



OWL 2 Web Ontology Language Quick Reference Guide

W3C Working Draft 21 April 2009

This version:

<http://www.w3.org/TR/2009/WD-owl2-quick-reference-20090421/>

Latest version:

<http://www.w3.org/TR/owl2-quick-reference/>

Previous version:

<http://www.w3.org/TR/2008/WD-owl2-quick-reference-20081202/>

Authors:

[Jie Bao](#), Rensselaer Polytechnic Institute

[Elisa F. Kendall](#), Sandpiper Software, Inc.

[Deborah L. McGuinness](#), Rensselaer Polytechnic Institute

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](#).

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.

This document provides a quick reference guide 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 \(UMBC\)](#) on earlier versions of a [Reference Card for the Semantic Web](#).

A draft [printable version](#) is available, but it is obsolete with respect to the current wiki version.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May Be Superseded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/>.

Summary of Changes

This Working Draft has numerous editorial changes since the previous version of 02 December 2008. In particular, the presentation has been significantly improved, and the content has been updated to reflect changes in the language specification.

Please Comment By 12 May 2009

The [OWL Working Group](#) seeks public feedback on this Working Draft. Please send your comments to 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 Namespaces](#)
- [2 OWL 2 constructs and axioms](#)
 - [2.1 Classes](#)
 - [2.1.1 Class Axioms](#)
 - [2.2 Properties](#)
 - [2.2.1 Property Expressions](#)
 - [2.2.2 Property Axioms](#)
 - [2.3 Data Ranges](#)
 - [2.4 Keys](#)
 - [2.5 Assertions](#)
 - [2.6 Declarations](#)
 - [2.7 Annotations](#)
 - [2.8 OWL Ontologies](#)
 - [2.9 Deprecated Vocabulary in OWL 2](#)
- [3 Built-in Datatypes and Facets](#)
 - [3.1 Built-in Datatypes](#)
 - [3.2 Facets](#)
- [4 Acknowledgments](#)

Editor's Note: To do list:

- Complete inline hyperlinks to [Mapping to RDF Graphs](#)
- Make a new pdf print version when the guide is finalized. ([Media:Quick Reference Guide.pdf](#))

Editor's Note: Markup suggestions from Ivan Herman

- the gray background shading is a little bit disturbing, maybe try other typographic trick, e.g., some lighter colour.
- whether we could find a trick so that we can switch on/off highlight the OWL 1/2 differences and the '?' links that refer to the NF&R.

1 Namespaces

The standard namespaces and prefixes in OWL 2 are

| 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# |

2 OWL 2 constructs and axioms

Features in bold are only available in OWL 2. Each table has columns:

- 1st: Feature's name and link to [Primer](#) and [New Features and Rationale](#) (if applicable)(as "(N)")
- 2nd: functional syntax and link to [Syntax](#)
- 3rd: RDF syntax with link to [Mapping to RDF Graphs](#)

We use the following notation conventions: unless stated otherwise, "C" is an OWL class, "D" is a data range, "P" and "Q" are object properties, "R" and "S" are data properties, "a" is an OWL individual, "u" and "v" are literals, "n" is a non-negative integer, "_:x" is anonymous individual. All names may have subscripts. "(a₁ ... a_n)" in the 3rd column stands for a [rdf list](#).

Editor's Note: Christine suggested to use same naming convention as in Syntax, e.g., CE, DR, OPE, DPE, a and It

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

2.1 Classes

Predefined Classes

| | | |
|---------------------|-------------|-------------|
| all OWL individuals | owl:Thing | owl:Thing |
| empty class | owl:Nothing | owl:Nothing |

Boolean Connectives and Enumeration of Individuals

| | | |
|------------------------------|---|--|
| intersection | ObjectIntersectionOf (C ₁ ... C _n) | :x rdf:type owl:Class. :x owl:intersectionOf (C ₁ ... C _n). |
| union | ObjectUnionOf (C ₁ ... C _n) | :x rdf:type owl:Class. :x owl:unionOf (C ₁ ... C _n). |
| complement | ObjectComplementOf (C) | :x rdf:type owl:Class. :x owl:complementOf C. |
| enumeration | ObjectOneOf (a ₁ ... a _n) | :x rdf:type owl:Class. :x owl:oneOf (a ₁ ... a _n). |

Object Property Restrictions

Every owl:Restriction is an owl:Class.

