W3C Editor's Draft

OWL 2 Web Ontology Language: Quick Reference Guide

W3C Editor's Draft 02 December 2008

This version:
http://www.w3.org/2007/OWL/draft/ED-owl2-quick-reference-20081202/
Latest editor's draft:
http://www.w3.org/2007/OWL/draft/owl2-quick-reference/
Previous version:
http://www.w3.org/2007/OWL/draft/ED-owl2-quick-reference-20081128/
(color-coded diff)
Authors:
Jie Bao, Rensselaer Polytechnic Institute
Elisa F. Kendall, Sandpiper Software, Inc.
Deborah L. McGuinness, Rensselaer Polytechnic Institute
Evan K. Wallace, NIST
Contributors:
Li Ding, Rensselaer Polytechnic Institute
Ankesh Khandelwal, Rensselaer Polytechnic Institute
Peter F. Patel-Schneider, Bell Labs Research, Alcatel-Lucent

This document is also available in these non-normative formats: PDF version.

Abstract

OWL 2 extends the W3C OWL Web Ontology Language with a small but useful set of features that have been requested by users, for which effective reasoning algorithms are now available, and that OWL tool developers are willing to support. The new features include extra syntactic sugar, additional property and qualified cardinality constructors, extended datatype support, simple metamodeling, and extended annotations.
This document is intended to provide a quick reference to the OWL 2 language, similar to what was provided in the Language Synopsis section of the OWL Web Ontology Language Overview. Inspiration for this effort includes work by the ebiquity Research Group at the University of Maryland Baltimore County (UMBC)

W3C Editor's Draft

on earlier versions of a Reference Card for the Semantic Web.
A draft printable version is available [1], but it is obsolete with respect to the current wiki version.

Status of this Document

May Be Superseded

This section describes the status of this document at the time of its publication. Other documents may supersede this document. A list of current W3C publications and the latest revision of this technical report can be found in the W3C technical reports index at http://www.w3.org/TR/.

Set of Documents

This document is being published as one of a set of 11 documents:

1. Structural Specification and Functional-Style Syntax
2. Direct Semantics
3. RDF-Based Semantics
4. Conformance and Test Cases
5. Mapping to RDF Graphs
6. XML Serialization
7. Profiles
8. Quick Reference Guide (this document)
9. New Features and Rationale
10. Manchester Syntax
11. rdf:text: A Datatype for Internationalized Text

First Public Working Draft

This document is intended to provide a quick reference to the OWL 2 language, similar to what was provided in the Language Synopsis section of the OWL Web Ontology Language Overview. It complements the OWL 2 Primer, an updated version of which will be published in due course. Inspiration for this effort includes work by the ebiquity Research Group at the University of Maryland Baltimore County (UMBC) on earlier versions of a Reference Card for the Semantic Web.

The intended final status of this document has not yet been determined; since it may become a Recommendation, it should be considered a Recommendation-Track document for now.

Please Comment By 2009-01-23

The OWL Working Group seeks public feedback on this First Public Working Draft. Please send your comments to public-owl-comments@w3.org (public archive). If

W3C Editor's Draft

possible, please offer specific changes to the text that would address your concern. You may also wish to check the Wiki Version of this document for internal-review comments and changes being drafted which may address your concerns.

No Endorsement

Publication as a Working Draft does not imply endorsement by the W3C Membership. This is a draft document and may be updated, replaced or obsoleted by other documents at any time. It is inappropriate to cite this document as other than work in progress.

Patents

This document was produced by a group operating under the 5 February 2004 W3C Patent Policy. W3C maintains a public list of any patent disclosures made in connection with the deliverables of the group; that page also includes instructions for disclosing a patent. An individual who has actual knowledge of a patent which the individual believes contains Essential Claim(s) must disclose the information in accordance with section 6 of the W3C Patent Policy.

Contents

Editor's Note: To do list:
- Create inline hyperlinks to Syntax
- Create hyperlinks to Direct Semantics and RDF-Based Semantics once the anchors in those documents are ready
- Complete inline hyperlinks to Primer when the missing ones are ready
- Create inline hyperlinks to New Features and Rationale
- Make a new 2-page print version.
- Design a similar guide for profiles with 1 page print version.

1 OWL/OWL 2 Vocabulary

Shaded constructs are only available in OWL 2.

