



## RIF Core

### W3C Working Draft 18 December 2008

**This version:**

<http://www.w3.org/TR/2008/WD-rif-core-20081218/>

**Latest version:**

<http://www.w3.org/TR/rif-core/>

**Previous version:**

<http://www.w3.org/TR/2007/WD-rif-core-20071030/>

**Editors:**

Harold Boley, National Research Council Canada  
Gary Hallmark, Oracle Corporation  
Michael Kifer, State University of New York at Stony Brook, USA  
Adrian Paschke, Free University Berlin  
Axel Polleres, DERI  
Dave Reynolds, Hewlett-Packard Laboratories, Bristol UK

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

---

[Copyright](#) © 2008 W3C® ([MIT](#), [ERCIM](#), [Keio](#)), All Rights Reserved. W3C [liability](#), [trademark](#) and [document use](#) rules apply.

---

## Abstract

This document, developed by the [Rule Interchange Format \(RIF\) Working Group](#), specifies RIF-Core, a common subset of RIF-BLD and RIF-PRD based on RIF-DTB 1.0. The RIF-Core presentation syntax and semantics are specified as restrictions on RIF-BLD. The XML serialization syntax of RIF-Core is specified via a mapping from the presentation syntax. A normative XML schema is also provided.

## Status of this Document

### May Be Superseded

*This section describes the status of this document at the time of its publication. Other documents may supersede this document. A list of current W3C publications and the latest revision of this technical report can be found in the [W3C technical reports index](http://www.w3.org/TR/) at <http://www.w3.org/TR/>.*

### Set of Documents

This document is being published as one of a set of 5 documents:

1. [RIF Use Cases and Requirements](#)
2. [RIF Core](#) (this document)
3. [RIF Datatypes and Built-Ins 1.0](#)
4. [RIF Production Rule Dialect](#)
5. [RIF Test Cases](#)

### Summary of Changes

The concept of RIF Core has changed considerably since the [30 March 2007 draft](#) which first offered a possible common-subset dialect. As stated in the [30 October 2007 placeholder](#), that first dialect was renamed "Basic Logic Dialect" and has since matured (see the [30 July 2008 Last Call Working Draft](#)). Meanwhile, the "Production Rule Dialect" (see today's [PRD draft](#)) has been developed, allowing the Working Group to design this new Core language as a subset of the intersection of BLD and PRD.

### Please Comment By 23 January 2009

The [Rule Interchange Format \(RIF\) Working Group](#) seeks public feedback on these Working Drafts. Please send your comments to [public-rif-comments@w3.org](mailto:public-rif-comments@w3.org) ([public archive](#)). If possible, please offer specific changes to the text that would address your concern. You may also wish to check the [Wiki Version](#) of this document for internal-review comments and changes being drafted which may address your concerns.

### No Endorsement

*Publication as a Working Draft does not imply endorsement by the W3C Membership. This is a draft document and may be updated, replaced or obsoleted by other documents at any time. It is inappropriate to cite this document as other than work in progress.*

## Patents

This document was produced by a group operating under the [5 February 2004 W3C Patent Policy](#). 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](#).

## Contents

- [1 Overview](#)
- [2 RIF-Core Presentation Syntax](#)
  - [2.1 Alphabet of RIF-Core](#)
  - [2.2 Terms of RIF-Core](#)
  - [2.3 Formulas of RIF-Core](#)
  - [2.4 Annotations and documents](#)
  - [2.5 Well-formed formulas](#)
  - [2.6 EBNF Grammar for the Presentation Syntax of RIF-Core \(Informative\)](#)
    - [2.6.1 EBNF for the RIF-Core Condition Language](#)
    - [2.6.2 EBNF for the RIF-Core Rule Language](#)
- [3 RIF-Core Semantics](#)
- [4 XML Serialization Syntax for RIF-Core](#)
- [5 Conformance and Safety](#)
  - [5.1 Safeness](#)
  - [5.2 Conformance Clauses](#)
- [6 RIF-Core as a Specialization of RIF-PRD](#)
- [7 Acknowledgements](#)
- [8 References](#)
  - [8.1 Normative References](#)
  - [8.2 Informational References](#)
- [9 Appendix: XML Schema for RIF-Core](#)
  - [9.1 Condition Language](#)
  - [9.2 Rule Language](#)