| | | |
|--|--|--|
| universal | ObjectAllValuesFrom (P C) | _:x rdf:type owl:Restriction. _:x owl:onProperty P. _:x owl:allValuesFrom C |
| existential | ObjectSomeValuesFrom (P C) | _:x rdf:type owl:Restriction. _:x owl:onProperty P. _:x owl:someValuesFrom C |
| individual value | ObjectHasValue (P a) | _:x rdf:type owl:Restriction. _:x owl:onProperty P. _:x owl:hasValue i. |
| local reflexivity (N) | ObjectHasSelf (P) | _:x rdf:type owl:Restriction. _:x owl:onProperty P. _:x owl:hasSelf "true"^^xsd:boolean. |
| exact cardinality | ObjectExactCardinality (n P) ObjectExactCardinality (n P C) (N) | _:x rdf:type owl:Restriction. _:x owl:onProperty P. <hr/> _:x owl:cardinality n. (without C) _:x owl:qualifiedCardinality n. _:x owl:onClass C. (with C) |
| maximum cardinality | ObjectMaxCardinality (n P) ObjectMaxCardinality (n P C) (N) | _:x rdf:type owl:Restriction. _:x owl:onProperty P. <hr/> _:x owl:minCardinality n. (without C) <hr/> _:x owl:minQualifiedCardinality n. _:x owl:onClass C. (with C) |
| minimum cardinality | ObjectMinCardinality (n P) ObjectMinCardinality (n P C) (N) | _:x rdf:type owl:Restriction. _:x owl:onProperty P. <hr/> _:x owl:maxCardinality n. (without C) <hr/> _:x owl:maxQualifiedCardinality n. _:x owl:onClass C. (with C) |

Data Property Restrictions

Every owl:Restriction is an owl:Class.

| | | |
|-------------|--|---|
| universal | DataAllValuesFrom (R D) | _:x rdf:type owl:Restriction. _:x owl:onProperty R. _:x owl:allValuesFrom D. |
| existential | DataSomeValuesFrom (R D) | _:x rdf:type owl:Restriction. _:x owl:onProperty R. _:x owl:someValuesFrom D. |

| | | |
|---------------------|--|--|
| individual value | DataHasValue (R u) | :x rdf:type owl:Restriction. :x owl:onProperty R. :x owl:hasValue u. |
| exact cardinality | DataExactCardinality (n R) | :x rdf:type owl:Restriction. :x owl:onProperty P. :x owl:cardinality "n"^^xsd:nonNegativeInteger. |
| | DataExactCardinality (n R D) (N) | :x owl:qualifiedCardinality "n"^^xsd:nonNegativeInteger. (without D) :x owl:onDataRange D. (with D) |
| maximum cardinality | DataMaxCardinality (n R) | :x rdf:type owl:Restriction. :x owl:onProperty P. :x owl:maxCardinality "n"^^xsd:nonNegativeInteger. (without D) |
| | DataMaxCardinality (n R D) (N) | :x owl:maxQualifiedCardinality "n"^^xsd:nonNegativeInteger. :x owl:onDataRange D. (with D) |
| minimum cardinality | DataMinCardinality (n R) | :x rdf:type owl:Restriction. :x owl:onProperty P.(without D) :x owl:minCardinality "n"^^xsd:nonNegativeInteger. |
| | DataMinCardinality (n R D) (N) | :x owl:minQualifiedCardinality "n"^^xsd:nonNegativeInteger. :x owl:onDataRange D. (with D) |

Restrictions Using n-ary Data Range

"Dⁿ" is a n-ary data range (cf [#Data Ranges](#)).

| | | |
|------------------------------|--|--|
| n-ary universal (N) | DataAllValuesFrom (R ₁ ... R _n D ⁿ) | :x rdf:type owl:Restriction. :x owl:onProperties (R ₁ ... R _n). :x owl:allValuesFrom D ⁿ . |
| n-ary existential (N) | DataSomeValuesFrom (R ₁ ... R _n D ⁿ) | :x rdf:type owl:Restriction. :x owl:onProperties (R ₁ ... R _n). :x owl:someValuesFrom D ⁿ . |

