A1. RDF/RDFS Language Reference

RDF Node

- **rdfs:Resource** — *the generic class of identified concept*

rdf:type	[rdfs:Resource → rdf:Class]	*membership*
rdfs:label	[rdfs:Resource → rdfs:Literal]	*annotation*
rdfs:comment	[rdfs:Resource → rdfs:Literal]	*annotation*
rdfs:seeAlso	[rdfs:Resource → rdfs:Resource]	*annotation*
rdfs:isDefinedBy	[rdfs:Resource → rdfs:Resource]	*annotation*
rdf:value	[rdfs:Resource → rdfs:Resource]	*complex values*

- **rdfs:Literal** — *the generic class of literal values*
- **rdf:XMLLiteral** — *the class of typed literals (c.f. XMLSchema)*

Class

- **rdfs:Class** — *the class of rdf classes*

rdfs:subClassOf	[rdfs:Class → rdfs:Class]	*subset relation*

Property

- **rdf:Property** — *the class of properties(i.e. binary relations)*

rdfs:subPropertyOf	[rdf:Property→ rdf:Property]
rdfs:domain	[rdf:Property→ rdfs:Class]
rdfs:range	[rdf:Property→ rdfs:Class]

Containers

- **rdfs:Container** — *the generic superclass of rdf resource containers*

rdfs:member	[rdfs:Resource → rdfs:Resrouce]	*membership*
rdf:_1, rdf_2, …	Sub-properties of rdf:member	

- **rdf:Alt** — *container of alternatives*
- **rdf:Bag** — *unordered container*
- **rdf:Seq** — *ordered container*
- **rdfs:ContainerMembershipProperty** — *all sub-properties of rdfs:member*

List

- **rdf:List** — *the class of RDF Lists*

rdf:first	[rdf:List → rdfs:Resource]	*car*
rdf:rest	[rdf:List → rdfs:List]	*cdr*

- **rdf:nil** — an instance of RDF:List representing the empty list

Datatype

- **rdfs:Datatype** — *the class of datatypes*

RDF Reification

- **rdf:Statement** — *the class of RDF statements*

rdf:subject	[rdf:Statement → rdfs:Resource]
rdf:predicate	[rdf:Statement → rdfs:Resource]
rdf:object	[rdf:Statement → rdfs:Resource]

Supported XML datatypes

xsd:decimal	xsd:negativeInteger	xsd:anyURI	xsd:date	xsd:string
xsd:double	xsd:positiveInteger	xsd:base64Binary	xsd:dateTime	xsd:normalizedString
xsd:float	xsd:nonPositiveInteger	xsd:boolean	xsd:time	xsd:token
xsd:int	xsd:nonNegativeInteger	xsd:byte	xsd:gYearMonth	xsd:language
xsd:integer	xsd:unsignedLong	xsd:hexBinary	xsd:gYear	xsd:NMTOKEN
xsd:long	xsd:unsignedInt	xsd:unsignedByte	xsd:gMonthDay	xsd:Name
xsd:short	xsd:unsignedShort		xsd:gDay	xsd:NCName
			xsd:gMonth	

A2. OWL Language Reference

Classes

- **owl:Class** — *all OWL classes, a sub-class of rdfs:Class*

owl:equivalentClass	[owl:Class → owl:Class]
owl:disjointWith *	[owl:Class → owl:Class]
owl:oneOf *	[rdfs:Class → rdf:List]
owl:intersectionOf -	[owl:Class → rdf:List]
owl:unionOf *	[owl:Class → rdf:List]
owl:complementOf *	[owl:Class → owl:Class]

  *NOTE:: * means only supported in OWL DL/FULL. – means supported in OWL Lite with restrictions*

- **owl:Restriction**

owl:onProperty	[owl:Restriction → rdf:Property]
owl:allValuesFrom	[owl:Restriction → rdfs:Class]
owl:someValuesFrom	[owl:Restriction → rdfs:Class]
owl:hasValue *	[owl:Restriction →] *no range constraint*
owl:cardinality -	[owl:Restriction → xsd:nonNegativeInteger]
owl:maxCardinality -	[owl:Restriction → xsd:nonNegativeInteger]
owl:minCardinality -	[owl:Restriction → xsd:nonNegativeInteger]

