

W3C: OWL 2 Web Ontology Language - Structural Specification and Functional-Style Syntax (second Edition)

[return to the W3C Standards](#)

Table 1: Data sheet for OWL 2 Web Ontology Language - Structural Specification and Functional-Style Syntax (second Edition)

Title	OWL 2 Web Ontology Language - Structural Specification and Functional-Style Syntax (second Edition)
Acronym	OWL 2
Version	2
Series	TR
Document Number	
Release Date	11 December 2012
Download	http://www.w3.org/2012/pdf/REC-owl2-syntax-20121211.pdf

Note: The following is an excerpt from the [W3C site](#). It is provided here as a convenience and is not authoritative. Refer to the original document as the authoritative reference.

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.

The meaningful constructs provided by OWL 2 are defined in terms of their structure. As well, a functional-style syntax is defined for these constructs, with examples and informal descriptions. One can reason with OWL 2 ontologies under either the RDF-Based Semantics [OWL 2 RDF-Based Semantics] or the Direct Semantics [OWL 2 Direct Semantics]. If certain restrictions on OWL 2 ontologies are satisfied and the ontology is in OWL 2 DL, reasoning under the Direct Semantics can be implemented using techniques well known in the literature.

Introduction

This document defines the OWL 2 language. The core part of this specification — called the structural specification — is independent of the concrete exchange syntaxes for OWL 2 ontologies. The structural specification describes the conceptual structure of OWL 2 ontologies and thus

provides a normative abstract representation for all (normative and nonnormative) syntaxes of OWL 2. This allows for a clear separation of the essential features of the language from issues related to any particular syntax. Furthermore, such a structural specification of OWL 2 provides the foundation for the implementation of OWL 2 tools such as APIs and reasoners. Each OWL 2 ontology represented as an instance of this conceptual structure can be converted into an RDF graph [OWL 2 RDF Mapping](#); conversely, most OWL 2 ontologies represented as RDF graphs can be converted into the conceptual structure defined in this document as OWL 2 RDF Mapping.

This document also defines the functional-style syntax, which closely follows the structural specification and allows OWL 2 ontologies to be written in a compact form. This syntax is used in the definitions of the semantics of OWL 2 ontologies, the mappings from and into the RDF/XML exchange syntax, and the different profiles of OWL 2. Concrete syntaxes, such as the functional-style syntax, often provide features not found in the structural specification, such as a mechanism for abbreviating IRIs.

Finally, this document defines OWL 2 DL — the subset of OWL 2 with favorable computational properties. Each RDF graph obtained by applying the RDF mapping to an OWL 2 DL ontology can be converted back into the conceptual structure defined in this document by means of the reverse RDF mapping as OWL 2 RDF Mapping.

An OWL 2 ontology is a formal description of a domain of interest. OWL 2 ontologies consist of the following three different syntactic categories:

- *Entities, such as classes, properties, and individuals, are identified by IRIs. They form the primitive terms of an ontology and constitute the basic elements of an ontology. For example, a class `a:Person` can be used to represent the set of all people. Similarly, the object property `a:parentOf` can be used to represent the parent-child relationship. Finally, the individual `a:Peter` can be used to represent a particular person called “Peter”.*
- *Expressions represent complex notions in the domain being described. For example, a class expression describes a set of individuals in terms of the restrictions on the individuals' characteristics.*
- *Axioms are statements that are asserted to be true in the domain being described. For example, using a subclass axiom, one can state that the class `a:Student` is a subclass of the class `a:Person`.*

These three syntactic categories are used to express the logical part of OWL 2 ontologies — that is, they are interpreted under a precisely defined semantics that allows useful inferences to be drawn. For example, if an individual `a:Peter` is an instance of the class `a:Student`, and `a:Student` is a subclass of `a:Person`, then from the OWL 2 semantics one can derive that `a:Peter` is also an instance of `a:Person`.

In addition, entities, axioms, and ontologies can be annotated in OWL 2. For example, a class can be given a human-readable label that provides a more descriptive name for the class. Annotations have no effect on the logical aspects of an ontology — that is, for the purposes of the OWL 2 semantics, annotations are treated as not being present. Instead, the use of annotations is left to the applications that use OWL 2. For example, a graphical user interface might choose to visualize

