

# Package ‘redland’

August 29, 2016

**Version** 1.0.17-8

**Title** RDF Library Bindings in R

**Date** 2016-04-25

**VignetteBuilder** knitr

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

This package supports RDF by implementing an R interface to the Redland RDF C library, described at <<http://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** knitr, testthat

**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** 5.0.1

**NeedsCompilation** yes

**Author** Matthew B. Jones [aut, cre],  
Peter Slaughter [aut],  
Jeroen Ooms [aut],  
Carl Boettiger [aut],  
Scott Chamberlain [aut],  
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** 2016-04-26 08:44:28

## R topics documented:

addStatement . . . . .	7
executeQuery . . . . .	8
freeModel . . . . .	8
freeParser . . . . .	9
freeQuery . . . . .	10
freeQueryResults . . . . .	11
freeSerializer . . . . .	12
freeStatement . . . . .	13
freeStorage . . . . .	13
freeWorld . . . . .	14
getBlankNodeId . . . . .	15
getNextResult . . . . .	15
getNodeTypes . . . . .	16
getQueryResultLimit . . . . .	17
getTermType . . . . .	17
initialize,Model-method . . . . .	18
initialize,Node-method . . . . .	18
initialize,Parser-method . . . . .	19
initialize,Query-method . . . . .	20
initialize,QueryResults-method . . . . .	20
initialize,Serializer-method . . . . .	21
initialize,Statement-method . . . . .	22
initialize,Storage-method . . . . .	22
initialize,World-method . . . . .	23
is.null.externalptr . . . . .	24
length,SWIGArray-method . . . . .	24
librdf_copyright_string . . . . .	25
librdf_copyright_string_get . . . . .	25
librdf_digest_final . . . . .	26
librdf_digest_init . . . . .	27
librdf_digest_to_string . . . . .	27
librdf_digest_update . . . . .	28
librdf_digest_update_string . . . . .	29
librdf_free_digest . . . . .	29
librdf_free_hash . . . . .	30
librdf_free_iterator . . . . .	31
librdf_free_model . . . . .	31
librdf_free_node . . . . .	32
librdf_free_parser . . . . .	33
librdf_free_query . . . . .	33
librdf_free_query_results . . . . .	34

librdf_free_serializer . . . . .	35
librdf_free_statement . . . . .	35
librdf_free_storage . . . . .	36
librdf_free_stream . . . . .	37
librdf_free_uri . . . . .	37
librdf_free_world . . . . .	38
librdf_hash_to_string . . . . .	39
librdf_internal_test_error . . . . .	39
librdf_internal_test_warning . . . . .	40
librdf_iterator_end . . . . .	41
librdf_iterator_get_context . . . . .	41
librdf_iterator_get_object . . . . .	42
librdf_iterator_next . . . . .	43
librdf_log_message_code . . . . .	43
librdf_log_message_facility . . . . .	44
librdf_log_message_level . . . . .	45
librdf_log_message_locator . . . . .	45
librdf_log_message_message . . . . .	46
librdf_model_add . . . . .	47
librdf_model_add_statement . . . . .	48
librdf_model_add_statements . . . . .	48
librdf_model_add_string_literal_statement . . . . .	49
librdf_model_add_typed_literal_statement . . . . .	50
librdf_model_as_stream . . . . .	51
librdf_model_contains_context . . . . .	52
librdf_model_contains_statement . . . . .	52
librdf_model_context_add_statement . . . . .	53
librdf_model_context_add_statements . . . . .	54
librdf_model_context_as_stream . . . . .	55
librdf_model_context_remove_statement . . . . .	55
librdf_model_context_remove_statements . . . . .	56
librdf_model_find_statements . . . . .	57
librdf_model_find_statements_in_context . . . . .	58
librdf_model_get_arc . . . . .	58
librdf_model_get_arcs . . . . .	59
librdf_model_get_arcs_in . . . . .	60
librdf_model_get_arcs_out . . . . .	61
librdf_model_get_contexts . . . . .	61
librdf_model_get_feature . . . . .	62
librdf_model_get_source . . . . .	63
librdf_model_get_sources . . . . .	63
librdf_model_get_target . . . . .	64
librdf_model_get_targets . . . . .	65
librdf_model_has_arc_in . . . . .	66
librdf_model_has_arc_out . . . . .	67
librdf_model_load . . . . .	68
librdf_model_query_execute . . . . .	69
librdf_model_remove_statement . . . . .	69

<code>librdf_model_set_feature</code> . . . . .	70
<code>librdf_model_size</code> . . . . .	71
<code>librdf_model_sync</code> . . . . .	72
<code>librdf_model_to_string</code> . . . . .	72
<code>librdf_model_transaction_commit</code> . . . . .	73
<code>librdf_model_transaction_rollback</code> . . . . .	74
<code>librdf_model_transaction_start</code> . . . . .	75
<code>librdf_new_digest</code> . . . . .	75
<code>librdf_new_hash</code> . . . . .	76
<code>librdf_new_hash_from_array_of_strings</code> . . . . .	77
<code>librdf_new_hash_from_string</code> . . . . .	77
<code>librdf_new_model</code> . . . . .	78
<code>librdf_new_model_from_model</code> . . . . .	79
<code>librdf_new_model_with_options</code> . . . . .	80
<code>librdf_new_node</code> . . . . .	80
<code>librdf_new_node_from_blank_identifier</code> . . . . .	81
<code>librdf_new_node_from_literal</code> . . . . .	82
<code>librdf_new_node_from_node</code> . . . . .	83
<code>librdf_new_node_from_normalised_uri_string</code> . . . . .	83
<code>librdf_new_node_from_typed_literal</code> . . . . .	84
<code>librdf_new_node_from_uri</code> . . . . .	85
<code>librdf_new_node_from_uri_local_name</code> . . . . .	86
<code>librdf_new_node_from_uri_string</code> . . . . .	86
<code>librdf_new_parser</code> . . . . .	87
<code>librdf_new_query</code> . . . . .	88
<code>librdf_new_query_from_query</code> . . . . .	89
<code>librdf_new_serializer</code> . . . . .	89
<code>librdf_new_statement</code> . . . . .	90
<code>librdf_new_statement_from_nodes</code> . . . . .	91
<code>librdf_new_statement_from_statement</code> . . . . .	92
<code>librdf_new_storage</code> . . . . .	92
<code>librdf_new_storage_from_storage</code> . . . . .	93
<code>librdf_new_uri</code> . . . . .	94
<code>librdf_new_uri_from_filename</code> . . . . .	95
<code>librdf_new_uri_from_uri</code> . . . . .	95
<code>librdf_new_world</code> . . . . .	96
<code>librdf_node_equals</code> . . . . .	97
<code>librdf_node_get_blank_identifier</code> . . . . .	97
<code>librdf_node_get_literal_value</code> . . . . .	98
<code>librdf_node_get_literal_value_as_latin1</code> . . . . .	99
<code>librdf_node_get_literal_value_datatype_uri</code> . . . . .	99
<code>librdf_node_get_literal_value_is_wf_xml</code> . . . . .	100
<code>librdf_node_get_literal_value_language</code> . . . . .	101
<code>librdf_node_get_li_ordinal</code> . . . . .	101
<code>librdf_node_get_type</code> . . . . .	102
<code>librdf_node_get_uri</code> . . . . .	103
<code>librdf_node_is_blank</code> . . . . .	103
<code>librdf_node_is_literal</code> . . . . .	104

