W3C Recommendation

Resource Description Framework (RDF): Concepts and Abstract Syntax

W3C Recommendation 10 February 2004

New Version Available: "RDF 1.1 Concepts and Abstract Syntax" (Document Status Update, 25 February 2014)

The RDF Working Group has produced a W3C Recommendation for a new version of RDF which adds features to this 2004 version, while remaining compatible. Please see "RDF 1.1 Concepts and Abstract Syntax" for a new version of this document, and the "What's New in RDF 1.1" document for the differences between this version of RDF and RDF 1.1.

This version:
http://www.w3.org/TR/2004/REC-rdf-concepts-20040210/
Latest version:
http://www.w3.org/TR/rdf-concepts/
Previous version:
http://www.w3.org/TR/2003/PR-rdf-concepts-20031215/
Editors:
Graham Klyne (Nine by Nine), <gk@ninebynine.org>
Jeremy J. Carroll (Hewlett Packard Labs), <jjc@hpl.hp.com>
Series editor:
Brian McBride (Hewlett Packard Labs) <bwm@hplb.hpl.hp.com>

Please refer to the **errata** for this document, which may include some normative corrections.

See also translations.

Abstract

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

RDF Concepts and Abstract Syntax defines an abstract syntax on which RDF is based, and which serves to link its concrete syntax to its formal semantics. It also

includes discussion of design goals, key concepts, datatyping, character normalization and handling of URI references.

Status of this Document

This document has been reviewed by W3C Members and other interested parties, and it has been endorsed by the Director as a W3C Recommendation. 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 is one document in a set of six (Primer, Concepts, Syntax, Semantics, Vocabulary, and Test Cases) intended to jointly replace the original Resource Description Framework specifications, RDF Model and Syntax (1999 Recommendation) and RDF Schema (2000 Candidate Recommendation). It has been developed by the RDF Core Working Group as part of the W3C Semantic Web Activity (Activity Statement, Group Charter) for publication on 10 February 2004.

Changes to this document since the Proposed Recommendation Working Draft are detailed in the change log.

The public is invited to send comments to www-rdf-comments@w3.org (archive) and to participate in general discussion of related technology on www-rdf-interest@w3.org (archive).

A list of implementations is available.

The W3C maintains a list of any patent disclosures related to this work.

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/.

Table of Contents

---

1. Introduction

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

This document defines an abstract syntax on which RDF is based, and which serves to link its concrete syntax to its formal semantics. This abstract syntax is quite distinct from XML's tree-based infoset [XML-INFOSET]. It also includes discussion of design goals, key concepts, datatyping, character normalization and handling of URI references.

Normative documentation of RDF falls into the following areas:

- XML serialization syntax [RDF-SYNTAX],
- formal semantics [RDF-SEMANTICS], and
- this document, (sections 4, 5 and 6).

Within this document, normative sections are explicitly labelled as such. Explicit notes are informative.

The framework is designed so that vocabularies can be layered. The RDF and RDF vocabulary definition (RDF schema) languages [RDF-VOCABULARY] are the first such vocabularies. Others (cf. OWL [OWL] and the applications mentioned in the primer [RDF-PRIMER]) are in development.

1.1 Structure of this Document

In section 2, the background rationale and design goals are introduced. Key concepts follow in section 3. Section 4 discusses URI references reserved for use by RDF.

Section 5 discusses datatypes. XML content of literals is described in section 5.1, and the abstract syntax is defined in section 6 of this document.

Section 7 discusses the role of fragment identifiers in URI references used with RDF.

2. Motivations and Goals

RDF has an abstract syntax that reflects a simple graph-based data model, and formal semantics with a rigorously defined notion of entailment providing a basis for well founded deductions in RDF data.

2.1 Motivation

The development of RDF has been motivated by the following uses, among others:

- Web metadata: providing information about Web resources and the systems that use them (e.g. content rating, capability descriptions, privacy preferences, etc.)
- Applications that require open rather than constrained information models (e.g. scheduling activities, describing organizational processes, annotation of Web resources, etc.)
- To do for machine processable information (application data) what the World Wide Web has done for hypertext: to allow data to be processed outside the particular environment in which it was created, in a fashion that can work at Internet scale.
- Interworking among applications: combining data from several applications to arrive at new information.
- Automated processing of Web information by software agents: the Web is moving from having just human-readable information to being a world-wide network of cooperating processes. RDF provides a world-wide lingua franca for these processes.

RDF is designed to represent information in a minimally constraining, flexible way. It can be used in isolated applications, where individually designed formats might be more direct and easily understood, but RDF's generality offers greater value from sharing. The value of information thus increases as it becomes accessible to more applications across the entire Internet.

2.2 Design Goals

The design of RDF is intended to meet the following goals:

- having a simple data model
- having formal semantics and provable inference
- using an extensible URI-based vocabulary
- using an XML-based syntax
- supporting use of XML schema datatypes
- allowing anyone to make statements about any resource

2.2.1 A Simple Data Model

RDF has a simple data model that is easy for applications to process and manipulate. The data model is independent of any specific serialization syntax.

