RDF 1.1 Concepts and Abstract Syntax

W3C Recommendation 25 February 2014

This version:
http://www.w3.org/TR/2014/REC-rdf11-concepts-20140225/
Latest published version:
http://www.w3.org/TR/rdf11-concepts/
Previous version:
http://www.w3.org/TR/2014/PR-rdf11-concepts-20140109/
Previous Recommendation:
http://www.w3.org/TR/rdf-concepts
Editors:
Richard Cyganiak, DERI, NUI Galway
David Wood, 3 Round Stones
Markus Lanthaler, Graz University of Technology
Previous Editors:
Graham Klyne
Jeremy J. Carroll
Brian McBride

Please check the **errata** for any errors or issues reported since publication.

The English version of this specification is the only normative version. Non-normative translations may also be available.

Abstract

The Resource Description Framework (RDF) is a framework for representing information in the Web. This document defines an abstract syntax (a data model) which serves to link all RDF-based languages and specifications. The abstract syntax has two key data structures: RDF graphs are sets of subject-predicate-object triples, where the elements may be IRIs, blank nodes, or datatyped literals. They are used to express descriptions of resources. RDF datasets are used to organize collections of RDF graphs, and comprise a default graph and zero or more named graphs.
RDF 1.1 Concepts and Abstract Syntax also introduces key concepts and terminology, and discusses datatyping and the handling of fragment identifiers in IRIs within RDF graphs.

Status of This Document

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/.

This document is part of the RDF 1.1 document suite. It is the central RDF 1.1 specification and defines the core RDF concepts. A new concept in RDF 1.1 is the notion of an RDF dataset to represent multiple graphs. Test suites and implementation reports of a number of RDF 1.1 specifications that build on this document are available through the RDF 1.1 Test Cases document [RDF11-TESTCASES]. There have been no changes to this document since its publication as Proposed Recommendation.

This document was published by the RDF Working Group as a Recommendation. If you wish to make comments regarding this document, please send them to public-rdf-comments@w3.org (subscribe, archives). All comments are welcome.

This document has been reviewed by W3C Members, by software developers, and by other W3C groups and interested parties, and is endorsed by the Director as a W3C Recommendation. It is a stable document and may be used as reference material or cited from another document. W3C's role in making the Recommendation is to draw attention to the specification and to promote its widespread deployment. This enhances the functionality and interoperability of the Web.

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.

Table of Contents

1. Introduction

This section is non-normative.

The *Resource Description Framework* (RDF) is a framework for representing information in the Web.

This document defines an abstract syntax (a data model) which serves to link all RDF-based languages and specifications, including:

- the formal model-theoretic semantics for RDF [RDF11-MT];
- serialization syntaxes for storing and exchanging RDF such as Turtle [TURTLE] and JSON-LD [JSON-LD];
- the SPARQL Query Language [SPARQL11-OVERVIEW];
- the RDF Schema vocabulary [RDF11-SCHEMA].

1.1 Graph-based Data Model

The core structure of the abstract syntax is a set of triples, each consisting of a subject, a predicate and an object. A set of such triples is called an RDF graph. An RDF graph can be visualized as a node and directed-arc diagram, in which each triple is represented as a node-arc-node link.

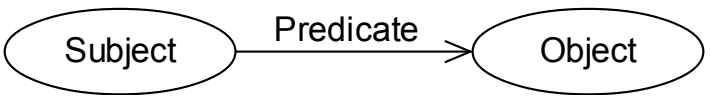

Fig. 1 An RDF graph with two nodes (Subject and Object) and a triple connecting them (Predicate)

There can be three kinds of nodes in an RDF graph: IRIs, literals, and blank nodes.

1.2 Resources and Statements

Any IRI or literal ***denotes*** something in the world (the "universe of discourse"). These

things are called ***resources***. Anything can be a resource, including physical things, documents, abstract concepts, numbers and strings; the term is synonymous with "entity" as it is used in the RDF Semantics specification [RDF11-MT]. The resource denoted by an IRI is called its referent, and the resource denoted by a literal is called its literal value. Literals have datatypes that define the range of possible values, such as strings, numbers, and dates. Special kind of literals, language-tagged strings, denote plain-text strings in a natural language.

Asserting an RDF triple says that *some relationship, indicated by the predicate, holds between the resources denoted by the subject and object*. This statement corresponding to an RDF triple is known as an ***RDF statement***. The predicate itself is an IRI and denotes a ***property***, that is, a resource that can be thought of as a binary relation. (Relations that involve more than two entities can only be indirectly expressed in RDF [SWBP-N-ARYRELATIONS].)

Unlike IRIs and literals, blank nodes do not identify specific resources. Statements involving blank nodes say that something with the given relationships exists, without explicitly naming it.

1.3 The Referent of an IRI

The resource denoted by an IRI is also called its ***referent***. For some IRIs with particular meanings, such as those identifying XSD datatypes, the referent is fixed by this specification. For all other IRIs, what exactly is denoted by any given IRI is not defined by this specification. Other specifications may fix IRI referents, or apply other constraints on what may be the referent of any IRI.

Guidelines for determining the referent of an IRI are provided in other documents, like *Architecture of the World Wide Web, Volume One* [WEBARCH] and *Cool URIs for the Semantic Web* [COOLURIS]. A very brief, informal, and partial account follows:

- By design, IRIs have global scope. Thus, two different appearances of an IRI denote the same resource. Violating this principle constitutes an IRI collision [WEBARCH].
- By social convention, the IRI owner [WEBARCH] gets to say what the intended (or usual) referent of an IRI is. Applications and users need not abide by this intended denotation, but there may be a loss of interoperability with other applications and users if they do not do so.
- The IRI owner can establish the intended referent by means of a specification or other document that explains what is denoted. For example, the Organization Ontology document [VOCAB-ORG] specifies the intended referents of various IRIs that start with `http://www.w3.org/ns/org#`.
- A good way of communicating the intended referent is to set up the IRI so that it dereferences [WEBARCH] to such a document.
- Such a document can, in fact, be an RDF document that describes the denoted resource by means of RDF statements.