## 1 Overview

This specification develops **RIF-Core** (the **Core** of the **Rule Interchange Format**). From a theoretical perspective, RIF-Core corresponds to the language of definite Horn rules without function symbols ('Datalog') and a standard first-order semantics. RIF-Core thus is a subset of RIF-BLD [[RIF-BLD](#)]. At the same time, RIF-Core is a language of production rules permitting only assert actions. RIF-Core thus also is a subset of RIF-PRD [[RIF-PRD](#)]. Moreover, RIF-Core is based on the

built-ins of RIF-DTB 1.0 [[RIF-DTB](#)]. The common subset of RIF-BLD and RIF-PRD is specified based on RIF-DTB 1.0.

Syntactically, RIF-Core has a number of Datalog extensions to support features such as objects and frames as in F-logic [[KLW95](#)], internationalized resource identifiers (or IRIs, defined by [[RFC-3987](#)]) as identifiers for concepts, and XML Schema datatypes [[XML-SCHEMA2](#)]. In addition, RIF RDF and OWL Compatibility [[RIF-RDF+OWL](#)] defines the syntax and semantics of integrated RIF-Core/RDF and RIF-Core/OWL languages. These features make RIF-Core a Web-aware language. However, it should be kept in mind that RIF is designed to enable interoperability among rule languages in general, and its uses are not limited to the Web.

RIF-Core is defined as a specialization of RIF-BLD (hence of [[RIF-FLD](#)] which is part of the RIF extensibility framework). It is a syntactic subset of RIF-BLD, so that a well-formed RIF-Core formula (including documents and condition formula) is also a well-formed RIF-BLD formula.

RIF-Core is also a syntactic subset of [[RIF-PRD](#)]. It is intended that a RIF-PRD consumer can treat a RIF-Core ruleset as if it were a RIF-PRD rule set and it would conform to the normative RIF-Core first order semantics. However, due to the presence of builtin functions and predicates there are RIF-Core rulesets which are unsafe and do not reach a stable fixed-point under RIF-PRD semantics. We define the conformance to RIF-Core so as to only require conformance over a safe subset of rules, in this way we permit RIF-PRD processors to be *safely conformant* while allowing RIF-Core documents to contain unsafe rules. Producers of RIF-Core who require maximum interchange are advised to restrict themselves to safe rules. These notions of safeness and safe conformance are defined formally in section 5 [Conformance and Safety](#).

RIF-Core is not a maximal intersection of RIF-BLD and RIF-PRD. It omits some features which do not significantly add to the expressiveness of the language and are perceived to be not widely supported by rule languages. For example, named argument uniterms are omitted from RIF-Core on such grounds.

To give a preview, here is a simple complete RIF-Core example deriving a ternary relation from its inverse.

**Example 1** (An introductory RIF-Core example).

A rule can be written in English to derive the `buy` relationships (rather than store them) from the `sell` relationships that are stored as facts (e.g., as exemplified by the English statement below):

- *A buyer buys an item from a seller if the seller sells the item to the buyer.*
- *John sells LeRif to Mary.*

The fact *Mary buys LeRif from John* can be logically derived by a *modus ponens* argument. Assuming Web IRIs for the predicates `buy` and `sell`, as well as for the

individuals John, Mary, and LeRif, the above English text can be represented in RIF-Core Presentation Syntax as follows.

```
Document (
  Prefix(cpt http://example.com/concepts#)
  Prefix(ppl http://example.com/people#)
  Prefix(bks http://example.com/books#)

  Group
  (
    Forall ?Buyer ?Item ?Seller (
      cpt:buy(?Buyer ?Item ?Seller) :- cpt:sell(?Seller ?Item ?Buyer)
    )

    cpt:sell(ppl:John bks:LeRif ppl:Mary)
  )
)
```

For the interchange of such rule (and fact) documents, an equivalent RIF-Core XML Syntax is given in this specification. To formalize their meaning, a RIF-Core Semantics is specified.

This document assumes familiarity with [\[RIF-BLD\]](#) and [\[RIF-PRD\]](#), as RIF-Core is derived from these documents via syntactic restrictions.