> **Note:** the term "model" used here in "data model" has a completely different sense to its use in the term "model theory". See [RDF-SEMANTICS] for more information about "model theory" as used in the literature of mathematics and logic.

2.2.2 Formal Semantics and Inference

RDF has a formal semantics which provides a dependable basis for reasoning about the meaning of an RDF expression. In particular, it supports rigorously defined notions of entailment which provide a basis for defining reliable rules of inference in RDF data.

2.2.3 Extensible URI-based Vocabulary

The vocabulary is fully extensible, being based on URIs with optional fragment identifiers (URI references, or URIrefs). URI references are used for naming all kinds of things in RDF.

The other kind of value that appears in RDF data is a literal.

2.2.4 XML-based Syntax

RDF has a recommended XML serialization form [RDF-SYNTAX], which can be used to encode the data model for exchange of information among applications.

2.2.5 Use XML Schema Datatypes

RDF can use values represented according to XML schema datatypes [XML-SCHEMA2], thus assisting the exchange of information between RDF and other XML applications.

2.2.6 Anyone Can Make Statements About Any Resource

To facilitate operation at Internet scale, RDF is an open-world framework that allows anyone to make statements about any resource.

In general, it is not assumed that complete information about any resource is available. RDF does not prevent anyone from making assertions that are nonsensical or inconsistent with other statements, or the world as people see it. Designers of applications that use RDF should be aware of this and may design their applications to tolerate incomplete or inconsistent sources of information.

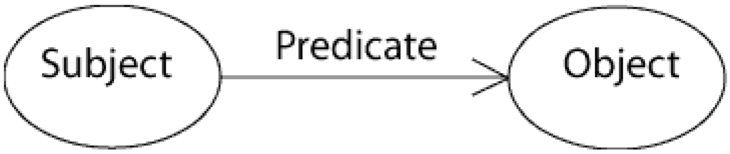
3. RDF Concepts

RDF uses the following key concepts:

- Graph data model
- URI-based vocabulary
- Datatypes
- Literals
- XML serialization syntax
- Expression of simple facts
- Entailment

3.1 Graph Data Model

The underlying structure of any expression in RDF is a collection of triples, each consisting of a subject, a predicate and an object. A set of such triples is called an RDF graph (defined more formally in section 6). This can be illustrated by a node and directed-arc diagram, in which each triple is represented as a node-arc-node link (hence the term "graph").



Each triple represents a statement of a relationship between the things denoted by the nodes that it links. Each triple has three parts:

1. a subject,
2. an object, and
3. a predicate (also called a property) that denotes a relationship.

The direction of the arc is significant: it always points toward the object.

The nodes of an RDF graph are its subjects and objects.

The assertion of an RDF triple says that some relationship, indicated by the predicate, holds between the things denoted by subject and object of the triple. The assertion of an RDF graph amounts to asserting all the triples in it, so the meaning of an RDF graph is the conjunction (logical AND) of the statements corresponding to all the triples it contains. A formal account of the meaning of RDF graphs is given in [RDF-SEMANTICS].

3.2 URI-based Vocabulary and Node Identification

A node may be a URI with optional fragment identifier (URI reference, or **URIref**), a literal, or blank (having no separate form of identification). Properties are URI references. (See [URI], section 4, for a description of URI reference forms, noting that relative URIs are not used in an RDF graph. See also section 6.4.)

A URI reference or literal used as a node identifies what that node represents. A URI reference used as a predicate identifies a relationship between the things represented by the nodes it connects. A predicate URI reference may also be a node in the graph.

A blank node is a node that is not a URI reference or a literal. In the RDF abstract syntax, a blank node is just a unique node that can be used in one or more RDF statements, but has no intrinsic name.

A convention used by some linear representations of an RDF graph to allow several statements to reference the same unidentified resource is to use a **blank node identifier**, which is a local identifier that can be distinguished from all URIs and literals. When graphs are merged, their blank nodes must be kept distinct if meaning is to be preserved; this may call for re-allocation of blank node identifiers. Note that such blank node identifiers are not part of the RDF abstract syntax, and the representation of triples containing blank nodes is entirely dependent on the particular concrete syntax used.

3.3 Datatypes

Datatypes are used by RDF in the representation of values such as integers, floating point numbers and dates.

A datatype consists of a lexical space, a value space and a lexical-to-value mapping, see section 5.

For example, the lexical-to-value mapping for the XML Schema datatype xsd:boolean, where each member of the value space (represented here as 'T' and 'F') has two lexical representations, is as follows:

Value Space	{T, F}
Lexical Space	{"0", "1", "true", "false"}
Lexical-to-Value Mapping	{<"true", T>, <"1", T>, <"0", F>, <"false", F>}

RDF predefines just one datatype `rdf:XMLLiteral`, used for embedding XML in RDF (see section 5.1).

There is no built-in concept of numbers or dates or other common values. Rather, RDF defers to datatypes that are defined separately, and identified with URI references. The predefined XML Schema datatypes [XML-SCHEMA2] are expected to be widely used for this purpose.

