Reliable Context

RELIABLE CONTEXT / RESEARCH & PRACTICAL GUIDANCE

What Does a Signed RDF Export Prove? A Verifier's Guide

> Reliable Context is a kernel of truth: a system of record for organizations that fuse private and public information and must defend every value they act on. One canonical record per real thing, a cited source and an observation date behind every value, unknowns declared rather than left blank, and any past state of the record reproducible.

Published by Reliable Context · 2026-10-01 · signed rdf export · rdf graph signature · rdf canonicalization · rdf dataset verification · graph provenance · data integrity · digital signatures · evidence review · chain of custody

Original article: https://reliablecontext.com/articles/signed-rdf-export-graph-evidence



In this article

1. Executive Summary
2. Introduction and Background
3. What a signed RDF export is
4. Define the export boundary before verifying
5. Verify the signature and authorization
6. A reproducible verifier fixture
7. Trace signed assertions to evidence
8. Data Analysis and Evidence
9. Implications and Future Directions
10. Frequently Asked Questions (FAQs)
11. Conclusion

Executive Summary

A signed Resource Description Framework (RDF) export can establish that a specified signer produced or endorsed a particular, bounded dataset and that the dataset presented for review matches the data covered by the proof. That conclusion is conditional. The reviewer must reconstruct the same RDF dataset, run the declared transformation and canonicalization, verify the signature with an authorized key, and assess the key at the relevant time. RDF Dataset Canonicalization 1.0 (RDFC-1.0) makes isomorphic datasets yield the same canonical result; it does not itself define a signature. (www.w3.org) A digital signature checks modification and claimed signatory, while the verifier's policy decides whether that signatory was entitled to make the relevant assertion. (csrc.nist.gov)

The decisive question is **what was inside the signed boundary**. An RDF dataset has a default graph and may have named graphs. A graph name does not necessarily identify the graph's contents. (www.w3.org) The export manifest should identify every included graph, its RDF version, transformation profile, canonicalization version, digest and signature suite, proof purpose, and external dependencies. A checksum for a transferred file tests byte identity; canonical RDF tests dataset identity across allowed serializations. Neither automatically covers a remotely retrieved ontology, JSON-LD context, source document, or later data unless it is included in the protected material or bound by a separately verified digest. (ndsa.org) (www.w3.org)

The proposed **verifier's fixture** contains two serializations of one isomorphic dataset, one dataset with a changed assertion, and a deliberately difficult blank-node dataset. The first should yield equal canonical output and a valid proof if the proof envelope is otherwise valid. The changed assertion should fail the integrity check. A resource limit or unresolved dependency yields **indeterminate**, not a claim that the assertion is false. The W3C test suite lists **86** cases, including a negative high-complexity blank-node case; these are conformance checks, not a throughput benchmark. (w3c.github.io) The standard itself recommends resource controls for expensive blank-node inputs. (www.w3.org)

Finally, **authenticity is narrower than truth**. The W3C Verifiable Credentials Data Model explicitly says verifiability does not imply that claims are true. (www.w3.org) A reviewer must inspect citations, observed source versions, custody events, conflicts, and the evidence behind selected assertions. PROV-O provides

terms to exchange such provenance, and data-citation guidance calls for a precise version or timeslice. (www.w3.org) (force11.org) The decision should record a pass, a demonstrated mismatch, or a request for evidence, with missing, blocked, and inapplicable fields distinguished. A signed snapshot is strong evidence of a bounded handoff only when the reviewer can reproduce both the cryptographic check and the evidentiary path from assertions to their sources.

Introduction and Background

Knowledge engineers increasingly exchange graph snapshots as files, archives, or API results. A recipient may receive an RDF dataset, a digest, a detached signature, and a statement naming the issuer. The practical question is whether this is the same snapshot the issuer exported. A second question asks whether the issuer controlled the signing key for this purpose. A third asks whether the statements in the graph are supported by the cited material. These are **dataset equality**, **signer authorization**, and **factual support**. Treating them as one yes-or-no test makes a technically valid proof sound stronger than it is. (csrc.nist.gov)