<code>librdf_node_is_resource</code>	105
<code>librdf_node_to_string</code>	105
<code>librdf_parser_check_name</code>	106
<code>librdf_parser_get_accept_header</code>	107
<code>librdf_parser_get_feature</code>	107
<code>librdf_parser_get_namespaces_seen_count</code>	108
<code>librdf_parser_get_namespaces_seen_prefix</code>	109
<code>librdf_parser_get_namespaces_seen_uri</code>	109
<code>librdf_parser_guess_name</code>	110
<code>librdf_parser_guess_name2</code>	111
<code>librdf_parser_parse_as_stream</code>	112
<code>librdf_parser_parse_counted_string_as_stream</code>	112
<code>librdf_parser_parse_counted_string_into_model</code>	113
<code>librdf_parser_parse_into_model</code>	114
<code>librdf_parser_parse_string_as_stream</code>	115
<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>	118
<code>librdf_query_get_offset</code>	119
<code>librdf_query_results_as_stream</code>	120
<code>librdf_query_results_finished</code>	120
<code>librdf_query_results_get_bindings_count</code>	121
<code>librdf_query_results_get_binding_name</code>	122
<code>librdf_query_results_get_binding_value</code>	122
<code>librdf_query_results_get_binding_value_by_name</code>	123
<code>librdf_query_results_get_boolean</code>	124
<code>librdf_query_results_get_count</code>	124
<code>librdf_query_results_is_bindings</code>	125
<code>librdf_query_results_is_boolean</code>	126
<code>librdf_query_results_is_graph</code>	126
<code>librdf_query_results_is_syntax</code>	127
<code>librdf_query_results_next</code>	128
<code>librdf_query_results_to_file</code>	128
<code>librdf_query_results_to_file2</code>	129
<code>librdf_query_results_to_string</code>	130
<code>librdf_query_results_to_string2</code>	131
<code>librdf_query_set_limit</code>	132
<code>librdf_query_set_offset</code>	132
<code>librdf_serializer_check_name</code>	133
<code>librdf_serializer_get_feature</code>	134
<code>librdf_serializer_serialize_model_to_file</code>	135
<code>librdf_serializer_serialize_model_to_string</code>	136
<code>librdf_serializer_serialize_stream_to_file</code>	136
<code>librdf_serializer_serialize_stream_to_string</code>	137
<code>librdf_serializer_set_feature</code>	138
<code>librdf_serializer_set_namespace</code>	139
<code>librdf_short_copyright_string</code>	140

<code>librdf_short_copyright_string_get</code>	140
<code>librdf_statement_equals</code>	141
<code>librdf_statement_get_object</code>	142
<code>librdf_statement_get_predicate</code>	142
<code>librdf_statement_get_subject</code>	143
<code>librdf_statement_is_complete</code>	144
<code>librdf_statement_match</code>	144
<code>librdf_statement_set_object</code>	145
<code>librdf_statement_set_predicate</code>	146
<code>librdf_statement_set_subject</code>	147
<code>librdf_statement_to_string</code>	147
<code>librdf_stream_end</code>	148
<code>librdf_stream_get_context</code>	149
<code>librdf_stream_get_object</code>	149
<code>librdf_stream_next</code>	150
<code>librdf_uri_compare</code>	151
<code>librdf_uri_equals</code>	151
<code>librdf_uri_to_string</code>	152
<code>librdf_version_decimal</code>	153
<code>librdf_version_decimal_get</code>	153
<code>librdf_version_major</code>	154
<code>librdf_version_major_get</code>	155
<code>librdf_version_minor</code>	155
<code>librdf_version_minor_get</code>	156
<code>librdf_version_release</code>	157
<code>librdf_version_release_get</code>	157
<code>librdf_version_string</code>	158
<code>librdf_version_string_get</code>	159
<code>librdf_world_get_feature</code>	159
<code>librdf_world_open</code>	160
<code>librdf_world_set_feature</code>	161
<code>librdf_world_set_logger</code>	162
<code>mergeNamespace_roclet</code>	162
<code>Model-class</code>	163
<code>Node-class</code>	164
<code>parseFileIntoModel</code>	165
<code>Parser-class</code>	166
<code>Query-class</code>	166
<code>QueryResults-class</code>	168
<code>raptor_locator_byte</code>	169
<code>raptor_locator_column</code>	169
<code>raptor_locator_file</code>	170
<code>raptor_locator_line</code>	171
<code>raptor_locator_uri</code>	171
<code>raptor_version_decimal</code>	172
<code>raptor_version_decimal_get</code>	173
<code>raptor_version_major</code>	173
<code>raptor_version_major_get</code>	174

raptor_version_minor . . . . .	175
raptor_version_minor_get . . . . .	175
raptor_version_release . . . . .	176
raptor_version_release_get . . . . .	177
raptor_version_string . . . . .	177
raptor_version_string_get . . . . .	178
rasqal_version_decimal . . . . .	179
rasqal_version_decimal_get . . . . .	179
rasqal_version_major . . . . .	180
rasqal_version_major_get . . . . .	181
rasqal_version_minor . . . . .	181
rasqal_version_minor_get . . . . .	182
rasqal_version_release . . . . .	183
rasqal_version_release_get . . . . .	183
rasqal_version_string . . . . .	184
rasqal_version_string_get . . . . .	185
redland . . . . .	185
roc_output.mergeNamespace . . . . .	187
roc_process.mergeNamespace . . . . .	187
Serializer-class . . . . .	188
serializeToCharacter . . . . .	189
serializeToFile . . . . .	189
setNameSpace . . . . .	190
setQueryResultLimit . . . . .	191
Statement-class . . . . .	191
Storage-class . . . . .	192
World-class . . . . .	193
[,ExternalReference-method . . . . .	193
[<-,ExternalReference-method . . . . .	194

**Index****195**


---

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	<i>Execute a query</i>
--------------	------------------------

---

**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

**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

*Free memory used by a librdf parser*


---

**Description**

Free memory used by a librdf parser

**Usage**

```
freeParser(.Object)

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

**Arguments**

.Object            a Node object

**Details**

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

## Examples

```
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)
queryResult <- executeQuery(query, model)
result <- getNextResult(queryResult)

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

```

---

freeQueryResults	<i>Free memory used by a librdf query results</i>
------------------	---

---

## 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

## 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)
queryResult <- executeQuery(query, model)
result <- getNextResult(queryResult)

```

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

---

freeSerializer	<i>Free memory used by a librdf serializer.</i>
----------------	---

---

### Description

Free memory used by a librdf serializer.

### Usage

```
freeSerializer(.Object)

## S4 method for signature 'Serializer'
freeSerializer(.Object)
```

### Arguments

.Object            a Serializer object

### Examples

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

---

freeStatement	<i>Free memory used by a librdf statement</i>
---------------	---

---

### 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	<i>Free memory used by a librdf storage object</i>
-------------	--

---

### 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

*Free memory used by a librdf world object*


---

**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)
```

---

getNextResult	<i>Get the next query result.</i>
---------------	-----------------------------------

---

### Description

The next query result is returned. .

**Usage**

```
getNextResult(.Object)

## S4 method for signature 'QueryResults'
getNextResult(.Object)
```

**Arguments**

.Object            a QueryResults object

---

getNodeType	<i>Determine the node type and return as a string</i>
-------------	---

---

**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)
```



---

getQueryResultLimit	<i>Get the query result limit</i>
---------------------	-----------------------------------

---

**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

---

getTermType	<i>Return the redland node type for the specified RDF term in a statement</i>
-------------	---

---

**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**

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

**Value**

the World object

---

```
initialize,Node-method
```

*Initialize a Node object.*

---

**Description**

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

**Usage**

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

**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. <a href="http://www.w3.org/2001/XMLSchema#string">http://www.w3.org/2001/XMLSchema#string</a>

**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

---

```
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
mimeType	a mime type of the syntax of the model
typeUri	a URI for the syntax of the model

**Details**

The serialization format that are supported by

**Value**

the Parser object

---

initialize, Query-method

*Initialize the Query object.*

---

**Description**

Initialize the Query object.

**Usage**

```
## 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))
```

### 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

### 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**

<http://librdf.org/docs>

**See Also**

This R function is a wrapper function 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**

<http://librdf.org/docs>

**See Also**

This R function is a wrapper function 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	<i>Finish the digesting of data.</i>
---------------------	--------------------------------------

---

**Description**

Finish the digesting of data.

**Usage**

```
librdf_digest_final ( digest )
```

**Arguments**

digest	the digest ("_p_librdf_digest_s")
--------	-----------------------------------

**Value**

void

**References**

<http://librdf.org/docs>

**See Also**

This R function is a wrapper function 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**

<http://librdf.org/docs>

**See Also**

This R function is a wrapper function 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**

<http://librdf.org/docs>

**See Also**

This R function is a wrapper function 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**

<code>digest</code>	the digest ("_p_librdf_digest_s")
<code>buf</code>	the data buffer ("character")
<code>length</code>	the length of the data ("integer")

**Value**

void

**References**

<http://librdf.org/docs>

**See Also**

This R function is a wrapper function 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**

<http://librdf.org/docs>

**See Also**

This R function is a wrapper function 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**

<http://librdf.org/docs>

**See Also**

This R function is a wrapper function 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	<i>Destructor - destroy a librdf_hash object.</i>
------------------	---

---

**Description**

Destructor - destroy a librdf\_hash object.

**Usage**

