_wf_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 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.

**Usage**

```
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 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 )
```

**Arguments**

model            the model object ("\_p\_librdf\_model\_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\_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 contest ("\_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_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**

<code>model</code>	the model object (" <code>_p_librdf_model_s</code> ")
<code>statement</code>	the statement (" <code>_p_librdf_statement_s</code> ")
<code>.copy</code>	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_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

<code>model</code>	librdf_model object (" <code>_p_librdf_model_s</code> ")
<code>context</code>	librdf_node context (" <code>_p_librdf_node_s</code> ")
<code>statement</code>	librdf_statement statement object (" <code>_p_librdf_statement_s</code> ")
<code>.copy</code>	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

<code>model</code>	librdf_model object (" <code>_p_librdf_model_s</code> ")
<code>context</code>	librdf_node context (" <code>_p_librdf_node_s</code> ")
<code>stream</code>	librdf_stream stream object (" <code>_p_librdf_stream_s</code> ")
<code>.copy</code>	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_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**

<code>model</code>	librdf_model object (" <code>_p_librdf_model_s</code> ")
<code>context</code>	librdf_node context (" <code>_p_librdf_node_s</code> ")

**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 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**

<code>model</code>	the model object (" <code>_p_librdf_model_s</code> ")
<code>statement</code>	the partial statement to match (" <code>_p_librdf_statement_s</code> ")

**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_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 (" <code>_p_librdf_model_s</code> ")
statement	librdf_statement partial statement to find (" <code>_p_librdf_statement_s</code> ")
inNodeOrNull	context librdf_node (or NULL) (" <code>_p_librdf_node_s</code> ")

### 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 )
```

**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 the Redland RDF C libraries. For more information about Redland RDF, view the online documentation indicated in the 'References' section.

---

librdf\_model\_get\_arcs *Return the arcs (predicates) of an arc in an RDF graph given source (subject) and target (object).*

---

**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/>

**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_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**

<code>model</code>	librdf_model object (" <code>_p_librdf_model_s</code> ")
<code>node</code>	librdf_node resource node (" <code>_p_librdf_node_s</code> ")

**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**

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

**Arguments**

model	librdf_model object (" <code>_p_librdf_model_s</code> ")
node	librdf_node resource node (" <code>_p_librdf_node_s</code> ")

**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**

```
librdf_model_get_contexts ( model )
```

**Arguments**

model            librdf\_model object ("\_p\_librdf\_model\_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\_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/>

**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_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**

<code>model</code>	librdf_model object (" <code>_p_librdf_model_s</code> ")
<code>arc</code>	librdf_node arc (" <code>_p_librdf_node_s</code> ")
<code>target</code>	librdf_node target (" <code>_p_librdf_node_s</code> ")

**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).

**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 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")

**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\_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 (" <code>_p_librdf_model_s</code> ")
source	librdf_node source (" <code>_p_librdf_node_s</code> ")
arc	librdf_node arc (" <code>_p_librdf_node_s</code> ")

**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_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**

<code>model</code>	librdf_model object (" <code>_p_librdf_model_s</code> ")
<code>node</code>	librdf_node resource node (" <code>_p_librdf_node_s</code> ")
<code>property</code>	librdf_node property node (" <code>_p_librdf_node_s</code> ")
<code>.copy</code>	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**

<code>model</code>	librdf_model object (" <code>_p_librdf_model_s</code> ")
<code>node</code>	librdf_node resource node (" <code>_p_librdf_node_s</code> ")
<code>property</code>	librdf_node property node (" <code>_p_librdf_node_s</code> ")
<code>.copy</code>	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

```
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**

<code>model</code>	librdf_model object (" <code>_p_librdf_model_s</code> ")
<code>query</code>	librdf_query object (" <code>_p_librdf_query</code> ")

**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 )
```

**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 the Redland RDF C libraries. For more information about Redland RDF, view the online documentation indicated in the 'References' section.

---

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**

model	librdf_model object ("_p_librdf_model_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\_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 the Redland RDF C libraries. For more information about Redland RDF, view the online documentation indicated in the 'References' section.

---

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 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 )
```

**Arguments**

model	librdf_model object (" <code>_p_librdf_model_s</code> ")
uri	base URI to use in serializing (or NULL if not used) (" <code>_p_librdf_uri_s</code> ")
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) (" <code>_p_librdf_uri_s</code> ")

**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\_model\_transaction\_commit  
*Commit a transaction.*

---

**Description**

Commit a transaction.

