Feedback from Dataset-JSON Submission Pilot Workshop and Prospects of Adoption in Japan

Yuichi Nakajima
Statistical Programming Group Head
Novartis Pharma K.K.

Meet the Speaker

Yuichi Nakajima

Title: Statistical Programming Group Head

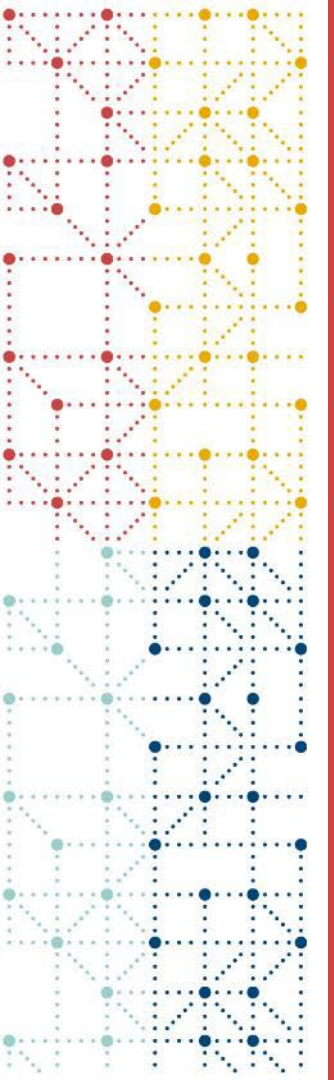
Organization: Analytics and CDM Japan, Novartis Pharma K.K.

Yuichi Nakajima is a head of statistical programming strategy & operations in Novartis Pharma K.K. In this position, he is responsible for statistical programming activities for clinical trial and post-marketing surveillance. In addition to his main responsibility, he supports digital innovation task in research & development organization and contributes digital technology implementation for their daily work.

He also pursues external engagement in order to raise visibility of clinical statistical programmer in Japan. He organizes PharmaSUG Japan and leads PHUSE Open Source Technology working group Japan.

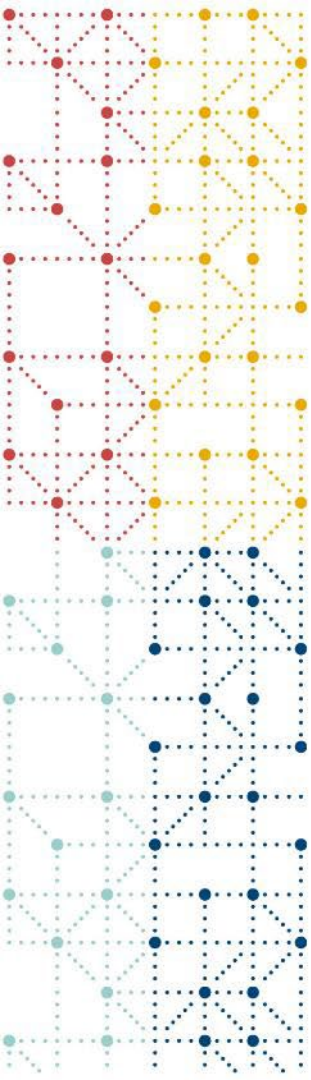
Disclaimer and Disclosures

- The views and opinions expressed in this presentation are those of the author(s) and do not necessarily reflect the official policy or position of CDISC.
- The views presented are the views of the presenter, not necessarily those of Novartis.