2.1.1 Class Axioms

| | | |
|----------------------------|---|---|
| subclasses | SubClassOf (C ₁ C ₂) | C ₁ rdfs:subClassOf C ₂ . |
|----------------------------|---|---|

| | | |
|---|--|---|
| equivalent classes | EquivalentClasses (C ₁ ... C _n) | C ₁ owl:equivalentClass C ₂ C _{n-1} owl:equivalentClass C _n . |
| disjoint classes | DisjointClasses (C ₁ C ₂) | C ₁ owl:disjointWith C ₂ . |
| pairwise disjoint classes (N) | DisjointClasses (C ₁ ... C _n) | _:x rdf:type owl:AllDisjointClasses. _:x owl:members (C ₁ ... C _n). |
| disjoint union (N) | DisjointUnionOf (C C ₁ ... C _n) | C owl:disjointUnionOf (C ₁ ... C _n). |

2.2 Properties

2.2.1 Property Expressions

- [Object Properties](#) are instances of owl:ObjectProperty

| | | |
|---|-------------------------------------|--------------------------|
| universal object property (N) | owl:topObjectProperty | owl:topObjectProperty |
| bottom object property (N) | owl:bottomObjectProperty | owl:bottomObjectProperty |
| inverse property | ObjectInverseOf (P) | _:x owl:inverseOf P |

- [Datatype Properties](#) are instances of owl:DatatypeProperty

| | | |
|---|------------------------|------------------------|
| universal datatype property (N) | owl:topDataProperty | owl:topDataProperty |
| bottom datatype property (N) | owl:bottomDataProperty | owl:bottomDataProperty |

2.2.2 Property Axioms

- [Object Property Axioms](#)

| | | |
|--|---|--|
| subproperty | SubObjectPropertyOf (P Q) | P rdfs:subPropertyOf Q. |
| property chain inclusion (N) | SubObjectPropertyOf (ObjectPropertyChain(P ₁ ... P _n) Q) | Q owl:propertyChainAxiom (P ₁ ... P _n). |
| property domain | ObjectPropertyDomain (P C) | P rdfs:domain C. |

| | | |
|--|---|---|
| property range | ObjectPropertyRange (P C) | P rdfs:range C. |
| equivalent properties | EquivalentObjectProperties (P ₁ ... P _n) | P ₁ owl:equivalentProperty P ₂ P _{n-1} owl:equivalentProperty P _n . |
| disjoint properties (N) | DisjointObjectProperties (P ₁ P ₂) | P ₁ owl:propertyDisjointWith P ₂ . |
| pairwise disjoint properties (N) | DisjointObjectProperties (P ₁ ... P _n) | _:x rdf:type owl:AllDisjointProperties. _:x owl:members (P ₁ ... P _n). |
| inverse properties | InverseObjectProperties (P Q) | P owl:inverseOf Q. |
| functional property | FunctionalObjectProperty (P) | P rdf:type owl:FunctionalProperty. |
| inverse functional property | InverseFunctionalObjectProperty (P) | P rdf:type owl:InverseFunctionalProperty. |
| reflexive property (N) | ReflexiveObjectProperty (P) | P rdf:type owl:ReflexiveProperty. |
| irreflexive property (N) | IrreflexiveObjectProperty (P) | P rdf:type owl:IrreflexiveProperty. |
| symmetric property | SymmetricObjectProperty (P) | P rdf:type owl:SymmetricProperty. |
| asymmetric property (N) | AsymmetricObjectProperty (P) | P rdf:type owl:AsymmetricProperty. |
| transitive property | TransitiveObjectProperty (P) | P rdf:type owl:TransitiveProperty. |

• **Datatype Property Axioms**

| | | |
|---------------------------------------|---|---|
| subproperty | SubDataPropertyOf (R S) | R rdfs:subPropertyOf S. |
| property domain | DataPropertyDomain (R C) | R rdfs:domain C. |
| property range | DataPropertyRange (R C) | R rdfs:range C. |
| equivalent properties | EquivalentDataProperties (R ₁ ... R _n) | R ₁ owl:equivalentProperty R ₂ R _{n-1} |

| | | |
|---|---|---|
| | | owl:equivalentProperty R _n . |
| <u>disjoint properties (N)</u> | <u>DisjointDataProperties</u> (R S) | R owl:propertyDisjointWith S. |
| <u>pairwise disjoint properties (N)</u> | <u>DisjointDataProperties</u> (R ₁ ... R _n) | _:x rdf:type owl:AllDisjointProperties. _:x owl:members (R ₁ ... R _n). |
| <u>functional property</u> | <u>FunctionalDataProperty</u> (R) | R rdf:type owl:FunctionalProperty. |

