

Breaking Free from the XPT: Exploration of Dataset-JSON as an Alternative Transport File to Regulatory Agencies

Ben Straub, GSK, Philadelphia, United States
Sam Parmar, Pfizer, New York City, United States
Nick Masel, Johnson and Johnson, Raleigh, United States

ABSTRACT

The SAS XPORT (XPT) transport file has been the de facto interchange format for SDTM and ADaM datasets in regulatory submissions for decades. While durable and widely supported, XPT is inherently constrained by legacy design decisions and is not a natural “first choice” format within modern open-source data ecosystems. The CDISC Dataset-JSON standard provides a contemporary alternative designed to represent clinical datasets and metadata using JavaScript Object Notation (JSON), enabling straightforward interaction across many programming languages and validation toolchains.

R Consortium R Submission Pilot 5 explored a fully reproducible, publicly accessible submission-like package centered on **R-generated ADaM datasets** and delivery of datasets in **Dataset-JSON v1.1**. The pilot demonstrates (1) converting study data inputs into analysis-ready datasets, (2) serializing datasets to Dataset-JSON using R, (3) regenerating key tables/figures from the R-derived datasets, and (4) implementing quality control comparisons to support confidence that artifacts match expectations and prior baselines.

This paper summarizes the background and motivations for Dataset-JSON, the R Consortium Working Group context, and the Pilot 5 implementation and lessons learned for teams evaluating Dataset-JSON as a viable alternative transport format for future submissions.

INTRODUCTION

What we are going to talk about

This paper covers four related topics:

1. Why the industry is reevaluating long-standing dataset transport assumptions (with XPT as the historical default).
2. What is Dataset-JSON, where it came from, and why it matters.
3. The R Consortium R Submissions Working Group context that enabled Pilot 5.
4. R Submission Pilot 5 use of Dataset-JSON as replacement for XPT for FDA Submission.

A high-level framing: why XPT feels like yesterday’s solution

XPT is not “bad”—as it is widely supported and deeply embedded in regulatory processes. However, when viewed through a modern software lens, it unduly burdens the acceptance of modern software practices and adoption of new data standards. In this section, we will lay out why the XPT is problematic.

Outdated Technology: The format was defined in the 1980s and was designed for data transfer between SAS applications on different operating systems. It is not aligned with modern data architectures or standards. The FDA adopted it as non-proprietary data format to submit for drug submissions and it has since stuck as the de-facto transport.

Limited Data Characteristics: Variable Names are restricted to a maximum of 8 characters and are case-insensitive. Variable Labels are limited to 40 characters. Character Variable Lengths cannot exceed 200 characters or bytes. Variable Types are limited to US ASCII for character variables and IBM INTEGER and DOUBLE for numeric variables.

Poor Handling of International Characters: It only supports US ASCII character encoding, meaning it cannot natively handle multibyte characters or UTF-8. Representing international characters requires “tricks” or workarounds, which can lead to encoding and interpretation issues.

Lack of Robust Metadata: The XPT format has no direct way to embed rich metadata within the file itself. It relies on separate define.xml files, which introduces the risk of data and metadata mismatches if not carefully managed.

Inefficient Storage: The format is inefficient, often leading to wasted storage space. It allocates fixed space for variables, and missing or shorter values are padded with blanks, which can result in up to 70% wasted space. It also does not support data compression.

Ecosystem asymmetry: XPT is universally known in SAS-centric workflows, but is less native to open-source tooling and data engineering stacks.

This is the motivation for exploring Dataset-JSON: not to declare a villain, but to ask whether an alternative transport can reduce friction and better fit modern, reproducible, open toolchains while continuing to support reviewer needs.

Dataset-JSON

A brief “history lesson”

In 2022, the FDA evaluated alternatives to serve as possible replacements to SAS V5 XPT. The FDA determined that JSON was the optimal modern format to serve as a replacement to SAS V5 XPT. CDISC Dataset-JSON v1.0 was released in August 2023 and a series of hackathons and workshops were spun up to investigate and build tooling. During this time, the [datasetjson](#) R package supporting Dataset-JSON v1.0 was released in October 2023, take note as this will be a crucial R package for later discussion. In 2024, additional workshops were held to finalize an updated v1.1 standard as well a Public Review. In the spring of 2025, the FDA via the [Federal Register notice](#) re-affirmed the standard and signaled openness to it being adopted.