Perhaps the most important characteristic of IRIs in web architecture is that they can be dereferenced, and hence serve as starting points for interactions with a remote server. This specification is not concerned with such interactions. It does not define an interaction model. It only treats IRIs as globally unique identifiers in a graph data

model that describes resources. However, those interactions are critical to the concept of *Linked Data* [LINKED-DATA], which makes use of the RDF data model and serialization formats.

1.4 RDF Vocabularies and Namespace IRIs

An ***RDF vocabulary*** is a collection of IRIs intended for use in RDF graphs. For example, the IRIs documented in [RDF11-SCHEMA] are the RDF Schema vocabulary. RDF Schema can itself be used to define and document additional RDF vocabularies. Some such vocabularies are mentioned in the Primer [RDF11-PRIMER].

The IRIs in an RDF vocabulary often begin with a common substring known as a ***namespace IRI***. Some namespace IRIs are associated by convention with a short name known as a ***namespace prefix***. Some examples:

Some example namespace prefixes and IRIs

Namespace prefix	Namespace IRI	RDF vocabulary
rdf	`http://www.w3.org/1999/02/22-rdf-syntax-ns#`	The RDF built-in vocabulary [RDF11-SCHEMA]
rdfs	`http://www.w3.org/2000/01/rdf-schema#`	The RDF Schema vocabulary [RDF11-SCHEMA]
xsd	`http://www.w3.org/2001/XMLSchema#`	The RDF-compatible XSD types

In some serialization formats it is common to abbreviate IRIs that start with namespace IRIs by using a namespace prefix in order to assist readability. For example, the IRI `http://www.w3.org/1999/02/22-rdf-syntax-ns#XMLLiteral` would be abbreviated as `rdf:XMLLiteral`. Note however that these abbreviations are *not* valid IRIs, and must not be used in contexts where IRIs are expected. Namespace IRIs and namespace prefixes are *not* a formal part of the RDF data model. They are merely a syntactic convenience for abbreviating IRIs.

The term "***namespace***" on its own does not have a well-defined meaning in the context of RDF, but is sometimes informally used to mean "namespace IRI" or "RDF vocabulary".

1.5 RDF and Change over Time

The RDF data model is *atemporal*: RDF graphs are static snapshots of information.

However, RDF graphs can express information about events and about temporal aspects of other entities, given appropriate vocabulary terms.

Since RDF graphs are defined as mathematical sets, adding or removing triples from an RDF graph yields a different RDF graph.

We informally use the term ***RDF source*** to refer to a persistent yet mutable source or container of RDF graphs. An RDF source is a resource that may be said to have a state that can change over time. A snapshot of the state can be expressed as an RDF graph. For example, any web document that has an RDF-bearing representation may be considered an RDF source. Like all resources, RDF sources may be named with IRIs and therefore described in other RDF graphs.

Intuitively speaking, changes in the universe of discourse can be reflected in the following ways:

- An IRI, once minted, should never change its intended referent. (See URI persistence [WEBARCH].)
- Literals, by design, are constants and never change their value.
- A relationship that holds between two resources at one time may not hold at another time.
- RDF sources may change their state over time. That is, they may provide different RDF graphs at different times.
- Some RDF sources may, however, be immutable snapshots of another RDF source, archiving its state at some point in time.

1.6 Working with Multiple RDF Graphs

As RDF graphs are sets of triples, they can be combined easily, supporting the use of data from multiple sources. Nevertheless, it is sometimes desirable to work with multiple RDF graphs while keeping their contents separate. RDF datasets support this requirement.

An RDF dataset is a collection of RDF graphs. All but one of these graphs have an associated IRI or blank node. They are called named graphs, and the IRI or blank node is called the graph name. The remaining graph does not have an associated IRI, and is called the default graph of the RDF dataset.

There are many possible uses for RDF datasets. One such use is to hold snapshots of multiple RDF sources.

1.7 Equivalence, Entailment and Inconsistency

An RDF triple encodes a statement—a simple ***logical expression***, or claim about the world. An RDF graph is the conjunction (logical *AND*) of its triples. The precise details of this meaning of RDF triples and graphs are the subject of the RDF Semantics specification [RDF11-MT], which yields the following relationships between RDF graphs:

Entailment
: An RDF graph *A* entails another RDF graph *B* if every possible arrangement of the world that makes *A* true also makes *B* true. When *A* entails *B*, if the truth of *A* is presumed or demonstrated then the truth of *B* is established.

Equivalence
: Two RDF graphs *A* and *B* are equivalent if they make the same claim about the world. *A* is equivalent to *B* if and only if *A* entails *B* and *B* entails *A*.

Inconsistency

An RDF graph is inconsistent if it contains an internal contradiction. There is no possible arrangement of the world that would make the expression true.

An ***entailment regime*** [RDF11-MT] is a specification that defines precise conditions that make these relationships hold. RDF itself recognizes only some basic cases of entailment, equivalence and inconsistency. Other specifications, such as RDF Schema [RDF11-SCHEMA] and OWL 2 [OWL2-OVERVIEW], add more powerful entailment regimes, as do some domain-specific vocabularies.

This specification does not constrain how implementations use the logical relationships defined by entailment regimes. Implementations may or may not detect inconsistencies, and may make all, some or no entailed information available to users.

1.8 RDF Documents and Syntaxes

An ***RDF document*** is a document that encodes an RDF graph or RDF dataset in a ***concrete RDF syntax***, such as Turtle [TURTLE], RDFa [RDFA-PRIMER], JSON-LD [JSON-LD], or TriG [TRIG]. RDF documents enable the exchange of RDF graphs and RDF datasets between systems.