- **OWL Classes**

all OWL individuals	owl:Thing
empty class	owl:Nothing

Connectives and Enumeration

intersection	IntersectionOf(C_1 … C_n)	x owl:intersectionOf [C_1 … C_n].	C_j an OWL class
union	UnionOf(C_1 … C_n)	x owl:unionOf [C_1 … C_n].	C_j an OWL class
complement	ComplementOf(C)	x owl:complementOf C.	C an OWL class
enumeration	OneOf(i_1 … i_n)	x owl:oneOf [i_1 … i_n].	i_j an OWL individual

Restrictions Using Object Properties owl:Restriction
Every owl:Restriction is an owl:Class.

universal	AllValuesFrom(P C)	x rdf:type owl:Restriction. x owl:onProperty P. x owl:allValuesFrom C	P an object property. C an OWL class
existential	SomeValuesFrom(P C)	x rdf:type owl:Restriction. x owl:onProperty P. x owl:someValuesFrom C	P an object property. C an OWL class
individual value	HasValue(P i)	x rdf:type owl:Restriction. x owl:onProperty P. x owl:hasValue i.	P an object property. i an individual
self	ExistsSelf(P)	x rdf:type owl:SelfRestriction. x owl:onProperty P.	P an object property
Cardinality Restrictions			
exact cardinality	ExactCardinality(n P)	x rdf:type owl:Restriction. x owl:onProperty P. x owl:cardinality n.	P an object property. n a nonNegativeInteger
maximum cardinality	MaxCardinality(n P)	x rdf:type owl:Restriction. x owl:onProperty P. x owl:maxCardinality n.	P an object property. n a nonNegativeInteger

minimum cardinality	MinCardinality(n P)	x rdf:type owl:Restriction. x owl:onProperty P. x owl:minCardinality n.	P an object property. n a nonNegativeInteger
Qualified Cardinality Restrictions			
exact cardinality	ExactCardinality(n P C)	x rdf:type owl:Restriction. x owl:onProperty P. x owl:qualifiedCardinality n. x owl:onClass C.	P an object property. n a nonNegativeInteger. C an OWL class
maximum cardinality	MaxCardinality(n P C)	x rdf:type owl:Restriction. x owl:onProperty P. x owl:minQualifiedCardinality n. x owl:onClass C.	P an object property. n a nonNegativeInteger; C an OWL class
minimum cardinality	MinCardinality(n P C)	x rdf:type owl:Restriction. x owl:onProperty P. x owl:maxQualifiedCardinality n. x owl:onClass C.	P an object property. n a nonNegativeInteger. C an OWL class

Restrictions Using Data Properties owl:Restriction
Every owl:Restriction is an owl:Class.

universal	AllValuesFrom(P D)	x rdf:type owl:Restriction. x owl:onProperty P. x owl:allValuesFrom D.	P a data property. D a datatype
existential	SomeValuesFrom(P D)	x rdf:type owl:Restriction. x owl:onProperty P. x owl:someValuesFrom D.	P a data property. D a datatype
n-ary universal	AllValuesFrom(P_1 … P_n D)	x rdf:type owl:Restriction. x owl:onProperties [P_1 … P_n]. x owl:allValuesFrom D.	P_i a data property. D an n-ary datatype
n-ary existential	SomeValuesFrom(P_1 … P_n D)	x rdf:type owl:Restriction. x owl:onProperties [P_1 … P_n]. x owl:someValuesFrom D.	P_i a data property. D an n-ary datatype
individual value	HasValue(P v)	x rdf:type owl:Restriction. x owl:onProperty P. x owl:hasValue v.	P an object property. v a literal
Cardinality Restrictions			
exact cardinality	ExactCardinality(n P)	x rdf:type owl:Restriction. x owl:onProperty P. x owl:cardinality n.	P a data property. n a nonNegativeInteger

maximum cardinality	MaxCardinality(n P)	x rdf:type owl:Restriction. x owl:onProperty P. x owl:maxCardinality n.	P a data property. n a nonNegativeInteger
minimum cardinality	MinCardinality(n P)	x rdf:type owl:Restriction. x owl:onProperty P. x owl:minCardinality n.	P a data property. n a nonNegativeInteger
Qualified Cardinality Restrictions			
exact cardinality	ExactCardinality(n P D)	x rdf:type owl:Restriction. x owl:onProperty P. x owl:qualifiedCardinality n. x owl:onDataRange D.	P a data property. n a nonNegativeInteger. D a datatype
maximum cardinality	MaxCardinality(n P D)	x rdf:type owl:Restriction. x owl:onProperty P. x owl:maxQualifiedCardinality n. x owl:onDataRange D.	P a data property. n a nonNegativeInteger. D a datatype
minimum cardinality	MinCardinality(n P D)	x rdf:type owl:Restriction. x owl:onProperty P. x owl:minQualifiedCardinality n. x owl:onDataRange D.	P a data property. n a nonNegativeInteger. D a datatype