RDF provides no mechanism for defining new datatypes. XML Schema Datatypes [XML-SCHEMA2] provides an extensibility framework suitable for defining new datatypes for use in RDF.

3.4 Literals

Literals are used to identify values such as numbers and dates by means of a lexical representation. Anything represented by a literal could also be represented by a URI, but it is often more convenient or intuitive to use literals.

A literal may be the object of an RDF statement, but not the subject or the predicate.

Literals may be plain or typed :

- A plain literal is a string combined with an optional language tag. This may be used for plain text in a natural language. As recommended in the RDF formal semantics [RDF-SEMANTICS], these plain literals are self-denoting.
- A typed literal is a string combined with a datatype URI. It denotes the member of the identified datatype's value space obtained by applying the lexical-to-value mapping to the literal string.

Continuing the example from section 3.3, the typed literals that can be defined using the XML Schema datatype xsd:boolean are:

Typed Literal	Lexical-to-Value Mapping	Value
<xsd:boolean, "true">	<"true", T>	T
<xsd:boolean, "1">	<"1", T>	T
<xsd:boolean, "false">	<"false", F>	F
<xsd:boolean, "0">	<"0", F>	F

For text that may contain markup, use typed literals with type rdf:XMLLiteral. If language annotation is required, it must be explicitly included as markup, usually by means of an `xml:lang` attribute. [XHTML] may be included within RDF in this way. Sometimes, in this latter case, an additional `span` or `div` element is needed to carry an `xml:lang` or `lang` attribute.

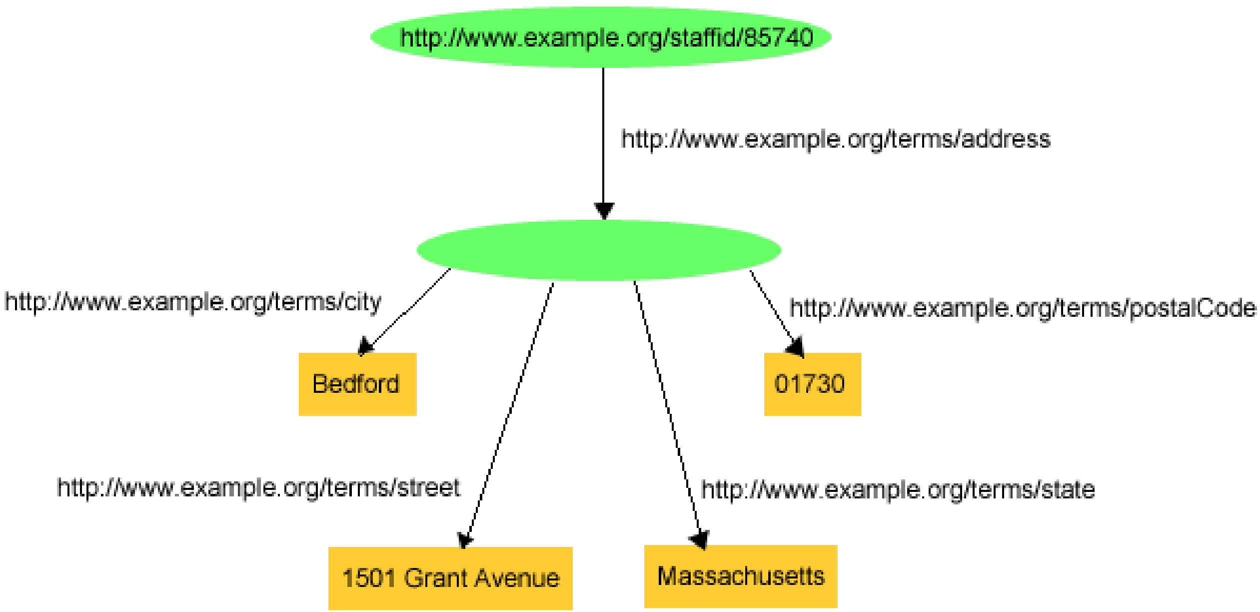
The string in both plain and typed literals is recommended to be in Unicode Normal Form C [NFC]. This is motivated by [CHARMOD] particularly section 4 Early Uniform Normalization.

3.5 RDF Expression of Simple Facts

Some simple facts indicate a relationship between two things. Such a fact may be represented as an RDF triple in which the predicate names the relationship, and the subject and object denote the two things. A familiar representation of such a fact might be as a row in a table in a relational database. The table has two columns, corresponding to the subject and the object of the RDF triple. The name of the table corresponds to the predicate of the RDF triple. A further familiar representation may be as a two place predicate in first order logic.

Relational databases permit a table to have an arbitrary number of columns, a row of which expresses information corresponding to a predicate in first order logic with an arbitrary number of places. Such a row, or predicate, has to be decomposed for representation as RDF triples. A simple form of decomposition introduces a new blank node, corresponding to the row, and a new triple is introduced for each cell in the row. The subject of each triple is the new blank node, the predicate corresponds to the column name, and object corresponds to the value in the cell. The new blank node may also have an `rdf:type` property whose value corresponds to the table name.