- **owl:DataRange *** — *sets of data values, range of data-valued property*
- **owl:DeprecatedClass** — *version control*

Properties

- **owl:DatatypeProperty** — *range is instance of rdfs:Datatype*
- **owl:ObjectProperty** — *range is instance of owl:Class*

owl:inverseOf	[owl:ObjectProperty → owl:ObjectProperty]

- **owl:OntologyProperty** — *domain/range are owl:Ontology*
- **owl:AnnotationProperty** — *range is rdfs:Literal*
- **owl:FunctionalProperty** — *(s,p,o1) , (s, p,o2) =>sameAs(o1, o2)*
- **owl:InverseFunctionalProperty** — *(s1,p,o) , (s2, p,o) => sameAs(s1, s2)*
- **owl:SymmetricProperty** — *(s,p,o) => (o,p,s)*
- **owl:TransitiveProperty** — *(a,p,b) (b,p,c) => (a,p,c)*
- **owl:DeprecatedProperty** — *version control*

owl:equivalentProperty	[rdf:Property → rdf:Property]

Special classes

- **owl:Thing** — *all OWL individuals*

owl:differentFrom	[owl:Thing → owl:Thing]
owl:sameAs	[owl:Thing → owl:Thing]

- **owl:Nothing** — *the complement of owl:Thing*
- **owl:AllDifferent** — *OWL built-in*

owl:distinctMembers	[owl:AllDifferent → rdf:List]	*OWL built-in*

Ontology

- **owl:Ontology** — *ontology description*

owl:backwardCompatibleWith	[owl:Ontology → owl:Ontology]
owl:imports	[owl:Ontology → owl:Ontology]
owl:incompatibleWith	[owl:Ontology → owl:Ontology]
owl:priorVersion	[owl:Ontology → owl:Ontology]
owl:versionInfo	[→] *no domain or range constraint*

A3. Popular Namespaces and Terms

prefix	URI and popular classes/properties	
rdf	http://www.w3.org/1999/02/22-rdf-syntax-ns#	
	Property	type
	Seq	first
	List	rest
rdfs	http://www.w3.org/2000/01/rdf-schema#	
	Resource	seeAlso
	Class	label
		subClassOf
		subPropertyOf
owl	http://www.w3.org/2002/07/owl#	
	Class	onProperty
	Restriction	imports
	ObjectProperty	equivalentClass
	DatatypeProperty	equivalentProperty
	Ontology	sameAs
	Thing	
admin	http://webns.net/mvcb/	
		generatorAgent
cc	http://web.resource.org/cc/	
	Agent	license
	License	
daml	http://www.daml.org/2001/03/daml+oil#	
dc	http://purl.org/dc/elements/1.1/	
		date
		creator
		language
		subject
		description
		title
		source
dcterms	http://purl.org/dc/terms/	
		created
		issued
foaf	http://xmlns.com/foaf/0.1/	
	Agent	mbox_sha1sum
	Person	name
	Document	homepage
	Image	mbox
		knows
		nick
		depiction
		firstName
		surname
		interest
geo	http://www.w3.org/2003/01/geo/wgs84_pos#	
	Point	lat
		long
rss	http://purl.org/rss/1.0/	
	channel	items
	image	description
	item	link
		title
ruml	http://www.w3.org/2003/11/ruleml#	
service	http://www.daml.org/services/owl-s/1.1/Service.owl#	
	ServiceModel	presents
	ServiceProfile	describedBy
	ServiceGrounding	supports
swrl	http://www.w3.org/2003/11/swrl#	
wn	http://xmlns.com/wordnet/1.6/	

ebiquity Research Group

Semantic Web Reference Card v1.0

http://ebiquity.umbc.edu/

B1. RDF/XML Syntax Reference

Reserved Terms

- **rdf:RDF** *special XML element containing a serialized RDF graph*
- **rdf:Description** *node element*
- **rdf:resource** *leaf node element in XML parse tree*
- **rdf:ID** *ID of node, local name, augmented by xml:base (global)*
  NOTE: the rdf:ID in property element will add a reified RDF statement for the triple
- **rdf:about** *ID of node, URIref, like hyperlink, (global)*
- **rdf:nodeID** *ID of blank node, local name (local)*
- **rdf:datatype** *shows the object node of a predicate is a typed literal*
- **rdf:parseType="Literal"** *what follows should be parsed as literal*
- **rdf:parseType="resource"** *omits a blank node (predicate→predicate)*
- **rdf:parseType="Collection"** *lets property element contain multiple nodes*
- **rdf:li** *container membership, similar to rdf:_1, rdf:_2…*
- **xml:base** *applies to rdf:about, rdf:resource, rdf:ID and rdf:datatype*
- **xml:lang** *identification of content language*

source: http://www.w3.org/TR/rdf-syntax-grammar/)

