

# Dataset-JSON: SAS<sup>®</sup> implementation

Lex Jansen

Senior Director, Data Science Development, CDISC (*contractor*)

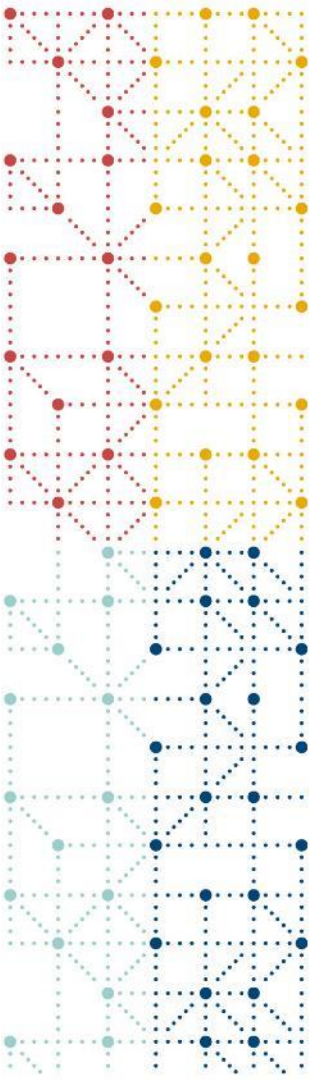


CDISC Q3 COSA Quarterly Spotlight Webinar - 2023-10-05



## Agenda

1. Dataset-JSON Document Structure
2. SAS and JSON
3. Writing Dataset-JSON with SAS
4. Reading Dataset-JSON with SAS
5. Comparing SAS datasets / Dataset-JSON files
6. Validating Dataset-JSON files
7. Demo



# Dataset-JSON Document Structure



# Dataset-JSON Document Structure

- Dataset-JSON was adapted from the Dataset-XML specification, but uses JSON format instead of XML
- Each Dataset-JSON file is connected with a Define-XML file, containing **detailed** metadata
- Each Dataset-JSON files contains **basic** information about dataset variables, so that it is possible to have a simple view of the contents of a dataset without the need of a Define-XML document
- Dataset-JSON Specification:
  - <https://www.cdisc.org/dataset-json>
  - <https://wiki.cdisc.org/display/PUB/Dataset-JSON>
- GitHub repository with JSON Schema and examples:  
<https://github.com/cdisc-org/DataExchange-DatasetJson>

# Dataset-JSON Document Structure

- At the top level of the Dataset-JSON object, there are some required attributes (**creationDateTime** and **datasetJSONVersion**)\*, and one of two optional attributes: **clinicalData** or **referenceData**.
- Subject data is stored in **clinicalData** and non-subject data is stored in **referenceData**.

```
{
  "creationDateTime": "2023-06-28T15:38:43",
  "datasetJSONVersion": "1.0.0",
  "clinicalData": {
    "studyOID": "cdisc.com/CDISCPIL0T01",
    "metaDataVersionOID": "MDV.MSGv2.0.SDTMIG.3.3.SDTM.1.7",
    "itemGroupData": {
      "IG.DM": { ...
    }
  }
}
```

```
{
  "creationDateTime": "2023-06-28T15:38:44",
  "datasetJSONVersion": "1.0.0",
  "referenceData": {
    "studyOID": "cdisc.com/CDISCPIL0T01",
    "metaDataVersionOID": "MDV.MSGv2.0.SDTMIG.3.3.SDTM.1.7",
    "itemGroupData": {
      "IG.TS": { ...
    }
  }
}
```

- **studyOID** and **metaDataVersionOID** must match corresponding values in Define-XML

# Dataset-JSON Document Structure - top level attributes

Attribute	Usage	Description
creationDateTime	Required	Time of creation of the file containing the document.
datasetJSONVersion	Required	Version of Dataset-JSON standard
fileOID	Optional	A unique identifier for this file.
asOfDateTime	Optional	The date/time at which the source database was queried in order to create this document.
originator	Optional	The organization that generated the Dataset-JSON file.
sourceSystem	Optional	The computer system or database management system that is the source of the information in this file.
sourceSystemVersion	Optional	The version of the "SourceSystem" above.
clinicalData	Optional	Contains datasets for clinical data across multiple subjects.
referenceData	Optional	Contains datasets for non-subject data domains.