```
librdf_free_hash ( hash )
```

**Arguments**

hash	hash object ("_p_librdf_hash_s")
------	----------------------------------

**Value**

void

**References**

<http://librdf.org/docs>

**See Also**

This R function is a wrapper function 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**

<http://librdf.org/docs>

**See Also**

This R function is a wrapper function 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**

<http://librdf.org/docs>

**See Also**

This R function is a wrapper function 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	<i>Destructor - destroy an librdf_node object.</i>
------------------	--

---

**Description**

Destructor - destroy an librdf\_node object.

**Usage**

```
librdf_free_node ( r )
```

**Arguments**

r librdf\_node object ("`_p_librdf_node_s`")

**Value**

void

**References**

<http://librdf.org/docs>

**See Also**

This R function is a wrapper function 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	<i>Destructor - destroys a librdf_parser object.</i>
--------------------	--

---

**Description**

Destructor - destroys a librdf\_parser object.

**Usage**

```
librdf_free_parser ( parser )
```

**Arguments**

parser	the parser ("_p_librdf_parser_s")
--------	-----------------------------------

**Value**

void

**References**

<http://librdf.org/docs>

**See Also**

This R function is a wrapper function 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	<i>Destructor - destroy a librdf_query object.</i>
-------------------	--

---

**Description**

Destructor - destroy a librdf\_query object.

**Usage**

```
librdf_free_query ( query )
```

**Arguments**

query	librdf_query object ("_p_librdf_query")
-------	---

**Value**

void

**References**

<http://librdf.org/docs>

**See Also**

This R function is a wrapper function 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**

<http://librdf.org/docs>

**See Also**

This R function is a wrapper function 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**

<http://librdf.org/docs>

**See Also**

This R function is a wrapper function 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**

<http://librdf.org/docs>

**See Also**

This R function is a wrapper function 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	<i>Destructor - destroy a librdf_storage object.</i>
---------------------	--

---

**Description**

Destructor - destroy a librdf\_storage object.

**Usage**

```
librdf_free_storage ( storage )
```

**Arguments**

storage	librdf_storage object (" <code>_p_librdf_storage_s</code> ")
---------	--

**Value**

void

**References**

<http://librdf.org/docs>

**See Also**

This R function is a wrapper function 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	<i>Destructor - destroy an librdf_stream object.</i>
--------------------	--

---

**Description**

Destructor - destroy an librdf\_stream object.

**Usage**

```
librdf_free_stream ( stream )
```

**Arguments**

stream	librdf_stream object ("_p_librdf_stream_s")
--------	---

**Value**

void

**References**

<http://librdf.org/docs>

**See Also**

This R function is a wrapper function 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	<i>Destructor - destroy a librdf_uri object.</i>
-----------------	--

---

**Description**

Destructor - destroy a librdf\_uri object.

**Usage**

```
librdf_free_uri ( uri )
```

**Arguments**

uri	librdf_uri object ("_p_librdf_uri_s")
-----	---------------------------------------

**Value**

void

## References

<http://librdf.org/docs>

## See Also

This R function is a wrapper function 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	<i>Terminate the library and frees all allocated resources.</i>
-------------------	---

---

## 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

<http://librdf.org/docs>

## See Also

This R function is a wrapper function 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 ("_p_librdf_hash_s")
filter	NULL terminated list of keys to ignore ("_p_p_char")

### Value

character

### References

<http://librdf.org/docs>

### See Also

This R function is a wrapper function 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**

<http://librdf.org/docs>

**See Also**

This R function is a wrapper function 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**

<http://librdf.org/docs>

**See Also**

This R function is a wrapper function 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 ("_p_librdf_iterator_s")
.copy	NA

**Value**

integer

**References**

<http://librdf.org/docs>

**See Also**

This R function is a wrapper function 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**

<http://librdf.org/docs>

**See Also**

This R function is a wrapper function 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**

<http://librdf.org/docs>

**See Also**

This R function is a wrapper function 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 librdf_iterator object ("_p_librdf_iterator_s")
<code>.copy</code>	NA

**Value**

integer

**References**

<http://librdf.org/docs>

**See Also**

This R function is a wrapper function 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**

<http://librdf.org/docs>

**See Also**

This R function is a wrapper function 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**

<http://librdf.org/docs>

**See Also**

This R function is a wrapper function 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**

<http://librdf.org/docs>

**See Also**

This R function is a wrapper function 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**

<http://librdf.org/docs>

**See Also**

This R function is a wrapper function 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**

<http://librdf.org/docs>

**See Also**

This R function is a wrapper function 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	Create and add a new statement about a resource to the model.
------------------	---

---

## Description

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

## Usage

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

## Arguments

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

## Value

integer

## References

<http://librdf.org/docs>

## See Also

This R function is a wrapper function 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**

<http://librdf.org/docs>

**See Also**

This R function is a wrapper function 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**<http://librdf.org/docs>**See Also**

This R function is a wrapper function 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**<http://librdf.org/docs>**See Also**

This R function is a wrapper function 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**

<http://librdf.org/docs>

**See Also**

This R function is a wrapper function 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**

<http://librdf.org/docs>

**See Also**

This R function is a wrapper function 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**

<code>model</code>	the model object (" <code>_p_librdf_model_s</code> ")
<code>context</code>	the contest (" <code>_p_librdf_node_s</code> ")
<code>.copy</code>	NA

**Value**

integer

**References**

<http://librdf.org/docs>

**See Also**

This R function is a wrapper function 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**

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

**Value**

integer

**References**

<http://librdf.org/docs>

**See Also**

This R function is a wrapper function 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**

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

**Value**

integer

**References**

<http://librdf.org/docs>

**See Also**

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

---

librdf\_model\_context\_add\_statements

*Add statements to a model with a context.*

---

**Description**

Add statements to a model with a context.

**Usage**

```
librdf_model_context_add_statements ( model,  
  context,  
  stream,  
  .copy )
```

**Arguments**

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

**Value**

integer

**References**

<http://librdf.org/docs>

**See Also**

This R function is a wrapper function 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**

<http://librdf.org/docs>

**See Also**

This R function is a wrapper function 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**

<http://librdf.org/docs>

**See Also**

This R function is a wrapper function 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**

<http://librdf.org/docs>



**See Also**

This R function is a wrapper function 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**

<http://librdf.org/docs>

**See Also**

This R function is a wrapper function 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 ("_p_librdf_model_s")
statement	librdf_statement partial statement to find ("_p_librdf_statement_s")
inNodeOrNull	context librdf_node (or NULL) ("_p_librdf_node_s")

### Value

\_p\_librdf\_stream\_s

### References

<http://librdf.org/docs>

### See Also

This R function is a wrapper function 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 (" <code>_p_librdf_model_s</code> ")
source	librdf_node source (" <code>_p_librdf_node_s</code> ")
target	librdf_node target (" <code>_p_librdf_node_s</code> ")

**Value**

`_p_librdf_node_s`

**References**

<http://librdf.org/docs>

**See Also**

This R function is a wrapper function 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 (" <code>_p_librdf_model_s</code> ")
source	librdf_node source (" <code>_p_librdf_node_s</code> ")
target	librdf_node target (" <code>_p_librdf_node_s</code> ")

**Value**

`_p_librdf_iterator_s`

**References**

<http://librdf.org/docs>

**See Also**

This R function is a wrapper function 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**

<http://librdf.org/docs>

**See Also**

This R function is a wrapper function 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**

<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**

<http://librdf.org/docs>

**See Also**

This R function is a wrapper function 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**

<http://librdf.org/docs>

**See Also**

This R function is a wrapper function 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**

<http://librdf.org/docs>

**See Also**

This R function is a wrapper function 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

<http://librdf.org/docs>

### See Also

This R function is a wrapper function 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**

<http://librdf.org/docs>

**See Also**

This R function is a wrapper function 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**<http://librdf.org/docs>**See Also**

This R function is a wrapper function 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**

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

**Value**`_p_librdf_iterator_s`**References**<http://librdf.org/docs>**See Also**

This R function is a wrapper function 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

<http://librdf.org/docs>

## See Also

This R function is a wrapper function 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

<http://librdf.org/docs>

## See Also

This R function is a wrapper function 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	<i>Load content from a URI into the model.</i>
-------------------	--

---

## 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

<http://librdf.org/docs>

## See Also

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

---

`librdf_model_query_execute`*Execute a query against the model.*

---

**Description**

Execute a query against the model.

**Usage**