- **Class Axioms**

subclasses	SubClassOf(C_1 C_2)	C_1 rdfs:subClassOf C_2.	C_i an OWL class
equivalent classes	EquivalentClasses(C_1 C_2)	C_1 owl:equivalentClass C_2.	C_i an OWL class
disjoint classes	DisjointClasses(C_1 C_2)	C_1 owl:disjointWith C_2.	C_i an OWL class
pairwise disjoint classes	DisjointClasses(C_1 … C_n)	x rdf:type owl:AllDisjointClasses. x owl:members [C_1 … C_n].	C_i an OWL class
disjoint union	DisjointUnionOf(C C_1 … C_n)	C owl:disjointUnionOf [C_1 … C_n].	C an OWL class. C_i an OWL class

- **Data Types** rdfs:Datatype

datatype complement	ComplementOf(D)	x owl:datatypeComplementOf D.	D a datatype

enumeration	OneOf(v_1 … v_n)	x owl:oneOf [v_1 … v_n].	v_j a literal
datatype restriction	DatatypeRestriction(D f_1 v_1 … f_n v_n)	x owl:onDatatype D. x owl:withRestrictions [y1 ... yn]. y1 f_1 v_1. ... yn f_n v_n.	f_j a constraining facet, v_j a restriction value

- **Object Properties** owl:ObjectProperty

universal object property	owl:topObjectProperty		= owl:Thing × owl:Thing
bottom object property	owl:bottomObjectProperty		empty binary relation
property chain	PropertyChain(P_1 … P_n)	x owl:propertyChain [P_1 … P_n].	P_i an object property

- **Object Property Axioms**

subproperty	SubPropertyOf(P Q)	P rdfs:subPropertyOf Q.	P,Q object properties
complex subproperty	SubPropertyOf(PropertyChain(P_1 … P_n) Q)	x rdfs:subPropertyOf Q. x owl:propertyChain [P_1 … P_n].	P_i and Q object properties
property domain	PropertyDomain(P C)	P rdfs:domain C.	P an object property, C an OWL class
property range	PropertyRange(P C)	P rdfs:range C.	P an object property, C an OWL class
equivalent properties	EquivalentProperties(P Q)	P owl:equivalentProperty Q.	P,Q object properties
disjoint properties	DisjointProperties(P Q)	P owl:propertyDisjointWith Q.	P,Q object properties

pairwise disjoint properties	DisjointProperties(P_1 … P_n)	x rdf:type owl:AllDisjointProperties. x owl:members [P_1 … P_n].	P_i an object property
inverse properties	InverseProperties(P Q)	P owl:inverseOf Q.	P,Q object properties
functional property	FunctionalProperty(P)	P rdf:type owl:FunctionalProperty.	P an object property s P,o1. s P o2 => o1=o2
inverse functional property	InverseFunctionalProperty(P)	P rdf:type owl:InverseFunctionalProperty.	P an object property s1 P,o. s2 P o => s1=s2
reflexive property	ReflexiveProperty(P)	P rdf:type owl:ReflexiveProperty.	P an object property for all individual i, i P i
irreflexive property	IrreflexiveProperty(P)	P rdf:type owl:IrreflexiveProperty.	P an object property i1 p i2 => i1≠i2
symmetric property	SymmetricProperty(P)	P rdf:type owl:SymmetricProperty.	P an object property s p o => o p s
asymmetric property	AsymmetricProperty(P)	P rdf:type owl:AsymmetricProperty.	P an object property s P o => not o P s

transitive property	TransitiveProperty(P)	P rdf:type owl:TransitiveProperty.	P an object property i1 P i2. i2 P i3 => i1 P i3

- **Datatype Properties** owl:DatatypeProperty

universal datatype property	owl:topDataProperty	= owl:Thing × rdfs:Literal
bottom datatype property	owl:bottomDataProperty	empty binary relation

