



# OWL 2 Web Ontology Language Quick Reference Guide

W3C Editor's Draft 10 June 2009

**This version:**

<http://www.w3.org/2007/OWL/draft/ED-owl2-quick-reference-20090610/>

**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-20090531/>  
([color-coded diff](#))

**Authors:**

[Jie Bao](#), Rensselaer Polytechnic Institute  
[Elisa F. Kendall](#), Sandpiper Software, Inc.  
[Deborah L. McGuinness](#), Rensselaer Polytechnic Institute  
[Peter F. Patel-Schneider](#), Bell Labs Research, Alcatel-Lucent

**Contributors:**

[Li Ding](#), Rensselaer Polytechnic Institute  
[Ankesh Khandelwal](#), Rensselaer Polytechnic Institute

This document is also available in these non-normative formats: [PDF version](#).

---

Copyright © 2009 W3C® ([MIT](#), [ERCIM](#), [Keio](#)), All Rights Reserved. W3C [liability](#), [trademark](#) and [document use](#) rules apply.

---

## Abstract

The OWL 2 Web Ontology Language, informally OWL 2, is an ontology language for the Semantic Web with formally defined meaning. OWL 2 ontologies provide classes, properties, individuals, and data values and are stored as Semantic Web documents. OWL 2 ontologies can be used along with information written in RDF, and OWL 2 ontologies themselves are primarily exchanged as RDF documents. The OWL 2 [Document Overview](#) describes the overall state of OWL 2, and should be read before other OWL 2 documents.

## 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### May Be Superseded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](http://www.w3.org/TR/) at <http://www.w3.org/TR/>.*

### Summary of Changes

This Working Draft has undergone significant presentation changes and a reorganization.

### Last Call

The Working Group believes it has completed its design work for the technologies specified in this document, so this is a "Last Call" draft. The design is not expected to change significantly, going forward, and now is the key time for external review, before the implementation phase.

### Please Comment By 29 July 2009

The [OWL Working Group](#) seeks public feedback on this Working Draft. Please send your comments to [public-owl-comments@w3.org](mailto:public-owl-comments@w3.org) ([public archive](#)). If 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 and see if the relevant text has already been updated.

### 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](#). This document is informative only. 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](#).*

---

## Table of Contents

- [1 Names, Prefixes, and Notation](#)
- [2 OWL 2 constructs and axioms](#)
  - [2.1 Class Expressions](#)
  - [2.2 Properties](#)
  - [2.3 Individuals & Literals](#)
  - [2.4 Data Ranges](#)
  - [2.5 Axioms](#)
  - [2.6 Declarations](#)
  - [2.7 Annotations](#)
  - [2.8 Ontologies](#)
- [3 Built-in Datatypes and Facets](#)
  - [3.1 Built-in Datatypes](#)
  - [3.2 Facets](#)
- [4 Appendix](#)
  - [4.1 New Features in OWL 2](#)
  - [4.2 Additional Vocabulary in OWL 2 RDF Syntax](#)
- [5 Acknowledgments](#)

**Editor's Note:** To do list:

- Make a new pdf print version when the guide is finalized. ([pdf](#)) ([wiki file](#))

## 1 Names, Prefixes, and Notation

Names in OWL 2 are IRIs, often written in a shorthand `prefix:local_name`, where `prefix:` is a [prefix name](#) that expands to an IRI, and `local_name` is the remainder of the name. The [standard prefix names](#) in OWL 2 are:

Prefix Name	Expansion
rdf:	<a href="http://www.w3.org/1999/02/22-rdf-syntax-ns#">http://www.w3.org/1999/02/22-rdf-syntax-ns#</a>
rdfs:	<a href="http://www.w3.org/2000/01/rdf-schema#">http://www.w3.org/2000/01/rdf-schema#</a>
owl:	<a href="http://www.w3.org/2002/07/owl#">http://www.w3.org/2002/07/owl#</a>
xsd:	<a href="http://www.w3.org/2001/XMLSchema#">http://www.w3.org/2001/XMLSchema#</a>

We use notation conventions in the following table\*:

Letters	Meaning	Letters	Meaning	Letters	Meaning	Letters	Meaning
C	class expression	CN	class name	D	data range	DN	datatype name

P	object property expression	PN	object property name	R	data property	A	annotation property
a	individual	aN	individual name	_:a	anonymous individual (a <a href="#">blank node label</a> )	v	literal
n	non-negative integer**	f	facet	ON	ontology name	U	IRI
s	IRI or anonymous individual	t	IRI, anonymous individual, or literal	p	prefix name	_:x	blank node
(a <sub>1</sub> ... a <sub>n</sub> )	<a href="#">RDF list</a>						

\* All of the above can have subscripts. \*\* as a shorthand for "n"^^xsd:nonNegativeInteger

## 2 OWL 2 constructs and axioms

For an OWL 2 DL ontology, there are some [global restrictions](#) on axioms.