2.3 Data Ranges

Built-in datatypes are unary data ranges. OWL 2 does not provide direct support for n-ary data ranges but provides syntactical hooks for applications to add them.

| | | |
|--|--|--|
| <u>data range complement (N)</u> | <u>DataComplementOf</u> (D) | _:x rdf:type rdfs:Datatype. _:x owl:datatypeComplementOf D. |
| <u>data range intersection</u> | <u>DataUnionOf</u> (D ₁ ...D _n) | _:x rdf:type rdfs:Datatype. _:x owl:unionOf (D ₁ ...D _n). |
| <u>data range union</u> | <u>DataIntersectionOf</u> (D ₁ ...D _n) | _:x rdf:type rdfs:Datatype. _:x owl:intersectionOf (D ₁ ...D _n). |
| <u>literal enumeration</u> | <u>DataOneOf</u> (v ₁ ... v _n) | _:x rdf:type rdfs:Datatype. _:x owl:oneOf (v ₁ ... v _n). |
| <u>datatype restriction (N)</u> | <u>DatatypeRestriction</u> (D f ₁ v ₁ ... f _n v _n) <i>D a built-in datatype</i> <i>f_j a constraining facet,</i> <i>v_j a restriction value</i> | _:x rdf:type rdfs:Datatype. _:x owl:onDatatype D. _:x owl:withRestrictions (y ₁ ... y _n). y ₁ f ₁ v ₁ y _n f _n v _n . |

2.4 Keys

| | | |
|---------------------------------|---|---|
| <u>Keys (N)</u> | <u>HasKey</u> (C (P ₁ ... P _m) (R ₁ ... R _n)) | C owl:hasKey (P ₁ ... P _m R ₁ ... R _n). |
|---------------------------------|---|---|

2.5 Assertions

| | | |
|--|--|---|
| individual equality | SameIndividual (a1 a2) | a1 owl:sameAs a2. |
| n-ary individual equality | SameIndividual (a1 ... an) | a _j owl:sameAs a _{j+1} . j=1...n-1 |
| individual inequality | DifferentIndividuals (a1 a2) | a1 owl:differentFrom a2. |
| pairwise individual inequality | DifferentIndividuals (a1 ... an) | _:x rdf:type owl:AllDifferent. _:x owl:members (a1 ... an). |
| class assertion | ClassAssertion (C a) | i rdf:type C. |
| positive object property assertion | ObjectPropertyAssertion (P a1 a2) | a1 P a2. |
| positive inverse object property assertion | ObjectPropertyAssertion (ObjectInverseOf (P) a1 a2) | a2 P a1. |
| positive data property assertion | DataPropertyAssertion (P a v) | a P v. |
| negative object property assertion (N) | NegativeObjectPropertyAssertion (P a1 a2) | _:x rdf:type owl:NegativePropertyAssertion. _:x owl:sourceIndividual a1. _:x owl:assertionProperty P. _:x owl:targetIndividual a2 |
| negative datatype property assertion (N) | NegativeDataPropertyAssertion (R a u) | _:x rdf:type owl:NegativePropertyAssertion. _:x owl:sourceIndividual a. _:x owl:assertionProperty R. _:x owl:targetValue u |

2.6 Declarations

| | | |
|---------------------------|--|-----------------------|
| class (N) | Declaration (Class(C)) | C rdf:type owl:Class. |
|---------------------------|--|-----------------------|

| | | |
|--|--|------------------------------------|
| <u>datatype (N)</u> | Declaration(Datatype(D)) | D rdf:type rdfs:Datatype. |
| <u>object property (N)</u> | Declaration(ObjectProperty(P)) | P rdf:type owl:ObjectProperty. |
| <u>datatype property (N)</u> | Declaration(DataProperty(R)) | R rdf:type owl:DatatypeProperty. |
| <u>annotation property (N)</u> | Declaration(AnnotationProperty(A)) | A rdf:type owl:AnnotationProperty. |
| <u>named individual (N)</u> | Declaration(NamedIndividual(a)) | a rdf:type owl:NamedIndividual. |

2.7 Annotations

Annotation of an object

s the annotation subject, v a resource; AP annotation property.

| | | |
|--|---|---|
| <u>annotation assertions</u> | <u>AnnotationAssertion</u> (AP s v) | s AP v. or (if the assertion itself has annotation) s AP v. _:x rdf:type owl:Annotation. _:x owl:subject s. _:x owl:predicate AP. _:x owl:object v. |
|--|---|---|

Note: an annotated object can be an ontology, an ontology entity, an anonymous individual, or another annotation.

