W3C Working Draft

RIF Test Cases

W3C Working Draft 11 May 2010

This version:
http://www.w3.org/TR/2010/WD-rif-test-20100511/
Latest version:
http://www.w3.org/TR/rif-test/
Previous version:
http://www.w3.org/TR/2009/WD-rif-test-20091001/ (color-coded diff)

Editors:
Stella Mitchell, Cornell University
Leora Morgenstern, Courant Institute of Mathematical Sciences
Adrian Paschke, Freie Universitaet Berlin

This document is also available in these non-normative formats: PDF version.

Abstract

This document describes the test cases developed by the Rule Interchange Format (RIF) Working Group in accordance with the Working Group's Charter. These test cases are intended to aid in the conformance evaluation of RIF implementations and thus promote interoperability.

Status of this Document

May Be Superseded

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

Set of Documents

This document is being published as one of a set of 10 documents:

1. RIF Overview
2. RIF Core Dialect
3. RIF Basic Logic Dialect
4. RIF Production Rule Dialect
5. RIF Framework for Logic Dialects
6. RIF Datatypes and Built-Ins 1.0
7. RIF RDF and OWL Compatibility
8. OWL 2 RL in RIF
9. RIF Combination with XML data
10. RIF Test Cases (this document)

XML Schema Datatypes Dependency

RIF is defined to use datatypes defined in the XML Schema Definition Language (XSD). As of this writing, the latest W3C Recommendation for XSD is version 1.0, with version 1.1 progressing toward Recommendation. RIF has been designed to take advantage of the new datatypes and clearer explanations available in XSD 1.1, but for now those advantages are being partially put on hold. Specifically, until XSD 1.1 becomes a W3C Recommendation, the elements of RIF which are based on it should be considered *optional*, as detailed in Datatypes and Builtins, section 2.3. Upon the publication of XSD 1.1 as a W3C Recommendation, those elements will cease to be optional and are to be considered required as otherwise specified.

We suggest that for now developers and users follow the XSD 1.1 Last Call Working Draft. Based on discussions between the Schema, RIF and OWL Working Groups, we do not expect any implementation changes will be necessary as XSD 1.1 advances to Recommendation.

Summary of Changes

There have been no substantive changes since the previous version. For details on the minor changes see the change log and color-coded diff.

Please Comment By 8 June 2010

The Rule Interchange Format (RIF) Working Group seeks public feedback on this Working Draft. Please send your comments to 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 and see if the relevant text has already been updated.

No Endorsement

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

Patents

This document was produced by a group operating under the 5 February 2004 W3C Patent Policy. The group does not expect this document to become a W3C Recommendation. 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 document describes the test cases developed by the Rule Interchange Format (RIF) Working Group in accordance with the Working Group's Charter. The sections below delineate the scope of the test cases, explain their limits in determining conformance, and present the different types of tests and the format in which they are represented.

The test cases are maintained in the RIF Test Repository, and the normative test suites can be downloaded from there. Formatted copies of all test cases in the repository, plus additional tests that have not yet been fully processed by the Working Group, can be viewed at the RIF Test Collection Site.

This set of test cases is intended to enable an empirical investigation of RIF implementations. They can help identify problems both with the software developed to implement RIF specifications and with the specifications themselves. A widely used test suite providing good coverage of the target RIF dialect makes it more likely that different implementations will interoperate correctly.

The format of the tests is designed to be suitable for use by RIF implementers using test harnesses. Developers will need to write their own test harnesses, and are encouraged to report their results as well as any problems they encounter with the tests.

If additional test cases are contributed to the W3C, the Consortium may add them to the set of RIF test cases.

1.1 Scope

This Working Draft documents test cases for the RIF Core Dialect, Basic Logic Dialect (BLD) and Production Rule Dialect (PRD), as well as for RIF-RDF and RIF-OWL combinations as specified in RIF RDF and OWL Compatibility. The current set of tests focuses on testing syntax, entailment and validity of import directives, as described in the Test Types section. Additional types of tests may be added in the future.

The test suite contains tests for RIF consumers as defined in the RIF-Core Conformance Clauses, RIF-BLD Conformance Clauses, and the RIF-PRD Conformance Clauses, but not for RIF producers. Producer tests are not included since producers translate from a particular rule language into RIF, and specifying tests to verify this translation without referring to specific rule languages would be difficult. However, partial syntactic validation of documents generated by producers can be performed using the XML schema for the target dialect (Core XML Schema, BLD XML Schema, or PRD XML Schema). For implementations that are both producers and consumers, some validation of the producer can be achieved by translating an entailment test from RIF to the target rule language, then back to RIF, then again to the target rule language, and then executing the entailment test.

The tests are designed to:

- aid in conformance evaluation by providing evidence that the specification has been implemented
- provide generally broad coverage of the language features
- focus on non-obvious features and behavior, and hard to implement features, since these types of tests are more likely to uncover problems in implementations.
- pinpoint omissions that can be corrected.
- illustrate the use and meaning of language features.