In the following tables the first column provides links to the [Primer](#) (if applicable) and the 2nd column provides links to the [Functional Syntax](#).

### 2.1 Class Expressions

#### [Predefined and Named Classes](#)

Language Feature	Functional Syntax	RDF Syntax
named class	CN	CN
universal class	<a href="#">owl:Thing</a>	owl:Thing
empty class	<a href="#">owl:Nothing</a>	owl:Nothing

#### **Boolean Connectives and Enumeration of Individuals**

Language Feature	Functional Syntax	RDF Syntax
<a href="#">intersection</a>	<a href="#">ObjectIntersectionOf</a> (C <sub>1</sub> ... C <sub>n</sub> )	_:x rdf:type owl:Class. _:x owl:intersectionOf ( C <sub>1</sub> ... C <sub>n</sub> ).
<a href="#">union</a>	<a href="#">ObjectUnionOf</a> (C <sub>1</sub> ... C <sub>n</sub> )	_:x rdf:type owl:Class. _:x owl:unionOf ( C <sub>1</sub> ... C <sub>n</sub> ).
<a href="#">complement</a>	<a href="#">ObjectComplementOf</a> (C)	_:x rdf:type owl:Class. _:x owl:complementOf C.
<a href="#">enumeration</a>	<a href="#">ObjectOneOf</a> (a <sub>1</sub> ... a <sub>n</sub> )	_:x rdf:type owl:Class. _:x owl:oneOf ( a <sub>1</sub> ... a <sub>n</sub> ).

**Object Property Restrictions**

Language Feature	Functional Syntax	RDF Syntax
<a href="#">universal</a>	<a href="#">ObjectAllValuesFrom</a> (P C)	<code>_:x rdf:type owl:Restriction. _:x owl:onProperty P. _:x owl:allValuesFrom C</code>
<a href="#">existential</a>	<a href="#">ObjectSomeValuesFrom</a> (P C)	<code>_:x rdf:type owl:Restriction. _:x owl:onProperty P. _:x owl:someValuesFrom C</code>
<a href="#">individual value</a>	<a href="#">ObjectHasValue</a> (P a)	<code>_:x rdf:type owl:Restriction. _:x owl:onProperty P. _:x owl:hasValue a.</code>
<a href="#">local reflexivity</a>	<a href="#">ObjectHasSelf</a> (P)	<code>_:x rdf:type owl:Restriction. _:x owl:onProperty P. _:x owl:hasSelf "true"^^xsd:boolean.</code>
<a href="#">exact cardinality</a>	<a href="#">ObjectExactCardinality</a> (n P)	<code>_:x rdf:type owl:Restriction. _:x owl:onProperty P. _:x owl:cardinality n.</code>
<a href="#">qualified exact cardinality</a>	<a href="#">ObjectExactCardinality</a> (n P C)	<code>_:x rdf:type owl:Restriction. _:x owl:onProperty P. _:x owl:qualifiedCardinality n. _:x owl:onClass C.</code>
maximum cardinality	<a href="#">ObjectMaxCardinality</a> (n P)	<code>_:x rdf:type owl:Restriction. _:x owl:onProperty P. _:x owl:minCardinality n.</code>
<a href="#">qualified maximum cardinality</a>	<a href="#">ObjectMaxCardinality</a> (n P C)	<code>_:x rdf:type owl:Restriction. _:x owl:onProperty P. _:x owl:minQualifiedCardinality n. _:x owl:onClass C.</code>
minimum cardinality	<a href="#">ObjectMinCardinality</a> (n P)	<code>_:x rdf:type owl:Restriction. _:x owl:onProperty P. _:x owl:maxCardinality n.</code>
<a href="#">qualified minimum cardinality</a>	<a href="#">ObjectMinCardinality</a> (n P C)	<code>_:x rdf:type owl:Restriction. _:x owl:onProperty P. _:x owl:maxQualifiedCardinality n. _:x owl:onClass C.</code>

