IntuitionLabs

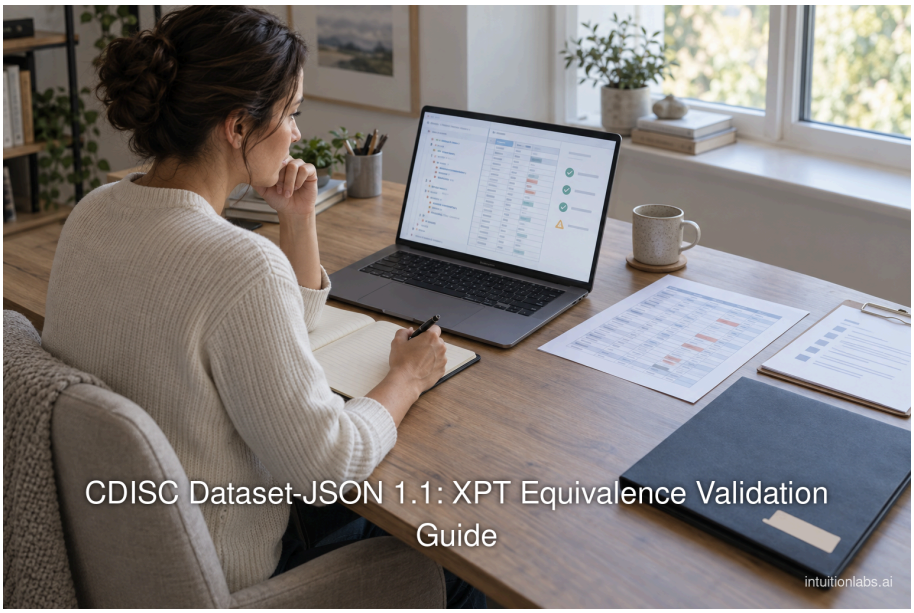
INTUITIONLABS / RESEARCH & PRACTICAL GUIDANCE

CDISC Dataset-JSON 1.1: XPT Equivalence Validation Guide

> Build practical AI for pharma and biotech with IntuitionLabs. We help life-science teams turn complex information and workflows into useful software, governed knowledge systems and AI tools.

Published by IntuitionLabs · 2026-09-26 · cdisc dataset-json 1.1 · xpt equivalence · dataset-json validation · clinical data standards · sas xport · define-xml · fda study data · submission readiness · data integrity

Original article: https://intuitionlabs.ai/articles/dataset-json-1-1-xpt-equivalence-validation



In this article

Executive Summary

Dataset-JSON 1.1 equivalence validation should answer a narrow question: can a sponsor generate Dataset-JSON and SAS Version 5 XPORT (XPT) from the same governed clinical dataset, reconstruct both, and demonstrate that the intended records, values, metadata, and missingness survived each path? The answer is an evidence package, not a byte-for-byte match. CDISC released Version 1.1 in December 2024 as a JSON transport for tabular datasets. (www.cdisc.org) (cdisc-org.github.io) The format has a defined schema and can refer to richer Define-XML metadata, so successful parsing alone is an incomplete acceptance test. (github.com) (www.cdisc.org)

Regulatory acceptance is a separate decision. FDA's 2024 feasibility pilot examined Dataset-JSON as a possible transport, and its April 2025 notice asked for comments on possible acceptance. (www.fda.gov) (www.federalregister.gov) Later evidence is more cautious: FDA's assessment of 2025 testing covered **five clinical and three nonclinical studies** with original XPT, converted JSON, and reconverted XPT packages, and concluded that Dataset-JSON had not demonstrated superiority over existing processes. (www.fda.gov) (www.fda.gov) As of September 26, 2026, FDA's June 2026 Study Data Technical Conformance Guide still specifies XPORT Version 5 for electronic datasets; the September action plan describes Dataset-JSON adoption and policy work as pending. (www.fda.gov) (www.fda.gov) A technical pass therefore cannot be presented as permission to replace an applicable XPT submission path. (www.fda.gov) (www.fda.gov)

The recommended pilot begins with a frozen canonical dataset and a versioned metadata contract. It emits both formats, then reads each back into a typed, ordered table. It checks file/schema validity, row and column counts, keys, metadata links, null patterns, exact strings, dates, and numeric values under documented type-aware comparison rules. CDISC says missing values are JSON `null`, distinct from empty strings, and permits some decimal values to travel as strings to avoid rounding. (cdisc-org.github.io) (cdisc-org.github.io) A preapproved tolerance applies only to named numeric fields whose source precision and target representation justify it; there is no universal tolerance in this report. JSON object-member order and row/column order require different treatment. (www.rfc-editor.org)