The distinction matters because graph data can be serialized in several forms. Apache Jena documents Turtle, TriG, N-Quads, and JSON-LD writers, among others. (jena.apache.org) Syntactic differences such as statement order or blank-node labels may leave the abstract dataset unchanged, while a changed predicate, object, or graph assignment changes it. RDFC-1.0 addresses this comparison by giving isomorphic datasets a shared canonical result. (dotnetrdf.org) The verifier still needs the exact transformation and proof suite, because a canonical form is only one input to a signature workflow. (www.w3.org)

This report uses **export** to mean a frozen, explicitly bounded RDF dataset plus its verification materials. It does not presume that the exported data is complete, that referenced resources were frozen, or that the issuer independently observed every fact. The task is to decide whether to **accept** the export as intact for its declared scope, **reject** it because a reproducible check fails, or **request more evidence** because the check cannot be completed. A network error, for example, need not mean a later verification would fail.

The guidance applies to standalone RDF snapshots and to constrained JSON-LD documents whose declared cryptosuites transform them to RDF. These are different packaging cases. The W3C Data Integrity suites discussed below specify proofs for transformed documents; their definitions do not, by themselves, standardize a proof envelope for every arbitrary standalone RDF dump. The export's own profile must therefore specify exactly what was signed and how the proof is carried.

What a signed RDF export is

Dataset, graph, and canonical form

An **RDF dataset** consists of exactly one default graph and zero or more named graphs under the RDF 1.2 Concepts candidate specification available as of October 2026. Blank nodes can be shared across graphs, so renaming a blank node in one graph without preserving its cross-graph identity can change the dataset. (rdf4j.org) A named graph is identified by a graph name, but the name is not required to denote the graph itself. It is therefore an error to infer that a graph identifier alone authenticates its contents or the party that supplied them.

Two datasets are **isomorphic** when an RDF-term mapping can rename blank nodes while preserving the dataset's graph structure and statements. RDFC-1.0 gives the same canonical result if and only if the datasets are isomorphic. (dotnetrdf.org) Its concrete canonical serialization sorts canonical N-Quads in code point order and concatenates them. (www.w3.org) That permits a verifier to compare semantically equivalent RDF dataset structures despite permitted changes in source serialization. A byte checksum of the original export file answers a different question: whether those exact file bytes changed. A 2023 Scientific Data paper describes RDF-specific content identifiers that ignore non-semantic bit-level differences. (www.nature.com) NIST describes message digests as a way to detect message changes; the National Digital Stewardship Alliance (NDSA) calls fixity evidence that one set of bits matches another. (csrc.nist.gov) (ndsa.org)

Canonicalization is not authentication. RDFC-1.0 explicitly leaves graph-signature definition to other specifications. A signature scheme must define the protected bytes, digest, signature algorithm, proof envelope, key representation, and verification steps. A verifier can reproduce canonical N-Quads perfectly and still lack any basis to attribute the dataset to an issuer. Conversely, a valid signature over a file can establish byte identity without proving that a different serialization denotes the same RDF dataset. (csrc.nist.gov) (www.rfc-editor.org)

Four identity questions

- **File identity:** Did the received bytes match a recorded digest or authenticated file manifest? The answer is about exact bytes and the stated algorithm. (csrc.nist.gov) (datatracker.ietf.org)
- **Dataset identity:** Did parsing, transformation, and RDFC-1.0 yield the expected canonical dataset? The answer tolerates serialization and blank-node-label differences allowed by the profile. (dotnetrdf.org)
- **Issuer identity:** Does a valid proof connect those protected bytes to a key that an accepted issuer authorized for this purpose? The answer requires a key relationship and verifier trust policy. (www.w3.org)
- **Source identity:** Do cited statements point to the precise source versions, excerpts, and observation dates they claim? This is an evidentiary review beyond signature verification. (force11.org)

As of October 2026, the 2024 RDFC-1.0 Recommendation references the RDF 1.1 dataset and N-Quads model, while RDF 1.2 Concepts is a Candidate Recommendation Snapshot dated **7 April 2026**. RDF 1.2 introduces triple terms. (www.w3.org) A verifier receiving such terms should require the exact RDF version and an explicit conversion or proof profile, then test that conversion for data loss. It should not silently assume an RDF 1.1 canonicalization path covers full RDF 1.2 structures. RDF4J's documentation, for example, distinguishes store support for triple terms, which may vary by backend. (rdf4j.org)

Define the export boundary before verifying

A signature protects the data fed to its proof algorithm, not everything a reader might later consult. W3C Data Integrity states that the protected data is the **transformed data**. The transformation may resolve a JSON-LD context, select statements, or combine a proof configuration with the document. If a term disappears or a remote context changes, the reviewer may be checking a different proposition from the issuer's intended one.

