Paper DS02

Next-Gen Transport Standards: Dataset-JSON from a Programmer's Lens

Venkata Koteswara Rao Rayala, Merck Specialities Private Limited, Hyderabad, India

ABSTRACT

The rapid advancement of AI and digital transformation is prompting a reassessment of traditional data submission methods in clinical research. In response, regulatory bodies such as the FDA and CDISC have introduced Dataset JSON-a modern transport format designed to enhance efficiency, improve metadata transparency, and enable seamless integration with evolving technologies. A key benefit of Dataset JSON is support for direct dataset access via Define.xml and enhanced human readability. This paper offers a programmer-centric analysis of Dataset JSON, comparing it with legacy formats like SAS XPT. It examines common implementation challenges, including schema, legacy system compatibility, and limited tool support, while proposing practical mitigation strategies. Tools such as Pinnacle 21 are discussed for validation and compliance. Additionally, the paper outlines practical approaches to generating Dataset JSON using SAS or R, equipping programmers with hands-on guidance. The objective is to facilitate a smooth transition toward more efficient, future-ready clinical data submissions.

INTRODUCTION

Regulatory submissions in clinical research are evolving rapidly to meet the demands of digital transformation. Legacy formats such as SAS XPT, though historically reliable, are increasingly inadequate due to their rigid structure, limited extensibility, and poor interoperability with modern platforms. To address these limitations, the FDA and CDISC have introduced Dataset-JSON, a modern data exchange standard designed to support regulatory submissions, API-driven workflows, and enhanced metadata transparency.

Key Features of Dataset-JSON

- **Lightweight and human-readable**: Built on the widely adopted JSON standard, making it easier for programmers and reviewers to interpret.
- **Define-XML integration**: Enables richer metadata and improves transparency for regulatory review.
- **Extensible design**: Supports new metadata and evolving use cases without structural constraints.
- **Future-ready alignment**: Compatible with cloud-native and AI-enabled ecosystems.

Benefits of Modernization

- **Efficiency**: Faster preparation and validation of datasets.
- **Transparency**: Metadata-rich submissions improve regulatory review.
- **Interoperability**: Seamless integration with APIs and modern tools.
- **Future readiness**: Extensible design supports evolving standards.