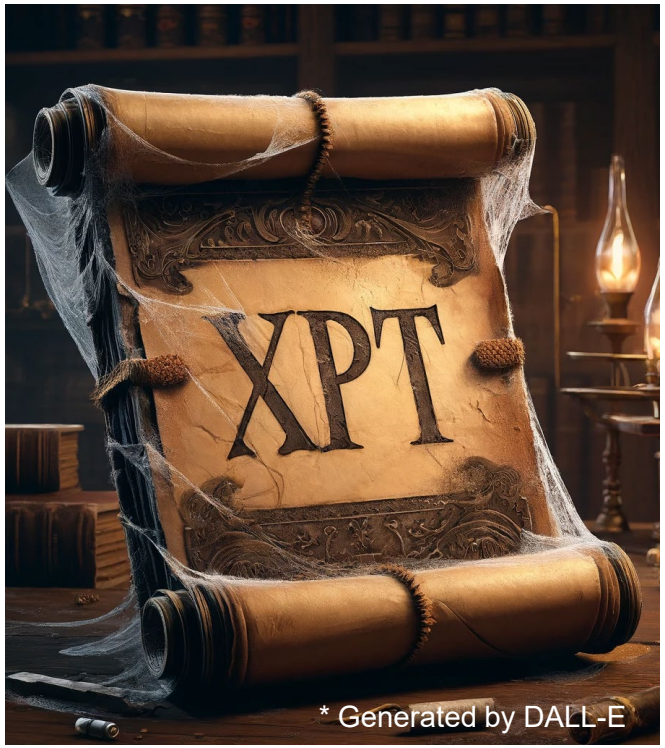


Agenda

1. Background
2. Dataset-JSON Submission Pilot Workshop in PHUSE
3. Available open-source solution of Dataset-JSON
4. Prospects of adoption in Japan
5. Summary



Background



XPT is still widely used but time to consider a “Successor”.

Limitation of xpt format

01

Data File Format Limitations

- The format **supports only US ASCII** for character encoding and lacks support for multibyte characters, which necessitates data translation or tra source.
- Variable names and l limitations, with variab **characters** and **label**
- Character field widths maximum of **200 char**

02

Storage Inefficiencies

- The format is inefficient in using storage space, often leading to up to 70% wasted space due to empty space allocated for columns that are not fully utilized.
- The format does not support data compression, which complicates file management, especially with **a maximum size limit of 5 Gigabytes per file**.

03

Content Limitations

- XPT is suited only for structures, which restr that can be transporte sub-optimal content st
- There is a lack of a ro within the format itself external files like define.xml to provide comprehensive data reviews, which necessitates keeping multiple files synchronized and updated.

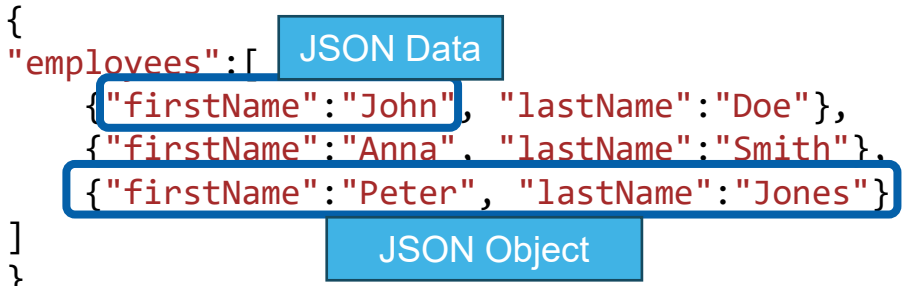
Lack of Extensibility

- The format does not support modern, extensible technologies, limiting its capability to adapt to new data types or evolving data handling requirements.
- The format does not inherently support features like user tracking or audit trails, limiting its utility in environments where data provenance and modification tracking are important.

Transport for the Next Generation version 1.0, 30 Apr 2017.

Advantage of JSON format

- JSON (JaveScript Object Notation) is the de facto standard for API data exchange.
- JSON is a lightweight format for storing and transporting data.
- JSON is often used when data is sent from a server to a web page.
- JSON is “self-describing” and easy to understand.



JSON Syntax Rules

- Data is in name/value pairs
- Data is separated by commas
- Curly braces hold objects
- Square brackets hold arrays

What is JSON (w3schools.com)

Specification of Dataset-JSON (as part of ODM v2.0)

- Top Level Attributes

```
{
    "creationDateTime": "2023-03-22T11:53:27",
    "datasetJSONVersion": "1.0.0",
    "fileOID": "www.sponsor.xyz.org.project123.final",
    "asOfDateTime": "2023-02-15T10:23:15",
    "originator": "Sponsor XYZ",
    "sourceSystem": "Software ABC",
    "sourceSystemVersion": "1.0.0",
    "clinicalData": { ... },
    "referenceData": { ... }
}
```