A concrete RDF syntax may offer many different ways to encode the same RDF graph or RDF dataset, for example through the use of namespace prefixes, relative IRIs, blank node identifiers, and different ordering of statements. While these aspects can have great effect on the convenience of working with the RDF document, they are not significant for its meaning.

2. Conformance

As well as sections marked as non-normative, all authoring guidelines, diagrams, examples, and notes in this specification are non-normative. Everything else in this specification is normative.

The key words MUST, MUST NOT, REQUIRED, SHOULD, SHOULD NOT, RECOMMENDED, MAY, and OPTIONAL in this specification are to be interpreted as described in [RFC2119].

This specification, *RDF 1.1 Concepts and Abstract Syntax*, defines a data model and related terminology for use in other specifications, such as concrete RDF syntaxes, API specifications, and query languages. Implementations cannot directly conform to *RDF 1.1 Concepts and Abstract Syntax*, but can conform to such other specifications that normatively reference terms defined here.

3. RDF Graphs

An ***RDF graph*** is a set of RDF triples.

3.1 Triples

An ***RDF triple*** consists of three components:

- the ***subject***, which is an IRI or a blank node

- the ***predicate***, which is an IRI
- the ***object***, which is an IRI, a literal or a blank node

An RDF triple is conventionally written in the order subject, predicate, object.

The set of ***nodes*** of an RDF graph is the set of subjects and objects of triples in the graph. It is possible for a predicate IRI to also occur as a node in the same graph.

IRIs, literals and blank nodes are collectively known as ***RDF terms***.

IRIs, literals and blank nodes are distinct and distinguishable. For example, `http://example.org/` as a string literal is neither equal to `http://example.org/` as an IRI, nor to a blank node with the blank node identifier `http://example.org/`.

3.2 IRIs

An ***IRI*** (Internationalized Resource Identifier) within an RDF graph is a Unicode string [UNICODE] that conforms to the syntax defined in RFC 3987 [RFC3987].

IRIs in the RDF abstract syntax MUST be absolute, and MAY contain a fragment identifier.

IRI equality: Two IRIs are equal if and only if they are equivalent under Simple String Comparison according to section 5.1 of [RFC3987]. Further normalization MUST NOT be performed when comparing IRIs for equality.

> NOTE
>
> **URIs and IRIs:** IRIs are a generalization of ***URIs*** [RFC3986] that permits a wider range of Unicode characters. Every absolute URI and URL is an IRI, but not every IRI is an URI. When IRIs are used in operations that are only defined for URIs, they must first be converted according to the mapping defined in section 3.1 of [RFC3987]. A notable example is retrieval over the HTTP protocol. The mapping involves UTF-8 encoding of non-ASCII characters, %-encoding of octets not allowed in URIs, and Punycode-encoding of domain names.
>
> **Relative IRIs:** Some concrete RDF syntaxes permit ***relative IRIs*** as a convenient shorthand that allows authoring of documents independently from their final publishing location. Relative IRIs must be resolved against a ***base IRI*** to make them absolute. Therefore, the RDF graph serialized in such syntaxes is well-defined only if a base IRI can be established [RFC3986].
>
> **IRI normalization:** Interoperability problems can be avoided by minting only IRIs that are normalized according to Section 5 of [RFC3987]. Non-normalized forms that are best avoided include:
>
> - Uppercase characters in scheme names and domain names
> - Percent-encoding of characters where it is not required by IRI syntax
> - Explicitly stated HTTP default port (`http://example.com:80/`); `http://example.com/` is preferable

- Completely empty path in HTTP IRIs (`http://example.com`); `http://example.com/` is preferable
- "`/./`" or "`/../`" in the path component of an IRI
- Lowercase hexadecimal letters within percent-encoding triplets ("`%3F`" is preferable over "`%3f`")
- Punycode-encoding of Internationalized Domain Names in IRIs
- IRIs that are not in Unicode Normalization Form C [NFC]

3.3 Literals

Literals are used for values such as strings, numbers, and dates.

A ***literal*** in an RDF graph consists of two or three elements:

- a ***lexical form***, being a Unicode [UNICODE] string, which SHOULD be in Normal Form C [NFC],
- a ***datatype IRI***, being an IRI identifying a datatype that determines how the lexical form maps to a literal value, and
- if and only if the datatype IRI is `http://www.w3.org/1999/02/22-rdf-syntax-ns#langString`, a non-empty ***language tag*** as defined by [BCP47]. The language tag MUST be well-formed according to section 2.2.9 of [BCP47].

A literal is a ***language-tagged string*** if the third element is present. Lexical representations of language tags MAY be converted to lower case. The value space of language tags is always in lower case.

Please note that concrete syntaxes MAY support ***simple literals*** consisting of only a lexical form without any datatype IRI or language tag. Simple literals are syntactic sugar for abstract syntax literals with the datatype IRI `http://www.w3.org/2001/XMLSchema#string`. Similarly, most concrete syntaxes represent language-tagged strings without the datatype IRI because it always equals `http://www.w3.org/1999/02/22-rdf-syntax-ns#langString`.

The ***literal value*** associated with a literal is:

1. If the literal is a language-tagged string, then the literal value is a pair consisting of its lexical form and its language tag, in that order.
2. If the literal's datatype IRI is in the set of recognized datatype IRIs, let *d* be the referent of the datatype IRI.
    a. If the literal's lexical form is in the lexical space of *d*, then the literal value is the result of applying the lexical-to-value mapping of *d* to the lexical form.
    b. Otherwise, the literal is ill-typed and no literal value can be associated with the literal. Such a case produces a semantic inconsistency but is not *syntactically* ill-formed. Implementations MUST accept ill-typed literals and produce RDF graphs from them. Implementations MAY produce warnings when encountering ill-typed literals.
3. If the literal's datatype IRI is *not* in the set of recognized datatype IRIs, then the literal value is not defined by this specification.

