# RDF Dataset Canonicalization and Hash 1.0 Processor Conformance

## EARL results from the RDF Dataset Canonicalization and Hash 1.0 Test Suite

## 11 March 2025

**Editor:**
Gregg Kellogg

This document is also available in these non-normative formats: JSON-LD .

## Abstract

This document report test subject conformance for and related specifications for RDF Dataset Canonicalization and Hash 1.0 Test Suite according to the requirements of the Evaluation and Report Language (EARL) 1.0 Schema [EARL10-SCHEMA].

This report is also available in alternate formats: Turtle and JSON-LD

## Status of This Document

This document is merely a W3C-internal document. It has no official standing of any kind and does not represent consensus of the W3C Membership.

This report describes the state of implementation conformance at the time of publication.

## Table of Contents

## Introduction

This implementation report covers the implementations of the RDF Dataset Canonicalization and Hash 1.0 specifications which have submitted test results. It is intended to be maintained by the RDF Dataset Canonicalization and Hash Working Group. The implementation report serves two purposes:

1. To demonstrate that there are multiple, independent implementations of the specifications. This is a prerequisite for progression of any standard to Recommendation.

2. To catalog the known, conforming implementations and which features each supports.

There may be other implementations which are not listed in this report, either due to not submitting tests or by not being intended as direct implementations of the specification, but instead layering on top of such libraries.

## Instructions for submitting implementation reports

Reports should be submitted in Turtle format to Public RCH WG or via a Pull Request to the w3c/rdf-canon.

Tests should be run using the test manifests defined in the Test Manifests Section.

Include an `earl:Assertion` for each test, referencing the test resource from the associated manifest and the test subject being reported upon. See the example below:

```
[ a earl:Assertion;
  earl:assertedBy <--your-developer-identifier-->;
  earl:subject <--your-software-identifier-->;
  earl:test <--uri-of-test-from-manifest>;
  earl:result [
    a earl:TestResult;
    earl:outcome earl:passed;
    dc:date "2023-01-25T10:18:04-08:00"^^xsd:dateTime];
  earl:mode earl:automatic ] .
```

The Test Subject should be defined as a `doap:Project`, including the name, homepage and developer(s) of the software (see DOAP). Optionally, including the project description and programming language. An example test subject description is the following:

```
<> foaf:primaryTopic <--your-software-identifier--> ;
  dc:issued "2016-12-26T10:18:04-08:00"^^xsd:dateTime ;
  foaf:maker <--your-developer-identifier--> .

<--your-software-identifier--> a doap:Project, earl:TestSubject, earl:Software ;
  doap:name          "My Cool RDF Canonicalizer" ;
  doap:release [
    doap:name     "--short name wih version number--";
    doap:revision "--Software version number--" ;
    doap:created  "2020-02-19"^^xsd:date;
  ] ;
  doap:developer     <--your-developer-identifier--> ;
  doap:homepage      <--your-software-homepage--> ;
  doap:description   "--your-project-description--"@en ;
  doap:programming-language "--your-implementation-language--" .
```

The software developer, either an organization or one or more individuals SHOULD be referenced from `doap:developer` using FOAF. For example:

```
<--your-developer-identifier--> a foaf:Person, earl:Assertor;
  foaf:name "--My Name--";
  foaf:homepage <--my homepage--> .
```

## 3. Test Manifests

### 3.1 RDF Dataset Canonicalization (RDFC-1.0) Test Suite

Tests the 1.0 version of RDF Dataset Canonicalization and the generation of canonical maps.