The deliverable is a **signed comparison manifest** containing hashes of actual transport files, semantic digests of reconstructed data, row/column totals, rule and schema versions, deviations, reviewer decisions, and the applicable submission-format decision. NIST describes digests as a means of detecting later change and digital signatures as a way to authenticate a signer. (csrc.nist.gov) (csrc.nist.gov) A pilot is ready when every required comparison passes or has a justified, approved exception, the evidence is reproducible, and the XPT fallback remains executable under the current regulator requirements. (database.ich.org)

Introduction and Background

Clinical Data Interchange Standards Consortium (CDISC) **Dataset-JSON** is a way to exchange tabular study data in JavaScript Object Notation (JSON). It does not itself determine whether a Study Data Tabulation Model (SDTM), Analysis Data Model (ADaM), or Standard for Exchange of Nonclinical Data (SEND) dataset is scientifically correct. CDISC's Version 1.1 specification describes one dataset per `.json` file, dataset and column metadata, and a `rows` array of data records. (cdisc-org.github.io) (cdisc-org.github.io) The sponsor still needs to evaluate its source standards, metadata, rules, and submission context.

The reason to run a pilot is practical. SAS Version 5 XPORT remains a familiar transport in established submission pipelines, while Dataset-JSON offers a different serialization and richer typed metadata. CDISC allows Dataset-JSON to reference a Define-XML document with more complete metadata; its `metaDataRef` identifies that file. (www.cdisc.org) (github.com) The comparison target is therefore the **meaning of the dataset under an explicit metadata contract**, not identical transport bytes. XPT and JSON use different encodings and headers by design. A matching hash of the two files would be surprising and is not an equivalence criterion.

The current FDA story must be read in order. The 2024 FDA, CDISC, and PHUSE pilot tested feasibility from September 2023 through April 2024 and aimed to transport study data without loss. (www.fda.gov) (www.fda.gov) A later Federal Register notice asked whether FDA should accept Dataset-JSON and described XPT replacement as a long-term possibility. (www.federalregister.gov) (www.federalregister.gov) In a 2025 exercise reported in 2026, FDA examined original XPT, JSON, and returned XPT for eight studies. Its assessment did not establish an advantage over existing processes. (www.fda.gov) These statements support a controlled internal pilot; they do not establish a new submission requirement.

The intended readers are clinical data platform architects, programmers, standards owners, and quality assurance (QA) leads. Their decision is whether their organization can produce a reproducible comparison pack without disrupting the accepted submission workflow. IntuitionLabs describes its work as data engineering and advisory services for life sciences; that perspective supports treating conversion as a governed integration problem rather than a product choice. (intuitionlabs.ai) (intuitionlabs.ai)

Key Changes

A transport with explicit dataset and column metadata

The Version 1.1 specification requires `datasetJSONVersion` and an `itemGroupOID` at dataset level. It carries a `columns` array whose order must match dataset variable order, and each column has an `itemOID`. (cdisc-org.github.io) (cdisc-org.github.io) (cdisc-org.github.io) (cdisc-org.github.io) This makes column-position drift observable, but only if the validation reads the declared metadata and the data together. A JSON parser that merely accepts the document cannot tell whether a value was placed under the intended clinical variable.

The column type vocabulary includes strings, integers, decimals, floats, doubles, booleans, date/time types, and URI. The specification also has a `targetDataType` for cases where a receiving system needs a different type, and recommends specifying it only when it differs from `dataType`. (cdisc-org.github.io) (cdisc-org.github.io) A pilot should record both the logical CDISC type and the physical type chosen by each tool. A character date in an XPT file, for example, needs an explicit comparison rule against the typed JSON representation; accepting either parser's inferred type silently would make the test irreproducible.

Missing values, dates, and decimals

CDISC's prose distinguishes **missing `null`** from the empty string `""`. (cdisc-org.github.io) JSON Schema itself treats `null` as a distinct primitive type, and a required-property check concerns presence of a property rather than the clinical meaning of its value. (json-schema.org) (json-schema.org) The pilot should therefore count nulls by field and by row, check that empty strings remain intentional, and compare missingness after reconstruction. It should document how each XPT library maps SAS missing values and blanks; a one-size mapping can hide a changed observation.