The test suite is not exhaustive: it does not completely cover the RIF specification.

1.2 Conformance

Conformance is defined as the fulfillment of specified requirements, which are detailed in the conformance clause of a specification. (See QA Framework specification Guide.) Admissable RIF documents, and conformant RIF consumers and producers are defined in the *conformance clauses* section of the relevant specification: RIF-Core Conformance Clauses, RIF-BLD Conformance Clauses, and RIF-PRD Conformance Clauses. A complete conformance test suite would be a set of test cases that cover this set of requirements, such that passing all the tests in the suite for a particlar RIF dialect indicates conformance to that dialect. However, the development of this type of comprehensive test suite is beyond the scope of the Working Group, and the RIF test cases do not constitute complete conformance test suites. Failure to pass all the tests in the suite indicates that the implementation does not meet the relevant specification. But passing all the tests in the suite indicates only that the implementation is conformant to that particular version of the test suite.

2 Deliverables

The RIF test package consists of:

- The RIF Test Cases document (this document)
- A zip file for each dialect test suite that includes Approved and Proposed tests:
    - Files containing the input (RIF/XML, RDF and OWL) documents used in the tests
    - Individual manifest files describing each test case
    - RIF Presentation Syntax versions of the RIF documents, as a human-readable version of the test. Specification of the presentation syntaxes can be found at RIF-Core Presentation Syntax, RIF-BLD Presentation Syntax, and RIF-PRD Presentation Syntax.
    - A Manifest file for each dialect test suite (CoreTests.xml, BLDTests.xml, PRDTests.xml), describing all the tests applicable to that dialect.
- A web site that includes all the files mentioned above, as well as individual tests organized into subdirectories.

3 Test Types

This section introduces a categorization of the tests included in the test suite. Each test case describes inputs that can be provided to a RIF processor, and specifies the behavior required to satisfy the conformance conditions in that situation. There are several different types of test cases detailed in the following sub-sections. The type of test determines the task, associated inputs, and expected outcome of the test.

3.1 Syntactic Tests

Syntactic tests validate a RIF consumer's recognition of an *admissible* RIF XML document. An admissible RIF document conforms to all the syntactic constraints of the relevant dialect, including those that cannot be checked with XML Schema validation.

3.1.1 Positive Syntax Tests

These tests involve a single RIF document, indicated by the InputDocument element in the manifest for the test. The document is a syntactically correct RIF document in each dialect indicated by a dialect element in the manifest.

The test is passed if the processor indicates that the document is syntactically correct, and failed otherwise.

- If there is a dialect property with a value of *rifTest:Core* then the test document is an admissible Core document.
- If there is a dialect property with a value of *rifTest:BLD* then the test document is an admissible BLD document in XML syntax.
- If there is a dialect property with a value of *rifTest:PRD* then the test document is an admissible PRD document.

Note that the premises of all positive and negative entailment tests (defined below) are syntactically correct RIF documents and so can be used as positive syntax tests.

Note also that a positive syntax test for a given dialect D is also a positive syntax test for any dialect that extends D. For dialects defined by the RIF Working Group, each dialect that a test applies to is indicated explicitly with a dialect element in the manifest.

3.1.2 Negative Syntax Tests

These tests involve a single RIF document, indicated by the InputDocument element in the manifest for the test. The document is not a syntactically correct RIF document in each dialect indicated by a dialect element in the manifest.

The test is passed if the processor indicates that the document is syntactically incorrect, and failed otherwise.

- If there is a dialect property with a value of *rifTest:Core* then the test document is not an admissible Core document.
- If there is a dialect property with a value of *rifTest:BLD* then the test document is not an admissible BLD document in XML syntax.
- If there is a dialect property with a value of *rifTest:PRD* then the test document is not an admissible PRD document.

Note that a negative syntax test for a given dialect D is also a negative syntax test for any dialect that restricts D. For dialects defined by the RIF Working Group, each dialect that a test applies to is indicated explicitly with a dialect element in the manifest.

3.2 Import Rejection Tests

Import rejection tests check a RIF consumer's recognition of an invalid imported document set. For RIF-RDF and RIF-OWL combinations, invalid import scenarios are specified in the Interpretation of Profiles section of the RIF RDF and OWL Compatibility document. For Core, BLD or PRD documents, import directives can result in a document set that doesn't satisfy the Core well-formedness, BLD well-formedness or PRD well-formedness requirements.

Each test of this type includes a RIF document, indicated by the InputDocument element in the manifest for the test, and one or more documents in the imports closure of the input document, indicated by ImportedDocument elements.

The test is passed if the processor indicates that the input document is rejected, and failed otherwise.

Note that an import rejection test for a given dialect D is also an import rejection test for any dialect that extends D. For dialects defined by the RIF Working Group, each dialect that a test applies to is indicated explicitly with a dialect element in the manifest.