As an example, consider Figure 6 from the [RDF-PRIMER]:



RDF Primer Figure 6: Using a Blank Node

This information might correspond to a row in a table `"STAFFADDRESSES"`, with a primary key `STAFFID`, and additional columns `STREET`, `STATE`, `CITY` and `POSTALCODE`.

Thus, a more complex fact is expressed in RDF using a conjunction (logical-AND) of simple binary relationships. RDF does not provide means to express negation (NOT) or disjunction (OR).

Through its use of extensible URI-based vocabularies, RDF provides for expression of facts about arbitrary subjects; i.e. assertions of named properties about specific named things. A URI can be constructed for any thing that can be named, so RDF facts can be about any such things.

3.6 Entailment

The ideas on meaning and inference in RDF are underpinned by the formal concept of entailment, as discussed in the RDF semantics document [RDF-SEMANTICS]. In brief, an RDF expression A is said to **entail** another RDF expression B if every possible arrangement of things in the world that makes A true also makes B true. On this basis, if the truth of A is presumed or demonstrated then the truth of B can be inferred .

4. RDF Vocabulary URI and Namespace (Normative)

RDF uses URI references to identify resources and properties. Certain URI references are given specific meaning by RDF. Specifically, URI references with the following leading substring are defined by the RDF specifications:

- `http://www.w3.org/1999/02/22-rdf-syntax-ns#` (conventionally associated with namespace prefix `rdf:`)

Used with the RDF/XML serialization, this URI prefix string corresponds to XML namespace names [XML-NS] associated with the RDF vocabulary terms.

> **Note:** this namespace name is the same as that used in the earlier RDF recommendation [RDF-MS].

Vocabulary terms in the `rdf:` namespace are listed in section 5.1 of the RDF syntax specification [RDF-SYNTAX]. Some of these terms are defined by the RDF specifications to denote specific concepts. Others have syntactic purpose (e.g. rdf:ID is part of the RDF/XML syntax).

5. Datatypes (Normative)

The datatype abstraction used in RDF is compatible with the abstraction used in XML Schema Part 2: Datatypes [XML-SCHEMA2].

A datatype consists of a lexical space, a value space and a lexical-to-value mapping.

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** of the datatype, and the second element belongs to the **value space** of the datatype:

- Each member of the lexical space is paired with (maps to) exactly one member of the value space.
- Each member of the value space may be paired with any number (including zero) of members of the lexical space (lexical representations for that value).

A datatype is identified by one or more URI references.

RDF may be used with any datatype definition that conforms to this abstraction, even if not defined in terms of XML Schema.

Certain XML Schema built-in datatypes are not suitable for use within RDF. For example, the QName datatype requires a namespace declaration to be in scope during the mapping, and is not recommended for use in RDF. [RDF-SEMANTICS] contains a more detailed discussion of specific XML Schema built-in datatypes.

> **Note:** When the datatype is defined using XML Schema:

- All values correspond to some lexical form, either using the lexical-to-value mapping of the datatype or if it is a union datatype with a lexical mapping associated with one of the member datatypes.
- XML Schema facets remain part of the datatype and are used by the XML Schema mechanisms that control the lexical space and the value space; however, RDF does not define a standard mechanism to access these facets.
- In [XML-SCHEMA1], white space normalization occurs during validation according to the value of the whiteSpace facet. The lexical-to-value mapping used in RDF datatyping occurs after this, so that the whiteSpace facet has no effect in RDF datatyping.

5.1 XML Content within an RDF Graph

RDF provides for XML content as a possible literal value. This typically originates from the use of `rdf:parseType="Literal"` in the RDF/XML Syntax [RDF-SYNTAX].

Such content is indicated in an RDF graph using a typed literal whose datatype is a special built-in datatype `rdf:XMLLiteral`, defined as follows.

A URI reference for identifying 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 [XML];
  - for which encoding as UTF-8 [RFC 2279] yields exclusive Canonical XML (with comments, with empty InclusiveNamespaces PrefixList) [XML-XC14N];
  - for which embedding between an arbitrary XML start tag and an end tag yields a document conforming to XML Namespaces [XML-NS]

The value space
: is a set of entities, called XML values, which is:

  - disjoint from the lexical space;
  - disjoint from the value space of any XML schema datatype [XML-SCHEMA2];
  - disjoint from the set of Unicode character strings [UNICODE] strings;
  - and in 1:1 correspondence with the lexical space.

The lexical-to-value mapping
: is a one-one mapping from the lexical space onto the value space, i.e. it is both injective and surjective.

  Note: Not all values of this datatype are compliant with XML 1.1 [XML 1.1]. If compliance with XML 1.1 is desired, then only those values that are fully normalized according to XML 1.1 should be used.

  Note: XML values can be thought of as the [XML-INFOSET] or the [XPATH] nodeset corresponding to the lexical form, with an appropriate equality function.

> **Note:** RDF applications may use additional equivalence relations, such as that which relates an `xsd:string` with an `rdf:XMLLiteral` corresponding to a single text node of the same string.