**Data Property Restrictions**

Language Feature	Functional Syntax	RDF Syntax
universal	<a href="#">DataAllValuesFrom</a> (R D)	<code>_:x rdf:type owl:Restriction. _:x owl:onProperty R. _:x owl:allValuesFrom D.</code>
existential	<a href="#">DataSomeValuesFrom</a> (R D)	<code>_:x rdf:type owl:Restriction. _:x owl:onProperty R. _:x owl:someValuesFrom D.</code>

literal value	<a href="#">DataHasValue</a> (R v)	_:x rdf:type owl:Restriction. _:x owl:onProperty R. _:x owl:hasValue v.
exact cardinality	<a href="#">DataExactCardinality</a> (n R)	_:x rdf:type owl:Restriction. _:x owl:onProperty R. _:x owl:cardinality n.
qualified exact cardinality	<a href="#">DataExactCardinality</a> (n R D)	_:x rdf:type owl:Restriction. _:x owl:onProperty R. _:x owl:qualifiedCardinality n. _:x owl:onDataRange D.
maximum cardinality	<a href="#">DataMaxCardinality</a> (n R)	_:x rdf:type owl:Restriction. _:x owl:onProperty R. _:x owl:maxCardinality n.
qualified maximum cardinality	<a href="#">DataMaxCardinality</a> (n R D)	_:x rdf:type owl:Restriction. _:x owl:onProperty R. _:x owl:maxQualifiedCardinality n. _:x owl:onDataRange D.
minimum cardinality	<a href="#">DataMinCardinality</a> (n R)	_:x rdf:type owl:Restriction. _:x owl:onProperty R. _:x owl:minCardinality n.
qualified minimum cardinality	<a href="#">DataMinCardinality</a> (n R D)	_:x rdf:type owl:Restriction. _:x owl:onProperty R. _:x owl:minQualifiedCardinality n. _:x owl:onDataRange D.

**Restrictions Using n-ary Data Range**

In the following table 'D<sup>n</sup>' is an n-ary data range.

Language Feature	Functional Syntax	RDF Syntax
n-ary universal	<a href="#">DataAllValuesFrom</a> (R <sub>1</sub> ... R <sub>n</sub> D <sup>n</sup> )	_:x rdf:type owl:Restriction. _:x owl:onProperties ( R <sub>1</sub> ... R <sub>n</sub> ). _:x owl:allValuesFrom D <sup>n</sup> .
n-ary existential	<a href="#">DataSomeValuesFrom</a> (R <sub>1</sub> ... R <sub>n</sub> D <sup>n</sup> )	_:x rdf:type owl:Restriction. _:x owl:onProperties ( R <sub>1</sub> ... R <sub>n</sub> ). _:x owl:someValuesFrom D <sup>n</sup> .

**2.2 Properties**

**Object Property Expressions**

Language Feature	Functional Syntax	RDF Syntax
<a href="#">named object property</a>	<a href="#">PN</a>	PN
<a href="#">universal object property</a>	<a href="#">owl:topObjectProperty</a>	owl:topObjectProperty
<a href="#">empty object property</a>	<a href="#">owl:bottomObjectProperty</a>	owl:bottomObjectProperty
<a href="#">inverse property</a>	<a href="#">ObjectInverseOf</a> (PN)	_:x owl:inverseOf PN

## Data Property Expressions

Language Feature	Functional Syntax	RDF Syntax
<a href="#">named data property</a>	<a href="#">R</a>	R
<a href="#">universal data property</a>	<a href="#">owl:topDataProperty</a>	owl:topDataProperty
<a href="#">empty data property</a>	<a href="#">owl:bottomDataProperty</a>	owl:bottomDataProperty

## 2.3 Individuals & Literals

Language Feature	Functional Syntax	RDF Syntax
<a href="#">named individual</a>	<a href="#">aN</a>	aN
anonymous individual	<a href="#">_:a</a>	_:a
<a href="#">literal</a> (datatype value)	<a href="#">"abc"^^DN</a>	"abc"^^DN