**Usage**

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

**Arguments**

model	the model object (" <code>_p_librdf_model_s</code> ")
.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\_rollback  
*Rollback a transaction.*

---

**Description**

Rollback a transaction.

**Usage**

```
librdf_model_transaction_rollback ( 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  
*Start a transaction*

---

**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 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 )
```

**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.

---

librdf_new_hash	<i>Constructor - create a new librdf_hash object.</i>
-----------------	---

---

**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**

```
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/>

**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.

**Usage**

```
librdf_new_hash_from_string ( world,
                             name,
                             string )
```

**Arguments**

world	redland world object (" <code>_p_librdf_world_s</code> ")
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	<i>Constructor - create a new storage librdf_model object.</i>
------------------	--

---

**Description**

Constructor - create a new storage librdf\_model object.

**Usage**

```
librdf_new_model ( world,
                  storage,
                  options_string )
```

**Arguments**

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

**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\_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**

```
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 (" <code>_p_librdf_world_s</code> ")
storage	librdf_storage storage to use (" <code>_p_librdf_storage_s</code> ")
options	librdf_hash of options to use (" <code>_p_librdf_hash_s</code> ")

### 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 )
```

**Arguments**

world            redland world object ("\_p\_librdf\_world\_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\_blank\_identifier

*Constructor - create a new blank node librdf\_node object from a blank node identifier.*

---

**Description**

Constructor - create a new blank node librdf\_node object from a blank node identifier.

**Usage**

```
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/>

**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\_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 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 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 )
```

**Arguments**

world	redland world object ("_p_librdf_world_s")
uri_string	UTF-8 encoded string representing a URI ("character")
source_uri	source URI ("_p_librdf_uri_s")
base_uri	base URI ("_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\_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**

```
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 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_local_name
```

*Constructor - create a new resource librdf\_node object with a given URI and local name.*

---

### 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
```

*Constructor - create a new librdf\_node object from a URI string.*

---

### Description

Constructor - create a new librdf\_node object from a 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.

---

librdf_new_parser	<i>Constructor - create a new librdf_parser object.</i>
-------------------	---

---

**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

**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	<i>Constructor - create a new librdf_query object.</i>
------------------	--

---

**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 the Redland RDF C libraries. For more information about Redland RDF, view the online documentation indicated in the 'References' section.

---

librdf\_new\_statement    *Constructor - create a new empty librdf\_statement.*

---

**Description**

Constructor - create a new empty librdf\_statement.

**Usage**

```
librdf_new_statement ( world )
```

**Arguments**

world	redland world object ("_p_librdf_world_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_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 the Redland RDF C libraries. For more information about Redland RDF, view the online documentation indicated in the 'References' section.

---

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**

```
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 the Redland RDF C libraries. For more information about Redland RDF, view the online documentation indicated in the 'References' section.

---

librdf_new_uri	<i>Constructor - create a new librdf_uri object from a URI string.</i>
----------------	--

---

**Description**

Constructor - create a new librdf\_uri object from a URI string.

**Usage**

```
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/>

**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_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**

```
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/>

**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_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**

```
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 the Redland RDF C libraries. For more information about Redland RDF, view the online documentation indicated in the 'References' section.

---

librdf\_new\_world            *Create a new Redland execution environment.*

---

**Description**

Create a new Redland execution environment.

**Usage**

```
librdf_new_world ( )
```

**Value**

\_p\_librdf\_world\_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\_equals      *Compare two librdf\_node objects for equality.*

---

**Description**

Compare two librdf\_node objects for equality.

**Usage**

```
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 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**

```
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`")

**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_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**

<code>node</code>	the node object (" <code>_p_librdf_node_s</code> ")
<code>.copy</code>	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

**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\_type *Get the type of the node.*

---

**Description**

Get the type of the node.