Literal term equality: Two literals are term-equal (the same RDF literal) if and only if

the two lexical forms, the two datatype IRIs, and the two language tags (if any) compare equal, character by character. Thus, two literals can have the same value without being the same RDF term. For example:

```
"1"^^xs:integer
"01"^^xs:integer
```

denote the same value, but are not the same literal RDF terms and are not term-equal because their lexical form differs.

3.4 Blank Nodes

Blank nodes are disjoint from IRIs and literals. Otherwise, the set of possible blank nodes is arbitrary. RDF makes no reference to any internal structure of blank nodes.

> NOTE
>
> ***Blank node identifiers*** are local identifiers that are used in some concrete RDF syntaxes or RDF store implementations. They are always locally scoped to the file or RDF store, and are *not* persistent or portable identifiers for blank nodes. Blank node identifiers are *not* part of the RDF abstract syntax, but are entirely dependent on the concrete syntax or implementation. The syntactic restrictions on blank node identifiers, if any, therefore also depend on the concrete RDF syntax or implementation. Implementations that handle blank node identifiers in concrete syntaxes need to be careful not to create the same blank node from multiple occurrences of the same blank node identifier except in situations where this is supported by the syntax.

3.5 Replacing Blank Nodes with IRIs

Blank nodes do not have identifiers in the RDF abstract syntax. The blank node identifiers introduced by some concrete syntaxes have only local scope and are purely an artifact of the serialization.

In situations where stronger identification is needed, systems MAY systematically replace some or all of the blank nodes in an RDF graph with IRIs. Systems wishing to do this SHOULD mint a new, globally unique IRI (a ***Skolem IRI***) for each blank node so replaced.

This transformation does not appreciably change the meaning of an RDF graph, provided that the Skolem IRIs do not occur anywhere else. It does however permit the possibility of other graphs subsequently using the Skolem IRIs, which is not possible for blank nodes.

Systems may wish to mint Skolem IRIs in such a way that they can recognize the IRIs as having been introduced solely to replace blank nodes. This allows a system to map IRIs back to blank nodes if needed.

Systems that want Skolem IRIs to be recognizable outside of the system boundaries

SHOULD use a well-known IRI [RFC5785] with the registered name `genid`. This is an IRI that uses the HTTP or HTTPS scheme, or another scheme that has been specified to use well-known IRIs; and whose path component starts with `/.well-known/genid/`.

For example, the authority responsible for the domain `example.com` could mint the following recognizable Skolem IRI:

```
http://example.com/.well-known/genid/d26a2d0e98334696f4ad70a677abc1f6
```

> NOTE
>
> RFC 5785 [RFC5785] only specifies well-known URIs, not IRIs. For the purpose of this document, a well-known IRI is any IRI that results in a well-known URI after IRI-to-URI mapping [RFC3987].

3.6 Graph Comparison

Two RDF graphs *G* and *G'* are ***isomorphic*** (that is, they have an identical form) if there is a bijection *M* between the sets of nodes of the two graphs, such that:

1. *M* maps blank nodes to blank nodes.
2. *M(lit)=lit* for all RDF literals *lit* which are nodes of *G*.
3. *M(iri)=iri* for all IRIs *iri* which are nodes of *G*.
4. The triple *(s, p, o)* is in *G* if and only if the triple *(M(s), p, M(o))* is in *G'*

See also: IRI equality, literal term equality.

With this definition, *M* shows how each blank node in *G* can be replaced with a new blank node to give *G'*. Graph isomorphism is needed to support the RDF Test Cases [RDF11-TESTCASES] specification.

4. RDF Datasets

An ***RDF dataset*** is a collection of RDF graphs, and comprises:

- Exactly one ***default graph***, being an RDF graph. The default graph does not have a name and MAY be empty.
- Zero or more ***named graphs***. Each named graph is a pair consisting of an IRI or a blank node (the ***graph name***), and an RDF graph. Graph names are unique within an RDF dataset.

Blank nodes can be shared between graphs in an RDF dataset.

> NOTE
>
> Despite the use of the word “name” in “named graph”, the graph name is not required to denote the graph. It is merely syntactically paired with the graph. RDF does not place any formal restrictions on what resource the graph name may denote, nor on the relationship between that resource and the graph. A

discussion of different RDF dataset semantics can be found in [RDF11-DATASETS].

Some RDF dataset implementations do not track empty named graphs. Applications can avoid interoperability issues by not ascribing importance to the presence or absence of empty named graphs.

SPARQL 1.1 [SPARQL11-OVERVIEW] also defines the concept of an RDF Dataset. The definition of an RDF Dataset in SPARQL 1.1 and this specification differ slightly in that this specification allows RDF Graphs to be identified using either an IRI or a blank node. SPARQL 1.1 Query Language only allows RDF Graphs to be identified using an IRI. Existing SPARQL implementations might not allow blank nodes to be used to identify RDF Graphs for some time, so their use can cause interoperability problems. Skolemizing blank nodes used as graph names can be used to overcome these interoperability problems.

4.1 RDF Dataset Comparison

Two RDF datasets (the RDF dataset *D1* with default graph *DG1* and any named graph *NG1* and the RDF dataset *D2* with default graph *DG2* and any named graph *NG2*) are ***dataset-isomorphic*** if and only if there is a bijection *M* between the nodes, triples and graphs in *D1* and those in *D2* such that:

1. *M* maps blank nodes to blank nodes;
2. *M* is the identity map on literals and URIs;
3. For every triple <s p o>, *M*(<s, p, o>)= <*M(s)*, *M(p)*, *M(o)*>;
4. For every graph *G*={t1, ..., tn}, *M(G)*={*M(t1)*, ..., *M(tn)*};
5. *DG2* = *M(DG1)*; and
6. <n, G> is in *NG1* if and only if <*M(n)*, *M(G)*> is in *NG2*.