| Test | Corese (Java) | rdf-canonicalization-cpp (C++) | rdf-canonize (JavaScript) | Titanium RDFC (Java) | Sophia (Rust) | RDF-URDNA (Java) | zkp-ld/rdf-canon (Rust) | RDF.ex (Elixir) | rdfjs-c14n (TypeScript) | Ruby RDF::Normalize (Ruby) |
|---|---|---|---|---|---|---|---|---|---|---|
| Test test001c: simple id | PASS | PASS | PASS | PASS | PASS | PASS | PASS | PASS | PASS | PASS |
| Test test002c: duplicate property iri values | PASS | PASS | PASS | PASS | PASS | PASS | PASS | PASS | PASS | PASS |
| Test test003c: bnode | PASS | PASS | PASS | PASS | PASS | PASS | PASS | PASS | PASS | PASS |
| Test test003m: bnode (map test) | PASS | PASS | PASS | PASS | PASS | PASS | PASS | PASS | PASS | PASS |
| Test test004c: bnode plus embed w/subject | PASS | PASS | PASS | PASS | PASS | PASS | PASS | PASS | PASS | PASS |
| Test test004m: bnode plus embed | PASS | PASS | PASS | PASS | PASS | PASS | PASS | PASS | PASS | PASS |

| | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|
| w/subject (map test) | | | | | | | | | | |
| Test test005c: bnode embed | PASS | PASS | PASS | PASS | PASS | PASS | PASS | PASS | PASS | PASS |
| Test test005m: bnode embed (map test) | PASS | PASS | PASS | PASS | PASS | PASS | PASS | PASS | PASS | PASS |
| Test test006c: multiple rdf types | PASS | PASS | PASS | PASS | PASS | PASS | PASS | PASS | PASS | PASS |
| Test test008c: single subject complex | PASS | PASS | PASS | PASS | PASS | PASS | PASS | PASS | PASS | PASS |
| Test test009c: multiple subjects - complex | PASS | PASS | PASS | PASS | PASS | PASS | PASS | PASS | PASS | PASS |
| Test test010c: type | PASS | PASS | PASS | PASS | PASS | PASS | PASS | PASS | PASS | PASS |
| Test test011c: type-coerced type | PASS | PASS | PASS | PASS | PASS | PASS | PASS | PASS | PASS | PASS |
| Test test013c: type-coerced type, cycle | PASS | PASS | PASS | PASS | PASS | PASS | PASS | PASS | PASS | PASS |
| Test test014c: check types | PASS | PASS | PASS | PASS | PASS | PASS | PASS | PASS | PASS | PASS |
| Test test016c: blank node - dual link - embed | PASS | PASS | PASS | PASS | PASS | PASS | PASS | PASS | PASS | PASS |
| Test test016m: blank node - dual link - embed (map test) | PASS | PASS | PASS | PASS | PASS | PASS | PASS | PASS | PASS | PASS |
| Test test017c: blank node - dual link - non-embed | PASS | PASS | PASS | PASS | PASS | PASS | PASS | PASS | PASS | PASS |
| Test test017m: blank node - dual link - non-embed (map test) | PASS | PASS | PASS | PASS | PASS | PASS | PASS | PASS | PASS | PASS |
| Test test018c: blank node - self link | PASS | PASS | PASS | PASS | PASS | PASS | PASS | PASS | PASS | PASS |
| Test test018m: blank node | PASS | PASS | PASS | PASS | PASS | PASS | PASS | PASS | PASS | PASS |

| | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|
| - self link (map test) | | | | | | | | | | |
| Test test019c: blank node - disjoint self links | PASS | PASS | PASS | PASS | PASS | PASS | PASS | PASS | PASS | PASS |
| Test test020c: blank node - diamond | PASS | PASS | PASS | PASS | PASS | PASS | PASS | PASS | PASS | PASS |
| Test test020m: blank node - diamond (map test) | PASS | PASS | PASS | PASS | PASS | PASS | PASS | PASS | PASS | PASS |
| Test test021c: blank node - circle of 2 | PASS | PASS | PASS | PASS | PASS | PASS | PASS | PASS | PASS | PASS |
| Test test022c: blank node - double circle of 2 | PASS | PASS | PASS | PASS | PASS | PASS | PASS | PASS | PASS | PASS |
| Test test023c: blank node - circle of 3 | PASS | PASS | PASS | PASS | PASS | PASS | PASS | PASS | PASS | PASS |
| Test test024c: blank node - double circle of 3 (0-1-2) | PASS | PASS | PASS | PASS | PASS | PASS | PASS | PASS | PASS | PASS |
| Test test025c: blank node - double circle of 3 (0-2-1) | PASS | PASS | PASS | PASS | PASS | PASS | PASS | PASS | PASS | PASS |
| Test test026c: blank node - double circle of 3 (1-0-2) | PASS | PASS | PASS | PASS | PASS | PASS | PASS | PASS | PASS | PASS |
| Test test027c: blank node - double circle of 3 (1-2-0) | PASS | PASS | PASS | PASS | PASS | PASS | PASS | PASS | PASS | PASS |
| Test test028c: blank node - double circle of 3 (2-1-0) | PASS | PASS | PASS | PASS | PASS | PASS | PASS | PASS | PASS | PASS |
| Test test029c: blank node - double circle of 3 (2-0-1) | PASS | PASS | PASS | PASS | PASS | PASS | PASS | PASS | PASS | PASS |
| Test test030c: blank node - point at circle of 3 | PASS | PASS | PASS | PASS | PASS | PASS | PASS | PASS | PASS | PASS |

| | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|
| Test test030m: blank node - point at circle of 3 (map test) | PASS | PASS | PASS | PASS | PASS | PASS | PASS | PASS | PASS | PASS |
| Test test033c: disjoint identical subgraphs (1) | PASS | PASS | PASS | PASS | PASS | PASS | PASS | PASS | PASS | PASS |
| Test test034c: disjoint identical subgraphs (2) | PASS | PASS | PASS | PASS | PASS | PASS | PASS | PASS | PASS | PASS |
| Test test035c: reordered w/strings (1) | PASS | PASS | PASS | PASS | PASS | PASS | PASS | PASS | PASS | PASS |
| Test test036c: reordered w/strings (2) | PASS | PASS | PASS | PASS | PASS | PASS | PASS | PASS | PASS | PASS |
| Test test038c: reordered 4 bnodes, reordered 2 properties (1) | PASS | PASS | PASS | PASS | PASS | PASS | PASS | PASS | PASS | PASS |
| Test test039c: reordered 4 bnodes, reordered 2 properties (2) | PASS | PASS | PASS | PASS | PASS | PASS | PASS | PASS | PASS | PASS |
| Test test040c: reordered 6 bnodes (1) | PASS | PASS | PASS | PASS | PASS | PASS | PASS | PASS | PASS | PASS |
| Test test043c: literal with language | PASS | PASS | PASS | PASS | PASS | PASS | PASS | PASS | PASS | PASS |
| Test test044c: poison – evil (1) | PASS | PASS | PASS | PASS | PASS | PASS | PASS | PASS | PASS | PASS |
| Test test045c: poison – evil (2) | PASS | PASS | PASS | PASS | PASS | PASS | PASS | PASS | PASS | PASS |
| Test test046c: poison – evil (3) | PASS | PASS | PASS | PASS | PASS | PASS | PASS | PASS | PASS | PASS |
| Test test047c: deep diff (1) | PASS | PASS | PASS | PASS | PASS | PASS | PASS | PASS | PASS | PASS |
| Test test047m: deep diff (1) (map test) | PASS | PASS | PASS | PASS | PASS | PASS | PASS | PASS | PASS | PASS |

| | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|
| Test test048c: deep diff (2) | PASS | PASS | PASS | PASS | PASS | PASS | PASS | PASS | PASS | PASS |
| Test test048m: deep diff (2) (map test) | PASS | PASS | PASS | PASS | PASS | PASS | PASS | PASS | PASS | PASS |
| Test test053c: @list | PASS | PASS | PASS | PASS | PASS | PASS | PASS | PASS | PASS | PASS |
| Test test053m: @list (map test) | PASS | PASS | PASS | PASS | PASS | PASS | PASS | PASS | PASS | PASS |
| Test test054c: t-graph | PASS | PASS | PASS | PASS | PASS | PASS | PASS | PASS | PASS | PASS |
| Test test055c: simple reorder (1) | PASS | PASS | PASS | PASS | PASS | PASS | PASS | PASS | PASS | PASS |
| Test test055m: simple reorder (1) (map test) | PASS | PASS | PASS | PASS | PASS | PASS | PASS | PASS | PASS | PASS |
| Test test056c: simple reorder (2) | PASS | PASS | PASS | PASS | PASS | PASS | PASS | PASS | PASS | PASS |
| Test test056m: simple reorder (2) (map test) | PASS | PASS | PASS | PASS | PASS | PASS | PASS | PASS | PASS | PASS |
| Test test057c: unnamed graph | PASS | PASS | PASS | PASS | PASS | PASS | PASS | PASS | PASS | PASS |
| Test test057m: unnamed graph (map test) | PASS | PASS | PASS | PASS | PASS | PASS | PASS | PASS | PASS | PASS |
| Test test058c: unnamed graph with blank node objects | PASS | PASS | PASS | PASS | PASS | PASS | PASS | PASS | PASS | PASS |
| Test test059c: n-quads parsing | PASS | PASS | PASS | PASS | PASS | PASS | PASS | PASS | PASS | PASS |
| Test test060c: n-quads escaping | PASS | PASS | PASS | PASS | PASS | UNTESTED | PASS | PASS | PASS | PASS |
| Test test060m: n-quads escaping (map test) | PASS | PASS | PASS | PASS | PASS | UNTESTED | PASS | PASS | PASS | PASS |
| Test test061c: same literal value with multiple languages | PASS | PASS | PASS | PASS | PASS | PASS | PASS | PASS | PASS | PASS |

| | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|
| Test test062c: same literal value with multiple datatypes | PASS | PASS | PASS | PASS | PASS | PASS | PASS | PASS | PASS | PASS |
| Test test063c: blank node - diamond (with _:b) | PASS | PASS | PASS | PASS | PASS | PASS | PASS | PASS | PASS | PASS |
| Test test063m: blank node - diamond (with _:b) (map test) | PASS | PASS | PASS | PASS | PASS | PASS | PASS | PASS | PASS | PASS |
| Test test064c: blank node - double circle of 3 (0-1-2, reversed) | PASS | PASS | PASS | PASS | PASS | PASS | PASS | PASS | PASS | PASS |
| Test test065c: blank node - double circle of 3 (0-2-1, reversed) | PASS | PASS | PASS | PASS | PASS | PASS | PASS | PASS | PASS | PASS |
| Test test066c: blank node - double circle of 3 (1-0-2, reversed) | PASS | PASS | PASS | PASS | PASS | PASS | PASS | PASS | PASS | PASS |
| Test test067c: blank node - double circle of 3 (1-2-0, reversed) | PASS | PASS | PASS | PASS | PASS | PASS | PASS | PASS | PASS | PASS |
| Test test068c: blank node - double circle of 3 (2-1-0, reversed) | PASS | PASS | PASS | PASS | PASS | PASS | PASS | PASS | PASS | PASS |
| Test test069c: blank node - double circle of 3 (2-0-1, reversed) | PASS | PASS | PASS | PASS | PASS | PASS | PASS | PASS | PASS | PASS |
| Test test070c: dataset - isomorphic default and iri named | PASS | PASS | PASS | PASS | PASS | PASS | PASS | PASS | PASS | PASS |
| Test test070m: dataset - isomorphic default and iri named (map test) | PASS | PASS | PASS | PASS | PASS | PASS | PASS | PASS | PASS | PASS |

| | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|
| Test test071c: dataset - isomorphic default and node named | PASS | PASS | PASS | PASS | PASS | PASS | PASS | PASS | PASS | PASS |
| Test test071m: dataset - isomorphic default and node named (map test) | PASS | PASS | PASS | PASS | PASS | PASS | PASS | PASS | PASS | PASS |
| Test test072c: dataset - shared blank nodes | PASS | PASS | PASS | PASS | PASS | PASS | PASS | PASS | PASS | PASS |
| Test test072m: dataset - shared blank nodes (map test) | PASS | PASS | PASS | PASS | PASS | PASS | PASS | PASS | PASS | PASS |
| Test test073c: dataset - referencing graph name | PASS | PASS | PASS | PASS | PASS | PASS | PASS | PASS | PASS | PASS |
| Test test073m: dataset - referencing graph name (map test) | PASS | PASS | PASS | PASS | PASS | PASS | PASS | PASS | PASS | PASS |
| Test test074c: poison - Clique Graph (negative test) | PASS | PASS | PASS | PASS | PASS | PASS | PASS | PASS | PASS | PASS |
| Test test075c: blank node - diamond (uses SHA-384) | PASS | PASS | PASS | PASS | PASS | PASS | PASS | PASS | PASS | PASS |
| Test test075m: blank node - diamond (uses SHA-384) (map test) | PASS | PASS | PASS | PASS | PASS | PASS | PASS | PASS | PASS | PASS |
| Test test076c: duplicate ground triple in input | PASS | PASS | PASS | PASS | PASS | PASS | PASS | PASS | PASS | PASS |
| Test test077c: duplicate triple with blank node in input | PASS | PASS | PASS | PASS | PASS | PASS | PASS | PASS | PASS | PASS |

| | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|
| **Percentage passed out of 86 Tests** | **100.0%** | **100.0%** | **100.0%** | **100.0%** | **100.0%** | **97.7%** | **100.0%** | **100.0%** | **100.0%** | **100.0%** |

## A. Test Subjects

This report was tested using the following test subjects:

**A.1 Corese**

**Description**
Software platform implementing and extending the standards of the Semantic Web.
**Release**
#fc18259
**Programming Language**
Java
**Home Page**
https://project.inria.fr/corese/
**Developer**
Wimmics Team
**Test Suite Compliance**

| | |
|---|---|
| RDF Dataset Canonicalization (RDFC-1.0) Test Suite | 86/86 (100.0%) |

**A.2 rdf-canonicalization-cpp**

**Description**
A RDF Dataset Canonicalization (RDFC-1.0) Implementation for C++
**Release**
1.0.0
**Programming Language**
C++
**Home Page**
https://github.com/dcdpr/rdf-canonicalization-cpp
**Developer**
Digital Contract Design, LLC https://github.com/dcdpr/
Dan Pape https://github.com/danpape/
**Test Suite Compliance**

| | |
|---|---|
| RDF Dataset Canonicalization (RDFC-1.0) Test Suite | 86/86 (100.0%) |

**A.3 rdf-canonize**

**Description**
A RDF Dataset Canonicalization processor for JavaScript
**Release**
4.0.1
**Programming Language**
JavaScript
**Home Page**
https://github.com/digitalbazaar/rdf-canonize
**Developer**
Digital Bazaar, Inc. https://digitalbazaar.com/
**Test Suite Compliance**

| | |
|---|---|
| RDF Dataset Canonicalization (RDFC-1.0) Test Suite | 86/86 (100.0%) |

**A.4 Titanium RDFC**

**Description**
A RDF Dataset Canonicalization (RDFC 1.0) in Java
**Release**
2.0.0
**Programming Language**
Java
**Home Page**
https://github.com/filip26/titanium-rdf-canon
**Developer**
Filip Kolarik https://github.com/filip26
**Test Suite Compliance**

| | |
|---|---|
| RDF Dataset Canonicalization (RDFC-1.0) Test Suite | 86/86 (100.0%) |