- ClinicalData and ReferenceData Attributes

```
"clinicalData": {
    "studyOID": "xxx",
    "metaDataVersionOID": "xxx",
    "metaDataRef": "https://metadata.location.org/api.link",
    "itemGroupData": { ... }
```

```
"itemGroupData": {
    "IG.DM": { ... }
}
```

```
"IG.DM": {
    "records": 100,
    "name": "DM",
    "label": "Demographics",
    "items": [ ... ],
    "itemData": [ ... ]
}
```

```
"items": [
   {
       "OID": "ITEMGROUPDATASEQ",
       "name": "ITEMGROUPDATASEQ",
       "label": "Record identifier",
       "type": "integer",
   },
   {
       "OID": "IT.DM.STUDYID",
       "name": "STUDYID",
       "label": "Study Identifier",
       "type": "string",
       "length": 12,
       "keySequence": 1,
   },
   ...
]
```

```
"itemData": [
   [1, "MyStudy", "001", "DM", 56],
   [2, "MyStudy", "002", "DM", 26],
   ...
]
```

Specification of Dataset-JSON (as part of ODM v2.0)


```
{
    "creationDateTime": "2023-03-22T11:53:27",
    "datasetJSONVersion": "1.0.0",
    "fileOID": "www.sponsor.org.project123.final",
    "asOfDateTime": "2023-02-15T10:23:15",
    "originator": "Sponsor XYZ",
    "sourceSystem": "Software ABC",
    "sourceSystemVersion": "1.2.3",
    "clinicalData": {
        "studyOID": "xxx",
        "metaDataVersionOID": "xxx",
        "metaDataRef": "https://metadata.location.org/api.link",
        "itemGroupData": {
            "IG.DM": {
                "records": 600,
                "name": "DM",
                "label": "Demographics",
                "items": [
                    {"OID": "ITEMGROUPDATASEQ", "name": "ITEMGROUPDATASEQ", "label": "Record identifier", "type": "integer"},
                    {"OID": "IT.STUDYID", "name": "STUDYID", "label": "Study identifier", "type": "string", "length": 7, "keySequence": 1},
                    {"OID": "IT.USUBJID", "name": "USUBJID", "label": "Unique Subject Identifier", "type": "string", "length": 3, "keySequence": 2},
                    {"OID": "IT.DOMAIN", "name": "DOMAIN", "label": "Domain Identifier", "type": "string", "length": 2},
                    {"OID": "IT.AGE", "name": "AGE", "label": "Subject Age", "type": "integer", "length": 2}
                ],
                "itemData": [
                    [1, "MyStudy", "001", "DM", 56],
                    [2, "MyStudy", "002", "DM", 26],
                    ...
                ]
            }
        }
    }
}
```

PHUSE: Dataset-JSON as Alternative Transport Format for Regulatory Submissions

- Project Scope

1. Demonstrate that Dataset-JSON(DSJSON) can transport information with no disruption to business.
2. Demonstrate the viability of DSJSON as the primary transport option.

- Sub Teams

Pilot Submission Report	Business Case	Technical Implementation	Strategy for Future Development

Development history

- **3Q**: DSJSON Hackathon led by COSA.
- **4Q**: DSJSON Hackathon Results shared at CDISC US Interchange / ODM v2.0 public review.

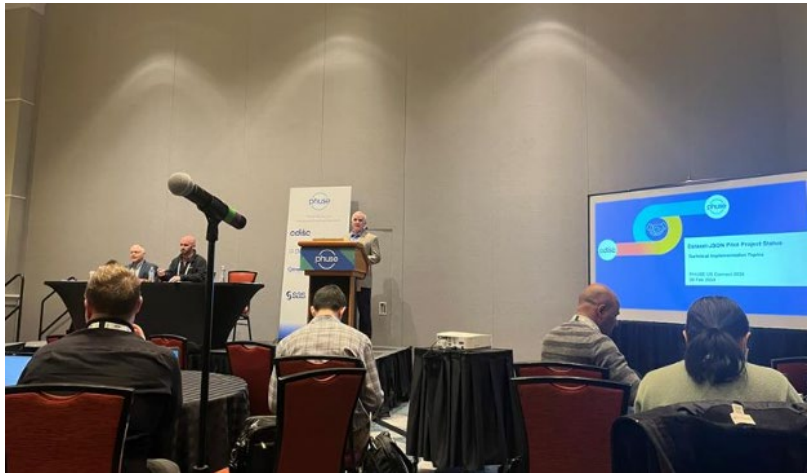
- **2Q**: PHUSE team formation
- **3Q**: ODM v2.0 including DSJSON v1.0 released / FDA Pilot Submission