## 2.4 Data Ranges

### Data Range Expressions

Language Feature	Functional Syntax	RDF Syntax
<a href="#">named datatype</a>	<a href="#">DN</a>	DN
<a href="#">data range complement</a>	<a href="#">DataComplementOf</a> (D)	_:x rdf:type rdfs:Datatype. _:x owl:datatypeComplementOf D.
<a href="#">data range intersection</a>	<a href="#">DataIntersectionOf</a> (D <sub>1</sub> ...D <sub>n</sub> )	_:x rdf:type rdfs:Datatype. _:x owl:intersectionOf (D <sub>1</sub> ...D <sub>n</sub> ).
<a href="#">data range union</a>	<a href="#">DataUnionOf</a> (D <sub>1</sub> ...D <sub>n</sub> )	_:x rdf:type rdfs:Datatype. _:x owl:unionOf (D <sub>1</sub> ...D <sub>n</sub> ).
<a href="#">literal enumeration</a>	<a href="#">DataOneOf</a> (v <sub>1</sub> ... v <sub>n</sub> )	_:x rdf:type rdfs:Datatype. _:x owl:oneOf ( v <sub>1</sub> ... v <sub>n</sub> ).
<a href="#">datatype restriction</a>	<a href="#">DatatypeRestriction</a> (DN f <sub>1</sub> v <sub>1</sub> ... f <sub>n</sub> v <sub>n</sub> )	_:x rdf:type rdfs:Datatype. _:x owl:onDatatype DN. _:x owl:withRestrictions ( _:x <sub>1</sub> ... _:x <sub>n</sub> ). _:x <sub>j</sub> f <sub>j</sub> v <sub>j</sub> . j=1...n

## 2.5 Axioms

### Class Expression Axioms

Language Feature	Functional Syntax	RDF Syntax
<a href="#">subclass</a>	<a href="#">SubClassOf</a> (C <sub>1</sub> C <sub>2</sub> )	C <sub>1</sub> rdfs:subClassOf C <sub>2</sub> .
<a href="#">equivalent classes</a>	<a href="#">EquivalentClasses</a> (C <sub>1</sub> ... C <sub>n</sub> )	C <sub>j</sub> owl:equivalentClass C <sub>j+1</sub> . j=1...n-1
<a href="#">disjoint classes</a>	<a href="#">DisjointClasses</a> (C <sub>1</sub> C <sub>2</sub> )	C <sub>1</sub> owl:disjointWith C <sub>2</sub> .
pairwise disjoint classes	<a href="#">DisjointClasses</a> (C <sub>1</sub> ... C <sub>n</sub> )	_:x rdf:type owl:AllDisjointClasses. _:x owl:members ( C <sub>1</sub> ... C <sub>n</sub> ).

disjoint union	<a href="#">DisjointUnionOf</a> (CN C <sub>1</sub> ... C <sub>n</sub> )	CN owl:disjointUnionOf ( C <sub>1</sub> ... C <sub>n</sub> ).
----------------	---	---

### Object Property Axioms

Language Feature	Functional Syntax	RDF Syntax
<a href="#">subproperty</a>	<a href="#">SubObjectPropertyOf</a> (P <sub>1</sub> P <sub>2</sub> )	P <sub>1</sub> rdfs:subPropertyOf P <sub>2</sub> .
<a href="#">property chain inclusion</a>	<a href="#">SubObjectPropertyOf</a> (ObjectPropertyChain(P <sub>1</sub> ... P <sub>n</sub> ) P)	P owl:propertyChainAxiom (P <sub>1</sub> ... P <sub>n</sub> ).
<a href="#">property domain</a>	<a href="#">ObjectPropertyDomain</a> (P C)	P rdfs:domain C.
<a href="#">property range</a>	<a href="#">ObjectPropertyRange</a> (P C)	P rdfs:range C.
<a href="#">equivalent properties</a>	<a href="#">EquivalentObjectProperties</a> (P <sub>1</sub> ... P <sub>n</sub> )	P <sub>j</sub> owl:equivalentProperty P <sub>j+1</sub> . j=1...n-1
<a href="#">disjoint properties</a>	<a href="#">DisjointObjectProperties</a> (P <sub>1</sub> P <sub>2</sub> )	P <sub>1</sub> owl:propertyDisjointWith P <sub>2</sub> .
<a href="#">pairwise disjoint properties</a>	<a href="#">DisjointObjectProperties</a> (P <sub>1</sub> ... P <sub>n</sub> )	_:x rdf:type owl:AllDisjointProperties. _:x owl:members ( P <sub>1</sub> ... P <sub>n</sub> ).
<a href="#">inverse properties</a>	<a href="#">InverseObjectProperties</a> (P <sub>1</sub> P <sub>2</sub> )	P <sub>1</sub> owl:inverseOf P <sub>2</sub> .
<a href="#">functional property</a>	<a href="#">FunctionalObjectProperty</a> (P)	P rdf:type owl:FunctionalProperty.
<a href="#">inverse functional property</a>	<a href="#">InverseFunctionalObjectProperty</a> (P)	P rdf:type owl:InverseFunctionalProperty.
<a href="#">reflexive property</a>	<a href="#">ReflexiveObjectProperty</a> (P)	P rdf:type owl:ReflexiveProperty.
<a href="#">irreflexive property</a>	<a href="#">IrreflexiveObjectProperty</a> (P)	P rdf:type owl:IrreflexiveProperty.
<a href="#">symmetric property</a>	<a href="#">SymmetricObjectProperty</a> (P)	P rdf:type owl:SymmetricProperty.
<a href="#">asymmetric property</a>	<a href="#">AsymmetricObjectProperty</a> (P)	P rdf:type owl:AsymmetricProperty.
<a href="#">transitive property</a>	<a href="#">TransitiveObjectProperty</a> (P)	P rdf:type owl:TransitiveProperty.