- **Datatype Property Axioms**

subproperty	SubPropertyOf(P Q)	P rdfs:subPropertyOf Q.	P,Q datatype properties
property domain	PropertyDomain(P C)	P rdfs:domain C.	P a datatype property, C an OWL class
property range	PropertyRange(P C)	P rdfs:range C.	P a datatype property, C a datatype
equivalent properties	EquivalentProperties(P Q)	P owl:equivalentProperty Q.	P,Q datatype properties
disjoint properties	DisjointProperties(P Q)	P owl:propertyDisjointWith Q.	P,Q datatype properties
pairwise disjoint properties	DisjointProperties(P_1 … P_n)	x rdf:type owl:AllDisjointProperties. x owl:members [P_1 … P_n].	P_i a datatype property
functional property	FunctionalProperty(P)	P rdf:type owl:FunctionalProperty.	P a datatype property s P,v1. s P v2 => v1=v2

- **Keys**

Keys	HasKey(C P_1 … P_n)	C owl:hasKey [P_1 … P_n].	C an OWL class, P_i an object property or datatype property

			x P_i z_i. y P_i z_i, for i => x=y

- **Assertions**

individual equality	SameIndividual(a_1,a_2)	a_1 owl:sameAs a_2.	a_1,a_2 individuals
individual equality	SameIndividual(a_1 … a_n)	a_i owl:sameAs a_{i+1}. i=1…n-1	a_i an individual
individual inequality	DifferentIndividuals(a_1 a_2)	a_1 owl:differentFrom a_2.	a_1,a_2 individuals
pairwise individual inequality	DifferentIndividuals(a_1 … a_n)	x rdf:type owl:AllDifferent. x owl:members [a_1 … a_n].	a_i an individual
class assertion	ClassAssertion(C a)	a rdf:type C.	a an individual, C an OWL class
positive object property assertion	PropertyAssertion(P a_1 a_2)	a_1 P a_2.	P an object property, a_i an individual
negative object property assertion	NegativePropertyAssertion(P a_1 a_2)	x rdf:type owl:NegativePropertyAssertion. x owl:sourceIndividual a_1. x owl:assertionProperty P. x owl:targetIndividual a_2	P an object property, a_i an individual
negative datatype property assertion	NegativePropertyAssertion(P a_1 a_2)	x rdf:type owl:NegativePropertyAssertion. x owl:sourceIndividual a_1. x owl:assertionProperty P. x owl:targetValue a_2	P a datatype property, a_1 an individual, a_2 a literal

- **Lists** rdf:List (an RDF feature)

Note: in everywhere else of the document, we use only the concise N3 syntax.

concise N3 syntax	[a_1 … a_n]
triple-based N3 syntax	x_1 rdf:type rdf:List. x_1 rdf:first a_1. x_1 rdf:rest x_2. … x_n rdf:type rdf:List.

	x_n rdf:first a_n. x_n rdf:rest rdf:nil.

- **Declarations**

class	Declaration(Class(x))	x rdf:type owl:Class.
datatype	Declaration(Datatype(x))	x rdf:type rdfs:Datatype.
object property	Declaration(ObjectProperty(x))	x rdf:type owl:ObjectProperty.
datatype property	Declaration(DataProperty(x))	x rdf:type owl:DatatypeProperty.
annotation property	Declaration(AnnotationProperty(x))	x rdf:type owl:AnnotationProperty.
named individual	Declaration(NamedIndividual(x))	x rdf:type owl:NamedIndividual.

- **Reification**

If an axiom, when removed of all annotations, can be translated in a single RDF triple s p o, it can also be mapped to the following reificated form

axioms	s p o	x rdf:type owl:Axiom. x owl:subject s. x owl:predicate p. x owl:object o.	s the subject, p the predicate, o the object of the triple.

- **Annotations**

annotations	Annotation(P v)	x rdf:type owl:Annotation. x owl:subject y. x owl:predicate P. x owl:object v.	y the annotated resource P annotation property v a resource.
annotations	Annotation(P v)	y P v.	y the annotated resource P annotation property v a resource

Note: an annotated resource can be an ontology, an ontology entity, an axiom, an anonymous individual, or even another annotation.

Annotation Properties owl:AnnotationProperty

human-readable name	Label	rdfs:label

human-readable description	Comment	rdfs:comment
additional information		rdfs:seeAlso
defining agent		rdfs:isDefinedBy
deprecation	Deprecated	owl:deprecated

Annotation Property Axioms

annotation subproperties	SubPropertyOf(P Q)	P rdfs:subPropertyOf Q.	P,Q annotation properties
annotation property domain	PropertyDomain(P U)	P rdfs:domain U.	P an annotation property, U a URI
annotation property range	PropertyRange(P U)	P rdfs:range U.	P an annotation property, U a URI

Deprecation

deprecated class	C Deprecated	C rdf:type owl:DeprecatedClass.	C an OWL class or a datatype
deprecated property	P Deprecated	P rdf:type owl:DeprecatedProperty.	P an object property, datatype property or annotation property.