**Usage**

```
librdf_node_get_type ( node,  
  .copy )
```

**Arguments**

node	the node object (" <code>_p_librdf_node_s</code> ")
.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\_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

**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\_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**

<code>node</code>	the node object (" <code>_p_librdf_node_s</code> ")
<code>.copy</code>	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 )
```

**Arguments**

world	redland world object ("_p_librdf_world_s")
name	name of parser ("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\_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 (" <code>_p_librdf_parser_s</code> ")
feature	librdf_Uri feature property (" <code>_p_librdf_uri_s</code> ")

**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_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 )
```

**Arguments**

parser	librdf_parser object ("_p_librdf_parser_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\_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 (" <code>_p_librdf_parser_s</code> ")
offset	index into list of namespaces (" <code>integer</code> ")

**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\_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 )
```

**Arguments**

world	librdf_world object ("_p_librdf_world_s")
mime_type	MIME type of syntax or NULL ("character")
buffer	content buffer or NULL ("character")
identifier	content identifier or NULL ("character")

**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\_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/>

**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_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**

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

**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 (" <code>_p_librdf_parser_s</code> ")
string	the content to parse ("character")
length	length of content (must be >0) ("integer")
base_uri	the base URI to use or NULL (" <code>_p_librdf_uri_s</code> ")
model	the model to use (" <code>_p_librdf_model_s</code> ")
.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**

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

**Arguments**

<code>parser</code>	the parser (" <code>_p_librdf_parser_s</code> ")
<code>uri</code>	the URI to read the content (" <code>_p_librdf_uri_s</code> ")
<code>inUriOrNull</code>	the base URI to use or NULL (" <code>_p_librdf_uri_s</code> ")
<code>model</code>	the model to use (" <code>_p_librdf_model_s</code> ")
<code>.copy</code>	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_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**

```
librdf_parser_parse_string_as_stream ( parser,  
  string,  
  base_uri )
```

**Arguments**

<code>parser</code>	the parser (" <code>_p_librdf_parser_s</code> ")
<code>string</code>	the string to parse ("character")
<code>base_uri</code>	the base URI to use or NULL (" <code>_p_librdf_uri_s</code> ")

**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_string_into_model`*Parse a string of content into an librdf\_model.*

---

**Description**

Parse a string of content into an librdf\_model.

**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 the Redland RDF C libraries. For more information about Redland RDF, view the online documentation indicated in the 'References' section.

---

librdf\_parser\_set\_feature

*Set the value of a parser feature.*

---

**Description**

Set the value of a parser feature.

**Usage**

```
librdf_parser_set_feature ( parser,  
feature,  
value,  
.copy )
```

**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 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**

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

**Arguments**

<code>query</code>	librdf_query query object (" <code>_p_librdf_query</code> ")
<code>.copy</code>	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_get_offset`*Get the query-specified offset on results.*

---

**Description**

Get the query-specified offset on results.

**Usage**

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

**Arguments**

<code>query</code>	librdf_query query object (" <code>_p_librdf_query</code> ")
<code>.copy</code>	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_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**

```
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\_finished

*Find out if binding results are exhausted.*

---

**Description**

Find out if binding results are exhausted.

**Usage**

```
librdf_query_results_finished ( 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\_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 (" <code>_p_librdf_query_results</code> ")
.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 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 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\_by\_name

*Get one binding value for a given name in the current result.*

---

### Description

Get one binding value for a given name in the current result.

### Usage

```
librdf_query_results_get_binding_value_by_name ( query_results,  
name )
```

### Arguments

query_results	librdf_query_results query results (" <code>_p_librdf_query_results</code> ")
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\_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")  
.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 )
```

**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\_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**

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\_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 (" <u>p</u> _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\_next

*Move to the next result.*

---

### Description

Move to the next result.