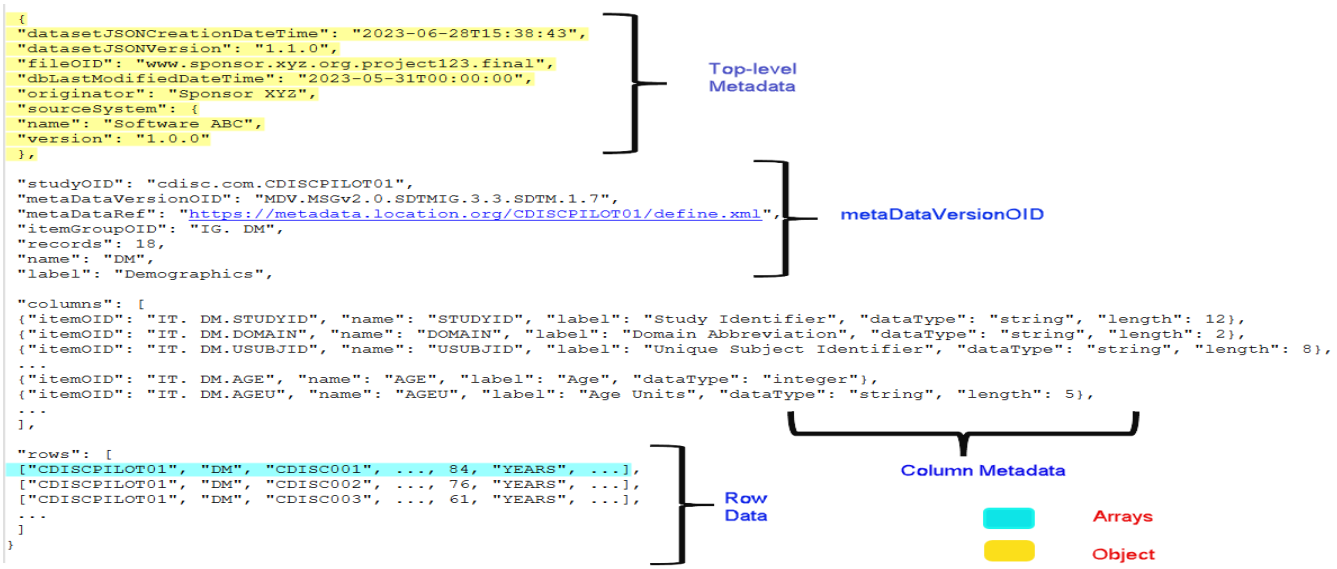
Challenges and Mitigation

- **Challenges**: Schema validation, limited tool support, and legacy system compatibility.
- **Mitigation strategies**: Tools such as Pinnacle 21 for compliance, and practical workflows in SAS and R for generating Dataset-JSON outputs.

Ultimately, modernizing regulatory submissions through Dataset-JSON represents a paradigm shift—moving from static, legacy formats to dynamic, interoperable, and transparent data pipelines that align with the digital evolution of clinical research.

HIERARCHICAL STRUCTURE OF DATASET-JSON

Dataset-JSON follows a hierarchical organization, where information flows from general metadata at the top level down to detailed subject-level data rows. This structure ensures clarity, transparency, and alignment with CDISC standards.



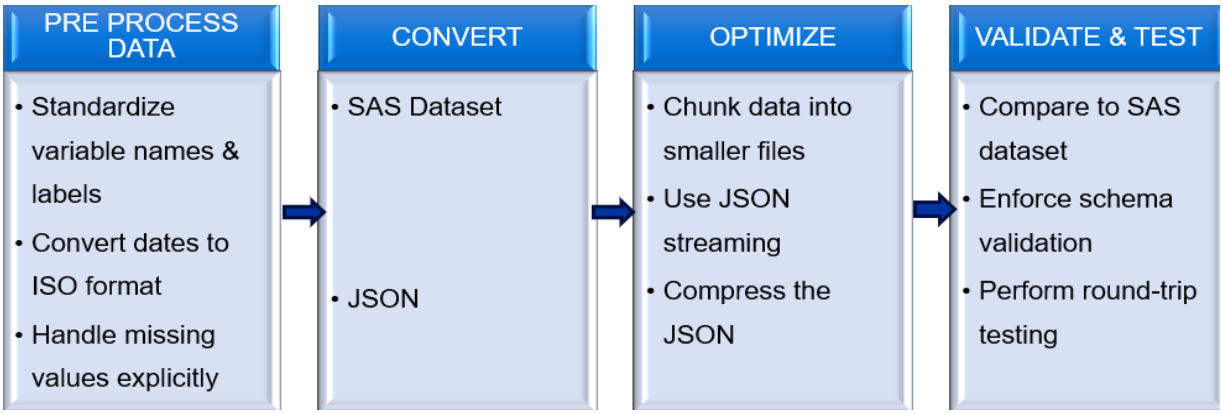
Dataset-JSON is a CDISC-compliant format used to represent clinical trial data in a structured, machine-readable way. It includes:

- **Top-level metadata**: Information about file creation, version, sponsor, and source system.
- **Study references**: Links to study identifiers and Define-XML metadata.
- **Dataset-level info**: Domain name (e.g., DM), label, and number of records.
- **Column metadata**: Definitions for each variable, including name, label, data type, and length.
- **Row data**: Subject-level values aligned with column definitions, stored as arrays.

This format improves transparency, interoperability, and regulatory readiness by aligning clinical data with modern digital standards.