**A.5 Sophia**

**Description**
A Rust toolkit for RDF and Linked Data.
**Release**
#4511da9
**Programming Language**
Rust
**Home Page**
https://github.com/pchampin/sophia_rs
**Developer**
Pierre-Antoine Champin

**Test Suite Compliance**

| | |
|---|---|
| RDF Dataset Canonicalization (RDFC-1.0) Test Suite | 86/86 (100.0%) |

### A.6 RDF-URDNA

**Description**
A Java implementation of the RDF URDNA-2015 algorithm. NOTE: The Titanium RDF library does not currently support IRIs which are required for test 060. That is why that test is skipped.

**Release**
1.3

**Programming Language**
Java

**Home Page**
https://github.com/setl/rdf-urdna

**Developer**
Simon Greatrix

**Test Suite Compliance**

| | |
|---|---|
| RDF Dataset Canonicalization (RDFC-1.0) Test Suite | 84/86 (97.7%) |

### A.7 zkp-ld/rdf-canon

**Description**
A Rust implementation of the RDF Dataset Canonicalization algorithm version 1.0 (RDFC-1.0) compatible with Oxigraph and Oxrdf.

**Release**
0.15.0-alpha.4

**Programming Language**
Rust

**Home Page**
https://github.com/zkp-ld/rdf-canon

**Developer**
Dan Yamamoto

**Test Suite Compliance**

| | |
|---|---|
| RDF Dataset Canonicalization (RDFC-1.0) Test Suite | 86/86 (100.0%) |

### A.8 RDF.ex

**Description**
RDF.ex is a pure-Elixir library for working with Resource Description Framework (RDF) data.

**Release**
1.2.0

**Programming Language**
Elixir

**Home Page**
https://rdf-elixir.dev/

**Developer**
Marcel Otto http://marcelotto.net/

**Test Suite Compliance**

| | |
|---|---|
| RDF Dataset Canonicalization (RDFC-1.0) Test Suite | 86/86 (100.0%) |

### A.9 rdfjs-c14n

**Description**
Implementation in Typescript of the RDF Canonicalization Algorithm RDFC-1.0, on top of the RDF/JS interfaces

**Release**
3.1.0

**Programming Language**
TypeScript

**Home Page**
https://iherman.github.io/rdfjs-c14n/

**Developer**
Ivan Herman https://www.ivan-herman.net/

**Test Suite Compliance**

| | |
|---|---|
| RDF Dataset Canonicalization (RDFC-1.0) Test Suite | 86/86 (100.0%) |

### A.10 Ruby RDF::Normalize

**Description**
RDF::Normalize performs Dataset Canonicalization for RDF.rb.

**Release**
0.7.0

**Programming Language**
Ruby

**Home Page**
https://github.com/ruby-rdf/rdf-normalize

**Developer**
Gregg Kellogg https://greggkellogg.net/

**Test Suite Compliance**

| | |
|---|---|
| RDF Dataset Canonicalization (RDFC-1.0) Test Suite | 86/86 (100.0%) |

## B. Individual Test Results

Individual test results used to construct this report are available here:

- elixir-earl-report.ttl
- java-corese-report.ttl
- java-rdf-urdna.ttl
- js-rdf-canonize-earl.ttl
- rdf-canonicalization-cpp.report.ttl
- rdfjs-c14n-report.ttl
- ruby-earl.ttl
- rust-sophia-rdfc10.ttl
- rust-zkp-ld-earl.ttl
- titanium-rdf-canon-earl.ttl

## C. Report Generation Software

This report generated by earl-report version 0.9.1 is a Ruby application freely available under the generous terms of the Unlicense. More information is available at https://github.com/gkellogg/earl-report .

This software is provided by Gregg Kellogg in hopes that it might make the lives of conformance testers easier.