- **1Q**: DSJSON Webinar / PHUSE US Connect Hands-On-Workshop
- **2Q**: PHUSE CSS
- **4Q**: Develop DSJSON v1.1 standard

2022 → 2023 → 2024

Dataset-JSON Submission Pilot Workshop in PHUSE

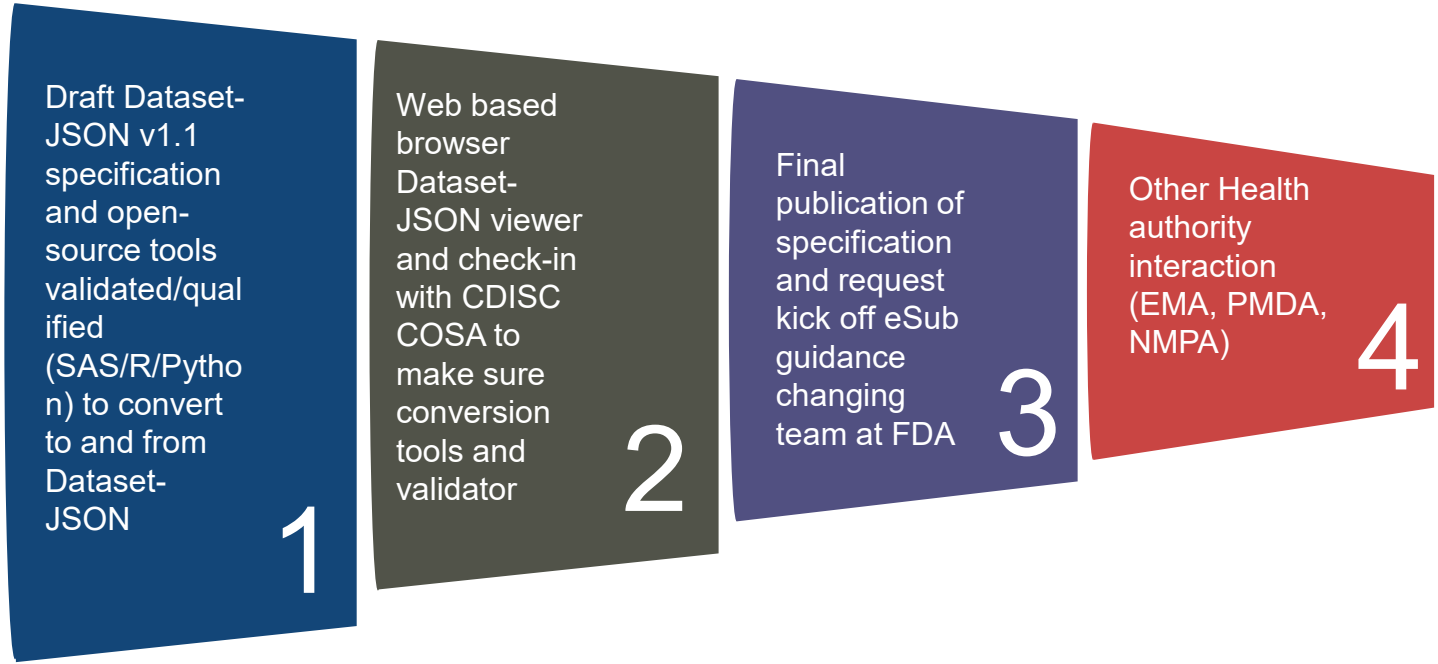
PHUSE US Connect 2024 in February

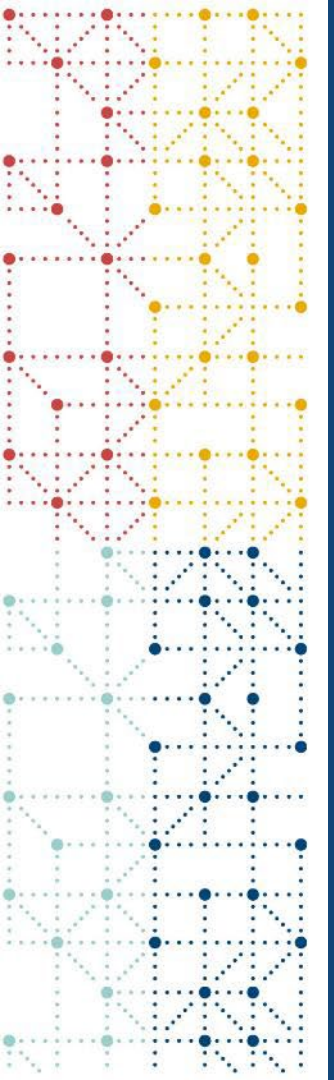


Key take aways

- FDA testing (Milestone 1) was successful.
  - Pilot using JSON format with existing XPT ingress/egress to carry the same data
  - Same content, different suitcase, no disruption to business process on either side
  - Allow FDA to evaluate how internal tools can support JSON format
- Assumptions.
  - Various data exchange for the datasets from EDC, ePRO, ...
  - Dataset-JSON API.
  - Aligns with DDF USDM, ARS, CORE, CDISC Library, OAK, HL7 FHIR, ...
- Dataset-JSON viewer must be provided.
- NDJSON for large datasets.
- Parquet.
  - Apache Parquet is an open source, column-oriented data file format designed for efficient data storage and retrieval.
  - Conversion tool from Dataset-JSON to Parquet, from Parquet to Dataset-JSON.

Priority of next steps





Available open-source solution of Dataset-JSON

Useful R packages for Dataset.json

R package / Developer	Description	GitHub

R4DSJSON / Ippei Akiya	Read DSJSON file and convert to R Dataframe and vice versa.
	`# Read dm.json file as tibbl`
	`dm_json_file <- system.file("testdata","dm.json", package="R4DSJSON")`
	`dm_tibble <- R4DSJSON::read.dataset.json(file = dm_json_file)`
	
	`# Write Dataset-Json file from R dataframe`
	`write.dataset.json(file="dm.json", dataframe=dm_tibble, studyOID="cdisc.com/CDISCPILOT01",`
	`metaDataVersionOID="MDV.MSGv2.0.SDTMIG.3.3.SDTM.1.7",`
	`dataset.name="DM", dataset.label="Demographics")`
		GitHub - i-akiya/R4DSJSON: R package for reading and writing CDISC Dataset-Json files.

datasetjson / Atorus Research	Apply attributes required for DSJSON and generate DSJSON file.
	`ds_updated <- ds_json |>`
	`set_data_type("referenceData") |>`
	`set_file_oid("/some/path") |>`