### Usage

```
librdf_query_results_next ( 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\_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

**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\_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 (" <code>_p_librdf_query_results</code> ")
name	format name ("character")
mime_type	format mime type (or NULL) ("character")
format_uri	URI of syntax to format to (or NULL) (" <code>_p_librdf_uri_s</code> ")
base_uri	Base URI of output formatted syntax (or NULL) (" <code>_p_librdf_uri_s</code> ")

**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_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**

<code>query</code>	librdf_query query object (" <code>_p_librdf_query</code> ")
<code>limit</code>	the limit on results, $\geq 0$ to set a limit, $< 0$ to have no limit ("integer")
<code>.copy</code>	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_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**

<code>query</code>	librdf_query query object (" <code>_p_librdf_query</code> ")
<code>offset</code>	offset for results, $\geq 0$ to set an offset, $< 0$ to have no offset ("integer")
<code>.copy</code>	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_check_name`*Check if a serializer name is known*

---

**Description**

Check if a serializer name is known

**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

**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\_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_file
```

*Write a librdf\_stream to a file.*

---

### Description

Write a librdf\_stream to a file.

**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 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

**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\_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  
*Short copyright string (one line).*

---

**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
-------	---------

**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\_statement\_equals

*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 (" <u>p</u> _librdf_statement_s")
statement2	second librdf_statement (" <u>p</u> _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\_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 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")

**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_statement_get_subject`

*Get the statement subject.*

---

**Description**

Get the statement subject.

**Usage**

```
librdf_statement_get_subject ( 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 the Redland RDF C libraries. For more information about Redland RDF, view the online documentation indicated in the 'References' section.

---

`librdf_statement_is_complete`*Check if statement is a complete and legal RDF triple.*

---

**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 the Redland RDF C libraries. For more information about Redland RDF, view the online documentation indicated in the 'References' section.

---

`librdf_statement_match`*Match a statement against a 'partial' statement.*

---

**Description**

Match a statement against a 'partial' statement.

**Usage**

```
librdf_statement_match ( statement,  
                        partial_statement,  
                        .copy )
```

**Arguments**

statement	statement (" <code>_p_librdf_statement_s</code> ")
partial_statement	statement with possible empty parts (" <code>_p_librdf_statement_s</code> ")
.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\_statement\_set\_object

*Set the statement object.*

---

**Description**

Set the statement object.

**Usage**

```
librdf_statement_set_object ( statement,
                             object )
```

**Arguments**

statement	librdf_statement object (" <code>_p_librdf_statement_s</code> ")
object	librdf_node of object (" <code>_p_librdf_node_s</code> ")

**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_statement_set_predicate  
    Set the statement predicate.
```

---

**Description**

Set the statement predicate.

**Usage**

```
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 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 (" <code>_p_librdf_statement_s</code> ")
subject	librdf_node of subject (" <code>_p_librdf_node_s</code> ")

**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\_stream\_end      *Test if the stream has ended.*

---

**Description**

Test if the stream has ended.

**Usage**

```
librdf_stream_end ( 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\_stream\_get\_object

*Get the current librdf\_statement in the stream.*

---

**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 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 )
```

**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/>

**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\_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 the Redland RDF C libraries. For more information about Redland RDF, view the online documentation indicated in the 'References' section.

---

librdf\_version\_decimal  
*Library full version as a decimal integer.*

---

**Description**

Library full version as a decimal integer.

**Usage**

```
librdf_version_decimal ( .copy )
```

**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\_decimal\_get  
*Return Redland librdf copyright*

---

**Description**

Return Redland librdf copyright

**Usage**

```
librdf_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.

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/>

**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\_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

**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\_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 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 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

**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\_release\_get

*Return the Redland librdf release version number*

---

**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 the Redland RDF C libraries. For more information about Redland RDF, view the online documentation indicated in the 'References' section.

librdf\_version\_string *Library full version as a string.*

---

**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 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

**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\_world\_get\_feature

*Get the value of a world feature.*

---

**Description**

Get the value of a world feature.

**Usage**

```
librdf_world_get_feature ( world,  
feature )
```

**Arguments**

world	librdf_world object ("_p_librdf_world_s")
feature	librdf_uri 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 the Redland RDF C libraries. For more information about Redland RDF, view the online documentation indicated in the 'References' section.

librdf\_world\_open      *Open a created redland world environment.*

---

**Description**

Open a created redland world environment.