A partial grammar (RELAX NG Compact)

```
rdf-xml-Doc          = RDF | nodeElement
RDF                  = element rdf:RDF { xmlbase?, nodeElementList }
nodeElementList      = nodeElement*
nodeElement          = element …… { ( idAttr | nodeIdAttr | aboutAttr )?,
                           xmllang?, xmlbase?, propertyAttr*, propertyEltList }
propertyEltList      = propertyElt*
propertyElt          = resourcePropertyElt | literalPropertyElt | ……
resourcePropertyElt  = element …… { idAttr?,
                           xmllang?, xmlbase?, nodeElement }
literalPropertyElt   = element* …… { (idAttr | datatypeAttr )?,
                           xmllang?, xmlbase?, text }
idAttr               = attribute rdf:ID { IDsymbol }
nodeIdAttr           = attribute rdf:nodeID { IDsymbol }
aboutAttr            = attribute rdf:about { URI-reference }
propertyAttr         = attribute * …… { string }
resourceAttr         = attribute rdf:resource { URI-reference }
datatypeAttr         = attribute rdf:datatype { URI-reference }
IDsymbol             = xsd:NMTOKEN
```
Source: http://www.w3.org/TR/rdf-syntax-grammar/#section-RELAXNG-Schema

Example. An RDF document ontology

```xml
<?xml version="1.0"  encoding="utf-8"?>
<rdf:RDF  xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#"
         xmlns:rdfs="http://www.w3.org/2000/01/rdf-schema#"
         xmlns:owl="http://www.w3.org/2002/07/owl#"
         xml:base = "http://example.org/ex.owl"
         xmlns = "http://example.org/ex.owl#">
<owl:Ontology rdf:about="">
   <rdfs:label xml:lang="en-US">Example Ontology</rdfs:label>
   <versionInfo>16 February 2005</versionInfo>
   <owl:imports rdf:resource="http://xmlns.com/foaf/0.1/" />
</owl:Ontology>
<owl:Class rdf:ID="RDFDocument">
   <rdfs:subClassOf  rdf:resource="http://xmlns.com/foaf/0.1/Document"/>
   <rdfs:label xml:lang="en-US">RDF Document</rdfs:label>
   <rdfs:comment xml:lang="en-US"> All RDF documents.  </rdfs:comment>
</owl:Class>
</rdf:RDF>
```

B2. Notation 3 Grammar for RDF (EBNF)

```
n3doc-rdf      ::=  (declaration '.')* (existential '.')* ( statement '.' )*
declaration    ::=  '@keywords' ( ',' barename)+ | '@prefix' qname
                        explicituri
existential    ::=  '@forSome' ( ',' symbol )+
statement      ::=  subject propertylist?
subject        ::=  (node_subject '!' pathtail* | (numericliteral |string)
                        pathtail+
object         ::=  node_object pathtail*
node_subject   ::=  symbol  | '(' object+ ')' | '[' propertylist ']'
node_object    ::=  numericliteral | string | symbol | '[' propertylist? ']'
pathtail       ::=  ('!' | '^' ) verb
propertylist   ::=  property  ( ';' property )*
property       ::=  verb object ( ',' object )*
verb           ::=  prop | '=' | '@a' | '@has' prop
symbol         ::=  explicituri | qname

alphanumeric   ::=  [a-zA-Z][a-zA-Z0-9_]*
barename       ::=  [a-zA-Z_][a-zA-Z0-9_]*
explicituri    ::=  <[^>]*>
langcode       ::=  [a-z]+(-[a-z0-9]+)*
numericliteral ::=  [-+]?[0-9]+(\.[0-9]+)?(e[-+]?[0-9]+)?
prop
qname          ::=  (([a-zA-Z_][a-zA-Z0-9_]*)?:)?([a-zA-Z_][a-zA-Z0-9_]*)?
string         ::=  ("""[^"\\]*(?:(?:\\.|"(?!""))[^"\\]*)*""")|("[^"\\]*(?:\\.[^"\\]*)*")
```
Source: http://www.w3.org/2000/10/swap/grammar/n3rdf-report.html#declaration_s