Timing variables are represented as ISO 8601 strings in Dataset-JSON. (cdisc-org.github.io) Decimal values can be serialized as strings where that preserves terminating decimal fractions without rounding. (cdisc-org.github.io) These rules imply type-aware tests: compare declared date/time semantics and original precision, and do not coerce every number through a binary floating-point value. JSON itself excludes NaN and Infinity, while canonical JSON schemes can impose their own numeric constraints. (www.rfc-editor.org) (www.rfc-editor.org) The pilot's decimal and exceptional-value cases should therefore record an explicit outcome: exact match, justified transformation, unsupported source state, or failure.

JSON, newline-delimited JSON, and compressed forms

CDISC publishes schemas for the JSON and newline-delimited JSON (NDJSON) representations. (github.com) Its NDJSON layout places metadata first and then one record array per line. (cdisc-org.github.io) The separate CDISC API specification identifies `application/x-ndjson` for a streamed endpoint, while the later compressed Dataset-JSON specification describes zLib-compressed NDJSON with `.dsjc` and a dedicated media type, `application/vnd.cdisc.dataset-json.compressed`. (cdisc-org.github.io) (cdisc-org.github.io) These distinct physical forms are useful for assessing streaming behavior, but a pilot should pin the exact representation, schema file, and tool versions used for each test. A success in one representation should not be assumed to establish success in every representation. (github.com) (www.cdisc.org)

The standard permits zero-record metadata transfer, and the top-level `rows` array is optional. (cdisc-org.github.io) Such files deserve separate cases: an empty data array, metadata-only content, and a genuinely absent row section are distinct parser paths. The test should show which forms each tool accepts and what it reconstructs; an omitted array must not be mistaken for silently lost records. CDISC recommends an attribute order even though JSON object order is not semantically significant. (cdisc-org.github.io) (www.rfc-editor.org)

Implementation Considerations and Process Changes

Establish a canonical contract

The pilot charter should name a **canonical source dataset** as the baseline, then freeze its schema, row identity rule, sort order, character encoding, precision policy, and metadata version. Prefer a source already governed by the organization's SDTM, ADaM, or SEND process. Define-XML provides the richer metadata relationship that a Dataset-JSON file may reference; compare the link and each expected identifier rather than treating a syntactically valid URI as sufficient. (github.com) (cdisc-org.github.io)

For every dataset, record the source file hash and the exact XPT writer, Dataset-JSON writer, XPT reader, and JSON reader versions. Treat conversion as four independent operations: canonical to XPT, canonical to JSON, XPT to reconstructed table, and JSON to reconstructed table. A fifth operation, JSON back to XPT, tests whether an established XPT path remains reproducible. FDA's 2025 exercise explicitly collected original XPT, converted JSON, and reconverted XPT packages, which makes this round-trip pattern a relevant pilot design.

Use two rule layers. **Format conformance** parses each file and checks the pinned Dataset-JSON schema or XPT reader constraints. **Dataset conformance** applies the study's CDISC rules and metadata expectations. CDISC CORE accepts SAS v5 XPORT and Dataset-JSON inputs and tests study data against CDISC conformance rules. (www.cdisc.org) (www.cdisc.org) FDA separately publishes validator and business rules for its review context. (www.fda.gov) (www.fda.gov) A green schema result cannot stand in for a passing CDISC or FDA rule result.

Build the case set before conversion

The case set must exercise the boundaries most likely to alter meaning: empty and long strings, nulls, extreme and high-precision numbers, dates and time zones, Unicode, duplicate and absent keys, row/column reorder, metadata mismatches, and a large file. SAS documentation describes an XPORT engine character-value limit of **200 characters**. (support.sas.com) That limit is a test fixture, not a reason to truncate the canonical source quietly. A value that cannot be carried by the chosen XPT profile needs an explicit, approved exception or a revised scope.