6. Abstract Syntax (Normative)

This section defines the RDF abstract syntax. The RDF abstract syntax is a set of triples, called the RDF graph.

This section also defines equivalence between RDF graphs. A definition of equivalence is needed to support the RDF Test Cases [RDF-TESTS] specification.

> **Implementation Note:** This abstract syntax is the syntax over which the formal semantics are defined. Implementations are free to represent RDF graphs in any other equivalent form. As an example: in an RDF graph, literals with datatype `rdf:XMLLiteral` can be represented in a non-canonical format, and canonicalization performed during the comparison between two such literals. In this example the comparisons may be being performed either between syntactic structures or between their denotations in the domain of discourse. Implementations that do not require any such comparisons can hence be optimized.

6.1 RDF Triples

An **RDF triple** contains three components:

- the **subject**, which is an RDF URI reference or a blank node
- the **predicate**, which is an RDF URI reference
- the **object**, which is an RDF URI reference, a literal or a blank node

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

The predicate is also known as the **property** of the triple.

6.2 RDF Graph

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

The set of **nodes** of an RDF graph is the set of subjects and objects of triples in the graph.

6.3 Graph Equivalence

Two RDF graphs G and G' are equivalent 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(uri)=uri for all RDF URI references uri 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'

With this definition, M shows how each blank node in G can be replaced with a new blank node to give G'.

6.4 RDF URI References

A **URI reference** within an RDF graph (an RDF URI reference) is a Unicode string [UNICODE] that:

- does not contain any control characters (#x00 - #x1F, #x7F-#x9F)
- and would produce a valid URI character sequence (per RFC2396 [URI], sections 2.1) representing an absolute URI with optional fragment identifier when subjected to the encoding described below.

The encoding consists of:

1. encoding the Unicode string as UTF-8 [RFC-2279], giving a sequence of octet values.
2. %-escaping octets that do not correspond to permitted US-ASCII characters.

The disallowed octets that must be %-escaped include all those that do not correspond to US-ASCII characters, and the excluded characters listed in Section 2.4 of [URI], except for the number sign (#), percent sign (%), and the square bracket characters re-allowed in [RFC-2732].

Disallowed octets must be escaped with the URI escaping mechanism (that is, converted to %HH, where HH is the 2-digit hexadecimal numeral corresponding to the octet value).

Two RDF URI references are equal if and only if they compare as equal, character by character, as Unicode strings.

> **Note:** RDF URI references are compatible with the anyURI datatype as defined by XML schema datatypes [XML-SCHEMA2], constrained to be an absolute rather than a relative URI reference.

> **Note:** RDF URI references are compatible with International Resource Identifiers as defined by [XML Namespaces 1.1].

> **Note:** this section anticipates an RFC on Internationalized Resource Identifiers. Implementations may issue warnings concerning the use of RDF URI References that do not conform with [IRI draft] or its successors.

> **Note:** The restriction to absolute URI references is found in this abstract syntax. When there is a well-defined base URI, concrete syntaxes, such as RDF/XML, may permit relative URIs as a shorthand for such absolute URI references.

> **Note:** Because of the risk of confusion between RDF URI references that would be equivalent if derefenced, the use of %-escaped characters in RDF

URI references is strongly discouraged. See also the URI equivalence issue of the Technical Architecture Group [TAG].

6.5 RDF Literals

A **literal** in an RDF graph contains one or two named components.

All literals have a **lexical form** being a Unicode [UNICODE] string, which SHOULD be in Normal Form C [NFC].

Plain literals have a lexical form and optionally a **language tag** as defined by [RFC-3066], normalized to lowercase.

Typed literals have a lexical form and a **datatype URI** being an RDF URI reference.

> **Note:** Literals in which the lexical form begins with a composing character (as defined by [CHARMOD]) are allowed however they may cause interoperability problems, particularly with XML version 1.1 [XML 1.1].

> **Note:** When using the language tag, care must be taken not to confuse language with locale. The language tag relates only to human language text. Presentational issues should be addressed in end-user applications.

> **Note:** The case normalization of language tags is part of the description of the abstract syntax, and consequently the abstract behaviour of RDF applications. It does not constrain an RDF implementation to actually normalize the case. Crucially, the result of comparing two language tags should not be sensitive to the case of the original input.

6.5.1 Literal Equality

Two literals are equal if and only if all of the following hold:

- The strings of the two lexical forms compare equal, character by character.
- Either both or neither have language tags.
- The language tags, if any, compare equal.
- Either both or neither have datatype URIs.
- The two datatype URIs, if any, compare equal, character by character.

> **Note:** RDF Literals are distinct and distinguishable from RDF URI references; e.g. http://example.org as an RDF Literal (untyped, without a language tag) is not equal to http://example.org as an RDF URI reference.

6.5.2 The Value Corresponding to a Typed Literal

The datatype URI refers to a datatype. For XML Schema built-in datatypes, URIs such as `http://www.w3.org/2001/XMLSchema#int` are used. The URI of the datatype `rdf:XMLLiteral` may be used. There may be other, implementation dependent, mechanisms by which URIs refer to datatypes.