```
librdf_model_query_execute ( model,  
  query )
```

**Arguments**

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

**Value**

\_p\_librdf\_query\_results

**References**

<http://librdf.org/docs>

**See Also**

This R function is a wrapper function 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**

<http://librdf.org/docs>

**See Also**

This R function is a wrapper function 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

<http://librdf.org/docs>

## See Also

This R function is a wrapper function 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	<i>Get the number of statements in the model.</i>
-------------------	---

---

## 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

<http://librdf.org/docs>

## See Also

This R function is a wrapper function 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	<i>Synchronise the model to the model implementation.</i>
-------------------	---

---

**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**

<http://librdf.org/docs>

**See Also**

This R function is a wrapper function 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	<i>Write serialized model to a string.</i>
------------------------	--

---

**Description**

Write serialized model to a string.

**Usage**

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



**Arguments**

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

**Value**

character

**References**

<http://librdf.org/docs>

**See Also**

This R function is a wrapper function 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 ("_p_librdf_model_s")
.copy	NA

**Value**

integer

**References**

<http://librdf.org/docs>

**See Also**

This R function is a wrapper function 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**

<http://librdf.org/docs>

**See Also**

This R function is a wrapper function 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

<http://librdf.org/docs>

### See Also

This R function is a wrapper function 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**

<http://librdf.org/docs>

**See Also**

This R function is a wrapper function 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**

<http://librdf.org/docs>

**See Also**

This R function is a wrapper function 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**

<http://librdf.org/docs>

**See Also**

This R function is a wrapper function 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**

<http://librdf.org/docs>

**See Also**

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

---

<code>librdf_new_model</code>	<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**

<http://librdf.org/docs>

**See Also**

This R function is a wrapper function 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**

<http://librdf.org/docs>

**See Also**

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

---

librdf\_new\_model\_with\_options

*Constructor - Create a new librdf\_model with storage.*


---

### Description

Constructor - Create a new librdf\_model with storage.

### Usage

```
librdf_new_model_with_options ( world,
storage,
options )
```

### Arguments

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

### Value

\_p\_librdf\_model\_s

### References

<http://librdf.org/docs>

### See Also

This R function is a wrapper function 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**

<http://librdf.org/docs>

**See Also**

This R function is a wrapper function 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**

<http://librdf.org/docs>

**See Also**

This R function is a wrapper function 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

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

## Value

`_p_librdf_node_s`

## References

<http://librdf.org/docs>

## See Also

This R function is a wrapper function 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**

<http://librdf.org/docs>

**See Also**

This R function is a wrapper function 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**

<http://librdf.org/docs>

**See Also**

This R function is a wrapper function 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

<http://librdf.org/docs>

## See Also

This R function is a wrapper function 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

<http://librdf.org/docs>

## See Also

This R function is a wrapper function 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

<http://librdf.org/docs>

### See Also

This R function is a wrapper function 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**

<http://librdf.org/docs>

**See Also**

This R function is a wrapper function 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**

<http://librdf.org/docs>

**See Also**

This R function is a wrapper function 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**

<http://librdf.org/docs>

**See Also**

This R function is a wrapper function 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**

<http://librdf.org/docs>

**See Also**

This R function is a wrapper function 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 (" <code>_p_librdf_world_s</code> ")
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) (" <code>_p_librdf_uri_s</code> ")

**Value**

`_p_librdf_serializer_s`

**References**

<http://librdf.org/docs>

**See Also**

This R function is a wrapper function 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 (" <code>_p_librdf_world_s</code> ")
-------	---

**Value**

`_p_librdf_statement_s`

**References**

<http://librdf.org/docs>

**See Also**

This R function is a wrapper function 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

<http://librdf.org/docs>

## See Also

This R function is a wrapper function 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

<http://librdf.org/docs>

### See Also

This R function is a wrapper function 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**

<http://librdf.org/docs>

**See Also**

This R function is a wrapper function 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**

<http://librdf.org/docs>

**See Also**

This R function is a wrapper function 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**

<http://librdf.org/docs>

**See Also**

This R function is a wrapper function 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**

<http://librdf.org/docs>

**See Also**

This R function is a wrapper function 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**

<http://librdf.org/docs>

**See Also**

This R function is a wrapper function 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	<i>Create a new Redland execution environment.</i>
------------------	--

---

**Description**

Create a new Redland execution environment.

**Usage**

```
librdf_new_world ( )
```

**Value**

\_p\_librdf\_world\_s

**References**

<http://librdf.org/docs>

**See Also**

This R function is a wrapper function 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	<i>Compare two librdf_node objects for equality.</i>
--------------------	--

---

**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**

<http://librdf.org/docs>

**See Also**

This R function is a wrapper function 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	<i>Get the blank node identifier as a UTF-8 encoded string.</i>
----------------------------------	---

---

**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**

<http://librdf.org/docs>

**See Also**

This R function is a wrapper function 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**

<http://librdf.org/docs>

**See Also**

This R function is a wrapper function 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**

<http://librdf.org/docs>

**See Also**

This R function is a wrapper function 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**

<http://librdf.org/docs>

**See Also**

This R function is a wrapper function 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**

<http://librdf.org/docs>

**See Also**

This R function is a wrapper function 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**

<http://librdf.org/docs>

**See Also**

This R function is a wrapper function 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**

<http://librdf.org/docs>

**See Also**

This R function is a wrapper function 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**

<http://librdf.org/docs>

**See Also**

This R function is a wrapper function 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**

<http://librdf.org/docs>

**See Also**

This R function is a wrapper function 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**

<http://librdf.org/docs>

**See Also**

This R function is a wrapper function 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**

<http://librdf.org/docs>

**See Also**

This R function is a wrapper function 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**

<http://librdf.org/docs>

**See Also**

This R function is a wrapper function 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_to_string` *Format the node as a string in a debugging format.*

---

**Description**

Format the node as a string in a debugging format.

**Usage**

```
librdf_node_to_string ( node )
```

**Arguments**

<code>node</code>	the node object (" <code>_p_librdf_node_s</code> ")
-------------------	---

**Value**

character

**References**

<http://librdf.org/docs>

**See Also**

This R function is a wrapper function 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 (" <code>_p_librdf_world_s</code> ")
name	name of parser ("character")
.copy	NA

**Value**

integer

**References**

<http://librdf.org/docs>

**See Also**

This R function is a wrapper function 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**

<http://librdf.org/docs>

**See Also**

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

---

`librdf_parser_get_feature`*Get the value of a parser feature.*

---

**Description**

Get the value of a parser feature.

**Usage**

```
librdf_parser_get_feature ( parser,  
feature )
```

**Arguments**

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

**Value**

`_p_librdf_node_s`

**References**

<http://librdf.org/docs>

**See Also**

This R function is a wrapper function 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**

<code>parser</code>	librdf_parser object (" <code>_p_librdf_parser_s</code> ")
<code>.copy</code>	NA

**Value**

integer

**References**

<http://librdf.org/docs>

**See Also**

This R function is a wrapper function 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**

<http://librdf.org/docs>

**See Also**

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

---

`librdf_parser_get_namespaces_seen_uri`*Get the uri of namespaces seen during parsing*

---

**Description**

Get the uri of namespaces seen during parsing

**Usage**

```
librdf_parser_get_namespaces_seen_uri ( parser,  
offset )
```

**Arguments**

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

**Value**

\_p\_librdf\_uri\_s

**References**

<http://librdf.org/docs>

**See Also**

This R function is a wrapper function 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\_name

*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_name ( mime_type,  
buffer,  
identifier )
```

**Arguments**

mime_type	MIME type of syntax or NULL ("character")
buffer	content buffer or NULL ("character")
identifier	content identifier or NULL ("character")

**Value**

character

**References**

<http://librdf.org/docs>

**See Also**

This R function is a wrapper function 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**

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

**Value**

character

**References**

<http://librdf.org/docs>

**See Also**

This R function is a wrapper function 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

<http://librdf.org/docs>

### See Also

This R function is a wrapper function 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**

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

**Value**

\_p\_librdf\_stream\_s

**References**

<http://librdf.org/docs>

**See Also**

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

---

librdf\_parser\_parse\_counted\_string\_into\_model

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

---

**Description**

Parse a counted string of content into an librdf\_model.