3.3 Semantic Tests

These tests validate a RIF processor's computation of the entailment relation. Each entailment test has one or more associated dialects, and is of the general form

Premises
    R
Conclusion
    C

Where ***R*** is a RIF document, ***C*** is a RIF condition, an RDF document or an OWL document, and the imports closure of ***R*** *entails* (for positive entailment tests) or *does not entail* (for negative entailment tests) ***C*** in the given dialects. The conclusions of semantic tests are defined to be RIF conditions because RIF Conformance is defined in terms of the entailment of closed RIF condition formulas. The complete specification of these tests is given in the sub-sections below.

Note that a positive or negative entailment test for a given dialect D also applies to any dialect that extends D. For dialects defined by the RIF Working Group, each dialect that a test applies to is indicated explicitly with a dialect element in the test's manifest.

3.3.1 Positive Entailment Tests

Each test of this type includes a premises document, indicated by the PremiseDocument element in the manifest for the test, a conclusion document, indicated by the ConclusionDocument element, and optionally one or more documents in the imports closure of the premises document, indicated by ImportedDocument elements. Note that the conclusion document is a RIF condition.

The applicable entailment regimes are indicated by dialect elements in the manifest file:

- if there is a dialect property with a value of "rifTest:Core" then the entailment holds according to the RIF-Core semantics.
- if there is a dialect property with a value of "rifTest:BLD" then the entailment holds according to the RIF-BLD definition of entailment.
- if there is a dialect property with a value of "rifTest:PRD" then the conclusion is satisfied according to the RIF-PRD definition of condition satisfaction.

In addition, an optional Combinations element in the manifest indicates the types of documents in the imports closure of the premises document.

Note that, in general, the conclusion will not include *everything* that is entailed by the premises.

A conformant RIF consumer should report that the conclusion is entailed by the premises, should not report that the answer is undecided, and must not report that the conclusion is not entailed by the premises.

3.3.2 Negative Entailment Tests

Each test of this type includes a premises document, indicated by the PremiseDocument element in the manifest for the test, a conclusion document, indicated by the NonConclusionDocument element, and optionally one or more documents in the imports closure of the premises document, indicated by ImportedDocument elements.

The applicable entailment regimes are indicated by dialect elements in the manifest file:

- if there is a dialect property with a value of "rifTest:Core" then the entailment does not hold according to the RIF-Core semantics.
- if there is a dialect property with a value of "rifTest:BLD" then the entailment does not hold according to the RIF-BLD definition of entailment.

- if there is a dialect property with a value of "rifTest:PRD" then the conclusion is not satisfied according to the RIF-PRD definition of condition satisfaction.

In addition, an optional Combinations element in the manifest indicate the types of documents in the imports closure of the premises document.

A conformant RIF consumer should report that the conclusion is not entailed by the premises, should not report that the answer is undecided, and must not report that the conclusion is entailed by the premises.

4 Test Case Format

Each test case has an associated manifest file that contains information needed to property execute the test. These files are in a machine-readable XML format in order to enable the development of automated testing frameworks. The information in the manifests can be used to select tests according to a variety of criteria, such as what dialect they apply to, and to locate the documents that participate in the test.

The format of the test cases follows the general guidelines of the W3C QA Working Group's [Test Metadata Note]. The RIF test case schema is given in rif-testcase.xsd.

This section presents a summary of the data elements, organized by the type of test to which they apply. The name of the root element of a test case document indicates the test type:

- PositiveEntailmentTest
- NegativeEntailmentTest
- PositiveSyntaxTest
- NegativeSyntaxTest
- ImportRejectionTest

The root element has two attributes:

- ***id*** is the unique name for the test case. It conforms to ***irelative-ref*** as defined in [*RFC-3987*] so that it can be appended to a base to generate a URI for the test.
- ***src*** is a link to the test case in the repository.

4.1 Properties of All Tests

4.1.1 status

The value of this property is the status of the test case according to the test case approval process. Values of this property are one of an enumerated list: *Proposed*, *Approved*, *Rejected* or *Obsolete*.

This is a required element, and there is one for each test.

4.1.2 dialect

This property is used to indicate the RIF dialects that the test applies to. There is one 'dialect' element for each dialect that the test is valid for. The possible dialect values are *Core*, *BLD* and *PRD*.

This is a required element, and there are one or more for each test.

4.1.3 purpose

The value of this property is a brief explanation of the purpose of the test, suitable for display in tabular format.

This is a required element, and there is one for each test.

4.1.4 description

A more detailed explanation, where appropriate, about the nature or characteristics of the test.

This is an optional element, and there is zero or one for each test.

4.1.5 Combinations

This element indicates the types of combinations that must be supported in order to properly execute the test. A Combination element contains one or more ***profile*** sub-elements; the value of a profile element is the iri of the profile.