## 2 RIF-Core Presentation Syntax

Like RIF-BLD and RIF-PRD, RIF-Core has both a **presentation** syntax *and an* XML syntax. The presentation syntax is normative, but is not intended to be a concrete syntax for RIF-BLD. It is defined in "mathematical English," a special form of English for communicating mathematical definitions, examples, etc. The presentation syntax deliberately leaves out details such as the delimiters of the various syntactic components, escape symbols, parenthesizing, precedence of operators, and the like. Since RIF is an interchange format, it uses XML as its concrete syntax and RIF-BLD conformance is described in terms of semantics-preserving transformations.

RIF-Core is a syntactic subset of RIF-BLD, and this section defines the presentation syntax of RIF-Core as a restriction on [the presentation syntax of RIF-BLD](#).

### 2.1 Alphabet of RIF-Core

The **alphabet** of the presentation language of RIF-Core is the alphabet of the RIF-BLD presentation language with the exclusion of the symbol ## and the set of symbols `ArgNames`.

**Editor's Note:** The status of membership (#) and subclass (##) formulas within Core is under debate in the working group. While there is a notion of membership within PRD it is restricted compared to that in BLD. This current draft for Core includes membership (#) but restricts its use to solely within the RIF Core Condition Formulas. Future drafts may extend its use, also include ## or may omit both entirely.

## 2.2 Terms of RIF-Core

The **Terms** of RIF-Core are the terms of RIF-BLD with the exclusion of *terms with named arguments*, *subclass terms*, and external terms of the form `External(fr)`, where `fr` is a frame term.

**Editor's Note:** This presumes that the At Risk #1 feature External frames is retained in RIF-BLD. If that is not the case the above will need updating.

## 2.3 Formulas of RIF-Core

The **Formulas** of the RIF-Core are the formulas of RIF-BLD with the following modifications.

- Subterms that occur inside *positional* atomic formulas can be either variables or constants. This implies that RIF-Core does not allow function symbols in positional formulas.
- Subterms that occur in an equality term can be variables, constants, or external positional terms. Thus, while function symbols are not allowed as arguments to predicates, built-in and externally defined functions are permitted inside equalities.
- Equality terms *cannot* occur in rule conclusions -- they are allowed only in rule premises.

**Editor's Note:** Built-in functions *and* predicates present problems for many bottom-up and top-down reasoners, which lack constraint-solving capabilities. This is because these reasoners can typically make use of a built-in only if certain arguments are bound. For instance, a bottom-up reasoner might have trouble with evaluating a condition formula `And(p(?X) External(foo(?Y, ?X)))`, where `p` is a database predicate and `foo` a builtin predicate, *if* the reasoner can invoke `foo` only when the first argument `?Y` is bound. The working group is currently discussing the implications of all this for RIF-Core compliance.

## 2.4 Annotations and documents

RIF-Core allows every term and formula to be optionally annotated in the same way as RIF-BLD. The frame formulas that are allowed as part of an annotation must be syntactically correct for RIF-Core. In particular, no function symbols are allowed in such a formula.

## 2.5 Well-formed formulas

A syntactically correct RIF-Core formula that passes the well-formedness test for RIF-BLD is also a well-formed RIF-Core formula. In this case, it simply means that

- each symbol in `Const` can be either an individual, a (non-external) predicate symbol, an external function, or an external predicate
- if a symbol occurs in a position of a predicate or (external) function of some arity then it cannot occur elsewhere in the formula with some other arity
- for every occurrence of `External(t)`, `t` must be an instance of the coherent set of external schemas (Section [Schemas for Externally Defined Terms](#) of [RIF-DTB]) associated with the language of RIF-Core.

For `rif:local` symbols, the first three of the above clauses apply only within the scope of a single document. That is, two different occurrences of the same `rif:local` symbol in different documents may have different arities, may occur both as predicates and individuals, etc. For other symbols, the above restrictions apply not only to the document that contains the particular occurrence of the symbol, but also to all documents that are imported by that document.

## 2.6 EBNF Grammar for the Presentation Syntax of RIF-Core (Informative)

Until now, we have used mathematical English to specify the syntax of RIF-Core as a restriction on RIF-BLD. Tool developers, however, may prefer EBNF notation, which provides a more succinct overview of the syntax. However, EBNF is unable to express all of the well-formedness conditions. For instance, the requirement that each symbol appear in only one context cannot be so expressed. As a result, the EBNF grammar defines a strict superset of RIF-Core. For this reason this section is not normative.