	`set_metadata_ref("some/define.xml") |>`
	`set_metadata_version("MDV.MSGv2.0.SDTMIG.3.3.SDTM.1.7") |>`
	`set_originator("Some Org") |>`
	`set_source_system("source system", "1.0") |>`
	`set_study_oid("SOMESTUDY")`
		GitHub - atorus-research/datasetjson: Read and write CDISC Dataset JSON files

Sample of DSJSON files are available in Dataset-JSON repository

How to get metadata for DSJ?

To get variable attributes (Name, label, type, ...)

1. Prepare company specific metadata.
2. Draft Define-xml for DSJSON.
3. Use CDISC Library API.

```
library(httr)
library(jsonlite)

myAPIkey <- “XXXXXXXXXXXXX(API key)"
PrimaryURL<- "https://library.cdisc.org/api/mdr/sdtmig/"
SDTMversion <- "3-2"
DOMAIN <- "DM"
endpoint <- paste0 (PrimaryURL, SDTMversion, "/datasets/", DOMAIN)
Response <- GET (endpoint, add_headers ("api-
key" = myAPIkey, "accept" = "application/json"))
responseBody <- httr::content (Response, as = "text", encoding = "UTF-
8")
data <- fromJSON(responseBody)

# Extracting the dataset variables component
variables <- data$datasetVariables
```

Create DSJSON from admiral test data using datasetjson package & CDISC Library

```r
library(datasetjson)
library(admiral.test)
library(httr)
library(jsonlite)

#####################################
####  Get CDISC Library metadata  #######
#####################################

# Initialize an empty list to store metadata for each variable
dataset_meta <- list()
# Get variable name of DM domain.
dm_vars <- names(admiral_dm)

# Loop through each variable to extract necessary details
for (i in seq_along(variables$name)) {
 if (variables$name[i] %in% dm_vars) {
  # Determine the correct type mapping
  type_value <- ifelse(variables$simpleDatatype[i] == "Char", "string", "float")
  variable_meta <- list(
   OID = paste("OID", variables$name[i], sep = ".")
   name = variables$name[i],
   label = variables$label[i],
   type = type_value
  )
  # Append to the main list
  dataset_meta[[length(dataset_meta) + 1]] <- variable_meta
 }
}
```