If all of a test's imported files are of the same dialect as the test, i.e. files imported with a single-argument imports directive, then the Combinations element is not present.

This is an optional element, and there are zero or one for each test.

4.1.6 ImportedDocument

This element provides a reference to a document imported by one of the other documents (Premise, Input or Imported) of the test.

ImportedDocument has one ***Normative*** sub-element, which indicates the document (file) that should be used to execute the test, and zero or more ***Presentation*** sub-elements, each of which contains the document in a presentation syntax.

The ***Normative*** element has one attribute and two sub-elements:

- the ***syntax:*** attribute indicates the syntax of the file; possible values are: *RIF/XML* or *RDF/XML*
- the ***name*** element contains the name of the file. In a local environment downloaded from the repository, the file will be in the same directory as the test case manifest
- the ***remote:*** attribute contains a link to the document in the RIF test cases repository
    - **Note**: if the syntax of the remote file is RDF/XML, then the link to the file given in the **remote** element will not include the .rdf suffix part of the file name. The software accessing this file can either append the .rdf suffix to the given uri, or can use an HTTP Accept header of *application/rdf+xml* to request the RDF/XML version of the file.

The ***Presentation*** element has text content and one attribute:

- ***syntax*** indicates the syntax of the content; possible values are: *RIFBLD-PS*, *RIFPRD-PS*, *Turtle*, or *OWL2 Functional Syntax*.

There are zero or more ImportedDocument elements for each test.

4.2 Properties of Import Rejection and Syntax Tests

4.2.1 InputDocument

This element provides a reference to the input document for the test.

InputDocument has one ***Normative*** sub-element, which indicates the document (file) that should be used to execute the test, and zero or more ***Presentation*** sub-elements, each of which contains the document in a presentation syntax.

The ***Normative*** element has one attribute and two sub-elements:

- the ***syntax:*** attribute indicates the syntax of the file; possible values are: *RIF/XML*.
- the ***name*** element contains the name of the file. In a local environment downloaded from the repository, the file will be in the same directory as the test case manifest
- the ***remote:*** attribute contains a link to the document in the RIF test cases repository

The ***Presentation*** element has text content and one attribute:

- ***syntax*** indicates the syntax of the content; possible values are: *RIFBLD-PS* or *RIFPRD-PS*.

There is one InputDocument element for each syntax and import rejection test.

4.3 Properties of Entailment Tests

4.3.1 PremiseDocument

This property provides a reference to the premise document for the test.

PremiseDocument has one ***Normative*** sub-element, which indicates the document (file) that should be used to execute the test, and zero or more ***Presentation*** sub-elements, each of which contains the document in a presentation syntax.

The ***Normative*** element has one attribute and two sub-elements:

- the ***syntax:*** attribute indicates the syntax of the file; possible values are: *RIF/XML*.
- the ***name*** element contains the name of the file. In a local environment downloaded from the repository, the file will be in the same directory as the test case manifest
- the ***remote:*** attribute contains a link to the document in the RIF test cases repository

The ***Presentation*** element has text content and one attribute:

- ***syntax*** indicates the syntax of the content; possible values are: *RIFBLD-PS* or *RIFPRD-PS*.

There is one PremiseDocument for each entailment test.

4.3.2 Properties of Positive Entailment tests

4.3.2.1 ConclusionDocument

This property provides a reference to the conclusion document for the test.

ConclusionDocument has one ***Normative*** sub-element, which indicates the document (file) that should be used to execute the test, and zero or more ***Presentation*** sub-elements, each of which contains the document in a presentation syntax.

The ***Normative*** element has one attribute and two sub-elements:

- the ***syntax:*** attribute indicates the syntax of the file; possible values are: *RIF/XML* or *Turtle*.
- the ***name*** element contains the name of the file. In a local environment downloaded from the repository, the file will be in the same directory as the test case manifest

- the ***remote:*** attribute contains a link to the document in the RIF test cases repository

The ***Presentation*** element has text content and one attribute:

- ***syntax*** indicates the syntax of the content; possible values are: *RIFBLD-PS*, *RIFPRD-PS* or *Turtle*.

There is one ConclusionDocument for each positive entailment test.

4.3.3 Properties of Negative Entailment tests

4.3.3.1 NonConclusionDocument

This property provides a reference to the conclusion document for the test.

NonConclusionDocument has one ***Normative*** sub-element, which indicates the document (file) that should be used to execute the test, and zero or more ***Presentation*** sub-elements, each of which contains the document in a presentation syntax.

The ***Normative*** element has one attribute and two sub-elements:

- the ***syntax:*** attribute indicates the syntax of the file; possible values are: *RIF/XML* or *Turtle*.
- the ***name*** element contains the name of the file. In a local environment downloaded from the repository, the file will be in the same directory as the test case manifest
- the ***remote:*** attribute contains a link to the document in the RIF test cases repository

The ***Presentation*** element has text content and one attribute:

- ***syntax*** indicates the syntax of the content; possible values are: *RIFBLD-PS*, *RIFPRD-PS* or *Turtle*.