The value associated with a typed literal is found by applying the lexical-to-value mapping associated with the datatype URI to the lexical form.

If the lexical form is not in the lexical space of the datatype associated with the datatype URI, then no literal value can be associated with the typed literal. Such a case, while in error, is not syntactically ill-formed.

> **Note:** In application contexts, comparing the values of typed literals (see section 6.5.2) is usually more helpful than comparing their syntactic forms (see section 6.5.1). Similarly, for comparing RDF Graphs, semantic notions of entailment (see [RDF-SEMANTICS]) are usually more helpful than syntactic equality (see section 6.3).

6.6 Blank Nodes

The **blank nodes** in an RDF graph are drawn from an infinite set. This set of blank nodes, the set of all RDF URI references and the set of all literals are pairwise disjoint.

Otherwise, this set of blank nodes is arbitrary.

RDF makes no reference to any internal structure of blank nodes. Given two blank nodes, it is possible to determine whether or not they are the same.

7. Fragment Identifiers

RDF uses an RDF URI Reference, which may include a fragment identifier, as a context free identifier for a resource. RFC 2396 [URI] states that the meaning of a fragment identifier depends on the MIME content-type of a document, i.e. is context dependent.

These apparently conflicting views are reconciled by considering that a URI reference in an RDF graph is treated with respect to the MIME type `application/rdf+xml` [RDF-MIME-TYPE]. Given an RDF URI reference consisting of an absolute URI and a fragment identifier, the fragment identifer identifies the same thing that it does in an `application/rdf+xml` representation of the resource identified by the absolute URI component. Thus:

- we assume that the URI part (i.e. excluding fragment identifier) identifies a resource, which is presumed to have an RDF representation. So when `eg:someurl#frag` is used in an RDF document, `eg:someurl` is taken to designate some RDF document (even when no such document can be retrieved).
- `eg:someurl#frag` means the thing that is indicated, according to the rules of the `application/rdf+xml` MIME content-type as a "fragment" or "view" of the RDF document at `eg:someurl`. If the document does not exist, or cannot be retrieved, or is available only in formats other than `application/rdf+xml`, then exactly what that view may be is somewhat undetermined, but that does not prevent use of RDF to say things about it.
- the RDF treatment of a fragment identifier allows it to indicate a thing that is entirely external to the document, or even to the "shared information space"

known as the Web. That is, it can be a more general idea, like some particular car or a mythical Unicorn.
- in this way, an `application/rdf+xml` document acts as an intermediary between some Web retrievable documents (itself, at least, also any other Web retrievable URIs that it may use, possibly including schema URIs and references to other RDF documents), and some set of possibly abstract or non-Web entities that the RDF may describe.

This provides a handling of URI references and their denotation that is consistent with the RDF model theory and usage, and also with conventional Web behavior. Note that nothing here requires that an RDF application be able to retrieve any representation of resources identified by the URIs in an RDF graph.

8. Acknowledgments

This document contains a significant contribution from Pat Hayes, Sergey Melnik and Patrick Stickler, under whose leadership was developed the framework described in the RDF family of specifications for representing datatyped values, such as integers and dates.

The editors acknowledge valuable contributions from the following: Frank Manola, Pat Hayes, Dan Brickley, Jos de Roo, Dave Beckett, Patrick Stickler, Peter F. Patel-Schneider, Jerome Euzenat, Massimo Marchiori, Tim Berners-Lee, Dave Reynolds and Dan Connolly.

Jeremy Carroll thanks Oreste Signore, his host at the W3C Office in Italy and Istituto di Scienza e Tecnologie dell'Informazione "Alessandro Faedo", part of the Consiglio Nazionale delle Ricerche, where Jeremy is a visiting researcher.

This document is a product of extended deliberations by the RDFcore Working Group, whose members have included: Art Barstow (W3C), Dave Beckett (ILRT), Dan Brickley (ILRT), Dan Connolly (W3C), Jeremy Carroll (Hewlett Packard), Ron Daniel (Interwoven Inc), Bill dehOra (InterX), Jos De Roo (AGFA), Jan Grant (ILRT), Graham Klyne (Nine by Nine), Frank Manola (MITRE Corporation), Brian McBride (Hewlett Packard), Eric Miller (W3C), Stephen Petschulat (IBM), Patrick Stickler (Nokia), Aaron Swartz (HWG), Mike Dean (BBN Technologies / Verizon), R. V. Guha (Alpiri Inc), Pat Hayes (IHMC), Sergey Melnik (Stanford University) and Martyn Horner (Profium Ltd).