**Usage**

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

**Arguments**

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

**Value**

integer

**References**

<http://librdf.org/docs>

**See Also**

This R function is a wrapper function 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**

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

**Value**

integer

**References**

<http://librdf.org/docs>

**See Also**

This R function is a wrapper function 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**

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

**Value**

\_p\_librdf\_stream\_s

**References**

<http://librdf.org/docs>

**See Also**

This R function is a wrapper function 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**

<http://librdf.org/docs>

**See Also**

This R function is a wrapper function 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

<code>parser</code>	librdf_parser object (" <code>_p_librdf_parser_s</code> ")
<code>feature</code>	librdf_uri feature property (" <code>_p_librdf_uri_s</code> ")
<code>value</code>	librdf_node feature property value (" <code>_p_librdf_node_s</code> ")
<code>.copy</code>	NA

## Value

integer

## References

<http://librdf.org/docs>

## See Also

This R function is a wrapper function 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

<code>query</code>	librdf_query object (" <code>_p_librdf_query</code> ")
<code>model</code>	model to operate query on (" <code>_p_librdf_model_s</code> ")

### Value

`_p_librdf_query_results`

### References

<http://librdf.org/docs>

### See Also

This R function is a wrapper function 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**

query	librdf_query query object ("_p_librdf_query")
.copy	NA

**Value**

integer

**References**

<http://librdf.org/docs>

**See Also**

This R function is a wrapper function 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**

query	librdf_query query object ("_p_librdf_query")
.copy	NA

**Value**

integer

**References**

<http://librdf.org/docs>

**See Also**

This R function is a wrapper function 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

<http://librdf.org/docs>

### See Also

This R function is a wrapper function 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**

<http://librdf.org/docs>

**See Also**

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

---

librdf\_query\_results\_get\_bindings\_count

*Get the number of bound variables in the result.*

---

**Description**

Get the number of bound variables in the result.

**Usage**

```
librdf_query_results_get_bindings_count ( query_results,  
  .copy )
```

**Arguments**

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

**Value**

integer

**References**

<http://librdf.org/docs>

**See Also**

This R function is a wrapper function 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

<http://librdf.org/docs>

### See Also

This R function is a wrapper function 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**

<http://librdf.org/docs>

**See Also**

This R function is a wrapper function 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 ("_p_librdf_query_results")
name	variable name ("character")

**Value**

\_p\_librdf\_node\_s

**References**

<http://librdf.org/docs>

**See Also**

This R function is a wrapper function 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**

<code>query_results</code>	<code>librdf_query_results query_results ("_p_librdf_query_results")</code>
<code>.copy</code>	<code>NA</code>

**Value**

integer

**References**

<http://librdf.org/docs>

**See Also**

This R function is a wrapper function 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**

<http://librdf.org/docs>

**See Also**

This R function is a wrapper function 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**

<http://librdf.org/docs>

**See Also**

This R function is a wrapper function 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

<http://librdf.org/docs>

### See Also

This R function is a wrapper function 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**

<http://librdf.org/docs>

**See Also**

This R function is a wrapper function 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 ("_p_librdf_query_results")
.copy	NA

**Value**

integer

**References**

<http://librdf.org/docs>

**See Also**

This R function is a wrapper function 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**

<http://librdf.org/docs>

**See Also**

This R function is a wrapper function 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_file`*Write a query results to a file.*

---

**Description**

Write a query results to a file.

**Usage**

```
librdf_query_results_to_file ( query_results,  
  name,  
  format_uri,  
  base_uri,  
  .copy )
```



**Arguments**

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

**Value**

integer

**References**<http://librdf.org/docs>**See Also**

This R function is a wrapper function 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**

<http://librdf.org/docs>

**See Also**

This R function is a wrapper function 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\_string

*Turn a query results into a string.*

---

**Description**

Turn a query results into a string.

**Usage**

```
librdf_query_results_to_string ( query_results,  
                                format_uri,  
                                base_uri )
```

**Arguments**

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

**Value**

character

**References**

<http://librdf.org/docs>

**See Also**

This R function is a wrapper function 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**

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

**Value**

character

**References**

<http://librdf.org/docs>

**See Also**

This R function is a wrapper function 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**

<http://librdf.org/docs>

**See Also**

This R function is a wrapper function 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**

query	librdf_query query object (" <code>_p_librdf_query</code> ")
offset	offset for results, $\geq 0$ to set an offset, $< 0$ to have no offset (" <code>integer</code> ")
.copy	NA

**Value**

integer

**References**

<http://librdf.org/docs>

**See Also**

This R function is a wrapper function 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 (" <code>_p_librdf_world_s</code> ")
name	name of serializer (" <code>character</code> ")
.copy	NA

**Value**

integer

**References**

<http://librdf.org/docs>

**See Also**

This R function is a wrapper function 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**

<http://librdf.org/docs>

**See Also**

This R function is a wrapper function 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

<http://librdf.org/docs>

## See Also

This R function is a wrapper function 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

<http://librdf.org/docs>

### See Also

This R function is a wrapper function 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**

<http://librdf.org/docs>

**See Also**

This R function is a wrapper function 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**

<http://librdf.org/docs>

**See Also**

This R function is a wrapper function 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

<http://librdf.org/docs>

## See Also

This R function is a wrapper function 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

<http://librdf.org/docs>

## See Also

This R function is a wrapper function 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**

<http://librdf.org/docs>

**See Also**

This R function is a wrapper function 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**

<http://librdf.org/docs>

**See Also**

This R function is a wrapper function 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 ("_p_librdf_statement_s")
statement2	second librdf_statement ("_p_librdf_statement_s")
.copy	NA

**Value**

integer

**References**

<http://librdf.org/docs>

**See Also**

This R function is a wrapper function 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**

<http://librdf.org/docs>

**See Also**

This R function is a wrapper function 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**

<http://librdf.org/docs>

**See Also**

This R function is a wrapper function 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**

<http://librdf.org/docs>

**See Also**

This R function is a wrapper function 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**

<code>statement</code>	librdf_statement object (" <code>_p_librdf_statement_s</code> ")
<code>.copy</code>	NA

**Value**

integer

**References**

<http://librdf.org/docs>

**See Also**

This R function is a wrapper function 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 ("_p_librdf_statement_s")
partial_statement	statement with possible empty parts ("_p_librdf_statement_s")
.copy	NA

**Value**

integer

**References**

<http://librdf.org/docs>

**See Also**

This R function is a wrapper function 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 ("_p_librdf_statement_s")
object	librdf_node of object ("_p_librdf_node_s")

**Value**

void

**References**

<http://librdf.org/docs>

**See Also**

This R function is a wrapper function 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 (" <code>_p_librdf_statement_s</code> ")
predicate	librdf_node of predicate (" <code>_p_librdf_node_s</code> ")

**Value**

void

**References**

<http://librdf.org/docs>

**See Also**

This R function is a wrapper function 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 ("_p_librdf_statement_s")
subject	librdf_node of subject ("_p_librdf_node_s")

**Value**

void

**References**

<http://librdf.org/docs>

**See Also**

This R function is a wrapper function 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\_to\_string  
*Format the librdf\_statement as a string.*

---

**Description**

Format the librdf\_statement as a string.

**Usage**

```
librdf_statement_to_string ( statement )
```

**Arguments**

statement      the statement ("\_p\_librdf\_statement\_s")

**Value**

character

**References**

<http://librdf.org/docs>

**See Also**

This R function is a wrapper function 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	<i>Test if the stream has ended.</i>
-------------------	--------------------------------------

---

**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**

<http://librdf.org/docs>

**See Also**

This R function is a wrapper function 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_context`*Get the context of the current object on the stream.*

---

**Description**

Get the context of the current object on the stream.

**Usage**

```
librdf_stream_get_context ( stream )
```

**Arguments**

`stream`                    the librdf\_stream object ("\_p\_librdf\_stream\_s")

**Value**

`_p_librdf_node_s`

**References**

<http://librdf.org/docs>

**See Also**

This R function is a wrapper function 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**

<http://librdf.org/docs>

**See Also**

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

---

<code>librdf_stream_next</code>	<i>Move to the next librdf_statement in the stream.</i>
---------------------------------	---

---

**Description**

Move to the next librdf\_statement in the stream.