There is one NonConclusionDocument for each negative entailment test.

4.4 Sample Test Case

A sample test case is shown below:

```
<?xml version="1.0" encoding="UTF-8"?>
<PositiveEntailmentTest id="RDF_Combination_SubClass_2"
       src="http://www.w3.org/2005/rules/test/repository/tc/RDF_Combination
       xmlns="http://www.w3.org/2009/10/rif-test#">
    <status>Approved</status>
    <dialect>Core</dialect>
    <dialect>BLD</dialect>
    <dialect>PRD</dialect>
    <Combinations>
        <profile>http://www.w3.org/2007/rif-import-profile#RDFS</profile>
    </Combinations>
    <purpose>Test interaction between rdfs:subClassOf, rdf:type, ## and # i
    <description>
       In RIF-RDF combinations, we have that rdf:type statements are equiva
       statements and RIF ## statements imply rdfs:subClassOf statements. B
       we have that ex:a rdf:type ex:D must hold and by the semantics of co
       ex:a rdf:type ex:D implies ex:a # ex:D. Therefore, ex:a # ex:D is de
    </description>
    <ImportedDocument>
        <Normative syntax="NTriples">
            <name>RDF_Combination_SubClass_2-import001</name>
            <remote>
              http://www.w3.org/2005/rules/test/repository/tc/RDF_Combinati
            </remote>
        </Normative>
        <Presentation syntax="NTriples"><![CDATA[
@prefix ex: <http://example.org/#> .
@prefix rdfs: <http://www.w3.org/2000/01/rdf-schema#> .
@prefix rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#> .

 ex:a rdf:type ex:C .
 ex:C rdfs:subClassOf ex:D .
]]>     </Presentation>
    </ImportedDocument>
    <PremiseDocument>
        <Normative syntax="RIF/XML">
            <name>RDF_Combination_SubClass_2-premise.rif</name>
            <remote>
              http://www.w3.org/2005/rules/test/repository/tc/RDF_Combinati
            </remote>
        </Normative>
        <Presentation syntax="RIFBLD-PS"><![CDATA[
Document(
    Import(<http://example.org/mygraph> <http://www.w3.org/2007/rif-import-
 )
]]>     </Presentation>
    </PremiseDocument>
    <ConclusionDocument>
        <Normative syntax="RIF/XML">
            <name>RDF_Combination_SubClass_2-conclusion.rif</name>
            <remote>
              http://www.w3.org/2005/rules/test/repository/tc/RDF_Combinati
            </remote>
        </Normative>
        <Presentation syntax="RIFBLD-PS"><![CDATA[
ex:a # ex:D
]]></Presentation>
    </ConclusionDocument>
</PositiveEntailmentTest>
```

5 Test Case Repository

The test cases are available from the Test Case Repository where they are each in an individual sub-directory named for the test case. The test cases are also available in dialect test suites as zip files. These zip files expand into a directory structure organized by status and test type. There is also a manifest file for each dialect(CoreTests.xml, BLDTests.xml, PRDTests.xml) that describes all the tests applicable to the dialect.

6 Running the Test Cases

The test suite is intended for use on a variety of rule systems, each with its own API. Therefore, the tasks of providing input to the rule engine in a system-dependent manner and of checking that the results are correct are left to the tester. The manifest contains a machine-readable description of the test cases, and is intended to enable automated processing of the test cases.

Users are expected to write their own test harnesses to carry out the following tasks:

- Reading test cases from the repository, and selecting the ones applicable to the implementation
- Applying customization if applicable (see below)
- Translating the RIF/XML in which the test case is written into the language of the system that is being tested
- Executing the tests, using the source files specified in the manifest
- Determining if the correct result has been achieved. (For example, for the particular case of Positive [Negative] Entailment tests, this means determining that the conclusion is [is not] entailed by the premises, as discussed above.)
- Producing a description of the test results that can be sent to the working group

The dialect and Combinations elements in the manifest file can be used to filter for appropriate tests. For example, a RIF-BLD implemention should select only tests that have a ***dialect*** property with a value of *BLD*, and implementations that don't support RIF-OWL combinations should skip tests that have a ***Combinations*** element containing ***profile*** sub-elements with a value of one of the OWL profiles.

6.1 Test Case Customization

Testers have the option of making certain changes to the test cases in order to facilitate use of the test suite. All changes made to the original test suite should be documented in free-text form as part of the results submission. Acceptable changes are described below.

6.1.1 Customizing Import Directives

For tests with input documents that contain import directives, the location of the imported document specified in the directive will be at the W3C website, and the imported document will be made available at that location. Test cases will also include all imported documents as part of their data (in the zip file), pointed to by elements in the manifest. Users who wish to access these files in their own environment can change the import directives in the input documents to reflect the updated location.