4.2 Content Negotiation of RDF Datasets

This section is non-normative.

Web resources may have multiple representations that are made available via content negotiation [WEBARCH]. A representation may be returned in an RDF serialization format that supports the expression of both RDF datasets and RDF graphs. If an RDF dataset is returned and the consumer is expecting an RDF graph, the consumer is expected to use the RDF dataset's default graph.

5. Datatypes

Datatypes are used with RDF literals to represent values such as strings, numbers and dates. The datatype abstraction used in RDF is compatible with XML Schema [XMLSCHEMA11-2]. Any datatype definition that conforms to this abstraction MAY be used in RDF, even if not defined in terms of XML Schema. RDF re-uses many of the XML Schema built-in datatypes, and defines two additional non-normative datatypes, `rdf:HTML` and `rdf:XMLLiteral`. The list of datatypes supported by an implementation

is determined by its recognized datatype IRIs.

A ***datatype*** consists of a lexical space, a value space and a lexical-to-value mapping, and is denoted by one or more IRIs.

The ***lexical space*** of a datatype is a set of Unicode [UNICODE] strings.

The ***lexical-to-value mapping*** of a datatype is a set of pairs whose first element belongs to the lexical space, and the second element belongs to the ***value space*** of the datatype. Each member of the lexical space is paired with exactly one value, and is a *lexical representation* of that value. The mapping can be seen as a function from the lexical space to the value space.

> NOTE
>
> Language-tagged strings have the datatype IRI `http://www.w3.org/1999/02/22-rdf-syntax-ns#langString`. No datatype is formally defined for this IRI because the definition of datatypes does not accommodate language tags in the lexical space. The value space associated with this datatype IRI is the set of all pairs of strings and language tags.

For example, the XML Schema datatype `xsd:boolean`, where each member of the value space has two lexical representations, is defined as follows:

Lexical space:
{"`true`", "`false`", "`1`", "`0`"}
Value space:
{***true***, ***false***}
Lexical-to-value mapping
{ <"`true`", ***true***>, <"`false`", ***false***>, <"`1`", ***true***>, <"`0`", ***false***>, }

The literals that can be defined using this datatype are:

This table lists the literals of type xsd:boolean.

Literal	Value
<"`true`", `xsd:boolean`>	***true***
<"`false`", `xsd:boolean`>	***false***
<"`1`", `xsd:boolean`>	***true***
<"`0`", `xsd:boolean`>	***false***

5.1 The XML Schema Built-in Datatypes

IRIs of the form `http://www.w3.org/2001/XMLSchema#xxx`, where *`xxx`* is the name of a datatype, denote the built-in datatypes defined in *XML Schema 1.1 Part 2: Datatypes* [XMLSCHEMA11-2]. The XML Schema built-in types listed in the following table are the ***RDF-compatible XSD types***. Their use is RECOMMENDED.

Readers might note that the xsd:hexBinary and xsd:base64Binary datatypes are the only safe datatypes for transferring binary information.

A list of the RDF-compatible XSD types, with short descriptions"

	Datatype	Value space (informative)
Core types	xsd:string	Character strings (but not all Unicode character strings)
	xsd:boolean	true, false
	xsd:decimal	Arbitrary-precision decimal numbers
	xsd:integer	Arbitrary-size integer numbers
IEEE floating-point numbers	xsd:double	64-bit floating point numbers incl. ±Inf, ±0, NaN
	xsd:float	32-bit floating point numbers incl. ±Inf, ±0, NaN
Time and date	xsd:date	Dates (yyyy-mm-dd) with or without timezone
	xsd:time	Times (hh:mm:ss.sss…) with or without timezone
	xsd:dateTime	Date and time with or without timezone
	xsd:dateTimeStamp	Date and time with required timezone
Recurring and partial dates	xsd:gYear	Gregorian calendar year
	xsd:gMonth	Gregorian calendar month
	xsd:gDay	Gregorian calendar day of the month
	xsd:gYearMonth	Gregorian calendar year and month
	xsd:gMonthDay	Gregorian calendar month and day
	xsd:duration	Duration of time
	xsd:yearMonthDuration	Duration of time (months and years only)
	xsd:dayTimeDuration	Duration of time (days, hours, minutes, seconds only)
Limited-range integer	xsd:byte	-128…+127 (8 bit)
	xsd:short	-32768…+32767 (16 bit)
	xsd:int	-2147483648…+2147483647 (32 bit)
	xsd:long	-9223372036854775808… +9223372036854775807 (64 bit)
	xsd:unsignedByte	0…255 (8 bit)
	xsd:unsignedShort	0…65535 (16 bit)

numbers	xsd:unsignedInt	0…4294967295 (32 bit)
	xsd:unsignedLong	0…18446744073709551615 (64 bit)
	xsd:positiveInteger	Integer numbers >0
	xsd:nonNegativeInteger	Integer numbers ≥0
	xsd:negativeInteger	Integer numbers <0
	xsd:nonPositiveInteger	Integer numbers ≤0
Encoded binary data	xsd:hexBinary	Hex-encoded binary data
	xsd:base64Binary	Base64-encoded binary data
Miscellaneous XSD types	xsd:anyURI	Absolute or relative URIs and IRIs
	xsd:language	Language tags per [BCP47]
	xsd:normalizedString	Whitespace-normalized strings
	xsd:token	Tokenized strings
	xsd:NMTOKEN	XML NMTOKENs
	xsd:Name	XML Names
	xsd:NCName	XML NCNames

The other built-in XML Schema datatypes are unsuitable for various reasons and SHOULD NOT be used:

- xsd:QName and xsd:ENTITY require an enclosing XML document context.
- xsd:ID and xsd:IDREF are for cross references within an XML document.
- xsd:NOTATION is not intended for direct use.
- xsd:IDREFS, xsd:ENTITIES and xsd:NMTOKENS are sequence-valued datatypes which do not fit the RDF datatype model.