# Dataset-JSON Document Structure

```
"itemGroupData": {  
  "IG.DM": {  
    "records": 18,  
    "name": "DM",  
    "label": "Demographics",  
    "items": [ ...  
  ],  
  "itemData": [ ...  
  ]  
}
```

- **itemGroupData** is an object with attributes corresponding to individual datasets.
- The attribute name (IG.DM) is the OID of a described dataset, which must be the same as the OID of the corresponding itemGroup in the Define-XML file
- **records, name, label:** basic dataset information (all required)
- **items** - basic information about variables
- **itemData** - dataset data

# Dataset-JSON Document Structure

```
"items": [  
  {  
    "OID": "ITEMGROUPDATASEQ",  
    "name": "ITEMGROUPDATASEQ",  
    "label": "Record Identifier",  
    "type": "integer",  
    "length": 8  
  },  
  {  
    "OID": "IT.DM.STUDYID",  
    "name": "STUDYID",  
    "label": "Study Identifier",  
    "type": "string",  
    "length": 12  
  },  
  {  
    "OID": "IT.DM.DOMAIN",  
    "name": "DOMAIN",  
    "label": "Domain Abbreviation",  
    "type": "string",  
    "length": 2  
  },  
]
```

- **items** - array of basic information about dataset variables.
- The **order of 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**)



# Dataset-JSON Document Structure

```
"items": [  
  {  
    "OID": "ITEMGROUPDATASEQ",  
    "name": "ITEMGROUPDATASEQ",  
    "label": "Record Identifier",  
    "type": "integer",  
    "length": 8  
  },  
  {  
    "OID": "IT.DM.STUDYID",  
    "name": "STUDYID",  
    "label": "Study Identifier",  
    "type": "string",  
    "length": 12  
  },  
  {  
    "OID": "IT.DM.DOMAIN",  
    "name": "DOMAIN",  
    "label": "Domain Abbreviation",  
    "type": "string",  
    "length": 2  
  },  
]
```

- **OID** - OID of a variable (must correspond to the variable OID in the Define-XML file)
- **name** - variable name
- **label** - variable description
- **type** - type of the variable. 'string', 'integer', 'decimal', 'float', 'double', 'boolean'
- **length** - variable length - most useful for the string type
- **displayFormat** - supports data visualization of numeric float and date values
- **keySequence** - indicates that this item is a key variable in the dataset structure

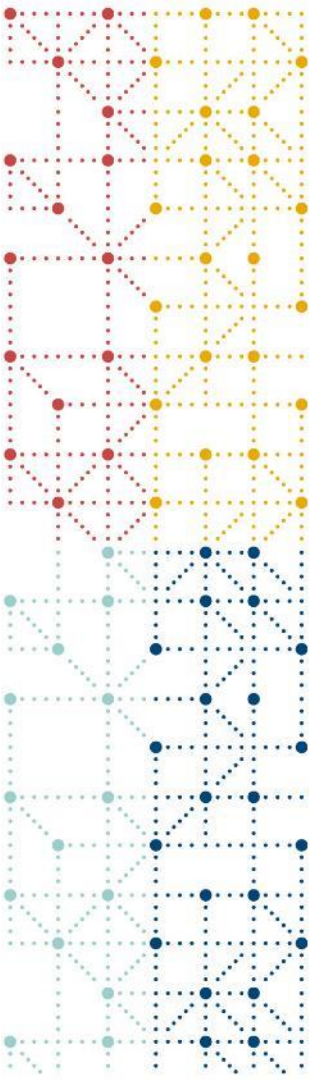
The last 3 attributes are optional

# Dataset-JSON Document Structure

- **itemData** is an array of records with variables values
- Each record itself is represented as an array of variables values
- The first value is a unique sequence number for each record in the dataset
- Missing values are represented by **null** in the case of numeric variables, and an **empty string** in case of character variables:

```
[1, "MyStudy", "", "DM", null]
```

```
"itemData": [  
  [1, "CDISCPILOT01", "DM", "CDISC001", "1115", "", "", "701", "1928", 84, "YEARS", "M", "WHITE", "NOT - HIS  
  [2, "CDISCPILOT01", "DM", "CDISC002", "1211", "2013-01-14", "Y", "701", "1936", 76, "YEARS", "F", "WHIT  
  [3, "CDISCPILOT01", "DM", "CDISC003", "1302", "", "", "701", "1951", 61, "YEARS", "M", "WHITE", "NOT - HIS  
  ...  
]
```