**Usage**

```
librdf_stream_next ( stream,  
  .copy )
```

**Arguments**

<code>stream</code>	librdf_stream object (" <code>_p_librdf_stream_s</code> ")
<code>.copy</code>	NA

**Value**

integer

**References**

<http://librdf.org/docs>

**See Also**

This R function is a wrapper function 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	<i>Compare two librdf_uri objects lexicographically.</i>
--------------------	--

---

**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**

<http://librdf.org/docs>

**See Also**

This R function is a wrapper function 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	<i>Compare two librdf_uri objects for equality.</i>
-------------------	---

---

**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**

<http://librdf.org/docs>

**See Also**

This R function is a wrapper function 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**

<http://librdf.org/docs>

**See Also**

This R function is a wrapper function 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**

<code>.copy</code>	NA
--------------------	----

**Value**

integer

**References**

<http://librdf.org/docs>

**See Also**

This R function is a wrapper function 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**

<code>.copy</code>	logical
--------------------	---------

**Value**

integer

**References**

<http://librdf.org/docs>

**See Also**

This R function is a wrapper function 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**

<http://librdf.org/docs>

**See Also**

This R function is a wrapper function 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**

<http://librdf.org/docs>

**See Also**

This R function is a wrapper function 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**

<http://librdf.org/docs>

**See Also**

This R function is a wrapper function 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**

<http://librdf.org/docs>

**See Also**

This R function is a wrapper function 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**

<code>.copy</code>	NA
--------------------	----

**Value**

integer

**References**

<http://librdf.org/docs>

**See Also**

This R function is a wrapper function 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**

<code>.copy</code>	logical
--------------------	---------

**Value**

integer

**References**

<http://librdf.org/docs>

**See Also**

This R function is a wrapper function 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**

<http://librdf.org/docs>

**See Also**

This R function is a wrapper function 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**

<code>.copy</code>	logical
--------------------	---------

**Value**

character

**References**

<http://librdf.org/docs>

**See Also**

This R function is a wrapper function 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**

<code>world</code>	librdf_world object (" <code>_p_librdf_world_s</code> ")
<code>feature</code>	librdf_uri feature property (" <code>_p_librdf_uri_s</code> ")

**Value**

`_p_librdf_node_s`

**References**

<http://librdf.org/docs>

**See Also**

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

---

<code>librdf_world_open</code>	<i>Open a created redland world environment.</i>
--------------------------------	--

---

**Description**

Open a created redland world environment.

**Usage**

```
librdf_world_open ( world )
```

**Arguments**

<code>world</code>	redland world object (" <code>_p_librdf_world_s</code> ")
--------------------	---

**Value**

void

**References**

<http://librdf.org/docs>

**See Also**

This R function is a wrapper function 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

<code>world</code>	librdf_world object (" <code>_p_librdf_world_s</code> ")
<code>feature</code>	librdf_uri feature property (" <code>_p_librdf_uri_s</code> ")
<code>value</code>	librdf_node feature property value (" <code>_p_librdf_node_s</code> ")
<code>.copy</code>	NA

## Value

integer

## References

<http://librdf.org/docs>

## See Also

This R function is a wrapper function 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

<code>world</code>	redland world object (" <code>_p_librdf_world_s</code> ")
<code>user_data</code>	user data to pass to function (" <code>_p_void</code> ")
<code>log_handler</code>	pointer to the function (" <code>_p_librdf_log_func</code> ")

### Value

`void`

### References

<http://librdf.org/docs>

### See Also

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

---

<code>mergeNamespace_roclet</code>	<i>A custom Roxygen roclet that adds Redland RDF functions to NAMESPACE file generated by Roxygen.</i>
------------------------------------	--

---

**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()
```

**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"))
```

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

**Examples**

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

## End(Not run)
```

---

Model-class

*A Redland Model object*


---

**Description**

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

**Details**

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

**Slots**

```
librdf_model A redland model object
```

**Methods**

- [Model-initialize](#): Initialize a Model object
- [addStatement](#): Add a Statement object to the Model
- [freeModel](#): Free memory used by a librdf model object

**See Also**

View examples of creating models by viewing the 'redland\_overview' vignette: `'vignette("redland_overview")'`  
[redland](#): redland package

**Examples**

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

---

Node-class	<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.
- [getBlankNodeId](#): Get the blank identifier that has been assigned for a specified Node object.

**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")
```

---

parseFileIntoModel	<i>Parse the contents of a file into a model</i>
--------------------	--

---

## 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 "rdxml" 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 <http://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

**Methods**

- [Query-initialize](#): Initialize a Query object.
- [executeQuery](#): Execute a query.
- [setQueryResultLimit](#): Set limit on returned query results.
- [getQueryResultLimit](#): Get the query result limit.
- [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)
queryResult <- executeQuery(query, model)
result <- getNextResult(queryResult)
```

---

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

---

## 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.
- `getNextResult`: Get the next query result.
- `freeQueryResults`: Free memory used by a librdf query result.

## 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)
queryResult <- executeQuery(query, model)
result <- getNextResult(queryResult)
```



---

raptor_locator_byte	<i>Get the locator byte offset from locator.</i>
---------------------	--

---

**Description**

Get the locator byte offset from locator

**Usage**

```
raptor_locator_byte ( locator, .copy )
```

**Arguments**

locator	raptor locator ("_p_raptor_locator")
.copy	logical

**Value**

character

**References**

<http://librdf.org/docs>

**See Also**

This R function is a wrapper function 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	<i>Get column number from locator</i>
-----------------------	---------------------------------------

---

**Description**

Get column number from locator

**Usage**

```
raptor_locator_column ( locator,  
  .copy )
```

**Arguments**

locator	raptor locator ("_p_raptor_locator")
.copy	logical

**Value**

integer

**References**

<http://librdf.org/docs>

**See Also**

This R function is a wrapper function 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	<i>Get file name from locator.</i>
---------------------	------------------------------------

---

**Description**

Get file name from locator.

**Usage**

```
raptor_locator_file ( locator )
```

**Arguments**

locator	raptor locator ("_p_raptor_locator")
---------	--------------------------------------

**Value**

character

**References**

<http://librdf.org/docs>

**See Also**

This R function is a wrapper function 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	<i>Get line number from locator.</i>
---------------------	--------------------------------------

---

**Description**

Get line number from locator.

**Usage**

```
raptor_locator_line ( locator, .copy )
```

**Arguments**

locator	raptor locator ("_p_raptor_locator")
.copy	logical

**Value**

integer

**References**

<http://librdf.org/docs>

**See Also**

This R function is a wrapper function 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	<i>Get URI from locator.</i>
--------------------	------------------------------

---

**Description**

Get URI from locator.

**Usage**

```
raptor_locator_uri ( locator )
```

**Arguments**

locator	raptor locator ("_p_raptor_locator")
---------	--------------------------------------

**Value**

character

**References**

<http://librdf.org/docs>

**See Also**

This R function is a wrapper function 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**

<http://librdf.org/docs>

**See Also**

This R function is a wrapper function 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**

<http://librdf.org/docs>

**See Also**

This R function is a wrapper function 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**

<http://librdf.org/docs>

**See Also**

This R function is a wrapper function 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**

<http://librdf.org/docs>

**See Also**

This R function is a wrapper function 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**

<http://librdf.org/docs>

**See Also**

This R function is a wrapper function 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**

<http://librdf.org/docs>

**See Also**

This R function is a wrapper function 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**

<http://librdf.org/docs>

**See Also**

This R function is a wrapper function 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**

<http://librdf.org/docs>

**See Also**

This R function is a wrapper function 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**

<http://librdf.org/docs>

**See Also**

This R function is a wrapper function 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**

<http://librdf.org/docs>

**See Also**

This R function is a wrapper function 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**

<http://librdf.org/docs>

**See Also**

This R function is a wrapper function 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**

<http://librdf.org/docs>

**See Also**

This R function is a wrapper function 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	<i>Rasqal major version number.</i>
----------------------	-------------------------------------

---

**Description**

Rasqal major version number.

**Usage**

```
rasqal_version_major ( .copy )
```

**Arguments**

.copy	logical
-------	---------

**Value**

integer

**References**

<http://librdf.org/docs>

**See Also**

This R function is a wrapper function 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**

<http://librdf.org/docs>

**See Also**

This R function is a wrapper function 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**

<http://librdf.org/docs>

**See Also**

This R function is a wrapper function 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**

<http://librdf.org/docs>

**See Also**

This R function is a wrapper function 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**

<http://librdf.org/docs>

**See Also**

This R function is a wrapper function 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**

<http://librdf.org/docs>

**See Also**

This R function is a wrapper function 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**

<http://librdf.org/docs>

**See Also**

This R function is a wrapper function 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

<http://librdf.org/docs>

### See Also

This R function is a wrapper function 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")', '?getNodeTypes', 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 <http://librdf.org/docs/api/index.html>.