Note: "Deprecated" is the short for *owl:deprecated "true"^^xsd:boolean*

- **OWL Ontologies** owl:Ontology

OWL ontology	Ontology (O V)	O rdf:type owl:Ontology. O owl:versionInfo V.	O a URI; V a URI (optional)
Ontology Properties owl:OntologyProperty			
importing	O Import(U)	O owl:imports U.	O an ontology, U an ontology URI
backwards compatibility		O owl:backwardCompatibleWith U.	O an ontology, U an ontology URI
incompatibility		O owl:incompatibleWith U.	O an ontology, U an ontology URI
prior version		O owl:priorVersion U.	O an ontology, U an ontology URI

- **Deprecated Vocabulary in OWL 2**

owl:DataRange	replaced by rdfs:Datatype
owl:distinctMembers	replaced by owl:members

2 Built-in Datatypes and Facets

2.1 Built-in Datatypes

The *value space* is a set determining the set of values of the datatype. A literal value "abc" of the datatype DT can be given in the form "abc"^^DT.

- **Numeric DataTypes**

owl:realPlus and owl:real: (base datatype for other numeric datatypes, should not be directly instantiated)

owl:realPlus	the set of all real numbers plus +INF(positive infinity), -INF(negative infinity), -0(negative zero), NaN (not-a-number)
owl:real	the set of all real numbers

Other numeric Datatypes

xsd:double	xsd:nonNegativeInteger	xsd:long	xsd:unsignedLong
xsd:float	xsd:nonPositiveInteger	xsd:int	xsd:unsignedInt
xsd:decimal	xsd:positiveInteger	xsd:short	xsd:unsignedShort
xsd:integer	xsd:negativeInteger	xsd:byte	xsd:unsignedByte

- **Strings**: *value space* is of the form <"abc", tag>

Strings with a Language Tag: tag is either an empty string or a lowercase language tag

rdf:text	internationalized strings

Strings without a Language Tag: tag is an empty string

xsd:string	xsd:NCName
xsd:normalizedString	xsd:NMTOKEN
xsd:token	xsd:language
xsd:Name	

- **Boolean Values**

xsd:Boolean	value space has only two values: *true* and *false*

- **Binary Data**

xsd:base64Binary
xsd:hexBinary

- **URIs**

xsd:anyURI

- **Time Instants**

OWL 2 datatype

owl:dateTime	*value space* is set of numbers that represent difference

	between the time instant and the first time instant of the 1st of January 1 AD in the proleptic Gregorian calendar, in seconds

XSD datatypes

xsd:date	xsd:gDay	xsd:gYear
xsd:dateTime	xsd:gMonth	xsd:gYearMonth
xsd:duration	xsd:gMonthDay	xsd:time

- **XML Fragment**

rdf:XMLLiteral

2.2 Facets

The *facet space* is a set of pairs of the form < *f v* >, where *f* is a URI called a constraining facet, and *v* is a value. Each such pair is mapped to a subset of the value space of the datatype.

Notations:

- Numeric Datatype, String Datatype and Binary datatype refer to a set of datatypes based on the classification done in the prior section.

Facet *f*	**Datatype**	**Value** *v*	**Explanation**
xsd:minInclusive, xsd:maxInclusive, xsd:minExclusive, xsd:maxExclusive	Numeric Datatype DT, owl:dateTime DT	Literal in DT	Restricts the value-space to greater than (equal to) or lesser than (equal to) a value
xsd:minLength, xsd:maxLength, xsd:length	String Datatype, Binary Datatype, xsd:anyURI	Nonnegative integer	Restricts the value-space based on the lengths of the literals
xsd:pattern	String Datatype, xsd:anyURI	xsd:string literal whose value is a regular expression	Restricts the value space to literals that match the regular expression
rdf:langPattern	rdf:text	xsd:string literal whose value is a regular expression	Restricts the value space to literals with language tags that match the regular expression

3 Name Spaces

Standard Namespaces and Prefixes used in OWL 2

Prefix	URI
rdf	http://www.w3.org/1999/02/22-rdf-syntax-ns#
rdfs	http://www.w3.org/2000/01/rdf-schema#
owl	http://www.w3.org/2002/07/owl#
xsd	http://www.w3.org/2001/XMLSchema#
ox	http://www.w3.org/2006/12/owl2-xml#