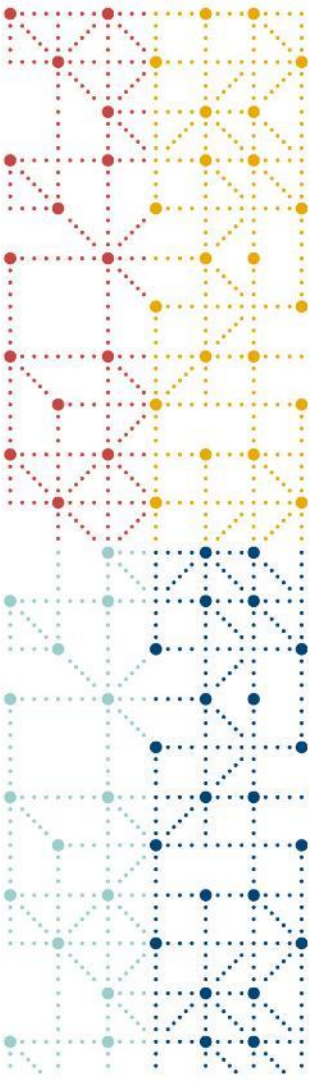


# SAS and JSON



# SAS and JSON

- Starting in SAS<sup>®</sup> 9.4, you can use **PROC JSON** to write SAS data sets to JSON files
- Starting in SAS<sup>®</sup> 9.4TS1M4, you can use the **JSON engine** to read JSON files into SAS data sets



# Writing Dataset-JSON with SAS

# Writing Dataset-JSON with SAS

- PROC JSON in SAS® gives the user control over the JSON output:
  - through the utilization of options
  - through the ability to control containers (objects or arrays)
  - by writing directly to the output file
  - by choosing exactly what to include or not include in the resulting JSON file

```
PROC JSON OUT=fileref | "external-file" <options>;  
  EXPORT <libref.>SAS-data-set <(data-set-options)> </options>;  
  WRITE VALUES value(s) </options>;  
  WRITE OPEN type;  
  WRITE CLOSE;  
RUN;
```

# Writing Dataset-JSON with SAS

```
PROC JSON OUT=jsonfout PRETTY;
  WRITE OPEN OBJECT;
  WRITE VALUES "creationDateTime" "&CurrentDateTime";
  WRITE VALUES "datasetJSONVersion" "1.0.0";
  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;
```

```
{
  "creationDateTime": "2023-06-28T15:38:43",
  "datasetJSONVersion": "1.0.0",
  "clinicalData": {
    "studyOID": "<study OID>",
    "metaDataVersionOID": "<MDV OID>",
    "itemGroupData": {
      "IG.DM": {
        "records": 18,
        "name": "DM",
        "label": "Demographics",
        "items": [
          ],
          "itemData": [
            ]
          }
        }
      }
    }
```

column metadata

Column data

# Writing Dataset-JSON with SAS

```
PROC JSON OUT=jsonfout PRETTY;  
  WRITE OPEN OBJECT;  
  WRITE VALUES "creationDateTime" "&CurrentDateTime";  
  WRITE VALUES "datasetJSONVersion" "1.0.0";  
  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;  
    %write_json_metadata_array(work.column_metadata);  
  WRITE CLOSE;  
  WRITE VALUES "itemData";  
  WRITE OPEN ARRAY;  
    EXPORT work.column_data / NOKEYS;  
  WRITE CLOSE;  
  WRITE CLOSE;  
  WRITE CLOSE;  
  WRITE CLOSE;  
  WRITE CLOSE;  
RUN;
```

```
{  
  "creationDateTime": "2023-06-28T15:38:43",  
  "datasetJSONVersion": "1.0.0",  
  "clinicalData": {  
    "studyOID": "<study OID>",  
    "metaDataVersionOID": "<MDV OID>",  
    "itemGroupData": {  
      "IG.DM": {  
        "records": 18,  
        "name": "DM",  
        "label": "Demographics",  
        "items": [  
          ],  
        "itemData": [  
          ]  
        }  
      }  
    }  
  }  
}
```

column metadata

Column data



# Using Define-XML metadata when creating Dataset-JSON

- Get OIDs from Define-XML when creating Dataset-JSON
- Use pre-specified metadata from Define-XML (label, datatype, length, and KeySequence) for creating Dataset-JSON
- Get variable display formats from Define-XML when creating SAS datasets from Dataset-JSON (especially for numeric date/time variables)
- Important to check that Define-XML is consistent with the data!

```
• /* Create metadata from Define-XML for ADaM */
%CreateMetadataFromDefineXML(
    definexml=&root/json/adam/define.xml,
    metadatalib=metaadam
);

/* Create metadata from Define-XML for SDTM */
%CreateMetadataFromDefineXML(
    definexml=&root/json/sdtm/define.xml,
    metadatalib=metasdtm
);

/* Some manual data type updates */
data metasdtm.metadata_columns;
set metasdtm.metadata_columns;
if missing(length) then do;
    if xml_datatype="date" then length=10;
    if xml_datatype="partialDate" then length=10;
    if xml_datatype="partialDatetime" then length=19;
    if xml_datatype="durationDatetime" then length=19;
    if xml_datatype="datetime" then length=19;
end;
run;
```