W3C requires a data-loss error in the specified JSON-LD processing case when undefined terms disappear during transformation. (www.w3.org) The JSON-LD specification advises vetting and caching critical remote contexts. (www.w3.org)

For a standalone RDF export, the issuer should provide a manifest that fixes the following before any digest is recomputed:

- **Dataset extent:** Identify the default graph, each named graph, any excluded graph, and whether the export is a complete snapshot or selected subset. (rdf4j.org)
- **RDF interpretation:** State the RDF version, input syntax, base Internationalized Resource Identifier (IRI), and parser settings. Relative identifiers without a stable base can change meaning across systems. (jena.apache.org)
- **Transformation:** Identify context expansion, filtering, skolemization, inference, or RDF 1.2 conversion, including the version and order of each step. (rdf4j.org)
- **Canonicalization:** Name RDFC-1.0 or another exact algorithm, implementation version, and expected canonical output or digest. Library documentation may cite a draft rather than the final Recommendation. (dotnetrdf.org)
- **Cryptography:** Record digest, signature suite, protected proof configuration, verification method, and proof purpose. A canonicalization algorithm's internal hash need not be the digest used by the signature suite. (www.w3.org)
- **External resources:** List referenced ontologies, shapes, contexts, vocabularies, and source files, marking each included, digest-bound, version-pinned, or outside the proof.
- **Time:** Record export time, proof creation or expiration if present, and the time basis for key and status checks. Proof creation time is optional in W3C Data Integrity.

Table 1 compares what common evidence artifacts establish. The Reliable Context row describes a system-of-record export as a source-custody input, using only its published claims; it does not imply that its export is an RDF signature.

Evidence artifact or layer	Input and test	What a pass supports	What remains open
File checksum or BagIt manifest	Recompute digests for the listed payload files. (datatracker.ietf.org)	Exact byte fixity for the manifested files.	Issuer authorization, RDF isomorphism, source truth.
RDFC-1.0 canonical output	Parse the declared dataset and compare canonical N-Quads. (dotnetrdf.org)	Dataset isomorphism under the selected RDF model.	Who issued the dataset and why it is credible.
Cryptographic proof	Reproduce the suite's transformed input and verify with an authorized key.	Integrity and claimed issuer for that protected input.	Unprotected links, completeness, and factual accuracy.
Provenance graph	Check entities, activities, agents, and derivation links. (force11.org)	A structured account of how asserted values arose.	Whether each cited source actually supports its value.

Evidence artifact or layer	Input and test	What a pass supports	What remains open
Reliable Context value export	Inspect the source, excerpt, locator, and dates exported with each value.	A reviewable chain of custody for that value, as the vendor describes it.	No published claim here that an RDF proof signs the export.

The table's rows are cumulative rather than interchangeable. A verified checksum can preserve a file whose RDF interpretation changes because a relative IRI is resolved under a new base. A valid graph proof can cover a dataset while leaving a cited source document outside the boundary. A provenance record can be detailed yet require independent inspection of its source quotations. The reviewer should write these limits into the decision, not silently promote one pass to every layer. (jena.apache.org) (ndsa.org)

Verify the signature and authorization

The verifier should begin from the issuer's published profile, not from a guess based on file extension or proof field names. For a W3C Data Integrity proof, the procedure transforms the document, hashes the relevant transformed material, and verifies the cryptographic proof. (www.w3.org) The `ecdsa-rdfc-2019` suite canonicalizes its input document as an RDF dataset and uses **SHA-256 with P-256** or **SHA-384 with P-384**, according to the selected curve. (www.w3.org) The `eddsa-rdfc-2022` suite also canonicalizes a document as RDF and uses **Ed25519** in its signature process. (www.w3.org) Those details matter because changing the suite changes the protected byte construction. They do not license applying either suite to an arbitrary N-Quads file without a declared wrapper and proof profile.