6.2 Reporting Test Results

Implementers are encouraged to report their test results by e-mail with a subject of "RIF Test Results" to public-rif-comments@w3.org (which has a public archive); the results will be summarized on the RIF Test Results page. This information will help to determine when the RIF specifications are ready to become W3C Recommendations.

The results should be formatted in XML according to rif-test-report.xsd. The requested information is described below, and an example test report is given at the end of this section.

6.2.1 Test Report fields

6.2.1.1 Implementation

There is one *Implemention* element per report. This information identifies the software that was tested.

- ***name*** (required) - short name of the implementation, suitable for display in a table
- ***version*** (required) - version of the software tested.
- ***website*** (optional) - website of the implementation software.
- ***description*** (optional) - XHTML description of the implementation software. This description will be included inline in the test results summary.
- ***organization name*** (required) - name of the organization that is responsible for the implementation.

- ***organization website*** (optional) - website of the organization that is responsible for the implementation.
- ***submitter name*** (required) - name of the person submitting the test results.
- ***submitter email*** (required) - email of the person submitting the test results.
- ***submitter website*** (optional) - website of the person submitting the test results.

6.2.1.2 Test Run

There is one *TestRun* element per report.

- ***Test Suite version*** (required) - The version of the RIF test suite that was used. The version can be found in the version.txt file in the RIF Test Case repository.
- ***platform*** (optional) - description of Hardware and OS used for test execution.
- ***comments*** (optional) - XHTML comments about the test run.
- ***date*** (required) - The date that the tests were performed, in XML Schema date format.

6.2.1.3 Test Result

There is one or more *TestResult* elements per report, one for each test case executed. If a test case is found to be in error, don't submit a test result for it, but rather report the bug as described the Reporting Problems section

- ***testId*** (required) - Test Case identifier. The identifier is given in the *id* attribute of the root element of the testcase manifest file.
- ***status*** (required) - The outcome of the test. This value must be one of an enumerated list:
    - ***pass*** - The test case passed according to the definition given in the Test Types section.
    - ***passSyntax*** - The premise of a Positive Entailment or Negative Entailment test was used as a Positive Syntax test, and the test passed according to the definition given for Positive Syntax tests.
    - ***fail*** - The test case failed according to the definition given in the Test Types section.
    - ***undecided*** - This value is applicable only to Entailment Tests and indicates that a passing result was not conclusively demonstrated.
    - ***notSupported*** - The features required to successfully execute this test case are not (or not yet) supported by the implementation.
- ***comments*** (optional) - XHTML comments about the test case or test result.
- ***msec*** (optional) - The number of milliseconds taken by the process to compute the test result.

- ***kbytes*** (optional) - Memory used by the process in computing the test result.

6.2.2 Example Test Report

An example test report is shown below:

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

<TestReport xmlns="http://www.w3.org/2009/10/rif-test-results#">
   <Implementation>
      <name>Riftia Pachyptila</name>
      <version>4.8</version>
      <website>http://www.example.com/gc/rp</website>
      <description>
         <span xmlns="http://www.w3.org/1999/xhtml"><em>Riftia</em> is part
         of the <a href="http://www.example.com/gc/rp">GC</a>
         Smart Computing© Suite.</span>
      </description>

      <Organization>
        <name>General Competencies, Inc</name>
        <website>http://www.example.com/gc</website>
      </Organization>

      <Submitter>
        <name>John Duguid</name>
        <email>jd@gc.com</email>
        <website>http://www.example.com/gc/people/jd</website> <!-- optiona
      </Submitter>
   </Implementation>

   <TestRun date="2009-10-27">
      <TestSuite>
        <version>1.0</version>
      </TestSuite>
      <platform>optional description of Hardware and OS being used.</platfo
      <comments>optional comments</comments>
   </TestRun>

   <TestResult>
      <testId>Core_Safeness</testId>
      <status>pass</status>
   </TestResult>

   <TestResult  msec="333">
      <testId>Equality_in_conclusion_3</testId>
      <status>pass</status>
      <comments>this was hard</comments>
   </TestResult>