The EBNF for the RIF-Core presentation syntax is given as follows. For convenience of reading we show the entire EBNF of its three parts (rules, conditions, and annotations); these are derived from the EBNF for RIF-BLD applying the restrictions described above.

**Rule Language:**

```

Document ::= IRIMETA? 'Document' '(' Base? Prefix* Import* Group? ')'
Base ::= 'Base' '(' IRI ')'
Prefix ::= 'Prefix' '(' Name IRI ')'
Import ::= IRIMETA? 'Import' '(' IRICONST PROFILE? ')'
Group ::= IRIMETA? 'Group' '(' (RULE | Group)* ')'
RULE ::= (IRIMETA? 'Forall' Var+ '(' CLAUSE ')') | CLAUSE
CLAUSE ::= Implies | ATOMIC
Implies ::= IRIMETA? ATOMIC ':-' FORMULA
PROFILE ::= TERM

```

**Condition Language:**

```

FORMULA ::= IRIMETA? 'And' '(' FORMULA* ')' |
          IRIMETA? 'Or' '(' FORMULA* ')' |
          IRIMETA? 'Exists' Var+ '(' FORMULA ')' |
          ATOMIC |
          IRIMETA? Equal |
          IRIMETA? Member |
          IRIMETA? 'External' '(' Atom ')'
ATOMIC ::= IRIMETA? (Atom | Frame )
Atom ::= UNITERM
UNITERM ::= Const '(' (TERM* ')'
Equal ::= TERM '=' ( TERM | 'External' '(' FUNC ')' )
FUNC ::= Const '(' (GENERAL_TERM* ')'
GENERAL_TERM ::= IRIMETA? (Const | Var | FUNC)
Member ::= TERM '#' TERM
Frame ::= TERM '[' (TERM '->' TERM)* ']'
TERM ::= IRIMETA? (Const | Var)
Const ::= '"' UNICODESTRING '"^^' SYMSPACE | CONSTSHORT
Name ::= UNICODESTRING
Var ::= '?' UNICODESTRING
SYMSPACE ::= ANGLEBRACKIRI | CURIE

```

**Annotations:**

```

IRIMETA ::= '(*' IRICONST? (Frame | 'And' '(' Frame* ')')? '*'

```

The following subsections explain and exemplify the Condition Language and Rule Language parts; the Annotations part is unchanged from RIF-BLD.

**2.6.1 EBNF for the RIF-Core Condition Language**

The RIF-Core Condition Language represents formulas that can be used in the premises of RIF-Core rules (also called rule bodies). The EBNF grammar for a superset of the RIF-Core condition language is shown in the above [conditions part](#).



This is a subset of the EBNF for the RIF-BLD condition language specified in RIF-BLD [conditions part](#) reflecting the syntax restrictions on RIF-Core described normatively in sections 2.1 through 2.5 above.

The [Example 2](#) from the RIF-BLD document illustrates a RIF-Core condition.

### 2.6.2 EBNF for the RIF-Core Rule Language

The presentation syntax for RIF-Core rules is based on the syntax in Section [EBNF for RIF-Core Condition Language](#) with the productions shown in the above [rules part](#).

Again, this is a subset of the EBNF for the RIF-BLD rule language specified in RIF-BLD [rules part](#) reflecting the syntax restrictions on RIF-Core described normatively in sections 2.1 through 2.5 above.

[Example 3](#) from the RIF-BLD document illustrates a set of RIF-Core rules and [Example 4](#) a RIF-Core document that contains an annotated group formula. In contrast, the [Example 6](#) from the RIF-BLD document shows a formula that is *not* in RIF-Core because it includes terms with named arguments, which are not allowed in this dialect.

## 3 RIF-Core Semantics

RIF-Core is a syntactic subset of RIF-BLD and the semantics of RIF-Core is identical that of RIF-BLD.

## 4 XML Serialization Syntax for RIF-Core

The XML syntax of RIF-Core is a subset of the [XML syntax of RIF-BLD](#). All XML tags of RIF-BLD (except `Subclass`, `sub` and `super`) are supported, but the XML schema of RIF-Core restricts their context with respect to what is allowed by the XML Schema of RIF-BLD. The semantics of the XML syntax for RIF-Core is defined through the same [RIF-BLD XML-to-presentation syntax mapping](#).