Annotation of an axiom

y the annotated object, AP annotation property, v a resource

| | | |
|--|---|---|
| <u>axiom annotations (N)</u> | AXIOM(<u>Annotation</u> (AP v)) | s p o. _:x rdf:type owl:Axiom. _:x owl:subject s. _:x owl:predicate p. _:x owl:object o. _:x AP v. If AXIOM(...) becomes s p o. _:x p o _:x AP v. If AXIOM(...) becomes _:x p o. |
|--|---|---|

Note: for n-ary axioms of type EquivalentClasses, EquivalentProperties or SameIndividual, they will first be broken up into several RDF triples of their binary forms, then each triple is reified using the above rule.

Annotation Properties

all annotation properties are instances of owl:AnnotationProperty

| | | |
|----------------------------|------------|----------------------------------|
| human-readable name | Label | rdfs:label |
| human-readable description | Comment | rdfs:comment |
| additional information | | rdfs:seeAlso |
| defining agent | | rdfs:isDefinedBy |
| version information | | owl:versionInfo |
| deprecation | Deprecated | owl:deprecated |

Annotation Axioms

AP annotation property, U an IRI.

| | | |
|---------------------------------------|--|--|
| annotation subproperties (N) | SubAnnotationPropertyOf (AP ₁ AP ₂) | AP ₁ rdfs:subPropertyOf AP ₂ . |
| annotation property domain (N) | AnnotationPropertyDomain (AP U) | AP rdfs:domain U. |
| annotation property range (N) | AnnotationPropertyRange (AP U) | AP rdfs:range U. |

Deprecation

C an OWL class or a datatype, P an object property, datatype property or annotation property.

| | | |
|---------------------|--------------|------------------------------------|
| deprecated class | C Deprecated | C rdf:type owl:DeprecatedClass. |
| deprecated property | P Deprecated | P rdf:type owl:DeprecatedProperty. |

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

2.8 OWL Ontologies

Annotations of Ontologies

O an ontology, U an ontology IRI, V an IRI.

| | | |
|---|--|---|
| OWL ontology (importing) | Ontology (O [V] Import (U)... Annotation(AP, v)...) | O rdf:type owl:Ontology. [O owl:versionIRI V.] O owl:imports U. ... O AP v. ... |
|---|--|---|

Note: if O is unnamed, then O is mapped to `_:x` (an anonymous individual).

Ontology Properties are instances of owl:OntologyProperty

| | |
|-------------------------|---------------------------------|
| backwards compatibility | O owl:backwardCompatibleWith U. |
| incompatibility | O owl:incompatibleWith U. |
| prior version | O owl:priorVersion U. |

2.9 Deprecated Vocabulary in OWL 2

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

3 Built-in Datatypes and Facets

3.1 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.

- **Numbers**

OWL Numeric Datatypes:

| |
|--|
| owl:rational (rational numbers) |
| owl:real (N)(real numbers) |

XSD 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: A Datatype for Internationalized Text](#) Jie Bao, Axel Polleres, Boris Motik. W3C Working Draft, 21 April 2009, <http://www.w3.org/TR/2009/WD-rdf-text-20090421/>. Latest version available at <http://www.w3.org/TR/rdf-text/>.

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: <i>true</i> and <i>false</i>) | |
| • Binary Data | |
| xsd:base64Binary | xsd:hexBinary |
| • IRIs | |
| xsd:anyURI IRIs as defined in XML Schema Datatypes | |
| • Time Instants | |
| xsd:dateTime | time instants with time zone offset |
| xsd:dateTimeStamp | time instants without time zone offset |
| • XML Literals | |
| rdf:XMLLiteral | Note: at risk in OWL 2 |

3.2 Facets

The *facet space* is a set of pairs of the form $\langle f v \rangle$, where f is an IRI 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(N) f | Datatype | Value v | Explanation |
|--|--|--|--|
| xsd:minInclusive , xsd:maxInclusive , xsd:minExclusive , xsd:maxExclusive | Numeric Datatype DT, Time instant 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:langRange | 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 |

4 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), Bijan Parsia (University of Manchester) and Peter F. Patel-Schneider (Bell Labs Research, Alcatel-Lucent) 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.