Table 1 is a proposed synthetic test-case matrix. Its cases and pass criteria are editorial implementation choices; they are not CDISC-mandated counts or universal numeric thresholds. The underlying format behaviors come from CDISC, the JSON standards, and SAS documentation. (support.sas.com) (www.rfc-editor.org)

Case	Canonical fixture and perturbation	Required observation	Decision rule
Baseline	Ordinary typed values, stable keys	Both transports reconstruct all rows and variables	Exact semantic equality
Missingness	Null, empty string, blank, and source-specific missing code	Field-level null and empty-string masks	No unexplained change
Decimal	Terminating decimal, near-boundary binary float	Original lexical value and reconstructed numeric value	Exact where representable; field-specific approved exception
Date/time	Date, partial or full timestamp, zone offset	Type, text, and intended instant or calendar value	Compare according to declared type
Text	Non-ASCII, combining mark, long value	UTF-8 decode, Unicode code points, XPT capacity	Preserve exact text or record transport limitation (www.rfc-editor.org) (support.sas.com)
Structure	Reordered object keys, reordered columns, duplicate row key	Column mapping and key diagnostics	Object keys ignored; wrong column mapping or duplicate key fails (www.rfc-editor.org)
Empty	Zero records, omitted `rows`, metadata-only	Reconstructed record count and metadata	Accept only documented forms and declared expectation
Scale	Incremental large dataset and interrupted write	Counts, final hashes, resource log, resumability	No truncated or silently skipped records (cdisc-org.github.io)

The matrix deliberately separates source values that both transports can represent from values that expose XPT limits. That distinction keeps a failure to encode a long value from being misreported as a JSON conversion defect. It also lets a QA lead see whether an accepted exception changes the submitted XPT path or only the experimental JSON path. (support.sas.com)

Canonical Equivalence Protocol

Compare reconstruction, not syntax

For each source dataset, define a stable compound key if the model has one, otherwise assign a row identity in the frozen canonical export. A sorted comparison can then align records across files even when physical row order changes. Keep a second, explicit row-order assertion if order matters to a consumer. JSON objects are unordered collections, while JSON arrays retain element sequence; the Dataset-JSON columns array binds positional values to variables. (www.rfc-editor.org) (www.rfc-editor.org) As a result, object-member order should not fail a semantic comparison, but an unexplained change in column sequence or row association should.

Normalize only under a written contract. Decode JSON as UTF-8, as required for interoperable exchange, and compare exact Unicode code-point sequences first. (www.rfc-editor.org) Unicode normalization can map equivalent strings to one binary representation, but applying it after the fact can conceal an upstream change; make any normalization a named, preapproved comparison mode. (www.unicode.org) (www.rfc-editor.org) Apply the same caution to trimming blanks and converting time zones. The raw value, normalized value, rule, and reason should remain in the exception record.

For numeric values, create per-column comparators from the declared source type, scale, and transport behavior. Integers and exact decimal strings should be tested exactly where both paths preserve them. Binary floats may need a tolerance, but only a documented absolute or relative bound for named fields, justified against the source precision and intended analysis. No source cited here sets a universal tolerance. A match caused by an overly broad tolerance is not proof of equivalence. (www.rfc-editor.org) (www.rfc-editor.org)

Table 2 defines three levels of equality and the evidence each produces. They answer different questions; reporting only a hash or only a row count would miss meaningful changes. NIST's digest and signature guidance supports preserving file integrity and signer identity, not inferring clinical equality from a file hash. (csrc.nist.gov) (csrc.nist.gov)

Level	Comparison	Example of pass	Example of failure or exception
Byte equality	Hash bytes of the same artifact at two handoffs	The delivered XPT matches the archived XPT	Different JSON and XPT hashes are expected; a changed hash of one artifact requires investigation (csrc.nist.gov)
Structural equality	Schema, dataset identifiers, column order/types, row width/count	Declared metadata matches reconstructed shape	Missing column, wrong `itemOID`, or record count mismatch (cdisc-org.github.io)
Semantic equality	Key-aligned values, null pattern, type-aware numeric/date/text rules	Every compared field is exact or has an approved exception	Changed value, lost null, or undocumented rounding

Byte equality is the right check for transfer integrity within one format. Structural equality catches malformed or misdescribed datasets. Semantic equality is the claim that the two transports represent the same intended data, subject to documented exceptions. A pilot should report all three, with failures carried forward instead of collapsing them into one pass flag. (csrc.nist.gov) (www.cdisc.org)