Dataset-JSON Conversion Workflow

This flowchart outlines the four key stages for preparing clinical trial data for Dataset-JSON submission:



This SAS code uses PROC JSON to generate a structured Dataset-JSON file that includes:

```
PROC JSON OUT=jsonfout PRETTY;
  WRITE OPEN OBJECT;
  WRITE VALUES "clinicalData";
  WRITE OPEN OBJECT;
  WRITE VALUES "studyOID" "<study OID>";
  WRITE VALUES "metaDataVersionOID" "<MDV OID>";
  WRITE VALUES "itemGroupData";
  WRITE OPEN OBJECT;
  WRITE VALUES "IG.DM";
  WRITE OPEN OBJECT;
  WRITE VALUES "records" 18;
  WRITE VALUES "name" "DM";
  WRITE VALUES "label" "Demographics";
  WRITE VALUES "items";
  WRITE OPEN ARRAY;
    EXPORT work.column_metadata / KEYS;
  WRITE CLOSE;
  WRITE VALUES "itemData";
  WRITE OPEN ARRAY;
    EXPORT work.column_data / NOKEYS;
  WRITE CLOSE;
  WRITE CLOSE;
  WRITE CLOSE;
  WRITE CLOSE;
  WRITE CLOSE;
RUN;
```

```
{
  "clinicalData": {
    "studyOID": "<study OID>",
    "metaDataVersionOID": "<MDV OID>",
    "itemGroupData": {
      "IG.DM": {
        "records": 18,
        "name": "DM",
        "label": "Demographics",
        "items": [
          column metadata
        ],
        "itemData": [
          Column data
        ]
      }
    }
  }
}
```

- KEYS → Exports data as **objects** with variable names (best for metadata).
- NOKEYS → Exports data as **arrays** without variable names (best for subject-level data).
- PRETTY → Adds indentation, line breaks, and spacing. The output is easy for humans to read.

  Output:
  ```
  {
    "studyOID": "CDISC01",
    "name": "DM"
  }
  ```
- NOPRETTY → Produces compact JSON with no extra whitespace. The output is efficient for machines to process.

  Output: **{"studyOID":"CDISC01","name":"DM"}**

DATASET JSON DATASET WITH R

THIS R CODE BUILDS A JSON DATASET BLOCK FOR THE DM DOMAIN USING NATIVE R STRUCTURES.

```r
library(jsonlite)
# Sample column metadata (similar to column_metadata in SAS)
column_metadata <- data.frame(
  itemOID = c("IT.DM.USUBJID", "IT.DM.AGE"),
  name = c("USUBJID", "AGE"),
  label = c("Subject ID", "Age"),
  dataType = c("string", "integer"),
  stringsAsFactors = FALSE
)
# Sample column data (similar to column_data in SAS)
column_data <- list(
  list("SUBJ001", 34),
  list("SUBJ002", 29)
)
# Construct Dataset-JSON structure
dataset_json <- list(
  clinicalData = list(
    studyOID = "<study OID>",
    metaDataVersionOID = "<MDV OID>",
    itemGroupData = list(
      `IG.DM` = list(
        records = length(column_data),
        name = "DM",
        label = "Demographics",
        items = column_metadata,
        itemData = column_data
      )
    )
  )
)
```

```json
{
  "clinicalData": {
    "studyOID": "<study OID>",
    "metaDataVersionOID": "<MDV OID>",
    "itemGroupData": {
      "IG.DM": {
        "records": 2,
        "name": "DM",
        "label": "Demographics",
        "items": [
          {
            "itemOID": "IT.DM.USUBJID",
            "name": "USUBJID",
            "label": "Subject ID",
            "dataType": "string"
          },
          {
            "itemOID": "IT.DM.AGE",
            "name": "AGE",
            "label": "Age",
            "dataType": "integer"
          }
        ],
        "itemData": [
          ["SUBJ001", 34],
          ["SUBJ002", 29]
        ]
      }
    }
  }
}
```

PACKAGE	STRENGTHS	LIMITATIONS
JSONLITE	BEST FOR NESTED CLINICAL DATA, SUPPORTS DATA FRAMES AND LISTS, CDISC-READY	NONE FOR THIS USE CASE
RJSON	LIGHTWEIGHT, SIMPLE	POOR SUPPORT FOR NESTED STRUCTURES AND DATA FRAMES
RJSONIO	FAST, LOW-LEVEL	COMPLEX SYNTAX, HARDER TO MAINTAIN
NDJSON	GOOD FOR LINE-DELIMITED JSON	NOT SUITABLE FOR HIERARCHICAL CDISC FORMATS
JQR	JSON MANIPULATION (LIKE JQ)	NOT FOR GENERATION, ONLY TRANSFORMATION