What Dataset-JSON is (practically)

Dataset-JSON is a CDISC standard intended to represent clinical tabular datasets using JSON. It exists in the context of increasing adoption of structured data standards and the need to support interoperability beyond a single vendor ecosystem. It is designed to meet a wide range of data exchange scenarios including regulatory submissions and API-based data sharing. Dataset-JSON can be thought of as a standard way to serialize a dataset plus its metadata into JSON. In practical terms it enables:

- transport of rows/records in JSON form
- representation of column metadata (name, label, type, etc.)
- consistent schema that can be validated
- contains one dataset per file
- each dataset can optionally reference a Define-XML file

The standard addresses requirements from the PHUSE 2017 “Transport for the Next Generation” paper. It follows a lowerCamelCase structure with three main components: top-level metadata (including creation datetime, version, and optional Define-XML references), column metadata (specifying variable names, labels, and data types), and row data arrays. It supports multiple data types with special handling for decimal precision (exchanged as strings to avoid rounding) and ISO 8601 format for date/time variables, with missing values represented as null. The standard also offers an NDJSON format (.ndjson) that enables streaming of large datasets by placing metadata on the first line and individual data rows on subsequent lines, allowing processing without loading entire datasets into memory.

Collaboration framing (CDISC / PHUSE / FDA / R Consortium)

Industry adoption of submission changes typically succeeds only where standards bodies, industry groups, and regulators align around feasibility and reviewer usability. Dataset-JSON exists within this collaborative ecosystem where CDISC defines standards, PHUSE supports applied use cases and community practice, and FDA engagement is critical to evaluate submission and review implications. The R Consortium Submissions Working Group has been working with the FDA for many years on R-based Submissions called Pilots. Pilot 5 is explicitly framed as an FDA–industry collaboration through the R Consortium and serves as a public “show your work” style example where Dataset-JSON was used rather than XPTs. Pilot 5 is covered in more detail later in the paper.

WHY DATASET-JSON OVER OTHER MODERN FORMATS

While modern formats like Parquet offer performance advantages for big data applications, Dataset-JSON provides specific benefits for regulatory submissions:

- **Self-contained metadata:** Variable labels, types, and CDISC-specific metadata are embedded directly in the file, eliminating external dependencies.
- **Human-readable and audit-friendly:** Text-based JSON enables review, comparison, and validation using standard development tools without specialized software.
- **Universal parsing:** JSON parsers exist in virtually every programming language, reducing implementation barriers across toolchains.
- **Data Exchange** Dataset-JSON supports both API and file based exchange.

Binary formats sacrifice these readability and auditability characteristics—considerations that matter critically in regulatory review contexts where transparency and reproducibility are paramount.

R CONSORTIUM WORKING GROUP

What is the focus?

The R Consortium R Submissions Working Group has focused on demonstrating that **open-source, R-based submission packages** can be organized, reproduced, and reviewed in a way that meets regulatory expectations. A key theme across pilots is transparency: making the full submission-like artifacts available (data, code, documentation, and outputs), enabling learning and reuse across industry. The mission is:

- Easier R-based clinical trial regulatory submissions **today**
 - by showing open examples of using current submission portals
- Easier R-based clinical trial regulatory submissions **tomorrow**
 - by collecting feedback and influencing future industry and agency decisions on system/process setup

Each Pilot's goals as well supporting documentation can be found on R Consortium's Submissions Working Group website. The link can be found below.

A Pilot Brief

A quick synopsis of the four Pilots:

- Pilot 1: TLFs in R, SDTM/ADaM from SAS XPT - 2021
- Pilot 2: TLFs are packaged into a Shiny App - 2022/2023
- Pilot 3: Rebuilt 5 ADaM datasets in R, still output XPT - 2023/2024
- Pilot 4: Pilot 2 delivered as webassembly and as a container- 2023-2026

Successes (as demonstrated by the pilots)

Across the pilot work, the working group has demonstrated:

- submission-like packaging conventions (eCTD-style organization),
- reproducible execution from raw/source materials through outputs,
- public documentation (e.g., ADRG) aligning to reviewer needs,
- practical workflows that teams can fork and adapt.