Map metadata for required variable and create metadata list.

```r
# Convert the list of lists into a dataframe with appropriate column names
dataset_meta_df <- do.call(rbind, lapply(dataset_meta, as.data.frame))
names(dataset_meta_df) <- c("OID", "name", "label", "type")

# Check the structure of dataset_meta_df to ensure it matches expected format
str(dataset_meta_df)

# Creating the dataset_json object
dmjson <- dataset_json(admiral_dm,
                       item_id = "IG.DM",
                       name = "DM",
                       label = "Demographics",
                       items = dataset_meta_df)

dmjson1 <- dmjson |>
 set_data_type("referenceData") |>
 set_file_oid("/some/path") |>
 set_metadata_ref("CDISC Library") |>
 set_metadata_version("MDV.MSGv2.0.SDTMIG.3.2.SDTM.1.4") |>
 set_originator("Some Org") |>
 set_source_system("source system", "1.0") |>
 set_study_oid("SOMESTUDY")

write_dataset_json(dmjson1, './datasetjson/dmjson.json')
```

Apply variable level metadata to DSJSON object.

Create DSJSON from admiral test data using datasetjson package & CDISC Library

```
> str(dataset_meta_df)
'data.frame':   25 obs. of  4 variables:
 $ OID  : chr  "OID.STUDYID" "OID.DOMAIN" "OID.USUBJID" "OID.SUBJID" ...
 $ name : chr  "STUDYID" "DOMAIN" "USUBJID" "SUBJID" ...
 $ label: chr  "Study Identifier" "Domain Abbreviation" "Unique Subject Identifier"
"Subject Identifier for the Study" ...
 $ type : chr  "string" "string" "string" "string" ...
>
> str(admiral_dm)
tibble [306 × 25] (S3: tbl_df/tbl/data.frame)
 $ STUDYID : chr [1:306] "CDISCPILOT01" "CDISCPILOT01" "CDISCPILOT01" "CDISCPILOT01"
...
  ..- attr(*, "label")= chr "Study Identifier"
 $ DOMAIN  : chr [1:306] "DM" "DM" "DM" "DM" ...
  ..- attr(*, "label")= chr "Domain Abbreviation"
 $ USUBJID : chr [1:306] "01-701-1015" "01-701-1023" "01-701-1028" "01-701-1033" ...
  ..- attr(*, "label")= chr "Unique Subject Identifier"
 $ SUBJID  : chr [1:306] "1015" "1023" "1028" "1033" ...
  ..- attr(*, "label")= chr "Subject Identifier for the Study"
 $ RFSTDTC : chr [1:306] "2014-01-02" "2012-08-05" "2013-07-19" "2014-03-18" ...
  ..- attr(*, "label")= chr "Subject Reference Start Date/Time"
 $ RFENDTC : chr [1:306] "2014-07-02" "2012-09-02" "2014-01-14" "2014-04-14" ...
  ..- attr(*, "label")= chr "Subject Reference End Date/Time"
```

dmjson1	list [8] (S3: datasetjson_v1_0_	List of length 8
creationDateTime	character [0]	
datasetJSONVersion	character [1]	'1.0.0'
fileOID	character [1]	'/some/path'
asOfDateTime	NULL	Pairlist of length 0
originator	character [1]	'Some Org'
sourceSystem	character [1]	'source system'
sourceSystemVersion	character [1]	'1.0'
referenceData	list [4] (S3: data_metadata, lis	List of length 4
studyOID	character [1]	'SOMESTUDY'
metaDataVersionOID	character [1]	'MDV.MSGv2.0.SDTMIG.3.2.SDTM.1.4'
metaDataRef	character [1]	'CDISC Library'
itemGroupData	list [1] (S3: dataset_metadata,	List of length 1
IG.DM	list [5]	List of length 5
records	integer [1]	306
name	character [1]	'DM'
label	character [1]	'Demographics'
items	list [26]	List of length 26
itemData	list [306 x 26] (S3: data.frame	A data.frame with 306 rows and 26 columns

items is an array of basic information about dataset variables. The order of the elements in the array must be the same as the order of variables in the described dataset. The first element always describes the Record Identifier (ITEMGROUPDATASEQ).