An introduction to RDF can be found at <http://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:

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

```
# Get first (and only, in this case) result
queryResult <- executeQuery(query, model)
result <- getNextResult(queryResult)
```

---

```
roc_output.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

```
roc_output.mergeNamespace(roclet, results, base_path, options = list(), check)
```

## Arguments

roclet	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
options	output options (not used)
check	a check flag (not used)

---

```
roc_process.mergeNamespace
```

*Roxygen process function for the 'mergeNamespace' roclet*

---

## Description

This function is called by the Roxygen2 roxygenize function.

## Usage

```
roc_process.mergeNamespace(roclet, partita, base_path, options = list())
```

## Arguments

roclet	the currently running roclet
partita	a list of all .R files in the package being roxygenized (not used by this roclet)
base_path	the top directory of the R package
options	unused by this roclet

## 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	<i>An RDF Serializer object.</i>
------------------	----------------------------------

---

## 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)
# Creat the default "rdxml" 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	<i>Serialize a model to a character vector.</i>
----------------------	---

---

**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 >= to have a limit, or a value < 0 to have no limit.

---

Statement-class	<i>An RDF Statement object</i>
-----------------	--------------------------------

---

**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")
```

---

Storage-class	<i>A Redland Storage object</i>
---------------	---------------------------------

---

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	<i>A Redland World object, used to initialize the Redland RDF library.</i>
-------------	--