5.2 The rdf:HTML Datatype

This section is non-normative.

RDF provides for HTML content as a possible literal value. This allows markup in literal values. Such content is indicated in an RDF graph using a literal whose datatype is set to ***rdf:HTML***. This datatype is defined as non-normative because it depends on [DOM4], a specification that has not yet reached W3C Recommendation status.

The rdf:HTML datatype is defined as follows:

The IRI denoting this datatype
: is http://www.w3.org/1999/02/22-rdf-syntax-ns#HTML.

The lexical space
: is the set of Unicode [UNICODE] strings.

The value space
: is a set of DOM DocumentFragment nodes [DOM4]. Two DocumentFragment nodes *A* and *B* are considered equal if and only if the DOM method

`A.isEqualNode(B)` [DOM4] returns `true`.

The lexical-to-value mapping

Each member of the lexical space is associated with the result of applying the following algorithm:

- Let `domnodes` be the list of DOM nodes [DOM4] that result from applying the HTML fragment parsing algorithm [HTML5] to the input string, without a context element.
- Let `domfrag` be a DOM `DocumentFragment` [DOM4] whose `childNodes` attribute is equal to `domnodes`
- Return `domfrag.normalize()`

> NOTE
>
> Any language annotation (`lang="…"`) or XML namespaces (`xmlns`) desired in the HTML content must be included explicitly in the HTML literal. Relative URLs in attributes such as `href` do not have a well-defined base URL and are best avoided. RDF applications may use additional equivalence relations, such as that which relates an `xsd:string` with an `rdf:HTML` literal corresponding to a single text node of the same string.

5.3 The `rdf:XMLLiteral` Datatype

This section is non-normative.

RDF provides for XML content as a possible literal value. Such content is indicated in an RDF graph using a literal whose datatype is set to ***`rdf:XMLLiteral`***. This datatype is defined as non-normative because it depends on [DOM4], a specification that has not yet reached W3C Recommendation status.

The `rdf:XMLLiteral` datatype is defined as follows:

The IRI denoting this datatype
is `http://www.w3.org/1999/02/22-rdf-syntax-ns#XMLLiteral`.

The lexical space
is the set of all strings which are well-balanced, self-contained XML content [XML10]; and for which embedding between an arbitrary XML start tag and an end tag yields a document conforming to XML Namespaces [XML-NAMES].

The value space
is a set of DOM `DocumentFragment` nodes [DOM4]. Two `DocumentFragment` nodes *A* and *B* are considered equal if and only if the DOM method `A.isEqualNode(B)` returns `true`.

The lexical-to-value mapping

Each member of the lexical space is associated with the result of applying the following algorithm:

- Let `domfrag` be a DOM `DocumentFragment` node [DOM4] corresponding

to the input string
- Return `domfrag.normalize()`

The canonical mapping

defines a canonical lexical form [XMLSCHEMA11-2] for each member of the value space. The `rdf:XMLLiteral` canonical mapping is the exclusive XML canonicalization method (*with comments, with empty InclusiveNamespaces PrefixList*) [XML-EXC-C14N].

> NOTE
>
> Any XML namespace declarations (`xmlns`), language annotation (`xml:lang`) or base URI declarations (`xml:base`) desired in the XML content must be included explicitly in the XML literal. Note that some concrete RDF syntaxes may define mechanisms for inheriting them from the context (e.g., `@parseType="literal"` in RDF/XML [RDF11-XML]).

5.4 Datatype IRIs

Datatypes are identified by IRIs. If *D* is a set of IRIs which are used to refer to datatypes, then the elements of *D* are called ***recognized datatype IRIs***. Recognized IRIs have fixed referents. If any IRI of the form `http://www.w3.org/2001/XMLSchema#xxx` is recognized, it MUST refer to the RDF-compatible XSD type named `xsd:xxx` for every XSD type listed in section 5.1. Furthermore, the following IRIs are allocated for non-normative datatypes:

- The IRI `http://www.w3.org/1999/02/22-rdf-syntax-ns#XMLLiteral` refers to the datatype `rdf:XMLLiteral`
- The IRI `http://www.w3.org/1999/02/22-rdf-syntax-ns#HTML` refers to the datatype `rdf:HTML`

> NOTE
>
> Semantic extensions of RDF might choose to recognize other datatype IRIs and require them to refer to a fixed datatype. See the RDF Semantics specification [RDF11-MT] for more information on semantic extensions.

RDF processors are not required to recognize datatype IRIs. Any literal typed with an unrecognized IRI is treated just like an unknown IRI, i.e. as referring to an unknown thing. Applications MAY give a warning message if they are unable to determine the referent of an IRI used in a typed literal, but they SHOULD NOT reject such RDF as either a syntactic or semantic error.

Other specifications MAY impose additional constraints on datatype IRIs, for example, require support for certain datatypes.

> NOTE

> The Web Ontology Language [OWL2-OVERVIEW] offers facilities for formally defining custom datatypes that can be used with RDF. Furthermore, a practice for identifying user-defined simple XML Schema datatypes is suggested in [SWBP-XSCH-DATATYPES]. RDF implementations are not required to support either of these facilities.

6. Fragment Identifiers

This section is non-normative.

RDF uses IRIs, which may include ***fragment identifiers***, as resource identifiers. The semantics of fragment identifiers is defined in RFC 3986 [RFC3986]: They identify a secondary resource that is usually a part of, view of, defined in, or described in the primary resource, and the precise semantics depend on the set of representations that might result from a retrieval action on the primary resource.

This section discusses the handling of fragment identifiers in representations that encode RDF graphs.

In RDF-bearing representations of a primary resource `<foo>`, the secondary resource identified by a fragment `bar` is the resource denoted by the full IRI `<foo#bar>` in the RDF graph. Since IRIs in RDF graphs can denote anything, this can be something external to the representation, or even external to the web.