Each Pilot has worked extensively with FDA reviewers to discuss issues from installation of packages, data issues, derivation in R and more. Once the Pilot is completed, the FDA provides recognition of the successful review with a "Statistical Review and Evaluation" letter. Pilot 3's letter is linked in the Reference materials.

Problems identified and iterations

Exploring new transport mechanisms and reproducible workflows surfaces practical issues that do not always appear in "standards only" discussions, including:

- ensuring variable labels, formats, and metadata persist predictably across conversions,
- managing numeric precision and representation choices,
- aligning generated deliverables with reviewer expectations and available tooling,
- maintaining robust QC evidence when changing transport mechanisms.

Pilot 5 contains concrete code and QC artifacts illustrating how these issues can be surfaced, addressed, and documented in a public workflow. Pilot 5 uses [Dataset-JSON v1.1](#) and demonstrates converting and producing these artifacts using R in a transparent manner.

PILOT 5

Focus of Pilot 5 (Dataset-JSON + R-generated ADaM)

Pilot 5 objectives were delivering a publicly accessible R-based submission package using Dataset-JSON, and expanding prior pilot approaches by generating ADaM datasets using R. The SDTM and ADaM datasets used in Pilot 1 were produced in SAS as XPT files. Pilot 3 built on these 3 tables and 1 figure by now rebuilding the ADaM datasets in R and outputting as XPT files. It is important to note that only 5 datasets were needed to produce the 3 tables and 1 figure. Remember the Pilots are more about investigating how the process works with R and less on the content created in R - hence the small submission package to help stay focused.

Now enter Pilot 5. Building off the success of Pilot 1 and Pilot 3, Pilot 5 wanted to see if all the XPT files in the submission package could be converted to Dataset-JSON using R. Luckily, the amazing `datasetjson` R package was available to help do a lot of the heavy lifting. See below for more information on it. As most of the code was in place for the tables, figures and ADaMs, we only had to add minimal code to produce the R created ADaMs into Dataset-JSON.

The `datasetjson` R package also had a handy conversion program for moving existing XPTs to Dataset-JSON. The Pilots do not have access to the raw data and so the SDTM could not be re-created from scratch. The location for this conversion program is provided in the references.

Pilot 5 artifacts, which are currently viewable, include a public repository for development work. This repository contains R programs, SDTM and ADaM data, QC Reports from Bots, closed pull requests with discussion, and meeting notes along with LLM-supported Development ADRG. Additionally, there is a public repository designated for final submission work, structured according to the M1/M5 framework, which houses final R programs, final JSON data, and the final ADRG along with other supporting documents. Please find the public GitHub Repos for development and submission work linked in the References.

Way of working: reproducibility-first

Pilot 5 is organized to be rerunnable and inspectable, including:

- dataset generation programs under `pilot5-submission/pilot5-programs/` (e.g., `adsl.r`, `adae.r`, `adlbc.r`, `adtte.r`),
- outputs and derived artifacts under `pilot5-submission/pilot5-output/`,
- documentation such as ADRG and eCTD README.

Most importantly, Pilot 5 treats comparability as an artifact. For example:

- `qcReport.qmd` compares Pilot 5 ADaM datasets against prior baseline artifacts using `diffdf`.
- `tlf-qc.qmd` supports comparing generated output artifacts, including text diffs and image-style comparisons.

AI infrastructure (where it helped)

Pilot 5 includes experimental automation to assist documentation workflows, including LLM-assisted pipelines for extracting information (e.g., variables, datasets, and filters) from analysis code to support reviewer documentation tables (see `adrg/llm-adrg-utils/llm_pipeline.qmd`). This is not positioned as a replacement for authoring, but as a mechanism to reduce manual effort and improve consistency when building reviewer-facing documentation from code.

Success of the Pilot (what Pilot 5 demonstrates)

Pilot 5 successfully submitted the submission package via the eCTD portal in the Fall of 2025 demonstrating, with a public reference implementation, that:

- Dataset-JSON can be produced in an R-driven pipeline with explicit metadata management.
- A submission-like artifact can be packaged in a reviewer-friendly structure.
- Outputs can be regenerated from the provided code and datasets.
- QC comparisons can be scripted and reported to support confidence in the fidelity of the produced artifacts.

Trials and tribulations (what you run into in the real world)

Pilot 5 encountered a couple of challenges along the way in bringing our submission package to the FDA grouped into standards and technical hurdles.

For standards, the first issue was that the standard of Dataset-JSON had just switched from 1.0 to 1.1 so we needed to understand the differences for discussion with FDA. Second, we make use of the Validation software Pinnacle 21 Community v4.1.0 (the latest available at the time of writing), which helps ensure compliance for submissions packages. Unfortunately, the [standard 1.1 was not available](#) in either the community or enterprise version of the software. A bit of setback, but we worked with our FDA partners to address and have documented this in the FDA submission. Once the standard 1.1 is available in the Pinnacle 21 we might re-submit.

For technical, the first issue was learning how the metadata is applied and stored in the Dataset-JSON format as we have to ensure consistency. The second issue was learning about converting float variables to decimal data type in the JSON output. This was a bit of a thorn in our side in the initial development. Eventually, a wrapper function was built to help provide consistency across all the datasets. There are also a few limitations to the R package `datasetjson` which are being [looked into by the team](#).

As of writing, we have not received confirmation from FDA on a successful review of the submission. The FDA requested some minor re-work of some of the code which required a re-submission in January of 2026. We hope that at the presentation that the FDA is successful in its of Pilot 5.

MINI EXAMPLE: FROM AN ADAM DATA FRAME TO DATASET-JSON

This section provides a minimal illustration of how Dataset-JSON can be produced in R in the same spirit as the Pilot 5 workflow. The goal is to show that Dataset-JSON creation is a reproducible, scriptable step that can be version-controlled and QC'd like any other artifact.

Pilot 5 conversion logic is exemplified by scripts such as `pilot5-submission/pilot5-programs/convert_xpt_to_datasetjson.r` (read source data, ensure labels, derive column metadata, write Dataset-JSON).

Below is a simplified “toy” example mirroring that concept.

```
library(datasetjson)
library(tibble)

adsl <- tibble(
  STUDYID = "CDISCPIL0T01",
  USUBJID = c("01-701-1015", "01-701-1023"),
  TRT01A = c("Placebo", "Xanomeline Low Dose"),
  AGE = c(67, 72)
)

# Dataset label (Pilot 5 ensures labels exist; may be derived from source metadata)
attr(adsl, "label") <- "Subject-Level Analysis Dataset"

# Minimal column metadata (Pilot 5 derives more comprehensive metadata)
columns <- tibble::tibble(
  itemOID = names(adsl),
  name = names(adsl),
  label = c("Study Identifier", "Unique Subject Identifier", "Actual Treatment for Period 01", "Age"),
  dataType = c("string", "string", "string", "integer")
)

ds_json <- datasetjson::dataset_json(
  adsl,
  item_oid = "ADSL",
```

```

name = "adsl",
dataset_label = attr(adsl, "label"),
columns = columns
)

json_text <- datasetjson::write_dataset_json(ds_json, pretty = TRUE)
# writeLines(json_text, "adsl.json")
cat(json_text)

{
  "datasetJSONCreationDateTime": "2026-02-09T21:31:30",
  "datasetJSONVersion": "1.1.0",
  "itemGroupOID": "ADSL",
  "records": 2,
  "name": "adsl",
  "label": "Subject-Level Analysis Dataset",
  "columns": [
    {
      "itemOID": "STUDYID",
      "name": "STUDYID",
      "label": "Study Identifier",
      "dataType": "string"
    },
    {
      "itemOID": "USUBJID",
      "name": "USUBJID",
      "label": "Unique Subject Identifier",
      "dataType": "string"
    },
    {
      "itemOID": "TRT01A",
      "name": "TRT01A",
      "label": "Actual Treatment for Period 01",
      "dataType": "string"
    },
    {
      "itemOID": "AGE",
      "name": "AGE",
      "label": "Age",
      "dataType": "integer"
    }
  ],
  "rows": [
    [
      "CDISCPILLOT01",
      "01-701-1015",
      "Placebo",
      67.0
    ],
    [
      "CDISCPILLOT01",
      "01-701-1023",
      "Xanomeline Low Dose",
      72.0
    ]
  ]
}

```

Even in this simplified example, Dataset-JSON highlights two operational advantages that Pilot 5 leverages: - metadata is handled explicitly and can be audited, - the resulting artifact is plain text JSON and fits naturally into reviews and automated checks.