### Data Property Axioms

Language Feature	Functional Syntax	RDF Syntax
<a href="#">subproperty</a>	<a href="#">SubDataPropertyOf</a> (R <sub>1</sub> R <sub>2</sub> )	R <sub>1</sub> rdfs:subPropertyOf R <sub>2</sub> .
<a href="#">property domain</a>	<a href="#">DataPropertyDomain</a> (R C)	R rdfs:domain C.
<a href="#">property range</a>	<a href="#">DataPropertyRange</a> (R D)	R rdfs:range D.
<a href="#">equivalent properties</a>	<a href="#">EquivalentDataProperties</a> (R <sub>1</sub> ... R <sub>n</sub> )	R <sub>j</sub> owl:equivalentProperty R <sub>j+1</sub> . j=1...n-1
disjoint properties	<a href="#">DisjointDataProperties</a> (R <sub>1</sub> R <sub>2</sub> )	R <sub>1</sub> owl:propertyDisjointWith R <sub>2</sub> .
pairwise disjoint properties	<a href="#">DisjointDataProperties</a> (R <sub>1</sub> ... R <sub>n</sub> )	_:x rdf:type owl:AllDisjointProperties. _:x owl:members ( R <sub>1</sub> ... R <sub>n</sub> ).



<a href="#">functional property</a>	<a href="#">FunctionalDataProperty</a> (R)	R rdf:type owl:FunctionalProperty.
-------------------------------------	--	------------------------------------

**Datatype Definitions**

Language Feature	Functional Syntax	RDF Syntax
<a href="#">datatype definition</a>	<a href="#">DatatypeDefinition</a> (DN D)	DN owl:equivalentClass D.

**Assertions**

Language Feature	Functional Syntax	RDF Syntax
<a href="#">individual equality</a>	<a href="#">SameIndividual</a> (a <sub>1</sub> ... a <sub>n</sub> )	a <sub>j</sub> owl:sameAs a <sub>j+1</sub> . j=1...n-1
<a href="#">individual inequality</a>	<a href="#">DifferentIndividuals</a> (a <sub>1</sub> a <sub>2</sub> )	a <sub>1</sub> owl:differentFrom a <sub>2</sub> .
pairwise individual inequality	<a href="#">DifferentIndividuals</a> (a <sub>1</sub> ... a <sub>n</sub> )	_:x rdf:type owl:AllDifferent. _:x owl:members (a <sub>1</sub> ... a <sub>n</sub> ).
<a href="#">class assertion</a>	<a href="#">ClassAssertion</a> (C a)	a rdf:type C.
<a href="#">positive object property assertion</a>	<a href="#">ObjectPropertyAssertion</a> ( P N a <sub>1</sub> a <sub>2</sub> )	a <sub>1</sub> P N a <sub>2</sub> .
<a href="#">positive data property assertion</a>	<a href="#">DataPropertyAssertion</a> ( R a v )	a R v.
<a href="#">negative object property assertion</a>	<a href="#">NegativeObjectPropertyAssertion</a> (P a <sub>1</sub> a <sub>2</sub> )	_:x rdf:type owl:NegativePropertyAssertion. _:x owl:sourceIndividual a <sub>1</sub> . _:x owl:assertionProperty P. _:x owl:targetIndividual a <sub>2</sub> .
<a href="#">negative data property assertion</a>	<a href="#">NegativeDataPropertyAssertion</a> (R a v )	_:x rdf:type owl:NegativePropertyAssertion. _:x owl:sourceIndividual a. _:x owl:assertionProperty R. _:x owl:targetValue v.