**Usage**

```
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\_feature  
                          *Set the value of a world feature.*

---

**Description**

Set the value of a world feature.

**Usage**

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

**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 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**

```
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/>

**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.

---

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	<i>A Redland Model object</i>
-------------	-------------------------------

---

**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	<i>A Redland Node, used to store one node in an RDF triple statement.</i>
------------	---

---

**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. **getNodeTypes**: 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**

```
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="G rard de La Martini re", 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**

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

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

```
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

```
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)
```

---

 Query-class

*Query an RDF 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](http://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")
status <- 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",
```

```

      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
results <- getResult(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 )
```

**Arguments**

locator	raptor locator (" <code>_p_raptor_locator</code> ")
.copy	logical

**Value**

character

**References**

<https://librdfs.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 (" <code>_p_raptor_locator</code> ")
.copy	logical

**Value**

integer

**References**

<https://librdfs.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

**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**

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_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/>

**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_get`*Raptor version as a decimal number.*

---

**Description**

Raptor version as a decimal number.

**Usage**

```
raptor_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.

---

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

**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\_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 the Redland RDF C libraries. For more information about Redland RDF, view the online documentation indicated in the 'References' section.

---

`raptor_version_release`*Raptor library release.*

---

**Description**

Raptor library release.

**Usage**

```
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 the Redland RDF C libraries. For more information about Redland RDF, view the online documentation indicated in the 'References' section.

---

`raptor_version_release_get`*Raptor library release.*

---

**Description**

Get Raptor library release.

**Usage**

```
raptor_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.

---

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 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/>

**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\_decimal  
*Rasqal version as a decimal number.*

---

**Description**

Rasqal version as a decimal number.

**Usage**

```
rasqal_version_decimal ( .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\_decimal\_get

*Get the Rasqal version as a decimal number.*

---

**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 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 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

*Get the Rasqal release version number.*

---

**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**

```
rasqal_version_string ( .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\_get

*Get the Rasqal version as a string*

---

**Description**

Get the Rasqal version as a string.

**Usage**

```
rasqal_version_string_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.

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:

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

**Note**

In order to communicate with the Redland RDF C libraries, the redland R package uses an interface 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 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\_', 'rasqal\_' or 'raptor\_' and are usually the same name as the underlying C library function. Documentation 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**

```
# This example creates the necessary R objects to hold an RDF model and reads
# in 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 <- getResult(query, model, "rdfxml")

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

---

roclet\_output.roclet\_mergeNamespace

*Roxygen output function that merges a base NAMESPACE file with the  
Roxygen dynamically created NAMESPACE file*

---

**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_output(x, results, base_path, ...)
```

**Arguments**

x	the currently running roclet
results	the list of items to process that was generated by the roc_process.mergedNamespace function
base_path	the base directory path of the package
...	additional parameters

---

```
roclet_process.roclet_mergeNamespace
```

*Roxygen process function for the 'mergeNamespace' roclet*

---

**Description**

This function is called by the Roxygen2 roxygenize 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
env	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

```
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" serizlizer
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

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

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

**Arguments**

.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

**Value**

a character vector containing the serialized model

---

serializeToFile	<i>Serialize a model to a file.</i>
-----------------	-------------------------------------

---

**Description**

Serialize a model to a file.

**Usage**

```
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	<i>Set a namespace for the serializer.</i>
--------------	--

---

**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	<i>Set limit on returned query results</i>
---------------------	--

---

**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.

---

Statement-class      *An RDF Statement object*

---

## 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

```
world <- new("World")
# Create nodes manually and add to the statment
subject <- new("Node", blank="_:myid1", world)
predicate <- new("Node", uri="http://www.example.com/isa", 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",
```

```
object="¡Hola, amigo! ¿Cómo estás?",
objectType="literal",
language="es")
```

---

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

**See Also**

[redland](#): redland package

**Examples**

```
world <- new("World")
```

---

writeResults	<i>Write query results to a file.</i>
--------------	---------------------------------------

---

**Description**

Write query results to a file.

**Usage**

```
writeResults(.Object, model, ...)