A practical check has four independently logged results:

1. **Parse and scope:** Record the received byte digest, syntax, base IRI, RDF version, graph inventory, and any parser errors. Keep the received file immutable for later comparison. (csrc.nist.gov) (jena.apache.org)
2. **Transform and canonicalize:** Apply only the declared steps, record software versions, and compare the canonical output or digest against the protected value. If the input requires a remote context, record the retrieved version and a trusted digest or snapshot. (dotnetrdf.org)
3. **Verify the proof:** Apply the exact cryptosuite to its specified proof configuration and transformed data, then check the signature bytes. A successful arithmetic check establishes a relation to a key, not an authorization policy. (csrc.nist.gov)
4. **Check the issuer:** Resolve the verification method, controller, and proof purpose under the relevant trust policy. W3C Controlled Identifiers calls for checking that the method is referenced by the appropriate verification relationship.

For an assertion, the `assertionMethod` relationship is the relevant W3C example. The verifier must also decide which issuers it trusts for which subjects, because the credential model leaves that decision to verifier policy. Record the time of proof creation if present and evaluate key expiration or revocation and document status separately. W3C distinguishes a proof's validity interval from that of the credential it secures, and

verification-method revocation from credential status. (www.w3.org) A present-day key document may not be sufficient evidence of a historical authorization state without an archived document or other trusted time record.

Failures need precise names. A malformed signature or changed protected assertion is a **reject** for the stated proof. A missing public key, unavailable status endpoint, unresolved context, unsupported RDF 1.2 construct, or exhausted resource limit is **indeterminate** until additional evidence or a compatible verifier is available. W3C explicitly notes that a network error need not imply future verification failure. The report should retain exact error codes, the stage reached, the fetched dependency versions, and the policy that converted technical results into an accept or reject decision. NIST notes that a digital signature does not itself supply confidentiality or replay protection, so distribution controls and freshness checks need separate treatment. (csrc.nist.gov)

A reproducible verifier fixture

The following fixture is **hypothetical**. It specifies expected decisions, not benchmark timings or a claim about a particular implementation. Imagine a dataset with four quads: two assertions in the default graph and two in a named evidence graph. One assertion connects an identified record to a blank node; the named graph carries a source locator and observation date. The issuer signs the canonical dataset under a stated profile and separately provides the source document. The reviewer preserves the original files and logs the canonical output and proof-verification result. (dotnetrdf.org) (force11.org)

Table 2 records the four manipulations and the result expected from a correct verifier. The first two rows use the same abstract dataset; the remaining rows change what can be concluded.

Fixture input	Controlled change	Canonical and proof result	Decision
A: original snapshot	Four quads in one default and one named graph; signed as declared.	Expected canonical output and authorized proof match.	Accept integrity for the stated boundary; inspect source support separately.
B: isomorphic serialization	Reorder statements and rename the blank node without changing its graph relationships.	Same canonical output; the same protected dataset can verify. (dotnetrdf.org)	Accept dataset identity if every other proof check passes.
C: altered assertion	Change one object's literal but keep the old proof.	Canonical output changes; signature check should fail. (rdflib.readthedocs.io)	Reject the presented snapshot for that proof.
D: hard blank-node input or missing context	Supply a high-symmetry blank-node component, or require an unavailable external context.	Resource limit or dependency error prevents completed verification. (rdflib.readthedocs.io)	Indeterminate; request a bounded, reproducible verification path.

The table is a **test design**, not a measured performance study. It isolates three properties. Isomorphism tolerates a serialization change. An assertion change should invalidate a proof tied to the original canonical dataset. A timeout tells the reviewer only that this verifier did not finish within its stated limits. RDFLib

documentation independently warns that blank-node canonicalization time may increase sharply on degenerate graphs, and the W3C test suite includes a **10-node clique** intended to exercise a high-complexity negative path. (rdflib.readthedocs.io) (w3c.github.io)

An unresolved ontology presents another boundary test. If the ontology file was inside the signed dataset, changing it changes the protected graph. If the export merely mentions the ontology's IRI, the signature of the dataset does not authenticate the current bytes retrieved from that IRI. The same issue applies to linked JSON-LD contexts; W3C states that linked resources are not cryptographically protected by a credential proof by default. A sound fixture therefore stores a frozen copy or authenticated digest for each dependency that affects interpretation. IPFS documentation also illustrates why an identifier should not be equated with a raw file digest: a content identifier includes codec information and can hash a root block rather than a file's simple byte stream. (docs.ipfs.tech)

Trace signed assertions to evidence

After the cryptographic checks, select assertions that matter to the decision and follow each one backward. A useful path is: signed quad, record identifier, source excerpt, exact source version, observation date, extraction or derivation step, and any conflicting source. PROV-O supplies RDF terms for exchanging provenance across systems and models quotation as a kind of derivation. (force11.org) Crossref describes relation types as a way to tag data and software citations precisely. (www.crossref.org) A provenance graph can express this path, but its presence is an assertion about history, not independent proof that the history occurred. The cited document must still be opened or retained and the claimed passage checked.

For each selected value, ask:

- **Location:** Is there a stable locator to the precise passage or data row? Dublin Core recommends enough bibliographic detail to identify a cited resource. (www.dublincore.org)
- **Version:** Does the citation name the source version or timeslice actually observed? FORCE11 calls for provenance and fixity in citation metadata, and the Research Data Alliance emphasizes precise dataset subsets and versions. (force11.org) (www.rd-alliance.org)
- **Observation:** Was the source read on a stated date, distinct from publication and export time? A later page edit can leave the original observation uncheckable unless a snapshot was retained. (force11.org)
- **Custody:** Are transfer and storage changes logged and verified with fixity checks? The NDSA recommends checking fixity when content moves. The UK National Archives also describes ongoing fixity checks for original files. (www.nationalarchives.gov.uk) (ndsa.org)
- **Conflict:** If two sources disagree, does the record retain both and explain the chosen value? A signature over one chosen value does not erase the alternative.
- **Unknown:** Does a missing fact state **not found**, **blocked**, or **not applicable**, with a reason, instead of silently becoming a negative claim?
- **Scope:** Is the cited evidence actually in the proof boundary, separately digest-bound, or merely linked? DCAT warns that checksum assurances depend on the integrity and authenticity of checksum metadata itself.

Reliable Context's published value-export guarantee is relevant at this layer: it says every value exports with a source, excerpt, locator, and dates, and that past record states can be reproduced. These are observable properties a reviewer can test on a received packet. They are not a substitute for an RDF signature or for inspecting the underlying source. The verification protocol should test the values actually needed for the decision, including one with a known conflict and one declared unknown, rather than infer factual support from a valid graph-level proof.

Table 3 specifies a compact packet that an independent reviewer can preserve. Each field should have a state of **present**, **not found**, **blocked**, or **not applicable**, with a reason for the latter three.

Packet item	Required evidence	Review if absent
Bounded dataset	Received bytes, graph inventory, base IRI, RDF version, and canonical digest.	Request a scope statement; do not infer completeness.
Proof profile	Transformation, canonicalization, suite, proof purpose, key and issuer.	Signature result is indeterminate.
Temporal record	Export, proof, observation, and key-status times with their sources.	Historical authorization or freshness remains unresolved.
Dependencies	Frozen contexts, vocabularies, shapes, and source files or their authenticated digests.	Mark interpretation or validation conditional.
Provenance trail	Assertion-to-source locator, excerpt, derivation, and conflicting evidence. (force11.org) (www.dublincore.org)	Request source support for material values.
Reproduction log	Tool versions, exact inputs, errors, canonical output, proof result, and reviewer decision. (dotnetrdf.org)	Another reviewer cannot repeat the check.

The packet gives a reviewer a way to say exactly what passed. For example, a received RDF dataset might pass canonical and signature checks while one source document remains blocked by access policy. That supports acceptance of the export's integrity for the declared boundary and a request for source evidence before accepting the affected value. Preservation guidance makes a similar distinction: fixity can show matching bits without ensuring that an object remains understandable. (ndsa.org)

Reviewer sign-off checklist

Use the following checklist as a compact decision record. A field can be present, not found, blocked, or not applicable; a missing field should carry a reason and the person or system that supplied the status.

- **Received object:** Preserve the exact package and compute its byte digest.
- **Graph inventory:** Name the default graph and every included named graph.
- **Selection rule:** State whether this is a whole snapshot or a selected subset.
- **Syntax and base:** Record the RDF version, syntax, and base IRI.
- **Transformation:** Pin every context, conversion, filter, and inference step.
- **Canonicalization:** Record algorithm, implementation version, output, and digest.
- **Proof:** Name the suite, protected configuration, signature bytes, and proof purpose.
- **Authority:** Record issuer, controller, key relationship, and trust policy.
- **Time:** Record proof, export, observation, and key-status time bases.

- **Dependencies:** Classify linked resources as included, digest-bound, or outside scope.
- **Value evidence:** Preserve source versions, locators, excerpts, and conflicts. (force11.org)
- **Decision:** Log accept, reject, or request for evidence, with reproducible reasons.

Data Analysis and Evidence

This section separates **specified conformance evidence** from performance claims. The W3C RDFC-1.0 test manifest lists **86 entries**. Its tests compare produced canonical N-Quads text with expected output; a negative test can pass by returning the specified high-complexity error. (w3c.github.io) The associated implementation report lists **11 test subjects**; **10** report **86 of 86**, and one reports **84 of 86**, with two cases untested. The report itself says it is an internal W3C document without official standing. (w3c.github.io) These figures show the scope and reported outcomes of that suite. They do not prove every dataset can be processed within a fixed time or that a receiving application configured its parser, transformation, and key policy correctly.

The fixture above has **four quads**, **two graphs**, and **one changed literal** in the tampered case. These are disclosed test inputs, not empirical population statistics. Its expected result is binary only when all declared dependencies and verification steps are available. A blank-node resource limit produces no successful proof check and no demonstrated signature mismatch. RDFC-1.0 recommends configurable timeouts and iteration limits for constructed datasets that can consume substantial computation. (rdflib.readthedocs.io) The distinction between **failed** and **not completed** should appear in the log and in any downstream decision rule.

Historical data show why blank nodes deserve explicit test coverage, though they do not predict current workloads. A **2014** empirical study of a **May 2012** web corpus reported **1.230 billion** unique quads across **8.373 million** RDF documents, with **3.758 million** documents, or **44.9%**, containing at least one blank node. (marceloarenas.cl) The corpus, collection date, and document mix bound that estimate. It is evidence that blank nodes were common in one historical crawl, not evidence of today's rate or of RDFC-1.0 runtime. A separate tool's experiment instructions use a **600-second** cutoff; that is an experimental limit, not a performance result for this verifier. (aidanhogan.com)

Cryptographic parameters are likewise conditional on the actual suite. The ECDSA RDF canonicalization suite specifies **SHA-256/P-256** or **SHA-384/P-384** pairings; the EdDSA RDF canonicalization suite specifies **Ed25519**. (www.w3.org) (www.rfc-editor.org) The IANA registry lists SHA-256 as a **256-bit** hash value. (www.iana.org) Those numbers describe algorithm choices and output size, not the truth probability of any assertion. NIST's digital signature standard describes signatures as a means of detecting unauthorized changes and authenticating the claimed signatory. (csrc.nist.gov) The verifier should record the configured suite and the exact digest bytes, then test authorization and source support as separate propositions.

The best quantitative conclusion is therefore narrow: a **pass** is reproducible equality and authorized proof for one declared boundary; a **fail** is a reproducible contradiction of that proof; an **indeterminate** result records an unmet precondition. Counts from a conformance suite, historical corpus, or toy fixture cannot be converted into a general accuracy percentage for signed exports. This three-way decision prevents unavailable dependencies and computation limits from being mislabeled as false assertions. (w3c.github.io) (ndsa.org)

Implications and Future Directions

The largest operational risk is an **underspecified boundary**. A file, a canonical RDF dataset, a transformed JSON-LD document, and a collection of linked source files are different objects. A verifier should demand a manifest or profile that says which one was protected, how it was transformed, and which dependencies are separately authenticated. DCAT's checksum model is useful for distribution fixity but explicitly notes that the checksum metadata itself also needs integrity and authenticity. BagIt similarly defines a complete package and manifest checks that can accompany, but do not replace, issuer authentication. The Library of Congress describes the manifest as an inventory of content files. (www.digitalpreservation.gov) (datatracker.ietf.org)

Future RDF 1.2 use makes version disclosure more important. The RDF 1.2 candidate specification adds triple terms; the 2024 RDFC-1.0 Recommendation was written against the then-current RDF 1.1 model. (rdf4j.org) A future export profile should specify whether RDF 1.2 constructs are excluded, converted by a named information-preserving encoding, or handled by a new canonicalization process. W3C's interoperability note cautions that one basic encoding is information preserving without preserving RDF semantics, a reminder to test both identity and interpretation. (www.w3.org)

Reviewers can also improve **evidence portability**. A dataset citation should identify the exact subset or version used, and an archive package should preserve source documents, hashes, and transfer checks. DataCite describes its metadata schema as supporting consistent resource identification for citation and retrieval. (schema.datacite.org) (www.rd-alliance.org) (datatracker.ietf.org) The Digital Preservation Coalition describes checksum comparisons as part of a chain of custody and advises generating new checksums after deliberate file-format migration. (www.dpconline.org) The US National Archives calls for transfer documentation sufficient to identify and interpret permanent electronic records. (www.archives.gov) Those practices make a later independent review possible, even when the original website changes or a current remote context is unavailable. They do not certify that the source's statement was correct when observed.

Finally, organizations should publish an explicit decision policy. It should identify which issuers may assert which kinds of information, how historical key status is established, when unavailable evidence triggers a request, and which assertions require source inspection. W3C leaves issuer trust and subject scope to verifier policy. That policy turns a technical proof result into a reviewable decision without overstating the proof's reach.

Frequently Asked Questions (FAQs)

How does a reviewer verify a signed RDF export?

Freeze the received package, enumerate the declared graphs and dependencies, reproduce the declared RDF transformation and canonicalization, verify the proof with the suite's procedure, check issuer authorization and temporal status, then inspect the cited evidence behind material assertions. Record each stage and its input. A mismatch is a rejection for that proof; an unavailable dependency is an indeterminate check. (csrc.nist.gov)

What does an RDF graph signature verify?

It verifies the protected input specified by its proof profile. For a canonical dataset signature, that can support dataset identity across isomorphic serializations and a relation to the signing key. RDFC-1.0 supplies canonicalization, while the signature scheme supplies authentication. The graph name, by itself, does not attest graph contents.

Can a digital signature prove that RDF data is true?

No. It can support integrity and claimed issuer for bounded data. The W3C credential model explicitly separates verifiability from truth. Factual review still requires source versions, quotations, observation dates, and assessment of conflicts. (force11.org)

Does an imported ontology or JSON-LD context become signed automatically?

No. A linked resource may influence interpretation or transformation while remaining outside the proof unless its bytes or digest are protected. The reviewer should pin the exact version or require it in the package. Remote contexts that matter to processing should be vetted and cached.

What if blank-node canonicalization times out?

Report **indeterminate**, preserve the input and resource limit, and ask for a reproducible profile or bounded input. A timeout is not evidence that the changed-data test failed. W3C calls out adversarial blank-node complexity, and implementation documentation also warns of degenerate cases. (rdflib.readthedocs.io)

How can a signed RDF export serve as audit evidence?

Keep the received package, its manifest and proof, the verification log, the applicable trust policy, and the precise sources behind material assertions. The signature result should name its protected boundary and verification time. For a later reviewer, fixity checks at each transfer and source-version identifiers make the chain reproducible; neither converts an unsupported assertion into a fact. Preservation guidance treats custody and processing history as separate from matching bits. (force11.org) (ndsa.org)

Conclusion

A signed RDF export proves a **bounded** proposition when the full verification path succeeds: the received dataset, after the declared transformation and canonicalization, matches the data protected by a valid proof from an authorized key. The proposition must name the dataset boundary, RDF version, proof suite, issuer, purpose, and relevant time. Without those details, a signature string is not a review decision.

The same result does not establish that every relevant graph was exported, that linked resources were protected, or that an asserted value is factually correct. Source quotations, observation dates, versions, custody history, conflicts, and explicit unknown states remain separate evidence. A reviewer should accept an intact export for its stated scope, reject a reproducible integrity mismatch, and request evidence when a dependency or authorization step cannot be checked. That limited conclusion is both more useful and more defensible than calling every signed statement true.

Source links

Links cited in this article, in order of first appearance. Full addresses are provided for readers using extracted text.

1. https://www.w3.org/TR/rdf-canon
2. https://csrc.nist.gov/Projects/digital-signatures
3. https://www.w3.org/TR/rdf12-concepts
4. https://ndsa.org/documents/NDSA-Fixity-Guidance-Report-final100214.pdf
5. https://www.w3.org/TR/vc-data-model
6. https://w3c.github.io/rdf-canon/tests
7. https://www.w3.org/TR/prov-o
8. https://force11.org/info/joint-declaration-of-data-citation-principles-final
9. https://jena.apache.org/documentation/io/rdf-output.html
10. https://dotnetrdf.org/docs/stable/user_guide/rdf_canonicalization_api.html
11. https://www.w3.org/TR/vc-data-integrity
12. https://rdf4j.org/documentation/tutorials/getting-started
13. https://www.nature.com/articles/s41597-023-02230-y
14. https://csrc.nist.gov/pubs/fips/180-4/upd1/final
15. https://www.rfc-editor.org/rfc/rfc9530.html
16. https://datatracker.ietf.org/doc/html/rfc8493
17. https://www.w3.org/TR/cid-1.0
18. https://rdf4j.org/documentation/programming/rdf12
19. https://www.w3.org/TR/json-ld11
20. https://www.w3.org/TR/vc-di-ecdsa
21. https://www.w3.org/TR/vc-di-eddsa
22. https://csrc.nist.gov/glossary/term/digital_signature
23. https://rdflib.readthedocs.io/en/stable/apidocs/rdflib.compare
24. https://docs.ipfs.tech/concepts/content-addressing
25. https://www.crossref.org/documentation/schema-library/markup-guide-metadata-segments/data-citation
26. https://www.dublincore.org/specifications/dublin-core/dcmi-terms
27. https://www.rd-alliance.org/groups/data-citation-wg/outputs?output=187079
28. https://www.nationalarchives.gov.uk/information-management/manage-information/preserving-digital-records/our-role
29. https://w3c.github.io/rdf-canon/reports
30. https://marceloarenas.cl/publications/jws14.pdf
31. https://aidanhogan.com/blabel
32. https://www.rfc-editor.org/info/rfc8032
33. https://www.iana.org/assignments/named-information
34. https://csrc.nist.gov/pubs/fips/186-5/final

35. https://www.digitalpreservation.gov/news/2008/20080602news_article_bagit.html
36. https://www.w3.org/TR/rdf12-interop
37. https://schema.datacite.org
38. https://www.dpconline.org/handbook//technical-solutions-and-tools/fixity-and-checksums?tmpl=component
39. https://www.archives.gov/records-mgmt/policy/transfer-guidance.html

THE PUBLISHER

About Reliable Context

Reliable Context is a system of record for organizations that fuse private and public information and must defend every value they act on.

The guarantees

One canonical record per real thing: duplicates resolve by rule, ambiguity is shown rather than hidden, and related organizations relate without ever being merged. A verbatim, cited source and an observation date sit behind every value. Unknowns are declared as not found, blocked or not applicable with a reason, never left as an empty cell. Any past state of the record is reproducible, together with what changed since.

Where it runs

It runs inside the customer's perimeter, one database per organization. Permitted sources, model routing, reader clearance and retention are set as the customer's own policies, and every value is exportable with its chain of custody for an auditor or regulator.

What it is not

It is not a vector database or a RAG framework, not a data catalog, MDM or data-quality suite, not a CRM or data-enrichment vendor, and not a chatbot or AI assistant. Where a reader needs one of those, the honest answer is to say so and point them there.

Visit

See Reliable Context.

Reliable Context is in production use in two sectors; more than six thousand of its identity decisions were made by rule and none has been reversed. Articles published here explain general practice, cite their primary and academic sources, and name the jurisdiction of any regulatory rule they describe. They are not legal, regulatory or compliance advice for a specific organization, and requirements change.

Website: https://reliablecontext.com

This article is educational. Verify consequential medical, legal, financial, regulatory and technical decisions with appropriate professionals and the cited primary sources. Company services are described separately from the article's research findings.