**Keys**

Language Feature	Functional Syntax	RDF Syntax
<a href="#">Key</a>	<a href="#">HasKey</a> (C (P <sub>1</sub> ... P <sub>m</sub> ) (R <sub>1</sub> ... R <sub>n</sub> ))	C owl:hasKey (P <sub>1</sub> ... P <sub>m</sub> R <sub>1</sub> ... R <sub>n</sub> ).

**2.6 Declarations**

Language Feature	Functional Syntax	RDF Syntax
<a href="#">class</a>	<a href="#">Declaration</a> ( Class( CN ) )	CN rdf:type owl:Class.

<a href="#">datatype</a>	<a href="#">Declaration</a> ( Datatype( DN ) )	DN rdf:type rdfs:Datatype.
<a href="#">object property</a>	<a href="#">Declaration</a> ( ObjectProperty( PN ) )	PN rdf:type owl:ObjectProperty.
<a href="#">data property</a>	<a href="#">Declaration</a> ( DataProperty( R ) )	R rdf:type owl:DatatypeProperty.
<a href="#">annotation property</a>	<a href="#">Declaration</a> ( AnnotationProperty( A ) )	A rdf:type owl:AnnotationProperty.
<a href="#">named individual</a>	<a href="#">Declaration</a> ( NamedIndividual( aN ) )	aN rdf:type owl:NamedIndividual.

## 2.7 Annotations

### Annotations

Language Feature	Functional Syntax	RDF Syntax
<a href="#">annotation assertion</a>	<a href="#">AnnotationAssertion</a> (A s t)	s A t.
<a href="#">annotation of an axiom</a> where the axiom in RDF is one or more triples with the same predicate $s_i U t_i$	AXIOM( <a href="#">Annotation</a> (A t) ...)	_:xi A t. s <sub>i</sub> U t <sub>i</sub> . ... _:xi rdf:type owl:Axiom. _:xi owl:annotatedSource s <sub>i</sub> . _:xi owl:annotatedProperty U. _:xi owl:annotatedTarget t <sub>i</sub> .
<a href="#">annotation of an axiom</a> where the axiom in RDF starts with $_:x$	AXIOM( <a href="#">Annotation</a> (A t) ...)	_:x A t. _:x ....
<a href="#">annotation of another annotation</a> (the other annotation in RDF starts with $s_1$ )	Annotation( <a href="#">Annotation</a> (A t) ... A <sub>1</sub> t <sub>1</sub> )	_:x A t. s <sub>1</sub> A <sub>1</sub> t <sub>1</sub> . _:x rdf:type owl:Annotation. _:x owl:annotatedSource s <sub>1</sub> . _:x owl:annotatedProperty A <sub>1</sub> . _:x owl:annotatedTarget t <sub>1</sub> .

### Annotation Properties

Language Feature	Functional Syntax	RDF Syntax
named annotation property	<a href="#">A</a>	A
human-readable name	<a href="#">rdfs:label</a>	<a href="#">rdfs:label</a>

human-readable comment	<a href="#">rdfs:comment</a>	<a href="#">rdfs:comment</a>
additional information	<a href="#">rdfs:seeAlso</a>	<a href="#">rdfs:seeAlso</a>
defining agent	<a href="#">rdfs:isDefinedBy</a>	<a href="#">rdfs:isDefinedBy</a>
version information	<a href="#">owl:versionInfo</a>	owl:versionInfo
deprecation	<a href="#">owl:deprecated</a>	owl:deprecated
backwards compatibility	<a href="#">owl:backwardCompatibleWith</a>	owl:backwardCompatibleWith
incompatibility	<a href="#">owl:incompatibleWith</a>	owl:incompatibleWith
prior version	<a href="#">owl:priorVersion</a>	owl:priorVersion

### Annotation Axioms

Language Feature	Functional Syntax	RDF Syntax
<a href="#">annotation subproperties</a>	<a href="#">SubAnnotationPropertyOf</a> (A <sub>1</sub> A <sub>2</sub> )	A <sub>1</sub> rdfs:subPropertyOf A <sub>2</sub> .
annotation property domain	<a href="#">AnnotationPropertyDomain</a> (A U)	A rdfs:domain U.
annotation property range	<a href="#">AnnotationPropertyRange</a> (A U)	A rdfs:range U.