This specification also draws upon an earlier RDF Model and Syntax document edited by Ora Lassilla and Ralph Swick, and RDF Schema edited by Dan Brickley and R. V. Guha. RDF and RDF Schema Working Group members who contributed to this earlier work are: Nick Arnett (Verity), Tim Berners-Lee (W3C), Tim Bray (Textuality), Dan Brickley (ILRT / University of Bristol), Walter Chang (Adobe), Sailesh Chutani (Oracle), Dan Connolly (W3C), Ron Daniel (DATAFUSION), Charles Frankston (Microsoft), Patrick Gannon (CommerceNet), R. V. Guha (Epinions, previously of Netscape Communications), Tom Hill (Apple Computer), Arthur van Hoff (Marimba), Renato Iannella (DSTC), Sandeep Jain (Oracle), Kevin Jones, (InterMind), Emiko Kezuka (Digital Vision Laboratories), Joe Lapp (webMethods Inc.), Ora Lassila (Nokia Research Center), Andrew Layman (Microsoft), Ralph

LeVan (OCLC), John McCarthy (Lawrence Berkeley National Laboratory), Chris McConnell (Microsoft), Murray Maloney (Grif), Michael Mealling (Network Solutions), Norbert Mikula (DataChannel), Eric Miller (OCLC), Jim Miller (W3C, emeritus), Frank Olken (Lawrence Berkeley National Laboratory), Jean Paoli (Microsoft), Sri Raghavan (Digital/Compaq), Lisa Rein (webMethods Inc.), Paul Resnick (University of Michigan), Bill Roberts (KnowledgeCite), i Tsuyoshi Sakata (Digital Vision Laboratories), Bob Schloss (IBM), Leon Shklar (Pencom Web Works), David Singer (IBM), Wei (William) Song (SISU), Neel Sundaresan (IBM), Ralph Swick (W3C), Naohiko Uramoto (IBM), Charles Wicksteed (Reuters Ltd.), Misha Wolf (Reuters Ltd.) and Lauren Wood (SoftQuad).

9. References

9.1 Normative References

[RDF-SEMANTICS]
: RDF Semantics, Patrick Hayes, Editor, W3C Recommendation, 10 February 2004, http://www.w3.org/TR/2004/REC-rdf-mt-20040210/ . Latest version available at http://www.w3.org/TR/rdf-mt/ .

[RDF-SYNTAX]
: RDF/XML Syntax Specification (Revised), Dave Beckett, Editor, W3C Recommendation, 10 February 2004, http://www.w3.org/TR/2004/REC-rdf-syntax-grammar-20040210/ . Latest version available at http://www.w3.org/TR/rdf-syntax-grammar/ .

[RDF-MIME-TYPE]
: MIME Media Types, The Internet Assigned Numbers Authority (IANA). This document is http://www.iana.org/assignments/media-types/ . The registration for `application/rdf+xml` is archived at http://www.w3.org/2001/sw/RDFCore/mediatype-registration .

[XML]
: Extensible Markup Language (XML) 1.0, Second Edition, T. Bray, J. Paoli, C.M. Sperberg-McQueen and E. Maler, Editors. World Wide Web Consortium. 6 October 2000. This version is http://www.w3.org/TR/2000/REC-xml-20001006. The latest version of XML is available at http://www.w3.org/TR/REC-xml.

[XML-NS]
: Namespaces in XML, T. Bray, D. Hollander and A. Layman, Editors. World Wide Web Consortium. 14 January 1999. This version is http://www.w3.org/TR/1999/REC-xml-names-19990114/. The latest version of Namespaces in XML is available at http://www.w3.org/TR/REC-xml-names/.

[RFC-2279]
: RFC 2279 - UTF-8, a transformation format of ISO 10646, F. Yergeau, IETF, January 1998. This document is http://www.isi.edu/in-notes/rfc2279.txt.

[URI]
: RFC 2396 - Uniform Resource Identifiers (URI): Generic Syntax, T. Berners-Lee, R. Fielding and L. Masinter, IETF, August 1998. This document is http://www.isi.edu/in-notes/rfc2396.txt.

[RFC-2732]

RFC 2732 - Format for Literal IPv6 Addresses in URL's, R. Hinden, B. Carpenter and L. Masinter, IETF, December 1999. This document is http://www.isi.edu/in-notes/rfc2732.txt.

[UNICODE]
The Unicode Standard, Version 3, The Unicode Consortium, Addison-Wesley, 2000. ISBN 0-201-61633-5, as updated from time to time by the publication of new versions. (See http://www.unicode.org/unicode/standard/versions/ for the latest version and additional information on versions of the standard and of the Unicode Character Database).

[NFC]
Unicode Normalization Forms, Unicode Standard Annex #15, Mark Davis, Martin Dürst. (See http://www.unicode.org/unicode/reports/tr15/ for the latest version).

[RFC-3066]
RFC 3066 - Tags for the Identification of Languages, H. Alvestrand, IETF, January 2001. This document is http://www.isi.edu/in-notes/rfc3066.txt.

[XML-XC14N]
Exclusive XML Canonicalization Version 1.0, J. Boyer, D.E. Eastlake 3rd, J. Reagle, Authors/Editors. W3C Recommendation. World Wide Web Consortium, 18 July 2002. This version of Exclusive XML Canonicalization is http://www.w3.org/TR/2002/REC-xml-exc-c14n-20020718/. The latest version of Canonical XML is at http://www.w3.org/TR/xml-exc-c14n.