**Editor's Note:** As noted earlier the status of membership (#) and subclass (##) formulas are under discussion in the working group. Depending on the resolution of this issue the tags relating to subclass may be reinstated or those relating to membership may be removed.

[Example 5](#), "A RIF condition and its XML serialization," in the RIF-BLD document illustrates XML serialization for a RIF-Core condition formula. XML serialization of a complete RIF-Core document appears in the RIF-BLD document as [Example 7](#).

## 5 Conformance and Safety

RIF-Core is a syntactic subset of both RIF-BLD and RIF-PRD. The semantics of a RIF-Core formula is the same as in RIF-BLD.

All RIF-Core documents are syntactically also RIF-PRD documents. However, some formulas may be *unsafe* and cannot be executed under the RIF-PRD operational semantics. Thus, in order to allow production rule systems and logic programming systems to interchange rulesets via RIF-Core, we define a *safe* subset of RIF-Core and the notion of *safely conformant* RIF-Core consumers or producers. A safely conformant RIF-Core consumer can reject non-safe RIF-Core documents even if they are otherwise well-formed. The notion of (general) conformance allows RIF-Core producers and consumers to produce and consume unsafe rulesets. For safe Core documents, the logical semantics of RIF-BLD and operational fixed-point semantics of RIF-PRD coincide. In the absence of data types and built-ins, this follows from equivalence of operational and declarative semantics for Datalog and discussed in [\[Vianu97\]](#).

**Editor's Note:** Since Vianu97 does not prove the equivalence of declarative/ fixed-point semantics for Datalog, just discusses it, the rewritten PRD section should add a more appropriate reference.

**Editor's Note:** A complete specification of Core as a specialization of RIF-PRD will be included in a future draft.

### 5.1 Safeness

#### Definition (Safeness).

- A variable  $?v$  is said to be *safe* with respect to a RIF condition formula  $\varphi$  if it appears in at least one *plain* atomic subformula of  $\varphi$  and is not in the scope of an existential quantifier within  $\varphi$ . A subformula of a RIF condition formula is *plain* if it is not an atom of the form `External(...)` and not within a disjunction.

- A RIF condition formula  $\varphi$  is called *safe* if all its variables are safe with respect to  $\varphi$  and for all its existential subformulas of the form  $\text{Exists } ?V_1 \dots ?V_n (\psi)$  each  $?V_1 \dots ?V_n$  is safe with respect to  $\psi$ .
- A RIF rule implication  $\varphi :- \psi$  is called *safe* if all its variables are safe with respect to  $\psi$ .
- A RIF Universal rule implication  $\text{Forall } ?V_1 \dots ?V_n (\varphi)$  is called *safe* if the implication  $\varphi$  is *safe*.
- A RIF Universal fact  $\text{Forall } ?V_1 \dots ?V_n (\varphi)$  is *safe* if  $\varphi$  contains no variables.

**Editor's Note:** In a future draft, there will be further discussion on the intuitions underlying this section's development of safeness conditions. Both positive and negative examples of various clauses in the definition of safeness will also be provided.

## 5.2 Conformance Clauses

RIF-Core conformance is described in terms of semantics-preserving transformations.

Let  $T$  be a set of datatypes that includes the datatypes specified in [RIF-DTB], and suppose  $E$  is a set of external predicates and functions that includes the built-ins listed in [RIF-DTB]. We say that a formula  $\varphi$  is a *Core<sub>T,E</sub>* formula iff

- it is a well-formed Core formula,
- all the datatypes used in  $\varphi$  are in  $T$ , and
- all the externally defined functions and predicates used in  $\varphi$  are in  $E$ .

A RIF processor is a **conformant Core<sub>T,E</sub> consumer** iff it implements a [semantics-preserving mapping](#) from the set of all Core<sub>T,E</sub> formulas to the language  $L$  of the processor.

A RIF processor is a **conformant Core<sub>T,E</sub> producer** iff it implements a [semantics-preserving mapping](#) from a subset of the language  $L$  of the processor to a set of Core<sub>T,E</sub> formulas.

A **conformant document** is an XML document that conforms to all the syntactic constraints of RIF-Core, including ones that cannot be checked by an XML Schema validator. Note that the concrete presentation syntax given in Section 2.6 is purely informative (to help implementers see the set of language structures supported by RIF-Core), the only normative concrete syntax for RIF-Core is the XML syntax.