Comparison pseudocode

The following pseudocode is a proposed runbook, not a CDISC validation rule. It assumes a pinned schema and a canonical metadata contract. Each comparator records counts and deviations rather than silently correcting them.

```
for dataset in frozen_manifest:
    source = read_canonical(dataset)
    xpt_bytes  = write_xpt(source, pinned_xpt_writer)
    json_bytes = write_dataset_json(source, pinned_json_writer)
    assert parse_json(json_bytes)
    assert validate_schema(json_bytes, pinned_cdisc_schema)

    xpt_table  = read_xpt(xpt_bytes, pinned_xpt_reader)
    json_table = read_dataset_json(json_bytes, pinned_json_reader)
    back_xpt   = write_xpt(json_table, pinned_xpt_writer)
    back_table = read_xpt(back_xpt, pinned_xpt_reader)

    for table in [xpt_table, json_table, back_table]:
```

```
        assert count_rows(table) == expected_rows(dataset)
        assert count_columns(table) == expected_columns(dataset)
        assert unique_keys(table, declared_key(dataset))
        assert metadata_matches(table, frozen_metadata(dataset))

    for key in sorted(source.keys):
        for field in frozen_metadata(dataset).fields:
            compare_null_state(source[key, field], xpt_table[key, field])
            compare_null_state(source[key, field], json_table[key, field])
            compare_value(source[key, field], xpt_table[key, field],
                          rule_for(field))
            compare_value(source[key, field], json_table[key, field],
                          rule_for(field))
            compare_value(xpt_table[key, field], back_table[key, field],
                          rule_for(field))

    emit_manifest(dataset, sha256(xpt_bytes), sha256(json_bytes),
                  sha256(back_xpt), counts, schema_version,
                  tool_versions, deviations, reviewer_signatures)
```

The manifest should also include source and reconstructed semantic digests computed from a documented canonical representation. A canonical JSON hash can be useful when its numeric restrictions fit the dataset; RFC 8785 requires numbers expressible as IEEE 754 double precision and preserves array order. (www.rfc-editor.org) (www.rfc-editor.org) Where exact decimal strings or other types do not fit that scheme, define an internal typed digest format rather than pretending an ordinary JSON file hash is format-neutral. Retain the bytes used to calculate every digest. (csrc.nist.gov)

Validation Runbook and Submission Gate

Execute and review the tests

Run the tests in fixed order so the error report preserves causality. First verify input inventory, source hashes, metadata version, and key rule. Second serialize both formats and record writer logs. Third parse and validate syntax and schema. Fourth reconstruct typed tables and run the structural and semantic comparisons. Fifth run CDISC conformance checks on each applicable representation. Sixth review exceptions and sign the result. CDISC CORE can process both XPT and Dataset-JSON, but its rule results still need to be tied to the selected rule set and study context. (www.cdisc.org) (www.cdisc.org) (www.cdisc.org)

Schema validation is necessary but narrow. JSON Schema's `required` keyword checks property presence, and its format assertion can be disabled by default under one validation vocabulary. (json-schema.org) (json-schema.org) The implementation should pin a validator configuration and include explicit tests of dates, links, row widths, nulls, and metadata references. Because CDISC's published prose distinguishes null from empty string, a schema pass should be corroborated against that behavior. Record whether every JSON and NDJSON instance was checked against its matching released schema.

Large-file checks should use increasing synthetic volumes and report file bytes, rows, columns, peak memory, elapsed time, parser mode, and interruption behavior. The purpose is to reveal the pilot environment's limits, not to publish an unsupported performance benchmark. CDISC publishes scripts for a roughly **5 GB** XPT example, but warns that its examples are illustrative and may not meet regulator requirements. (cdisc-org.github.io) (cdisc-org.github.io) NDJSON's metadata-first and row-per-line layout permits a streaming test design. A truncated stream must fail the final count and digest checks; a resumed stream must preserve row identity and a complete, revalidated manifest. The test pack should include a deliberately interrupted write to prove that incomplete output cannot be signed as complete. (csrc.nist.gov)

Classify deviations and preserve evidence

Every discrepancy needs a location and a decision owner. Record dataset, key, field, source value, reconstructed XPT value, reconstructed JSON value, comparator rule, machine-readable reason, and reviewer disposition. Useful categories are unsupported XPT source value, declared transformation, metadata inconsistency, parser disagreement, expected ordering change, precision exception, and true data mismatch. Avoid a broad “acceptable difference” bucket. FDA's technical guide asks sponsors to explain meaningful discrepancies in a Reviewer Guide; the internal evidence should be detailed enough to support that explanation if relevant. (www.fda.gov)

ICH E6(R3) says data reconciliation should be implemented as appropriate after electronic exchange or migration. A signed exception record should preserve the actual artifacts, tool configurations, comparison outputs, approval identity, date, and the reason an exception is or is not acceptable. A digest detects later change, while a digital signature can authenticate who approved the report. (csrc.nist.gov) (csrc.nist.gov) Re-running the same pinned inputs should reproduce the comparison result; if it does not, the run cannot be treated as complete.