[XML-SCHEMA2]
XML Schema Part 2: Datatypes, W3C Recommendation, World Wide Web Consortium, 2 May 2001.This version is http://www.w3.org/TR/2001/REC-xmlschema-2-20010502/. The latest version is available at http://www.w3.org/TR/xmlschema-2/.

9.2 Informational References

[RDF-TESTS]
RDF Test Cases, Jan Grant and Dave Beckett, Editors, W3C Recommendation, 10 February 2004, http://www.w3.org/TR/2004/REC-rdf-testcases-20040210/ . Latest version available at http://www.w3.org/TR/rdf-testcases/ .

[RDF-VOCABULARY]
RDF Vocabulary Description Language 1.0: RDF Schema, Dan Brickley and R. V. Guha, Editors, W3C Recommendation, 10 February 2004, http://www.w3.org/TR/2004/REC-rdf-schema-20040210/ . Latest version available at http://www.w3.org/TR/rdf-schema/ .

[RDF-PRIMER]
RDF Primer, Frank Manola and Eric Miller, Editors, W3C Recommendation, 10 February 2004, http://www.w3.org/TR/2004/REC-rdf-primer-20040210/ . Latest version available at http://www.w3.org/TR/rdf-primer/ .

[CHARMOD]
Character Model for the World Wide Web 1.0, M. Dürst, F. Yergeau, R. Ishida, M. Wolf, T. Texin, Editors, World Wide Web Consortium Working Draft, work in progress, 22 August 2003. This version of the Character Model is http://www.w3.org/TR/2003/WD-charmod-20030822/. The latest version of the Character Model is at http://www.w3.org/TR/charmod/.

[XML-1.1]
Extensible Markup Language (XML) 1.1, John Cowan, Editor. W3C Candidate Recommendation 15 October 2002. This version is http://www.w3.org/TR/2002/CR-xml11-20021015/. The latest version is available at http://www.w3.org/TR/xml11/.

[XML-SCHEMA1]
XML Schema Part 1: Structures W3C Recommendation, World Wide Web Consortium, 2 May 2001. This version is http://www.w3.org/TR/2001/REC-xmlschema-1-20010502/. The latest version is available at http://www.w3.org/TR/xmlschema-1/.

[XML-NAMESPACES-1.1]
Namespaces in XML 1.1, Tim Bray, Dave Hollander, Andrew Layman, Richard Tobin, Editors. W3C Proposed Recommendation 05 November 2003. This version is http://www.w3.org/TR/2003/PR-xml-names11-20031105/. The latest version is available at http://www.w3.org/TR/xml-names11/.

[XML-INFOSET]
XML Information Set, John Cowan and Richard Tobin, W3C Recommendation, 24 October 2001. This document is http://www.w3.org/TR/2001/REC-xml-infoset-20011024/. The latest version is available at http://www.w3.org/TR/xml-infoset/.

[XPATH]
XML Path Language (XPath) Version 1.0, J. Clark and S. DeRose, Editors. World Wide Web Consortium, 16 November 1999. This version of XPath is http://www.w3.org/TR/1999/REC-xpath-19991116. The latest version of XPath is at http://www.w3.org/TR/xpath.

[OWL]
OWL Web Ontology Language Reference, Mike Dean and Guus Schreiber, Editors, W3C Recommendation, 10 February 2004, http://www.w3.org/TR/2004/REC-owl-ref-20040210/ . Latest version available at http://www.w3.org/TR/owl-ref/ .

[RDF-MS]
Resource Description Framework (RDF) Model and Syntax Specification, O. Lassila and R. Swick, Editors. World Wide Web Consortium. 22 February 1999. This version is http://www.w3.org/TR/1999/REC-rdf-syntax-19990222/. The latest version of RDF M&S is available at http://www.w3.org/TR/REC-rdf-syntax/.

[XHTML]
XHTML 1.0 The Extensible HyperText Markup Language (Second Edition), World Wide Web Consortium. 26 January 2000, revised 1 August 2002. This version is http://www.w3.org/TR/2002/REC-xhtml1-20020801/. The latest version of XHTML 1 is available at http://www.w3.org/TR/xhtml1/.

[IRI draft]
Internationalized Resource Identifiers (IRIs), M. Dürst and M. Suignard, Internet-Draft, June 2003, expires December 2003. This document is http://www.w3.org/International/iri-edit/draft-duerst-iri-04.

[TAG]
TAG Issues List, W3C Technical Architecture Group. This document is http://www.w3.org/2001/tag/issues.

Appendix A: Revisions Since Last Call Working Draft of 10 October 2003

There were no substantive changes.

The following editorial changes have been made:

Wording of Graph Equivalence
Following a suggestion from ter Horst, the wording of Section 6.3 has been improved.

Avoid 'globally'
In response to a comment from ter Horst, the wording of Section 3.2 has been changed, replacing a single phrase to avoid the word 'globally', which was undefined and unclear.

%s in RDF URI References
Following a comment from Patel-Schneider, an additional note has been added, concerning %-escapes in section 6.4 RDF URI References. A new informative reference to the [TAG] issue list has been added.

References
Dated references RDF and OWL documents have been updated.