P21 COMMUNITY: SUPPORTING DATASET-JSON ADOPTION

1. Validator Support

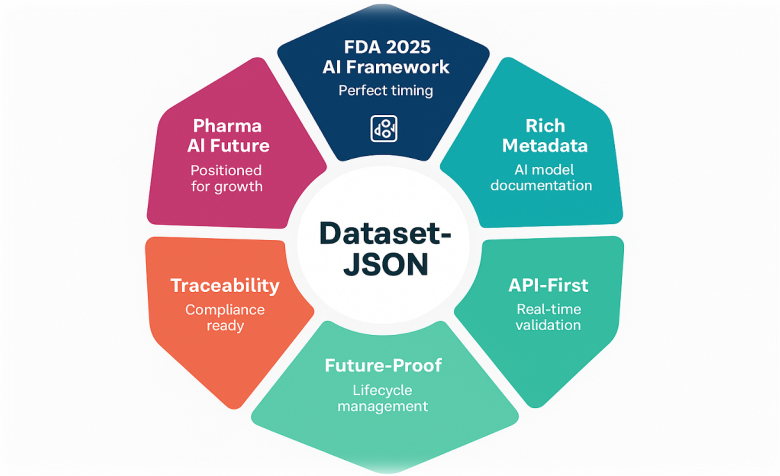
P21 now validates Dataset-JSON files directly, ensuring they meet CDISC and FDA compliance standards. This allows sponsors to check structure, metadata, and data integrity without reverting to legacy XPT files, streamlining workflows and reducing risk of submission errors.

2. Define.xml Integration

Dataset-JSON files can be linked directly to Define.xml, which contains detailed metadata definitions. This integration improves transparency by giving reviewers immediate context for variables and domains, eliminating the need for older XPT viewers and enhancing metadata traceability.

3. Future-Proof Submissions

Dataset-JSON aligns with emerging regulatory expectations for digital-first, machine-readable submissions. It supports cloud-native workflows, API-based exchanges, and automated review systems, ensuring that data pipelines remain compliant and scalable as standards evolve.



CONCLUSION

The introduction of Dataset-JSON marks a pivotal shift in clinical data submission, aligning regulatory workflows with the demands of digital transformation. Compared to legacy formats like SAS XPT, Dataset-JSON offers enhanced metadata transparency, human readability, and seamless integration with modern platforms and APIs. These advantages position it as a future-ready standard for regulatory submissions.

Despite its benefits, adoption challenges persist—particularly around schema validation, legacy system compatibility, and limited tool support. However, with the aid of validation tools like Pinnacle 21 and practical implementation strategies using SAS and R, programmers can navigate these hurdles effectively.

As the industry continues to embrace AI-driven automation and cloud-native infrastructures, Dataset-JSON is poised to become a cornerstone of efficient, transparent, and interoperable data exchange. Programmers and sponsors are encouraged to adopt this format proactively, ensuring smoother transitions and long-term alignment with evolving regulatory expectations.

REFERENCES

1. WWW.CDISC.ORG/STANDARDS/DATA_EXCHANGE/DATASET_JSON
2. HTTPS://WWW.LEXJANSEN.COM/PHARMASUG/2022/AD/PHARMASUG-2022-AD-150_PPT.PDF
3. HTTPS://WWW.LEXJANSEN.COM/CSS_US/2023/PRE_01.PDF
4. HTTPS://WWW.FDA.GOV/INDUSTRY/FDA_DATA_STANDARDS_ADVISORY_BOARD
5. HTTPS://WWW.ECMA_INTERNATIONAL.ORG/PUBLICATIONS_AND_STANDARDS/STANDARDS/ECMA-404/

CONTACT INFORMATION

Author Name: Venkata Koteswara Rao Rayala
Company: Merck Specialities Private Limited
Address: Hyderabad
Work Phone:
Email: venkata-koteswara-rao.rayala@merckgroup.com
Website: