



RIF Test Cases

W3C Editor's Draft 18 December 2008

This version:

<http://www.w3.org/2005/rules/wg/draft/ED-rif-test-20081218/>

Latest editor's draft:

<http://www.w3.org/2005/rules/wg/draft/rif-test/>

Previous version:

<http://www.w3.org/2005/rules/wg/draft/ED-rif-test-20081125/> ([color-coded diff](#))

Editors:

Stella Mitchell, IBM

[Leora Morgenstern](#), IBM

Adrian Paschke, Free University Berlin

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 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May Be Superseded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 5 documents:

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

First Public Working Draft

This First Public Working Draft conveys the current state of the Working Group's evolving approach to developing test cases for RIF dialects and testing procedures which can be used by RIF system implementors. The actual tests are not part of this document but are maintained on a Wiki and then can be exported to other (still-evolving) formats for download or over-the-Web use. The group expects to publish relatively stable, approved test suites for its dialects when issuing its [Call for Implementations](#) in the coming months.

Please Comment By 23 January 2009

The [Rule Interchange Format \(RIF\) Working Group](#) seeks public feedback on this First Public 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 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 Introduction](#)
 - [1.1 Scope](#)
 - [1.2 Conformance](#)
- [2 Deliverables](#)
- [3 Test Types](#)
 - [3.1 Syntactic Tests](#)
 - [3.1.1 Positive Syntax Tests](#)
 - [3.1.2 Negative Syntax Tests](#)
 - [3.2 Semantic Tests](#)
 - [3.2.1 Positive Entailment Tests](#)
 - [3.2.2 Negative Entailment Tests](#)
- [4 Test Case Format](#)
 - [4.1 Properties of All Tests](#)
 - [4.1.1 rifTest:identifier](#)
 - [4.1.2 rifTest:status](#)
 - [4.1.3 rifTest:dialect](#)
 - [4.1.4 rifTest:purpose](#)
 - [4.1.5 dc:description](#)
 - [4.1.6 rifTest:specRef](#)
 - [4.1.7 rifTest:seeAlso](#)
 - [4.1.8 dc:contributor](#)
 - [4.2 Properties of Syntactic Tests](#)
 - [4.2.1 rifTest:inputDocument](#)
 - [4.3 Properties of Entailment Tests](#)
 - [4.3.1 rifTest:premisesDocument](#)
 - [4.3.2 rifTest:conclusionDocument](#)
 - [4.3.3 rifTest:importedDocument](#)
 - [4.3.4 rifTest:importSupport](#)
- [5 Test Case Repository](#)
- [6 Running the Test Cases](#)
 - [6.1 Test Case Customization](#)
 - [6.1.1 Customizing Import Directives](#)
 - [6.2 Reporting Test Results](#)
 - [6.3 Reporting Problems](#)
- [7 References](#)
 - [7.1 Normative References](#)
 - [7.2 Informational References](#)
- [8 Appendix: Schema for RIF Test Case Manifest Files](#)

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 in 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.

Editor's Note: We may set up a publicly accessible collection site.

1.1 Scope

This first Working Draft documents test cases for the [Basic Logic Dialect \(BLD\)](#) of RIF, and for BLD-RDF and BLD-OWL combinations as specified in [RIF RDF and OWL Compatibility](#). Future drafts of this document will cover test cases for other RIF dialects as those dialects are finalized. The current set of tests focuses on testing entailment and syntax, as described in the [Test Types](#) section. Additional types of tests may be added in the future.

The BLD test suite contains tests for RIF consumers as defined in the [RIF-BLD Conformance Clauses](#), but not for RIF-BLD-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 [RIF-BLD XML Schema](#). For implementations that are both producers and consumers, some validation of the producer can be achieved by translating entailment tests from RIF to the target rule language, then back to RIF, then again to the target rule language, and then performing 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-BLD 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](#).) Conformant RIF-BLD documents, consumers and producers are defined in the [conformance clauses](#) section of the RIF BLD specification. 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 indicates conformance. 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 a complete conformance test suite. 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 test suite, that includes Approved 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 \[RIF-BLD\]](#) versions of the RIF input documents, as a human readable version of the test.
- A Manifest file for each test suite, describing all the tests in the suite, and containing all the data needed to execute the tests.
- 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 processor's recognition of a [conformant BLD document in XML syntax](#) [RIF-BLD]. A RIF processor can use XML Schema validation against the [RIF-BLD Schemas](#) to verify that input documents are [valid BLD documents](#), and so most of the syntax tests included in the test suite address syntax requirements that are not enforced by the XML schema.

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 conformant RIF document in the dialect indicated by the [dialect](#) element.

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

If the value of the dialect property is *rifTest:BLD* then the test document is a [conformant BLD document XML syntax](#).

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

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 conformant RIF document in the dialect indicated by the [dialect](#) element.

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

If the value of the dialect property is *rifTest:BLD* then the test document is not a [conformant RIF-BLD document](#)

3.2 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

Premise
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 complete specification of these tests is given in the sub-sections below.

Editor's Note: In the future, the test suite may include tests for $BLD_{T,E}$ consumers, which include datatypes and external terms not included in [RIF-DTB](#)

3.2.1 Positive Entailment Tests

Each test of this type includes a premises document, indicated by the [premisesDocument](#) 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.

The applicable entailment regimes are indicated by [dialect](#) elements in the manifest file. For example, if there is a dialect property with a value of "rifTest:BLD" then the entailment holds according to the [RIF-BLD definition of entailment](#).

In addition, optional [importSupport](#) elements in the manifest indicate the types of documents in the imports closure of the premises document. If an importSupport property is present with the value *rifTest:RDF* then the test constitutes a RIF-RDF combination as defined in [RIF RDF and OWL compatibility](#) and there are additional [rules of entailment](#) in effect. If an importSupport property is present with the value *rifTest:OWL-DL* or *rifTest:OWL-Full* then the test constitutes a RIF-OWL combination and the associated entailment rules are defined in [OWL compatibility](#).

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 unknown, and must not report that the conclusion is not entailed by the premises.

3.2.2 Negative Entailment Tests

Each test of this type includes a premises document, indicated by the [premisesDocument](#) 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.

The applicable entailment regimes are indicated by [dialect](#) elements in the manifest file. For example, 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](#).

In addition, optional [importSupport](#) elements in the manifest indicate the types of documents in the imports closure of the premises document. If an importSupport property is present with the value "rifTest:RDF" then the test constitutes a RIF-RDF combination as defined in [RIF RDF and OWL compatibility](#) and there are additional [rules of entailment](#) in effect. If an importSupport property is present with the value "rifTest:OWL-DL" or "rifTest:OWL-Full" then the test constitutes a RIF-OWL combination and the associated entailment rules are defined in [OWL compatibility](#).

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

Note that while ideally the RIF consumer would be able to conclusively demonstrate that the conclusion cannot be drawn from the premises, in practice a failure to draw the conclusion after a thorough attempt to do so can be considered a successful outcome.

4 Test Case Format

Editor's Note: The test case format is under discussion and is likely to change

Each test case has an associated manifest file that contains information needed to properly execute the test. This file is in a machine-readable format, RDF/XML, in order to enable the development of automated testing frameworks. The manifest file also contains information, such as contributor and status, used for test case management. Some examples of how the data in the manifest can be used are:

- selecting tests according to a variety of criteria, such as what dialect they apply to
- locating the documents that participate in the test
- identifying sections of specifications that are relevant to the test
- tracking the tests during the development and review process

The format of the RIF test case manifests follows the general guidelines of the W3C QA Working Group's [\[Test Metadata Note\]](#). A schema for the test case format is given in [Appendix 8](#). This section presents a summary of the property elements, organized by the [type of test](#) to which they apply.

In the following, dc is bound to <http://purl.org/dc/elements/1.1/>, and rifTest is bound to <http://www.w3.org/2005/rules/test/rifTestSchema#>.