## 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**

```

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)

```

---

[,ExternalReference-method

*Subset a list of ExternalReferences*


---

**Description**

Subset a list of ExternalReferences

**Usage**

```

## 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

---

[<- ,ExternalReference-method  
*Assign values in a list of ExternalReferences*

---

**Description**

Assign values in a list of ExternalReferences

**Usage**

```
## 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, [9](#), [161](#)
- freeModel, Model-method (freeModel), [9](#)
- 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, [12](#), [185](#)
- freeSerializer, Serializer-method  
(freeSerializer), [12](#)
- 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](#)
- getNodeTypes, [15](#), [162](#)
- getNodeTypes, Node-method (getNodeTypes),  
[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, 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\_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\_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\_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\_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
- 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, 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, [157](#)
- librdf\_world\_open, [158](#)
- librdf\_world\_set\_feature, [158](#)
- librdf\_world\_set\_logger, [159](#)
  
- mergeNamespace\_roclet, [160](#)
- Model, [182](#)
- Model (Model-class), [161](#)
- Model-class, [161](#)
- Model-initialize
  - (initialize, Model-method), [19](#)
  
- Node, [182](#)
- Node (Node-class), [161](#)
- Node-class, [161](#)
- Node-initialize
  - (initialize, Node-method), [19](#)
  
- parseFileIntoModel, [161](#), [162](#), [163](#)
- parseFileIntoModel, Parser, World, character, Model-method
  - (parseFileIntoModel), [162](#)
- Parser, [182](#)
- Parser (Parser-class), [163](#)
- Parser-class, [163](#)
- Parser-initialize
  - (initialize, Parser-method), [20](#)
  
- Query, [161](#), [182](#)
- Query (Query-class), [164](#)
- Query-class, [164](#)
- Query-initialize
  - (initialize, Query-method), [21](#)
- QueryResults, [182](#)
- QueryResults (QueryResults-class), [165](#)
- QueryResults-class, [165](#)
- QueryResults-initialize
  - (initialize, QueryResults-method), [22](#)
  
- raptor\_locator\_byte, [165](#)
- raptor\_locator\_column, [166](#)
- raptor\_locator\_file, [167](#)
- raptor\_locator\_line, [167](#)
- raptor\_locator\_uri, [168](#)
- raptor\_version\_decimal, [169](#)
- raptor\_version\_decimal\_get, [169](#)
- raptor\_version\_major, [170](#)
- raptor\_version\_major\_get, [171](#)
- raptor\_version\_minor, [171](#)
- raptor\_version\_minor\_get, [172](#)
- raptor\_version\_release, [173](#)
- raptor\_version\_release\_get, [173](#)
- raptor\_version\_string, [174](#)
- raptor\_version\_string\_get, [175](#)
- rasqal\_version\_decimal, [175](#)
- rasqal\_version\_decimal\_get, [176](#)
- rasqal\_version\_major, [177](#)
- rasqal\_version\_major\_get, [177](#)
- rasqal\_version\_minor, [178](#)
- rasqal\_version\_minor\_get, [179](#)
- rasqal\_version\_release, [179](#)
- rasqal\_version\_release\_get, [180](#)
- rasqal\_version\_string, [181](#)
- rasqal\_version\_string\_get, [181](#)
- redland, [161–165](#), [182](#), [185](#), [188–190](#)
- roclet\_output.roclet\_mergeNamespaces, [183](#)
- roclet\_process.roclet\_mergeNamespaces, [184](#)
- Serializer, [182](#)
- Serializer (Serializer-class), [185](#)
- Serializer-class, [185](#)
- Serializer-initialize
  - (initialize, Serializer-method), [22](#)
- serializeToCharacter, [185](#), [185](#)
- serializeToCharacter, Serializer, World, Model-method
  - (serializeToCharacter), [185](#)
- serializeToFile, [185](#), [186](#)
- serializeToFile, Serializer, World, Model, character-method
  - (serializeToFile), [186](#)
- setNameSpace, [185](#), [187](#)
- setNameSpace, Serializer, World, character, character-method
  - (setNameSpace), [187](#)
- setQueryResultLimit, [164](#), [187](#)
- setQueryResultLimit, Query-method
  - (setQueryResultLimit), [187](#)

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