CONCLUSION

CDISC and PHUSE working with the FDA laid the groundwork to re-imagine how drug submissions are delivered to the FDA via transport files with Dataset-JSON. Building off of their work, the R Consortium R Submission Pilot 5 provides a concrete, public example demonstrating that Dataset-JSON can be used as a realistic alternative transport format in a submission-like package while maintaining reproducibility and reviewability. The pilot shows that Dataset-JSON can support the same scientific intent and reproducibility expectations traditionally associated with XPT-based submissions. This is achieved by pairing Dataset-JSON with transparent dataset derivation code, explicit metadata management, and reviewer-oriented packaging and documentation (eCTD-like structure + ADRG). First-class QC reporting (dataset and output comparisons) further strengthens confidence in the approach. Together, these elements align more naturally with modern open-source data workflows while maintaining regulatory standards.

REFERENCES

CDISC. (n.d.). CDISC SDTM & ADaM Pilot Project (CDISCPilot01 source materials). GitHub. Retrieved February 9, 2026, from <https://github.com/cdisc-org/sdtm-adam-pilot-project>

Electronic Study Data Submission; Data Standards; Clinical Data Interchange Standards Consortium Dataset-JavaScript Object Notation; Request for Comments. (2025, April 9). Federal Register, 90(69), 2025-06051. <https://www.federalregister.gov/documents/2025/04/09/2025-06051/electronic-study-data-submission-data-standards-clinical-data-interchange-standards-consortium>

GitHub. (2025, April 14). GitHub Copilot (GPT-4.1 version) [Large language model]. <https://github.com/copilot/>

OpenAI. (2025, April 14). GPT-4.1 (version) [Large language model]. <https://platform.openai.com>

OpenAI. (2025, November 12). GPT-5.1 (version) [Large language model]. <https://platform.openai.com>

R Consortium. (2024). Statistical review and evaluation. GitHub. https://github.com/RConsortium/submissions-wg/blob/main/_Documents/Summary_R_Pilot3_Submission.pdf

R Consortium. (2025). R Consortium R submission Pilot 5 development repository. GitHub. <https://github.com/RConsortium/submissions-pilot5-datasetjson>

R Consortium. (2025). R Consortium Pilot 5 eCTD package overview (Quarto source). GitHub. https://github.com/RConsortium/submissions-pilot5-datasetjson/blob/main/ectd_readme/README.qmd

R Consortium. (2026). Pilot 5 development repository. GitHub. <https://github.com/RConsortium/submissions-pilot5-datasetjson>

R Consortium. (2026). Pilot 5 submission repository. GitHub. <https://github.com/RConsortium/submissions-pilot5-datasetjson-to-fda>

U.S. Food and Drug Administration. (n.d.). Electronic Common Technical Document (eCTD) overview. Retrieved February 9, 2026, from <https://www.fda.gov/drugs/electronic-regulatory-submission-and-review/electronic-common-technical-document-ectd>

Atorus Research. (2025). Converting from XPT. In datasetjson R package. https://atorus-research.github.io/datasetjson/articles/converting_files.html

RECOMMENDED READING

Pilot 5 ADRG preview (public links referenced by the Pilot 5 repo README):

- HTML: <https://rpodcast.quarto.pub/pilot-5-aanalysis-data-reviewers-guide/>
- PDF: <https://rsubmission-draft.us-east-1.linodeobjects.com/pilot5-adrg-quarto-pdf.pdf>

QC reports in the Pilot 5 repo:

- qcReport.qmd
- tlf-qc.qmd

LLM-assisted ADRG utilities (Pilot 5 repo):

- `adrg/llm-adrg-utils/llm_pipeline.qmd`

ACKNOWLEDGMENTS

The authors acknowledge the work of **PHUSE**, **CDISC**, **R Consortium** and the **R Submissions Working Group**, as well as the broader set of contributors who made the Pilot 5 repository, documentation, and reproducible workflows publicly available for the benefit of regulators, industry, and the open-source community.

CONTACT INFORMATION

Ben Straub
GSK
Philadelphia, United States

Sam Parmar
Pfizer
New York City, United States

Nick Masel
Johnson and Johnson
Raleigh, United States

Brand and product names are trademarks of their respective companies.