4.1 Properties of All Tests

4.1.1 `rifTest:identifier`

The value of this property is a unique identifier for the test case. It conforms to *irrelative-ref* as defined in [\[RFC-3987\]](#) so that it can be appended to a base to generate a URI for the test. Values of this property are of type `xs:string`.

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

4.1.2 `rifTest: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: `rifTest:Proposed`, `rifTest:Accepted`, or `rifTest:Rejected`.

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

4.1.3 `rifTest: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 `rifTest:BLD`.

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

4.1.4 `rifTest:purpose`

The value of this property is a brief explanation of the purpose of the test, suitable for display in tabular format. Values of this property are of type `xs:string`.

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

4.1.5 `dc: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.

4.1.6 `rifTest:specRef`

a URI reference to section(s) of a RIF specification(s) relevant to this test. Values of this property are of type `rdfs:Literal`.

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

4.1.7 `rifTest:seeAlso`

a URL reference to material related to this test. This property is used to indicate other RIF test cases, issues from the RIF issues list, or in general other documents that provide useful background information for the test. Values of this property are of type `rdfs:Literal`.

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

4.1.8 `dc:contributor`

This property indicates the individual that contributed the test. This is a required element, and there are one or more for each test.

4.2 Properties of Syntactic Tests

4.2.1 `rifTest:inputDocument`

This property provides a reference to the input document for the test. Subproperties of `:inputDocument` are used to reference the input document in different syntaxes.

There is one `inputDocument` for each test.

4.3 Properties of Entailment Tests

4.3.1 `rifTest:premisesDocument`

This property provides a reference to the premises document for the test. Subproperties of `:premisesDocument` are used to reference the premises document in different syntaxes.

There is one premises document for each test.

4.3.2 `rifTest:conclusionDocument`

This property provides a reference to the conclusion document for the test. Subproperties of `:conclusionDocument` are used to reference the conclusion document in different syntaxes.

There is one conclusion document for each test.

4.3.3 `rifTest:importedDocument`

This property provides a reference to a document imported by one of the other documents of the test. Subproperties of `:importedDocument` are used to provide importedDocuments in different syntaxes.

There can be zero or more imported documents for each test.

4.3.4 `rifTest:importSupport`

This property is used to indicate what type of imported documents this test includes. This can be used in filtering for applicable tests. If all the imported files are of the same type as the dialect of the test, this element is not present. The value of this property is one of an enumerated list: `rifTest:RDF`, `rifTest:OWL-DL`, `rifTest:OWL-Full`

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

5 Test Case Repository

Test suites can be downloaded from the [[Test Case Repository](#)] as zip files that contain all the approved tests for a RIF dialect. Each zip file expands into subdirectories, one for each test, containing the test input files and individual manifest files. In addition, each test suite is available as a single RDF file containing all the tests for the suite, each in the format described in [Test Case Format](#). Test cases are also available individually in the repository, organized into directories by 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 catalog, 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 *rifTest:dialect* and *rifTest:importSupport* elements in the manifest file can be used to filter for appropriate tests. In the current set of test cases, all tests are in the *rifTest:BLD* dialect. Implementations that do not support BLD-RDF combinations should skip tests that have an *:importSupport* property with a value of *:RDF*. Implementations that do not support BLD-OWL combinations should skip tests that have an *:importSupport* property with a value of *:OWL-DL* or *:OWL-Full*.

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 send their test results to public-rif-comments@w3.org.

Editor's Note: We are working on defining a structured format for reporting test results.

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. 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](#) hosted at the W3C.

[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 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, 30 July 2008, <http://www.w3.org/TR/2008/WD-rif-dtb-20080730/>. Latest version available at <http://www.w3.org/TR/rif-dtb/>.

[RIF-RDF+OWL]

RIF RDF and OWL Compatibility, Jos de Bruijn, eds. 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/>.

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, 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/>.

[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 for the test case manifest files is under development as [rifTestSchema.rdf](#) in the RIF Test Cases Repository.