Example. John's homepage (N3 version)

```
@prefix rdfs:   <http://www.w3.org/2000/01/rdf-schema#> .
@prefix rdf:    <http://www.w3.org/1999/02/22-rdf-syntax-ns#> .
@prefix foaf:   <http://xmlns.com/foaf/0.1/> .
@prefix :       <#> .

<http://example.org/john.html>
    a       foaf:Document .

[ ]  a       foaf:Person ;
     foaf:homepage <http://example.org/john.html> ;
     foaf:name "John Doe" .
```

Example. John's homepage (RDF/XML version)

```xml
<?xml version="1.0"  encoding="utf-8"?>
<rdf:RDF  xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#"
         xmlns:rdfs="http://www.w3.org/2000/01/rdf-schema#"
         xmlns:foaf = "http://xmlns.com/foaf/0.1/" >
<rdf:Description>
   <rdf:type rdf:resource="http://xmlns.com/foaf/0.1/Person"/>
   <foaf:name>John Doe</foaf:name>
   <foaf:homepage>
      <foaf:Document rdf:about="http://example.org/john.html"/>
   </foaf:homepage>
</rdf:Description>
</rdf:RDF>
```

B3. N-Triples Grammar (EBNF)

```
ntripleDoc   ::=  line*
line         ::=  ws* ( comment | triple )? eoln
comment      ::=  '#' ( character - ( cr | lf ) )*
triple       ::=  subject ws+ predicate ws+ object ws* '.' ws*
subject      ::=  uriref | nodeID
predicate    ::=  uriref
object       ::=  uriref | nodeID | literal
uriref       ::=  '<' absoluteURI '>'
nodeID       ::=  '_:' name
literal      ::=  langString | datatypeString
langString   ::=  '"' string '"' ( '@' language )?
datatypeString ::= '"' string '"' '^^' uriref
language     ::=  [a-z]+ ('-' [a-z0-9]+ )*
                     encoding a language tag.
ws           ::=  space | tab
eoln         ::=  cr | lf | cr lf
space        ::=  #x20 /* US-ASCII space - decimal 32 */
cr           ::=  #xD /* US-ASCII carriage return - decimal 13 */
lf           ::=  #xA /* US-ASCII line feed - decimal 10 */
tab          ::=  #x9 /* US-ASCII horizontal tab - decimal 9 */
string       ::=  character* with escapes as defined in Strings
name         ::=  [A-Za-z][A-Za-z0-9]*
absoluteURI  ::=  character+ with escapes (c.f. URI References)
character    ::=  [#x20-#x7E] /* US-ASCII space to decimal 126 */
```
Source: http://www.w3.org/TR/rdf-testcases/#ntriples

Example. John's homepage (N-Triples version)

Line1:
```
_:x  <http://www.w3.org/1999/02/22-rdf-syntax-ns#type>
     <http://xmlns.com/foaf/0.1/Person> .
```
Line2:
```
_:x <http://xmlns.com/foaf/0.1/homepage>
    <http://example.org/john.html> .
```
Line3:
```
_:x  <http://xmlns.com/foaf/0.1/name>   "John Doe".
```
Line4:
```
<http://example.org/john.html>
<http://www.w3.org/1999/02/22-rdf-syntax-ns#type>
<http://xmlns.com/foaf/0.1/Document > .
```

Tips

- MIME type for RDF/XML is "application/rdf+xml".
- Extensions: RDF/XML → ".rdf", N3 → ".n3", N-Triple → ".nt", OWL → ".owl"
- Popular RDF editors: Swoop, Protégé
- Popular SW search engines: Swoogle, Semantic Web Search, SchemaWeb
- Popular RDF APIs: Jena's ARP, Redland, Sesame's Rio, RDF Lib
- Popular RDF/OWL-DL reasoners: Jena, JTP, Pellet, FaCT++, Racer
- Popular RDF storage: Kowari, Sesame, 3store, DLDB, Jena