   <TestResult msec="10000" kbytes="220">
      <testId>NestedListsAreNotFlatLists</testId>
      <status>undecided</status>
   </TestResult>
</TestReport>
```

6.3 Reporting Problems

If you believe that a test case is incorrect please send a bug report to public-rif-comments@w3.org. Please be specific about the problem you are reporting. Clearly identify the test case or test cases that are problematic and include the repository version, which can be found in the version.txt file in the RIF Test Case repository. If you believe that the expected result is wrong, then tell us what you believe the expected result should be. Please cite portions of one or more of the RIF specifications to support your position whenever possible.

7 References

7.1 Normative References

[REPOSITORY]
Repository for RIF Test Cases, http://www.w3.org/2005/rules/test/repository

[RFC-3987]
RFC 3987: Internationalized Resource Identifiers (IRIs). M. Duerst, M. Suignard. IETF, January 2005, http://www.ietf.org/rfc/rfc3987.txt.

[RIF-BLD]
RIF Basic Logic Dialect Harold Boley, Michael Kifer, eds. W3C Proposed Recommendation, 11 May 2010, http://www.w3.org/TR/2010/PR-rif-bld-20100511/. Latest version available at http://www.w3.org/TR/rif-bld/.

[RIF-Core]
RIF Core Dialect Harold Boley, Gary Hallmark, Michael Kifer, Adrian Paschke, Axel Polleres, Dave Reynolds, eds. W3C Proposed Recommendation, 11 May 2010, http://www.w3.org/TR/2010/PR-rif-core-20100511/. Latest version available at http://www.w3.org/TR/rif-core/.

[RIF-DTB]
RIF Datatypes and Built-Ins 1.0 Axel Polleres, Harold Boley, Michael Kifer, eds. W3C Proposed Recommendation, 11 May 2010, http://www.w3.org/TR/2010/PR-rif-dtb-20100511/. Latest version available at http://www.w3.org/TR/rif-dtb/.

[RIF-RDF+OWL]
[*RIF RDF and OWL Compatibility*](http://www.w3.org/TR/2010/PR-rif-rdf-owl-20100511/) Jos de Bruijn, editor. W3C Proposed Recommendation, 11 May 2010, http://www.w3.org/TR/2010/PR-rif-rdf-owl-20100511/. Latest version available at http://www.w3.org/TR/rif-rdf-owl/.

[RIF-PRD]
[*RIF Production Rule Dialect*](http://www.w3.org/TR/2010/PR-rif-prd-20100511/) Christian de Sainte Marie, Gary Hallmark, Adrian Paschke, eds. W3C Proposed Recommendation, 11 May 2010, http://www.w3.org/TR/2010/PR-rif-prd-20100511/. Latest version available at http://www.w3.org/TR/rif-prd/.

7.2 Informational References

[OWL Tests]
OWL Web Ontology Language Test Cases, Jeremy J. Carroll and Jos De Roo, Editors, W3C Recommendation, 10 February 2004, http://www.w3.org/TR/2004/REC-owl-test-20040210/ . Latest version available at http://www.w3.org/TR/owl-test/ .

[QA Framework]
QA Framework: Specification Guidelines, Karl Dubost, Lynne Rosenthal, Dominique Hazael-Massieux, and Lofton Henderson eds. W3C Recommendation 17 August 2005, http://www.w3.org/TR/2005/REC-qaframe-spec-20050817/ .

[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/

[RIF-CHARTER]
Rule Interchange Format Working Group Charter, Sandro Hawke, ed. W3C document, 17 November 2005, http://www.w3.org/2005/rules/wg/charter.html .

[RIF-FLD]
[*RIF Framework for Logic Dialects*](http://www.w3.org/TR/2010/PR-rif-fld-20100511/) Harold Boley, Michael Kifer, eds. W3C Proposed Recommendation, 11 May 2010, http://www.w3.org/TR/2010/PR-rif-fld-20100511/. Latest version available at http://www.w3.org/TR/rif-fld/.

[Test Metadata]
Test Metadata, Edited by Patrick Curran (Sun Microsystems) and Karl Dubost (W3C). Produced by members of the the W3C Quality Assurance Working Group, W3C Working Group Note. 14-September-2005, http://www.w3.org/TR/test-metadata/.

8 Appendix: Schema for RIF Test Case Manifest Files

The schema is available online as rif-testcase.xsd in the [RIF Test Case Repository], and is reproduced below.

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

<xs:schema
  xmlns:xs="http://www.w3.org/2001/XMLSchema"
  xmlns="http://www.w3.org/2009/10/rif-test#"
  targetNamespace="http://www.w3.org/2009/10/rif-test#"
  elementFormDefault="qualified"
  version="1.0">

  <xs:group name="TestCase">
    <xs:choice>
      <xs:element ref="PositiveEntailmentTest"/>
      <xs:element ref="NegativeEntailmentTest"/>
      <xs:element ref="PositiveSyntaxTest"/>
      <xs:element ref="NegativeSyntaxTest"/>
      <xs:element ref="ImportRejectionTest"/>
    </xs:choice>
  </xs:group>


<!-- Properties common to all test cases -->

  <xs:group name="commonTestCaseInfo">
    <xs:sequence>
      <xs:element ref="status"/>
      <xs:element ref="dialect" maxOccurs="unbounded"/>
      <xs:element ref="Combinations" minOccurs="0"/>
      <xs:element ref="purpose"/>
      <xs:element ref="description" minOccurs="0"/>