In addition:

- Conformant Core producers and consumers are required to support only the entailments of [closed RIF condition formulas](#).

- A **conformant RIF-Core consumer** is a conformant  $\text{Core}_{T,E}$  consumer in which T consists only of the datatypes and E consists only of the externally defined functions and predicates that are required by RIF-Core. These datatypes and externally defined terms (called built-ins) are specified in [RIF-DTB]. A conformant RIF-Core consumer must reject all inputs that do not match the syntax of Core. If it implements extensions, it may do so under user control -- having a "strict Core" mode and a "run-with-extensions" mode.
- A **conformant Core producer** is a conformant  $\text{Core}_{T,E}$  producer, which produces documents that include only the datatypes and externals that are required by Core.

### Feature At Risk #3: Strictness Requirement

Note: This feature is "[at risk](#)" and may be removed from this specification based on feedback. Please send feedback to [public-rif-comments@w3.org](mailto:public-rif-comments@w3.org).

The two preceding clauses are features **AT RISK**. In particular, the "strictness" requirement is under discussion.

- **Safely conformant.** A Core consumer or producer is safely conformant if it supports only entailments of the form  $\varphi \models_{\text{Core}} \psi$ , where  $\psi$  is a *closed RIF condition formula* and  $\varphi$  is a *safe formula*. Since such a consumer/producer might not preserve entailment over unsafe formulas, it might not be Core-conformant in the more general sense defined earlier. Thus, safe conformance is a weaker requirement than (general) conformance.

## 6 RIF-Core as a Specialization of RIF-PRD

**Editor's Note:** In a future draft the specification of RIF-Core as a specialization of RIF-PRD will be provided.

## 7 Acknowledgements

This document is the product of the Rules Interchange Format (RIF) Working Group (see below) whose members deserve recognition for their time and commitment. The editors extend special thanks to: \*\*\*, for their thorough reviews and insightful discussions; the working group chairs, Chris Welty and Christian de Sainte-Marie, for their invaluable technical help and inspirational leadership; and W3C staff contact Sandro Hawke, a constant source of ideas, help, and feedback.

The regular attendees at meetings of the Rule Interchange Format (RIF) Working

Group at the time of the publication were: Adrian Paschke (REWERSE), Axel Polleres (DERI), Chris Welty (IBM), Christian de Sainte Marie (ILOG), Dave Reynolds (HP), Gary Hallmark (ORACLE), Harold Boley (NRC), Hassan Aït-Kaci (ILOG), Igor Mozetic (JSI), John Hall (OMG), Jos de Bruijn (FUB), Leora Morgenstern (IBM), Michael Kifer (Stony Brook), Mike Dean (BBN), Sandro Hawke (W3C/MIT), and Stella Mitchell (IBM). We would also like to thank \*\*\* past members of the working group, \*\*\*.

## 8 References

### 8.1 Normative References

#### [RDF-CONCEPTS]

*Resource Description Framework (RDF): Concepts and Abstract Syntax*, Klyne G., Carroll J. (Editors), W3C Recommendation, 10 February 2004, <http://www.w3.org/TR/2004/REC-rdf-concepts-20040210/>. Latest version available at <http://www.w3.org/TR/rdf-concepts/>.

#### [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>.

#### [RFC-3987]

*RFC 3987 - Internationalized Resource Identifiers (IRIs)*, M. Duerst and M. Suignard, IETF, January 2005. This document is <http://www.ietf.org/rfc/rfc3987.txt>.

#### [RIF-BLD]

*RIF Basic Logic Dialect* Harold Boley, Michael Kifer, eds. W3C Working Draft, 30 July 2008, <http://www.w3.org/TR/2008/WD-rif-bld-20080730/>. Latest version available at <http://www.w3.org/TR/rif-bld/>.

#### [RIF-DTB]

*RIF Datatypes and Built-Ins 1.0* Axel Polleres, Harold Boley, Michael Kifer, eds. W3C Working Draft, 18 December 2008, <http://www.w3.org/TR/2008/WD-rif-dtb-20081218/>. Latest version available at <http://www.w3.org/TR/rif-dtb/>.

#### [RIF-FLD]

*RIF Framework for Logic Dialects* Harold Boley, Michael Kifer, eds. W3C Working Draft, 30 July 2008, <http://www.w3.org/TR/2008/WD-rif-fld-20080730/>. Latest version available at <http://www.w3.org/TR/rif-fld/>.