In this way, the RDF-bearing representation acts as an intermediary between the web-accessible primary resource, and some set of possibly non-web or abstract entities that the RDF graph may describe.

In cases where other specifications constrain the semantics of fragment identifiers in RDF-bearing representations, the encoded RDF graph should use fragment identifiers in a way that is consistent with these constraints. For example, in an HTML+RDFa document [HTML-RDFA], the fragment `chapter1` may identify a document section via the semantics of HTML's `@name` or `@id` attributes. The IRI `<#chapter1>` should then be taken to denote that same section in any RDFa-encoded triples within the same document. Similarly, fragment identifiers should be used consistently in resources with multiple representations that are made available via content negotiation [WEBARCH]. For example, if the fragment `chapter1` identifies a document section in an HTML representation of the primary resource, then the IRI `<#chapter1>` should be taken to denote that same section in all RDF-bearing representations of the same primary resource.

7. Generalized RDF Triples, Graphs, and Datasets

This section is non-normative.

It is sometimes convenient to loosen the requirements on RDF triples. For example, the completeness of the RDFS entailment rules is easier to show with a generalization of RDF triples.

A ***generalized RDF triple*** is a triple having a subject, a predicate, and object, where

each can be an IRI, a blank node or a literal. A ***generalized RDF graph*** is a set of generalized RDF triples. A ***generalized RDF dataset*** comprises a distinguished generalized RDF graph, and zero or more pairs each associating an IRI, a blank node or a literal to a generalized RDF graph.

Generalized RDF triples, graphs, and datasets differ from normative RDF triples, graphs, and datasets only by allowing IRIs, blank nodes and literals to appear in any position, i.e., as subject, predicate, object or graph names.

> NOTE
>
> Any users of generalized RDF triples, graphs or datasets need to be aware that these notions are non-standard extensions of RDF and their use may cause interoperability problems. There is no requirement on the part of any RDF tool to accept, process, or produce anything beyond standard RDF triples, graphs, and datasets.

8. Acknowledgments

This section is non-normative.

The editors acknowledge valuable contributions from Thomas Baker, Tim Berners-Lee, David Booth, Dan Brickley, Gavin Carothers, Jeremy Carroll, Pierre-Antoine Champin, Dan Connolly, John Cowan, Martin J. Dürst, Alex Hall, Steve Harris, Sandro Hawke, Pat Hayes, Ivan Herman, Peter F. Patel-Schneider, Addison Phillips, Eric Prud'hommeaux, Nathan Rixham, Andy Seaborne, Leif Halvard Silli, Guus Schreiber, Dominik Tomaszuk, and Antoine Zimmermann.

The membership of the RDF Working Group included Thomas Baker, Scott Bauer, Dan Brickley, Gavin Carothers, Pierre-Antoine Champin, Olivier Corby, Richard Cyganiak, Souripriya Das, Ian Davis, Lee Feigenbaum, Fabien Gandon, Charles Greer, Alex Hall, Steve Harris, Sandro Hawke, Pat Hayes, Ivan Herman, Nicholas Humfrey, Kingsley Idehen, Gregg Kellogg, Markus Lanthaler, Arnaud Le Hors, Peter F. Patel-Schneider, Eric Prud'hommeaux, Yves Raimond, Nathan Rixham, Guus Schreiber, Andy Seaborne, Manu Sporny, Thomas Steiner, Ted Thibodeau, Mischa Tuffield, William Waites, Jan Wielemaker, David Wood, Zhe Wu, and Antoine Zimmermann.

A. Changes between RDF 1.0 and RDF 1.1

This section is non-normative.

A detailed overview of the differences between RDF versions 1.0 and 1.1 can be found in What's New in RDF 1.1 [RDF11-NEW].

B. References

B.1 Normative references

[BCP47]
A. Phillips; M. Davis. *Tags for Identifying Languages*. September 2009. IETF Best Current Practice. URL: http://tools.ietf.org/html/bcp47

[NFC]
M. Davis, Ken Whistler. *TR15, Unicode Normalization Forms.*. 17 September 2010, URL: http://www.unicode.org/reports/tr15/

[RFC2119]
S. Bradner. *Key words for use in RFCs to Indicate Requirement Levels.* March 1997. Internet RFC 2119. URL: http://www.ietf.org/rfc/rfc2119.txt

[RFC3987]
M. Dürst; M. Suignard. *Internationalized Resource Identifiers (IRIs)*. January 2005. RFC. URL: http://www.ietf.org/rfc/rfc3987.txt

[UNICODE]
The Unicode Standard. URL: http://www.unicode.org/versions/latest/

[XMLSCHEMA11-2]
David Peterson; Sandy Gao; Ashok Malhotra; Michael Sperberg-McQueen; Henry Thompson; Paul V. Biron et al. *W3C XML Schema Definition Language (XSD) 1.1 Part 2: Datatypes*. 5 April 2012. W3C Recommendation. URL: http://www.w3.org/TR/xmlschema11-2/

B.2 Informative references

[COOLURIS]
Leo Sauermann; Richard Cyganiak. *Cool URIs for the Semantic Web*. 3 December 2008. W3C Note. URL: http://www.w3.org/TR/cooluris

[DOM4]
Anne van Kesteren; Aryeh Gregor; Ms2ger; Alex Russell; Robin Berjon. *W3C DOM4*. 4 February 2014. W3C Last Call Working Draft. URL: http://www.w3.org/TR/dom/

[HTML-RDFA]
Manu Sporny. *HTML+RDFa 1.1*. 22 August 2013. W3C Recommendation. URL: http://www.w3.org/TR/html-rdfa/

[HTML5]
Robin Berjon; Steve Faulkner; Travis Leithead; Erika Doyle Navara; Edward O'Connor; Silvia Pfeiffer. *HTML5*. 4 February 2014. W3C Candidate Recommendation. URL: http://www.w3.org/TR/html5/