Table 3 is a proposed submission-readiness gate. Its submission column should be refreshed against the applicable FDA catalog and guide immediately before filing.

Gate	Technical pilot decision	Submission decision
Inventory and scope	All declared datasets, versions, and cases accounted for	Confirm study applicability and current supported format
Format and metadata	JSON schema, IDs, row widths, Define-XML reference pass	Apply current technical conformance guide and catalog
Semantic comparison	Exact matches or signed, field-specific exceptions	Retain XPORT Version 5 for FDA electronic dataset submissions (FDA Study Data Technical Conformance Guide)
Rules and review	CDISC rule outputs and discrepancy log reviewed	Apply FDA validator/business rules and reviewer-guide process as applicable (www.fda.gov) (www.fda.gov)
Reproducibility	Files, hashes, tools, logs, signatures archived	Submit only through an authorized format and route (csrc.nist.gov)

A passing pilot means the technical evidence is coherent and repeatable. It is not a regulatory waiver. The format decision should be rechecked against the applicable guide and catalog at filing; FDA's study-data resources describe further Dataset-JSON testing. (www.fda.gov) Thus the go/no-go record should have two signatures: one for equivalence evidence and one for the submission-format determination under the documents effective at filing.

Data Analysis and Evidence

The public evidence supports a **pilot decision**, not a universal performance or replacement claim. FDA described its 2024 project as a feasibility test whose aim was lossless transport without significant operational impact. A PHUSE/CDISC poster reported that the 2024 pilot met its objectives and that Dataset-JSON could function as an XPT alternative in that test setting. (phuse.s3.eu-central-1.amazonaws.com) (advance.hub.phuse.global) PHUSE also noted that regulatory review tools would need updating to take advantage of added metadata and flexibility. (advance.hub.phuse.global) These are bounded project findings, not a sponsor-independent error rate.

FDA's later evidence is broader and more cautious. Its 2025 exercise included **five clinical studies and three nonclinical studies**, with original XPT, converted JSON, and converted-back XPT packages. (www.fda.gov) FDA's published assessment concluded that Dataset-JSON did not demonstrate superiority over existing processes. (www.fda.gov) This sequence is not contradictory if interpreted by question: the earlier pilot asked whether transport was feasible; the later assessment asked whether adoption improved current processes. A sponsor's equivalence test should likewise distinguish data preservation from operational benefit.

The quantitative core of an internal report should contain raw denominators. For each dataset, report **expected rows**, rows parsed from XPT, rows parsed from JSON, rows parsed after the return trip, expected columns, missing-value counts by field, key collisions, exact field matches, approved tolerance matches, and unresolved mismatches. Report the number of files and bytes actually processed, plus tool and schema versions. These are proposed measures, not published universal thresholds. Row counts alone cannot prove that the right rows or values survived; key-aligned field checks supply the missing evidence. (csrc.nist.gov)

If a pilot evaluates numeric tolerances, publish both the number of values that needed the exception and the largest observed absolute and relative difference by field. Keep the original lexical value when decimal strings are used, since CDISC explicitly permits strings to avoid rounding. For text, publish exact mismatches and any approved normalization mode. JSON's UTF-8 rule and Unicode's normalization mechanism explain why a visually similar value can still differ at the byte level. (www.rfc-editor.org) (www.unicode.org) For files, keep separate hashes for source, delivered XPT, JSON, and returned XPT; the digest answers artifact integrity, not semantic equality. (csrc.nist.gov) (www.rfc-editor.org)

No public source used here establishes a general conversion error rate, latency threshold, or cost saving for Dataset-JSON 1.1. The internal acceptance threshold should therefore be defined from each sponsor's data criticality, XPT limitations, and intended use before results are seen. The small, published FDA study count and the changing policy context make a transparent exception log more informative than a single pass percentage. (support.sas.com)

Implications and Future Directions

The immediate architectural choice is to make the canonical dataset independent of either transport. That allows the same curated records and metadata to feed XPT and Dataset-JSON and lets the comparison isolate serialization behavior. It also prevents a JSON-to-XPT conversion from becoming the only copy of source truth. FDA's three-version exercise provides a public example of a round-trip evaluation structure, while ICH calls for appropriate reconciliation.

Standards governance should pin the **Dataset-JSON 1.1 release**, schema artifact, metadata version, CDISC rule set, conversion software, and comparator policy in one change-controlled manifest. CDISC's schema covers JSON and NDJSON forms, and CDISC CORE accepts both XPT and Dataset-JSON; those capabilities make dual-path tests feasible, but they do not remove the need to review every field-level deviation. (www.cdisc.org) Re-run the case matrix when any pinned component changes, particularly a writer, reader, schema, or date/decimal policy.

As of September 2026, FDA's Study Data Technical Conformance Guide specifies XPORT Version 5 for electronic dataset submissions, and FDA's September 2026 action plan describes testing Dataset-JSON as a potential replacement for XPT v5. The policy watch list should include the Data Standards Catalog, the Study Data Technical Conformance Guide, CDISC release and errata pages, and public FDA pilot updates. A future accepted format would change the submission gate, not the value of a transport-neutral equivalence pack. (www.fda.gov) (www.cdisc.org)

IntuitionLabs' published service descriptions include data engineering, implementation, and advisory work; for an adjacent consultancy, the useful role is to help teams define the data contract, instrument the comparison, and transfer evidence into their quality process. The decision to submit a format still belongs to the sponsor under the regulator's current requirements.

Frequently Asked Questions (FAQs)

Is Dataset-JSON 1.1 currently required for FDA study-data submission?

No. As of September 26, 2026, the FDA's June 2026 technical guide specifies XPORT Version 5 for electronic datasets, and FDA describes further Dataset-JSON testing. (www.fda.gov) (www.fda.gov) The applicable catalog and guide should be checked again immediately before any actual submission. A pilot result by itself does not alter the accepted format.

How should Dataset-JSON be validated against XPT?

Start from one frozen canonical dataset. Generate both transports, parse both back to typed tables, and compare dataset identity, variable metadata, row and column counts, keys, null masks, exact strings and dates, and numeric values under field-specific rules. Then run CDISC conformance checks and archive files, logs, hashes, and approvals. CDISC CORE accepts both inputs, and the Dataset-JSON specification defines the relevant metadata and null behavior. (csrc.nist.gov)

Does a schema pass prove equivalence?

No. A schema pass establishes that the JSON instance meets the selected structural schema rules under the selected validator settings. JSON Schema's required-property and format behavior shows why that is narrower than record-level meaning. (json-schema.org) (json-schema.org) A valid file can still have a wrong column mapping, changed value, or unresolved metadata link, so the runbook separately checks structure, semantics, and conformance rules.

Must JSON and XPT hashes match?

No. Hashes are for detecting changes to a particular artifact, such as whether the archived XPT is identical to the delivered XPT. NIST's guidance describes message digests as a change-detection mechanism. (csrc.nist.gov) Compare reconstructed, typed values for cross-format equivalence. If a semantic digest is used, publish its canonicalization rules and numeric limitations. (www.rfc-editor.org) (www.rfc-editor.org)

What should happen when a value cannot round-trip?

Capture the source value, both reconstructed values, the relevant type and comparator policy, and the exact transport limitation or software behavior. A long character value that exceeds the selected XPT profile's capability should be classified explicitly; SAS documents a 200-character XPORT engine limit. (support.sas.com) Reject unexplained change. If an exception is justified, require a named owner, approval, and an unchanged accepted submission path. FDA recommends explaining meaningful discrepancies in the relevant Reviewer Guide.

Conclusion

Dataset-JSON 1.1 can be piloted responsibly when the sponsor treats **equivalence as an evidenced claim about data meaning**. The core design is simple: freeze a canonical dataset, serialize XPT and JSON independently, reconstruct both, compare by stable keys and declared types, and preserve the bytes, hashes, rule results, and signed exceptions. CDISC's metadata and missing-value conventions provide necessary checks, while general JSON rules explain why textual similarity and object ordering are not enough. (www.rfc-editor.org)

The current public record supports caution at the submission boundary. FDA has tested Dataset-JSON but continues to evaluate whether to adopt it as a submission transport. (www.fda.gov) A technical pass is useful evidence for architecture and standards governance. It should be signed separately from the decision that a particular regulator accepts a particular submission format.