Prospects of adoption in Japan

FDA Statement

CBER-CDER Data Standards Program Action Plan (version 1.4), 15Feb2024

Project Title & Description	Project Status	Project Stages						
		REQT	ALT	DEV	TEST	ADOPT	IMPL	POLICY
Dataset-JSON Standard This CDER project is a collaboration with PHUSE and CDISC to test the use of Dataset-JSON as potential replacement for XPT.	**Q1:** Received test submissions from industry demonstrating that FDA can receive dataset packages in Dataset-JSON format with no impact to data integrity as compared to the XPT files.	Complete	Complete	In Progress	In Progress	Pending	Pending	Pending

Study Data Standards Resources

As of 05Apr2024

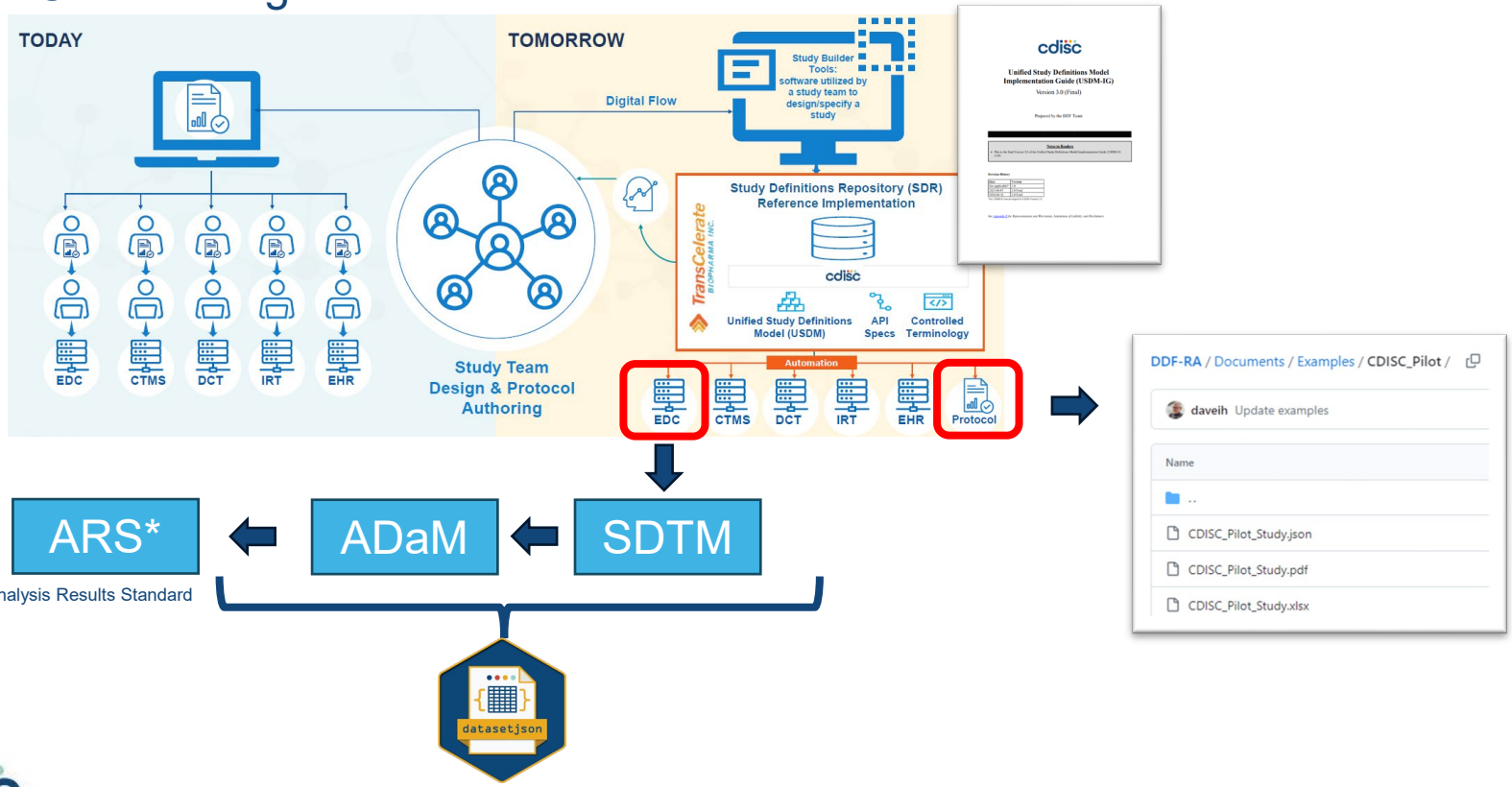
CDER and CBER, in collaboration with CDISC and PhUSE, has conducted preliminary testing of CDISC's Dataset JSON message exchange standard. Initial results indicate potential use as a replacement for XPT v5. As such, CBER and CDER will conduct further testing to evaluate Dataset JSON's capability to support the submission of regulatory study data. Results will be communicated, and we will engage stakeholders for input as we progress through this evaluation.