#### [RIF-RDF+OWL]

*RIF RDF and OWL Compatibility* Jos de Bruijn, editor. W3C Working Draft, 30 July 2008, <http://www.w3.org/TR/2008/WD-rif-rdf-owl-20080730/>. Latest version available at <http://www.w3.org/TR/rif-rdf-owl/>.

**[RIF-PRD]**

*RIF Production Rule Dialect* Christian de Sainte Marie, Adrian Paschke, Gary Hallmark, eds. W3C Working Draft, 18 December 2008, <http://www.w3.org/TR/2008/WD-rif-prd-20081218/>. Latest version available at <http://www.w3.org/TR/rif-prd/>.

**[XML1.0]**

*Extensible Markup Language (XML) 1.0 (Fourth Edition)*, W3C Recommendation, World Wide Web Consortium, 16 August 2006, edited in place 29 September 2006. This version is <http://www.w3.org/TR/2006/REC-xml-20060816/>.

**[XML-Base]**

*XML Base*, W3C Recommendation, World Wide Web Consortium, 27 June 2001. This version is <http://www.w3.org/TR/2001/REC-xmlbase-20010627/>. The latest version is available at <http://www.w3.org/TR/xmlbase/>.

**[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/>.

## 8.2 Informational References

**[ANF01]**

*Normal Form Conventions for XML Representations of Structured Data*, Henry S. Thompson. October 2001. Available at <http://www.ltg.ed.ac.uk/~ht/normalForms.html>.

**[CL73]**

*Symbolic Logic and Mechanical Theorem Proving*, C.L. Chang and R.C.T. Lee. Academic Press, 1973.

**[CURIE]**

*CURIE Syntax 1.0: A syntax for expressing Compact URIs*, Mark Birbeck, Shane McCarron. W3C Working Draft 2 April 2008. Available at <http://www.w3.org/TR/curie/>.

**[Enderton01]**

*A Mathematical Introduction to Logic, Second Edition*, H. B. Enderton. Academic Press, 2001.

**[KLW95]**

*Logical foundations of object-oriented and frame-based languages*, M. Kifer, G. Lausen, J. Wu. Journal of ACM, July 1995, pp. 741--843.

**[Mendelson97]**

*Introduction to Mathematical Logic, Fourth Edition*, E. Mendelson. Chapman & Hall, 1997.

**[OWL-Reference]**

[OWL Web Ontology Language Reference](#), M. Dean, G. Schreiber, Editors, W3C Recommendation, 10 February 2004. Latest version available at <http://www.w3.org/TR/owl-ref/>.

**[RDFSYN04]**

*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/>.

**[RIF-UCR]**

[RIF Use Cases and Requirements](#) Adrian Paschke, David Hirtle, Allen Ginsberg, Paula-Lavinia Patranjan, Frank McCabe, eds. W3C Working Draft, 18 December 2008, <http://www.w3.org/TR/2008/WD-rif-ucr-20081218/>. Latest version available at <http://www.w3.org/TR/rif-ucr/>.

**[TRT03]**

*Object-Oriented RuleML: User-Level Roles, URI-Grounded Clauses, and Order-Sorted Terms*, H. Boley. Springer LNCS 2876, Oct. 2003, pp. 1-16. Preprint at [http://it-iti.nrc-cnrc.gc.ca/publications/nrc-46502\\_e.html](http://it-iti.nrc-cnrc.gc.ca/publications/nrc-46502_e.html).

**[vEK76]**

*The semantics of predicate logic as a programming language*, M. van Emden and R. Kowalski. Journal of the ACM 23 (1976), pp. 733-742.

**[Vianu97]**

*Rule-Based Languages*, Vianu V.. Annals of Mathematics and Artificial Intelligence 19 (1997), pp. 215-259.

## 9 Appendix: XML Schema for RIF-Core

The **namespace** of RIF is *http://www.w3.org/2007/rif#*.

XML schemas for the RIF-Core sublanguages are defined below.

<b>Editor's Note:</b> The schemas will be made available on line in a future working draft.
---

## 9.1 Condition Language

```
<?xml version="1.0" encoding="UTF-8"?>
```

## 9.2 Rule Language

```
<?xml version="1.0" encoding="UTF-8"?>
```