## 2.8 Ontologies

### Ontologies

Language Feature	Functional Syntax	RDF Syntax
<a href="#">OWL ontology (importing)*</a>	<a href="#">Ontology</a> ([ON [U]] <a href="#">Import</a> (ON <sub>1</sub> )... Annotation(A t) ... )	ON rdf:type owl:Ontology. [ON owl:versionIRI U.] ON owl:imports ON <sub>1</sub> . ... ON A t. ...
prefix declaration**	<a href="#">Prefix</a> (p=U)	@prefix p U.

Note \*: in the RDF syntax `_:x` is used in place of ON if there is no ontology name. \*\* RDF syntax is in Turtle, other RDF serializations may vary.

## 3 Built-in Datatypes and Facets

### 3.1 Built-in Datatypes

Universal Datatype	<a href="#">rdfs:Literal</a>
--------------------	------------------------------

<a href="#">Numbers</a>	<a href="#">owl:rational</a>	<a href="#">owl:real</a>		
	<a href="#">xsd:double</a>	<a href="#">xsd:float</a>	<a href="#">xsd:decimal</a>	<a href="#">xsd:integer</a>
	<a href="#">xsd:long</a>	<a href="#">xsd:int</a>	<a href="#">xsd:short</a>	<a href="#">xsd:byte</a>
	<a href="#">xsd:nonNegativeInteger</a>	<a href="#">xsd:nonPositiveInteger</a>		
	<a href="#">xsd:positiveInteger</a>	<a href="#">xsd:negativeInteger</a>		
	<a href="#">xsd:unsignedLong</a>	<a href="#">xsd:unsignedInt</a>		
	<a href="#">xsd:unsignedShort</a>	<a href="#">xsd:unsignedByte</a>		
<a href="#">Strings</a>	<a href="#">rdf:PlainLiteral</a> (RDF plain literals)			
	<a href="#">xsd:string</a>	<a href="#">xsd:NCName</a>	<a href="#">xsd:Name</a>	<a href="#">xsd:NMToken</a>
	<a href="#">xsd:token</a>	<a href="#">xsd:language</a>	<a href="#">xsd:normalizedString</a>	
<a href="#">Boolean Values</a>	<a href="#">xsd:boolean</a> (value space: <i>true</i> and <i>false</i> )			
<a href="#">Binary Data</a>	<a href="#">xsd:base64Binary</a>	<a href="#">xsd:hexBinary</a>		
<a href="#">IRIs</a>	<a href="#">xsd:anyURI</a>			
<a href="#">Time Instants</a>	<a href="#">xsd:dateTime</a> (optional time zone offset)			
	<a href="#">xsd:dateTimeStamp</a> (required time zone offset)			
<a href="#">XML Literals</a>	<a href="#">rdf:XMLLiteral</a>			

Note: owl:rational and rdf:XMLLiteral are at Risk in OWL 2.

### 3.2 Facets

Facet	Value	Applicable Datatypes	Explanation
<a href="#">xsd:minInclusive</a> <a href="#">xsd:maxInclusive</a> <a href="#">xsd:minExclusive</a> <a href="#">xsd:maxExclusive</a>	literal in the corresponding datatype	Numbers, Time Instants	Restricts the value-space to greater than (equal to) or lesser than (equal to) a value
<a href="#">xsd:minLength</a> <a href="#">xsd:maxLength</a> <a href="#">xsd:length</a>	Non-negative integer	Strings, Binary Data, IRIs	Restricts the value-space based on the lengths of the literals
<a href="#">xsd:pattern</a>	xsd:string literal as a regular expression	Strings, IRIs	Restricts the value space to literals that match the regular expression
<a href="#">rdf:langRange</a>	xsd:string literal as a regular expression	rdf:PlainLiteral	Restricts the value space to literals with language tags that match the regular expression

## 4 Appendix

### 4.1 New Features in OWL 2

Class Expressions	<ul style="list-style-type: none"> <li><a href="#">local reflexivity</a> (self restriction)</li> <li><a href="#">object</a> and <a href="#">data</a> qualified exact/maximum/minimal cardinality restriction</li> <li><a href="#">universal</a> and <a href="#">existential</a> restriction on n-ary data range</li> </ul>
Class Axioms	<ul style="list-style-type: none"> <li><a href="#">pairwise disjoint classes</a></li> <li><a href="#">class disjoint union</a></li> </ul>
Property Expressions	<ul style="list-style-type: none"> <li><a href="#">universal</a> and <a href="#">empty</a> object property</li> <li><a href="#">universal</a> and <a href="#">empty</a> data property</li> </ul>