---

**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")
```

---

[,ExternalReference-method	<i>Subset a list of ExternalReferences</i>
----------------------------	--

---

**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

## \*Topic **classes**

- Model-class, [163](#)
- Node-class, [164](#)
- Parser-class, [166](#)
- Query-class, [166](#)
- QueryResults-class, [168](#)
- Serializer-class, [188](#)
- Statement-class, [191](#)
- Storage-class, [192](#)
- World-class, [193](#)
- [, ExternalReference-method, [193](#)
- [<-, ExternalReference-method, [194](#)
- addStatement, [7](#), [163](#), [164](#)
- addStatement, Model, Statement-method  
(addStatement), [7](#)
- executeQuery, [8](#), [167](#)
- executeQuery, Query-method  
(executeQuery), [8](#)
- freeModel, [8](#), [164](#)
- freeModel, Model-method (freeModel), [8](#)
- freeParser, [9](#), [166](#), [167](#)
- freeParser, Parser-method (freeParser), [9](#)
- freeQuery, [10](#)
- freeQuery, Query-method (freeQuery), [10](#)
- freeQueryResults, [11](#), [168](#)
- freeQueryResults, QueryResults-method  
(freeQueryResults), [11](#)
- freeSerializer, [12](#), [188](#)
- freeSerializer, Serializer-method  
(freeSerializer), [12](#)
- freeStatement, [13](#), [191](#)
- freeStatement, Statement-method  
(freeStatement), [13](#)
- freeStorage, [13](#), [192](#)
- freeStorage, Storage-method  
(freeStorage), [13](#)
- freeWorld, [14](#), [193](#)
- freeWorld, World-method (freeWorld), [14](#)
- getBlankNodeId, [15](#), [164](#)
- getBlankNodeId, Node-method  
(getBlankNodeId), [15](#)
- getNextResult, [15](#), [168](#)
- getNextResult, QueryResults-method  
(getNextResult), [15](#)
- getNodeType, [16](#), [164](#)
- getNodeType, Node-method (getNodeType),  
[16](#)
- getQueryResultLimit, [17](#), [167](#)
- getQueryResultLimit, Query-method  
(getQueryResultLimit), [17](#)
- getTermType, [17](#), [191](#)
- getTermType, Statement, character-method  
(getTermType), [17](#)
- initialize, Model-method, [18](#)
- initialize, Node-method, [18](#)
- initialize, Parser-method, [19](#)
- initialize, Query-method, [20](#)
- initialize, QueryResults-method, [20](#)
- initialize, Serializer-method, [21](#)
- initialize, Statement-method, [22](#)
- initialize, Storage-method, [22](#)
- initialize, World-method, [23](#)
- is.null.externalptr, [24](#)
- length, SWIGArray-method, [24](#)
- librdf\_copyright\_string, [25](#)
- librdf\_copyright\_string\_get, [25](#)
- librdf\_digest\_final, [26](#)
- librdf\_digest\_init, [27](#)
- librdf\_digest\_to\_string, [27](#)
- librdf\_digest\_update, [28](#)
- librdf\_digest\_update\_string, [29](#)
- librdf\_free\_digest, [29](#)
- librdf\_free\_hash, [30](#)
- librdf\_free\_iterator, [31](#)

- `librdf_free_model`, 31
- `librdf_free_node`, 32
- `librdf_free_parser`, 33
- `librdf_free_query`, 33
- `librdf_free_query_results`, 34
- `librdf_free_serializer`, 35
- `librdf_free_statement`, 35
- `librdf_free_storage`, 36
- `librdf_free_stream`, 37
- `librdf_free_uri`, 37
- `librdf_free_world`, 38
- `librdf_hash_to_string`, 39
- `librdf_internal_test_error`, 39
- `librdf_internal_test_warning`, 40
- `librdf_iterator_end`, 41
- `librdf_iterator_get_context`, 41
- `librdf_iterator_get_object`, 42
- `librdf_iterator_next`, 43
- `librdf_log_message_code`, 43
- `librdf_log_message_facility`, 44
- `librdf_log_message_level`, 45
- `librdf_log_message_locator`, 45
- `librdf_log_message_message`, 46
- `librdf_model_add`, 47
- `librdf_model_add_statement`, 48
- `librdf_model_add_statements`, 48
- `librdf_model_add_string_literal_statement`, 49
- `librdf_model_add_typed_literal_statement`, 50
- `librdf_model_as_stream`, 51
- `librdf_model_contains_context`, 52
- `librdf_model_contains_statement`, 52
- `librdf_model_context_add_statement`, 53
- `librdf_model_context_add_statements`, 54
- `librdf_model_context_as_stream`, 55
- `librdf_model_context_remove_statement`, 55
- `librdf_model_context_remove_statements`, 56
- `librdf_model_find_statements`, 57
- `librdf_model_find_statements_in_context`, 58
- `librdf_model_get_arc`, 58
- `librdf_model_get_arcs`, 59
- `librdf_model_get_arcs_in`, 60
- `librdf_model_get_arcs_out`, 61
- `librdf_model_get_contexts`, 61
- `librdf_model_get_feature`, 62
- `librdf_model_get_source`, 63
- `librdf_model_get_sources`, 63
- `librdf_model_get_target`, 64
- `librdf_model_get_targets`, 65
- `librdf_model_has_arc_in`, 66
- `librdf_model_has_arc_out`, 67
- `librdf_model_load`, 68
- `librdf_model_query_execute`, 69
- `librdf_model_remove_statement`, 69
- `librdf_model_set_feature`, 70
- `librdf_model_size`, 71
- `librdf_model_sync`, 72
- `librdf_model_to_string`, 72
- `librdf_model_transaction_commit`, 73
- `librdf_model_transaction_rollback`, 74
- `librdf_model_transaction_start`, 75
- `librdf_new_digest`, 75
- `librdf_new_hash`, 76
- `librdf_new_hash_from_array_of_strings`, 77
- `librdf_new_hash_from_string`, 77
- `librdf_new_model`, 78
- `librdf_new_model_from_model`, 79
- `librdf_new_model_with_options`, 80
- `librdf_new_node`, 80
- `librdf_new_node_from_blank_identifier`, 81
- `librdf_new_node_from_literal`, 82
- `librdf_new_node_from_node`, 83
- `librdf_new_node_from_normalised_uri_string`, 83
- `librdf_new_node_from_typed_literal`, 84
- `librdf_new_node_from_uri`, 85
- `librdf_new_node_from_uri_local_name`, 86
- `librdf_new_node_from_uri_string`, 86
- `librdf_new_parser`, 87
- `librdf_new_query`, 88
- `librdf_new_query_from_query`, 89
- `librdf_new_serializer`, 89
- `librdf_new_statement`, 90
- `librdf_new_statement_from_nodes`, 91
- `librdf_new_statement_from_statement`, 92
- `librdf_new_storage`, 92
- `librdf_new_storage_from_storage`, 93

librdf\_new\_uri, 94  
librdf\_new\_uri\_from\_filename, 95  
librdf\_new\_uri\_from\_uri, 95  
librdf\_new\_world, 96  
librdf\_node\_equals, 97  
librdf\_node\_get\_blank\_identifier, 97  
librdf\_node\_get\_li\_ordinal, 101  
librdf\_node\_get\_literal\_value, 98  
librdf\_node\_get\_literal\_value\_as\_latin1, 99  
librdf\_node\_get\_literal\_value\_datatype\_uri, 99  
librdf\_node\_get\_literal\_value\_is\_wf\_xml, 100  
librdf\_node\_get\_literal\_value\_language, 101  
librdf\_node\_get\_type, 102  
librdf\_node\_get\_uri, 103  
librdf\_node\_is\_blank, 103  
librdf\_node\_is\_literal, 104  
librdf\_node\_is\_resource, 105  
librdf\_node\_to\_string, 105  
librdf\_parser\_check\_name, 106  
librdf\_parser\_get\_accept\_header, 107  
librdf\_parser\_get\_feature, 107  
librdf\_parser\_get\_namespaces\_seen\_count, 108  
librdf\_parser\_get\_namespaces\_seen\_prefix, 109  
librdf\_parser\_get\_namespaces\_seen\_uri, 109  
librdf\_parser\_guess\_name, 110  
librdf\_parser\_guess\_name2, 111  
librdf\_parser\_parse\_as\_stream, 112  
librdf\_parser\_parse\_counted\_string\_as\_stream, 112  
librdf\_parser\_parse\_counted\_string\_into\_model, 113  
librdf\_parser\_parse\_into\_model, 114  
librdf\_parser\_parse\_string\_as\_stream, 115  
librdf\_parser\_parse\_string\_into\_model, 116  
librdf\_parser\_set\_feature, 117  
librdf\_query\_execute, 118  
librdf\_query\_get\_limit, 118  
librdf\_query\_get\_offset, 119  
librdf\_query\_results\_as\_stream, 120  
librdf\_query\_results\_finished, 120  
librdf\_query\_results\_get\_binding\_name, 122  
librdf\_query\_results\_get\_binding\_value, 122  
librdf\_query\_results\_get\_binding\_value\_by\_name, 123  
librdf\_query\_results\_get\_bindings\_count, 121  
librdf\_query\_results\_get\_boolean, 124  
librdf\_query\_results\_get\_count, 124  
librdf\_query\_results\_is\_bindings, 125  
librdf\_query\_results\_is\_boolean, 126  
librdf\_query\_results\_is\_graph, 126  
librdf\_query\_results\_is\_syntax, 127  
librdf\_query\_results\_next, 128  
librdf\_query\_results\_to\_file, 128  
librdf\_query\_results\_to\_file2, 129  
librdf\_query\_results\_to\_string, 130  
librdf\_query\_results\_to\_string2, 131  
librdf\_query\_set\_limit, 132  
librdf\_query\_set\_offset, 132  
librdf\_serializer\_check\_name, 133  
librdf\_serializer\_get\_feature, 134  
librdf\_serializer\_serialize\_model\_to\_file, 135  
librdf\_serializer\_serialize\_model\_to\_string, 136  
librdf\_serializer\_serialize\_stream\_to\_file, 136  
librdf\_serializer\_serialize\_stream\_to\_string, 137  
librdf\_serializer\_set\_feature, 138  
librdf\_serializer\_set\_namespace, 139  
librdf\_short\_copyright\_string, 140  
librdf\_short\_copyright\_string\_get, 140  
librdf\_statement\_equals, 141  
librdf\_statement\_get\_object, 142  
librdf\_statement\_get\_predicate, 142  
librdf\_statement\_get\_subject, 143  
librdf\_statement\_is\_complete, 144  
librdf\_statement\_match, 144  
librdf\_statement\_set\_object, 145  
librdf\_statement\_set\_predicate, 146  
librdf\_statement\_set\_subject, 147  
librdf\_statement\_to\_string, 147  
librdf\_stream\_end, 148  
librdf\_stream\_get\_context, 149

- librdf\_stream\_get\_object, [149](#)
- librdf\_stream\_next, [150](#)
- librdf\_uri\_compare, [151](#)
- librdf\_uri\_equals, [151](#)
- librdf\_uri\_to\_string, [152](#)
- librdf\_version\_decimal, [153](#)
- librdf\_version\_decimal\_get, [153](#)
- librdf\_version\_major, [154](#)
- librdf\_version\_major\_get, [155](#)
- librdf\_version\_minor, [155](#)
- librdf\_version\_minor\_get, [156](#)
- librdf\_version\_release, [157](#)
- librdf\_version\_release\_get, [157](#)
- librdf\_version\_string, [158](#)
- librdf\_version\_string\_get, [159](#)
- librdf\_world\_get\_feature, [159](#)
- librdf\_world\_open, [160](#)
- librdf\_world\_set\_feature, [161](#)
- librdf\_world\_set\_logger, [162](#)
  
- mergeNamespace\_roclet, [162](#)
- Model (Model-class), [163](#)
- Model-class, [163](#)
- Model-initialize
  - (initialize, Model-method), [18](#)
  
- Node (Node-class), [164](#)
- Node-class, [164](#)
- Node-initialize
  - (initialize, Node-method), [18](#)
  
- parseFileIntoModel, [163](#), [165](#), [166](#)
- parseFileIntoModel, Parser, World, character, Model-method
  - (parseFileIntoModel), [165](#)
- Parser (Parser-class), [166](#)
- Parser-class, [166](#)
- Parser-initialize
  - (initialize, Parser-method), [19](#)
  
- Query, [163](#)
- Query (Query-class), [166](#)
- Query-class, [166](#)
- Query-initialize
  - (initialize, Query-method), [20](#)
- QueryResults (QueryResults-class), [168](#)
- QueryResults-class, [168](#)
- QueryResults-initialize
  - (initialize, QueryResults-method), [20](#)
  
- raptor\_locator\_byte, [169](#)
- raptor\_locator\_column, [169](#)
- raptor\_locator\_file, [170](#)
- raptor\_locator\_line, [171](#)
- raptor\_locator\_uri, [171](#)
- raptor\_version\_decimal, [172](#)
- raptor\_version\_decimal\_get, [173](#)
- raptor\_version\_major, [173](#)
- raptor\_version\_major\_get, [174](#)
- raptor\_version\_minor, [175](#)
- raptor\_version\_minor\_get, [175](#)
- raptor\_version\_release, [176](#)
- raptor\_version\_release\_get, [177](#)
- raptor\_version\_string, [177](#)
- raptor\_version\_string\_get, [178](#)
- rasqal\_version\_decimal, [179](#)
- rasqal\_version\_decimal\_get, [179](#)
- rasqal\_version\_major, [180](#)
- rasqal\_version\_major\_get, [181](#)
- rasqal\_version\_minor, [181](#)
- rasqal\_version\_minor\_get, [182](#)
- rasqal\_version\_release, [183](#)
- rasqal\_version\_release\_get, [183](#)
- rasqal\_version\_string, [184](#)
- rasqal\_version\_string\_get, [185](#)
- redland, [164](#), [166–168](#), [185](#), [188](#), [192](#), [193](#)
- redland-package (redland), [185](#)
- redland::Model, [186](#)
- redland::Node, [186](#)
- redland::Parser, [186](#)
- redland::Query, [186](#)
- redland::QueryResults, [186](#)
- redland::Serializer, [186](#)
- redland::Statement, [186](#)
- redland::Storage, [186](#)
- redland::World, [186](#)
- roc\_output.mergeNamespace, [187](#)
- roc\_process.mergeNamespace, [187](#)
  
- Serializer (Serializer-class), [188](#)
- Serializer-class, [188](#)
- Serializer-initialize
  - (initialize, Serializer-method), [21](#)
- serializeToCharacter, [188](#), [189](#)
- serializeToCharacter, Serializer, World, Model-method
  - (serializeToCharacter), [189](#)
- serializeToFile, [188](#), [189](#)

serializeToFile, Serializer, World, Model, character-method  
    (serializeToFile), [189](#)  
setNameSpace, [188](#), [190](#)  
setNameSpace, Serializer, World, character, character-method  
    (setNameSpace), [190](#)  
setQueryResultLimit, [167](#), [191](#)  
setQueryResultLimit, Query-method  
    (setQueryResultLimit), [191](#)  
setQueryResultsLimit  
    (setQueryResultLimit), [191](#)  
Statement, [163](#)  
Statement (Statement-class), [191](#)  
Statement-class, [191](#)  
Statement-initialize  
    (initialize, Statement-method),  
    [22](#)  
Storage (Storage-class), [192](#)  
Storage-class, [192](#)  
Storage-initialize  
    (initialize, Storage-method), [22](#)  
  
World (World-class), [193](#)  
World-class, [193](#)  
World-initialize  
    (initialize, World-method), [23](#)