[JSON-LD]
Manu Sporny, Gregg Kellogg, Markus Lanthaler, Editors. *JSON-LD 1.0*. 16 January 2014. W3C Recommendation. URL: http://www.w3.org/TR/json-ld/

[LINKED-DATA]
Tim Berners-Lee. *Linked Data*. Personal View, imperfect but published. URL: http://www.w3.org/DesignIssues/LinkedData.html

[OWL2-OVERVIEW]
W3C OWL Working Group. *OWL 2 Web Ontology Language Document Overview (Second Edition)*. 11 December 2012. W3C Recommendation. URL: http://www.w3.org/TR/owl2-overview/

[RDF11-DATASETS]
Antoine Zimmermann. *RDF 1.1: On Semantics of RDF Datasets*. W3C Working Group Note, 25 February 2014. The latest version is available at

http://www.w3.org/TR/rdf11-datasets/.

[RDF11-MT]

Patrick J. Hayes, Peter F. Patel-Schneider. *RDF 1.1 Semantics.* W3C Recommendation, 25 February 2014. URL: http://www.w3.org/TR/2014/REC-rdf11-mt-20140225/. The latest edition is available at http://www.w3.org/TR/rdf11-mt/

[RDF11-NEW]

David Wood. *What's New in RDF 1.1*. W3C Working Group Note, 25 February 2014. The latest version is available at http://www.w3.org/TR/rdf11-new/.

[RDF11-PRIMER]

Guus Schreiber, Yves Raimond. *RDF 1.1 Primer*. W3C Working Group Note, 25 February 2014. The latest version is available at http://www.w3.org/TR/rdf11-primer/.

[RDF11-SCHEMA]

Dan Brickley, R. V. Guha. *RDF Schema 1.1*. W3C Recommendation, 25 February 2014. URL: http://www.w3.org/TR/2014/REC-rdf-schema-20140225/. The latest published version is available at http://www.w3.org/TR/rdf-schema/.

[RDF11-TESTCASES]

Gregg Kellogg, Markus Lanthaler. *RDF 1.1 Test Cases*. W3C Working Group Note, 25 February 2014. The latest published version is available at http://www.w3.org/TR/rdf11-testcases/.

[RDF11-XML]

Fabien Gandon, Guus Schreiber. *RDF 1.1 XML Syntax*. W3C Recommendation, 25 February 2014. URL: http://www.w3.org/TR/2014/REC-rdf-syntax-grammar-20140225/. The latest published version is available at http://www.w3.org/TR/rdf-syntax-grammar/.

[RDFA-PRIMER]

Ivan Herman; Ben Adida; Manu Sporny; Mark Birbeck. *RDFa 1.1 Primer - Second Edition*. 22 August 2013. W3C Note. URL: http://www.w3.org/TR/rdfa-primer/

[RFC3986]

T. Berners-Lee; R. Fielding; L. Masinter. *Uniform Resource Identifier (URI): Generic Syntax (RFC 3986)*. January 2005. RFC. URL: http://www.ietf.org/rfc/rfc3986.txt

[RFC5785]

Mark Nottingham; Eran Hammer-Lahav. *Defining Well-Known Uniform Resource Identifiers (URIs) (RFC 5785)*. April 2010. RFC. URL: http://www.rfc-editor.org/rfc/rfc5785.txt

[SPARQL11-OVERVIEW]

The W3C SPARQL Working Group. *SPARQL 1.1 Overview*. 21 March 2013. W3C Recommendation. URL: http://www.w3.org/TR/sparql11-overview/

[SWBP-N-ARYRELATIONS]

Natasha Noy; Alan Rector. *Defining N-ary Relations on the Semantic Web*. 12 April 2006. W3C Note. URL: http://www.w3.org/TR/swbp-n-aryRelations

[SWBP-XSCH-DATATYPES]

Jeremy Carroll; Jeff Pan. *XML Schema Datatypes in RDF and OWL*. 14 March 2006. W3C Note. URL: http://www.w3.org/TR/swbp-xsch-datatypes

[TRIG]

Gavin Carothers, Andy Seaborne. *TriG: RDF Dataset Language*. W3C Recommendation, 25 February 2014. URL: http://www.w3.org/TR/2014/REC-

trig-20140225/. The latest edition is available at http://www.w3.org/TR/trig/

[TURTLE]

Eric Prud'hommeaux, Gavin Carothers. *RDF 1.1 Turtle: Terse RDF Triple Language.* W3C Recommendation, 25 February 2014. URL: http://www.w3.org/TR/2014/REC-turtle-20140225/. The latest edition is available at http://www.w3.org/TR/turtle/

[VOCAB-ORG]

Dave Reynolds. *The Organization Ontology*. 16 January 2014. W3C Recommendation. URL: http://www.w3.org/TR/vocab-org/

[WEBARCH]

Ian Jacobs; Norman Walsh. *Architecture of the World Wide Web, Volume One*. 15 December 2004. W3C Recommendation. URL: http://www.w3.org/TR/webarch/

[XML-EXC-C14N]

John Boyer; Donald Eastlake; Joseph Reagle. *Exclusive XML Canonicalization Version 1.0*. 18 July 2002. W3C Recommendation. URL: http://www.w3.org/TR/xml-exc-c14n

[XML-NAMES]

Tim Bray; Dave Hollander; Andrew Layman; Richard Tobin; Henry Thompson et al. *Namespaces in XML 1.0 (Third Edition)*. 8 December 2009. W3C Recommendation. URL: http://www.w3.org/TR/xml-names

[XML10]

Tim Bray; Jean Paoli; Michael Sperberg-McQueen; Eve Maler; François Yergeau et al. *Extensible Markup Language (XML) 1.0 (Fifth Edition)*. 26 November 2008. W3C Recommendation. URL: http://www.w3.org/TR/xml