	<ul style="list-style-type: none"> <li>• <a href="#">inverse object property expression</a></li> </ul>
Property Axioms	<ul style="list-style-type: none"> <li>• <a href="#">property chain inclusion</a></li> <li>• <a href="#">disjoint object properties</a></li> <li>• <a href="#">disjoint data properties</a></li> <li>• <a href="#">reflexive</a>, <a href="#">irreflexive</a>, and <a href="#">asymmetric</a> object property.</li> </ul>
Data Ranges	<ul style="list-style-type: none"> <li>• <a href="#">datatype definition</a></li> <li>• <a href="#">data range complement</a>, <a href="#">intersection</a> and <a href="#">union</a></li> <li>• <a href="#">datatype restriction</a> and <a href="#">facets</a></li> <li>• <a href="#">hook for n-ary datatype</a></li> </ul>
Assertions	<ul style="list-style-type: none"> <li>• <a href="#">negative object property assertion</a></li> <li>• <a href="#">negative data property assertion</a></li> </ul>
Annotation	<ul style="list-style-type: none"> <li>• <a href="#">annotation assertion</a></li> <li>• <a href="#">annotation of an axiom or an annotation</a></li> <li>• <a href="#">annotation subproperties</a></li> <li>• annotation property <a href="#">domain</a> and <a href="#">range</a></li> <li>• owl:deprecated annotation property</li> </ul>
<a href="#">Extra Built-in Datatypes</a>	<ul style="list-style-type: none"> <li>• owl:rational, owl:real, xsd:dateTimeStamp, rdf:PlainLiteral</li> </ul>
Others	<ul style="list-style-type: none"> <li>• <a href="#">key</a></li> <li>• <a href="#">declaration</a></li> <li>• <a href="#">metamodeling capabilities</a> (Punning)</li> <li>• <a href="#">anonymous individual</a></li> </ul>

## 4.2 Additional Vocabulary in OWL 2 RDF Syntax

Feature	Vocabulary	Note
data range	owl:DataRange	deprecated in OWL 2, replaced by <a href="#">rdfs:Datatype</a>
membership of a set of pairwise different individuals	owl:distinctMembers	can alternatively use owl:members
ontology property	owl:OntologyProperty	
deprecation	owl:DeprecatedClass, owl:DeprecatedProperty	alternative RDF syntax: s rdf:type owl:DeprecatedClass . or s rdf:type owl:DeprecatedProperty . can be replaced by s owl:deprecated "true"^^xsd:boolean .

## 5 Acknowledgments

The starting point for the development of OWL 2 was the [OWL1.1 member submission](#), itself a result of user and developer feedback, and in particular of information gathered during the [OWL Experiences and Directions \(OWLED\)](#)

[Workshop series](#). The working group also considered [postponed issues](#) from the [WebOnt Working Group](#).

This document has been produced by the OWL Working Group (see below), and its contents reflect extensive discussions within the Working Group as a whole. The editors extend special thanks to Bernardo Cuenca Grau (Oxford University), Christine Golbreich (Université de Versailles St-Quentin and LIRMM), Ivan Herman (W3C/ERCIM), and Bijan Parsia (University of Manchester) for their thorough reviews.

The regular attendees at meetings of the OWL Working Group at the time of publication of this document were: Jie Bao (RPI), Diego Calvanese (Free University of Bozen-Bolzano), Bernardo Cuenca Grau (Oxford University), Martin Dzbor (Open University), Achille Fokoue (IBM Corporation), Christine Golbreich (Université de Versailles St-Quentin and LIRMM), Sandro Hawke (W3C/MIT), Ivan Herman (W3C/ERCIM), Rinke Hoekstra (University of Amsterdam), Ian Horrocks (Oxford University), Elisa Kendall (Sandpiper Software), Markus Krötzsch (FZI), Carsten Lutz (Universität Bremen), Deborah L. McGuinness (RPI), Boris Motik (Oxford University), Jeff Pan (University of Aberdeen), Bijan Parsia (University of Manchester), Peter F. Patel-Schneider (Bell Labs Research, Alcatel-Lucent), Alan Ruttenberg (Science Commons), Uli Sattler (University of Manchester), Michael Schneider (FZI), Mike Smith (Clark & Parsia), Evan Wallace (NIST), and Zhe Wu (Oracle Corporation). We would also like to thank past members of the working group: Jeremy Carroll, Jim Hendler, Vipul Kashyap.