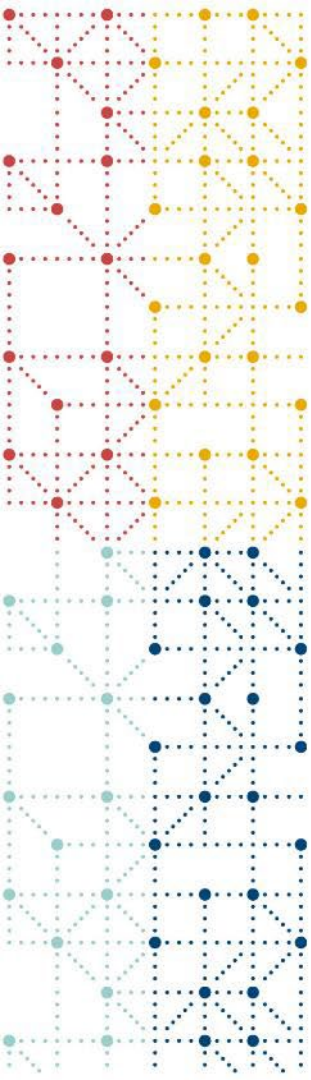
What we prepare in Japan ?

	Health Authority (PMDA)	Sponsors / Industry
Short term	• To have a common understanding of DSJSON. • To identify the impacts to existing process.	
Mid term	• Update eSub guidance and technical conformance guide. • Implement conformance rules. • Gateway system update. • Potential acceptance of data in Japanese.	• Develop / Update system (conversion tools) to generate dataset in JSON format. • Fill the gap within software to generate DS JSON such as date epochs / rounding issues.
Long term		• Implement DSJSON as transport data format into E2E process of clinical trial (From data capture to NDA) • To be utilized in RWD.

Future: JSON format in E2E process

- TransCelerate Digital Data Flow





Summary

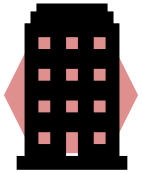
Three key factors

1

New guidance released: DSJSON v1.1 specification

2

Technical implementation: Open-source solution (conversion tool / viewer)



3

Regulation for new data format by Health Authorities

Dataset-JSON

- This possible new data format will impact clinical study data flow.

- Concept of DSJSON is not just a discussion of transport format but a process optimization of clinical studies.

Acknowledgement

- **Sam Hume**, Vice President of Data Science, CDISC
- PHUSE OST WG, Dataset-JSON sub team
  - **Ippei Akiya**
  - **Yutaka Morioka**
  - **Tomoyuki Namai**

References

- PHUSE WORKING GROUP: Dataset-JSON as Alternative Transport Format Dataset-JSON as Alternative Transport Format for Regulatory Submissions - WORKING GROUPS - PHUSE Advance Hub
- CDISC Dataset-JSON Dataset-JSON | CDISC
- Dataset-JSON v1.1 Dataset-JSON v1.1 - Dataset-JSON v1.1 - Wiki (cdisc.org)
- COSA Repository Directory CDISC Open Source Alliance Directory
- Dataset-JSON Hackathon & Pilot Home Dataset-JSON Hackathon & Pilot Home - Dataset-JSON Hackathon - Wiki (cdisc.org)

Thank You!