The most defensible result is a reproducible package that reveals exactly where bytes, structure, or meaning differ, and why each difference was accepted or rejected. That package can be rerun when schemas, software, or regulatory policy change. For FDA electronic dataset submissions, the June 2026 guide specifies XPORT Version 5 (FDA Study Data Technical Conformance Guide). (csrc.nist.gov) (www.fda.gov)

Source links

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

1. https://www.cdisc.org/standards/data-exchange/dataset-json/dataset-json-v1-1
2. https://cdisc-org.github.io/DataExchange-DatasetJson/doc/dataset-json1-1.html
3. https://github.com/cdisc-org/DataExchange-DatasetJson/blob/master/doc/dataset-json1-1.md
4. https://www.fda.gov/drugs/news-events-human-drugs/dataset-json-pilot-report-and-next-steps-07252024
5. https://www.federalregister.gov/documents/2025/04/09/2025-06051/electronic-study-data-submission-data-standards-clinical-data-interchange-standards-consortium
6. https://www.fda.gov/media/193422/download
7. https://www.fda.gov/media/153632/download
8. https://www.fda.gov/media/194922/download?attachment=
9. https://www.fda.gov/industry/study-data-standards-resources/study-data-submission-cder-and-cber
10. https://www.fda.gov/industry/fda-data-standards-advisory-board/study-data-standards-resources
11. https://www.rfc-editor.org/rfc/rfc8259.html
12. https://csrc.nist.gov/pubs/fips/180-4/upd1/final
13. https://csrc.nist.gov/pubs/fips/186-5/final
14. https://database.ich.org/sites/default/files/ICH_E6%28R3%29_Step4_FinalGuideline_2025_0106.pdf
15. https://intuitionlabs.ai/
16. https://json-schema.org/draft/2020-12/json-schema-validation
17. https://www.rfc-editor.org/rfc/rfc8785.html
18. https://cdisc-org.github.io/DataExchange-DatasetJson/doc/dataset-json-ndjson1-1.html
19. https://cdisc-org.github.io/DataExchange-DatasetJson-API/
20. https://cdisc-org.github.io/DataExchange-DatasetJson/doc/compressed-dataset-json1-1.html
21. https://www.cdisc.org/core
22. https://intuitionlabs.ai/articles/define-xml-2-1-implementation-guide
23. https://support.sas.com/techsup/notes/v8/1/528.html
24. https://www.unicode.org/reports/tr15/tr15-58.html
25. https://cdisc-org.github.io/DataExchange-DatasetJson/examples/
26. https://phuse.s3.eu-central-1.amazonaws.com/Archive/2024/Connect/EU/Strasbourg/POS_PP33.pdf
27. https://advance.hub.phuse.global/wiki/plugins/viewsource/viewpagesrc.action?pageId=26817392
28. https://www.fda.gov/regulatory-information/search-fda-guidance-documents/data-standards-catalog

THE PUBLISHER

About IntuitionLabs

IntuitionLabs is an AI consulting, custom software development and data engineering firm serving pharmaceutical, biotechnology, medical-device and other life-science organizations. We work with clinical, regulatory, medical-affairs, commercial, quality and IT teams to connect technology decisions with the work people need to accomplish.

AI consulting and adoption

Our AI enablement services cover readiness assessments, use-case selection, governance and policies, team workshops, adoption measurement and ongoing advisory support. We help organizations structure the information layer behind AI: source material, context, permissions and maintained knowledge that make generated answers useful and reviewable. Private LLM inference and hosted AI options support teams evaluating how to operate AI with appropriate control over their data and infrastructure.

Software, data and life-science workflows

IntuitionLabs develops custom software for pharma and biotech, integrates enterprise systems, and builds data engineering and business intelligence solutions. Areas of focus include AI agents, regulatory research, medical writing, medical affairs, CMC information, competitive intelligence and clinical-document workflows. Our eTMF intelligence work includes cross-system reconciliation and inspection-readiness support.

Enterprise platforms and regulated delivery

We provide Veeva services, application support, managed services, integrations and custom applications, alongside enterprise content work involving platforms such as Egnyte. For regulated workflows, our services include GxP enablement, computer-system validation and software development addressing 21 CFR Part 11 requirements. The applicable controls, validation responsibilities and acceptance criteria are defined for each engagement.

Work with IntuitionLabs

Explore AI enablement, pharma and biotech software development, data engineering and BI, and Veeva services. Contact IntuitionLabs to discuss your workflow, information sources and implementation needs.

IntuitionLabs publishes educational research to help life-science teams make informed technology decisions. Coverage of a product or organization does not imply a client relationship, endorsement or partnership.

Website: https://intuitionlabs.ai