      <xs:element name="ImportedDocument" type="documentType" minOccurs="0"
    </xs:sequence>
  </xs:group>

<!-- The different test case types -->

 <xs:element name="PositiveEntailmentTest">
  <xs:complexType>
    <xs:sequence>
      <xs:group ref="commonTestCaseInfo"/>
      <xs:element name="PremiseDocument" type="documentType"/>
      <xs:element name="ConclusionDocument" type="documentType"/>
    </xs:sequence>
```

```
   <xs:attribute name="id" type="xs:string" use="required"/>
   <xs:attribute name="src" type="xs:anyURI" use="required"/>
  </xs:complexType>
</xs:element>

<xs:element name="NegativeEntailmentTest">
 <xs:complexType>
   <xs:sequence>
     <xs:group ref="commonTestCaseInfo"/>
     <xs:element name="PremiseDocument" type="documentType"/>
     <xs:element name="NonConclusionDocument" type="documentType"/>
   </xs:sequence>
   <xs:attribute name="id" type="xs:string" use="required"/>
   <xs:attribute name="src" type="xs:anyURI" use="required"/>
  </xs:complexType>
</xs:element>

<xs:element name="PositiveSyntaxTest" type="syntaxTest"/>
<xs:element name="NegativeSyntaxTest" type="syntaxTest"/>
<xs:element name="ImportRejectionTest" type="syntaxTest"/>

<xs:complexType name="syntaxTest">
  <xs:sequence>
    <xs:group ref="commonTestCaseInfo"/>
    <xs:element name="InputDocument" type="documentType"/>
  </xs:sequence>
   <xs:attribute name="id" type="xs:string" use="required"/>
   <xs:attribute name="src" type="xs:anyURI" use="required"/>
</xs:complexType>

<!-- other element definitions -->

<xs:element name="Combinations">
  <xs:complexType>
    <xs:sequence>
       <xs:element name="profile" type="xs:anyURI" maxOccurs="unbounded"/>
    </xs:sequence>
  </xs:complexType>
</xs:element>

<xs:element name="status" type="statusType"/>
<xs:element name="dialect" type="dialectType"/>
<xs:element name="purpose" type="descriptionType"/>
<xs:element name="description" type="descriptionType"/>

<!-- other type definitions -->

<xs:complexType name="documentType">
```

```
    <xs:sequence>
      <xs:element name="Normative">
        <xs:complexType>
           <xs:sequence>
             <xs:element name="name" type="xs:string"/>
             <xs:element name="remote" type="xs:anyURI" maxOccurs="unbounde
           </xs:sequence>
           <xs:attribute name="syntax" type="nSyntaxType"/>
        </xs:complexType>
      </xs:element>
      <xs:element name="Presentation" minOccurs="0" maxOccurs="unbounded">
         <xs:complexType>
             <xs:simpleContent>
               <xs:extension base="xs:string">
                 <xs:attribute name="syntax" type="pSyntaxType" use="requir
               </xs:extension>
             </xs:simpleContent>
         </xs:complexType>
      </xs:element>
    </xs:sequence>
 </xs:complexType>

 <xs:simpleType name="statusType">
    <xs:restriction base="xs:string">
       <xs:enumeration value="Approved" />
       <xs:enumeration value="Proposed" />
    </xs:restriction>
 </xs:simpleType>

 <xs:simpleType name="dialectType">
    <xs:restriction base="xs:string">
       <xs:enumeration value="Core" />
       <xs:enumeration value="BLD" />
       <xs:enumeration value="PRD" />
    </xs:restriction>
 </xs:simpleType>

<xs:simpleType name="nSyntaxType">
    <xs:restriction base="xs:string">
       <xs:enumeration value="RIF/XML" />
       <xs:enumeration value="RDF/XML" />
       <xs:enumeration value="NTriples" />
       <xs:enumeration value="Turtle" />
    </xs:restriction>
</xs:simpleType>

<xs:simpleType name="pSyntaxType">
    <xs:restriction base="xs:string">
```

```
       <xs:enumeration value="RIFBLD-PS" />
       <xs:enumeration value="RIFPRD-PS" />
       <xs:enumeration value="NTriples" />
       <xs:enumeration value="Turtle" />
       <xs:enumeration value="OWL2 Functional Syntax" />
    </xs:restriction>
</xs:simpleType>

 <xs:complexType name="descriptionType" mixed="true">
    <xs:sequence>
      <xs:any namespace="http://www.w3.org/1999/xhtml"
      minOccurs="0" maxOccurs="unbounded"
      processContents="skip"/>
    </xs:sequence>
 </xs:complexType>


</xs:schema>
```

9 Appendix: Change Log (Informative)

This appendix summarizes the main changes to this document since the draft of 18 December, 2008.

- A new test type, Import Rejection, was added.
- A new test case status, Obsolete, was added.
- Documentation of test cases for the Production Rule Dialect (PRD) and the Core dialect (Core) was added.
- The statement saying that, in practice, for negative entailment tests, a conformant RIF system may not be able to conclusively demonstrate that the conclusion cannot be drawn from the premises was removed.
- Some typographical and editorial changes were made