# Writing Dataset-JSON - %write\_datasetjson()

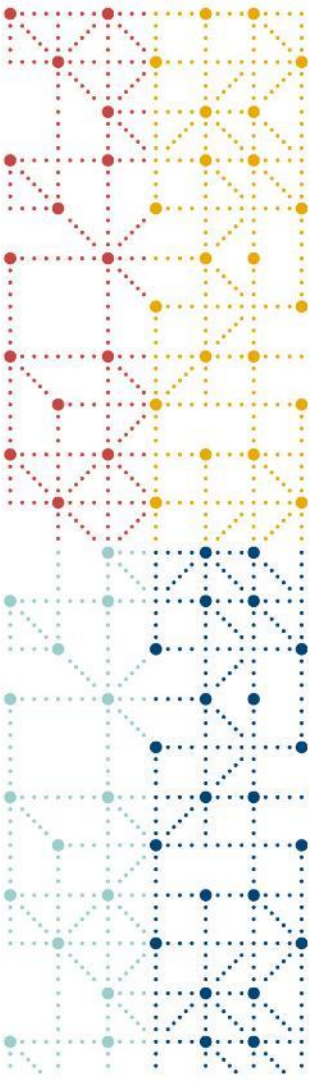
```
%write_datasetjson(  
    dataset=                /* (libname.)memname of the data set */  
    xptpath=,               /* Path to XPT file */  
    jsonpath=,             /* Path to Dataset-JSON file */  
    usemetadata=N,         /* Use Define-XML metadata? (Y/N) */  
    metadatalib=,          /* Define-XML metadata library */  
    datasetJSONVersion=1.0.0,  
    fileOID=,  
    asOfDateTime=,  
    originator=,  
    sourceSystem=,  
    sourceSystemVersion=,  
    studyOID=,  
    metaDataVersionOID=,  
    metaDataRef=,  
    pretty=NOPRETTY);
```

# Writing Dataset-JSON - %write\_datasetjson() - loop

```
data _null_;
  length datasetname $64 jsonpath $512 fileoid $128 code $2048;
  set work.dirtree_adam;
  datasetname=scan(filename, 1, ".");
  jsonpath=cats("&project_folder/json_out/adam/", datasetname, ".json");
  fileoid=cats("&_fileOID", "/", "%sysfunc(date(), is8601da.)", "/", datasetname);
  code=cats('%nrstr(%write_datasetjson('
            , 'dataset=dataadam.', datasetname, ','
            /* , 'xptpath=', fullpath, ',' */
            , 'jsonpath=', jsonpath, ','
            , 'usemetadata=N,'
            , 'metadatalib=metaadam,'
            , "fileOID=", fileoid, ","
            , "asOfDateTime=2023-05-31T00:00:00, "
            , "originator=CDISC ADaM MSG Team", ","
            , "sourceSystem=Sponsor System,"
            , "sourceSystemVersion=1.0,"
            , "studyOID=&_studyOID,"
            , "metaDataVersionOID=&_metaDataVersionOID,"
            , "metaDataRef=define.xml"
            , ');)');
  call execute(code);
run;
```

# Writing Dataset-JSON - %write\_datasetjson() - loop

```
data _null_;
  length datasetname $64 jsonpath $512 fileoid $128 code $2048;
  set work.dirtree_sdtm;
  datasetname=scan(filename, 1, ".");
  jsonpath=cats("&project_folder/json_out/sdtm/", datasetname, ".json");
  fileoid=cats("&_fileOID", "/", "%sysfunc(date(), is8601da.)", "/", datasetname);
  code=cats('%nrstr(%write_datasetjson('
            /* , 'dataset=datasdtm.', name, ',' */
            , 'xptpath=', fullpath, ','
            , 'jsonpath=', jsonpath, ','
            , 'usemetadata=Y,'
            , 'metadatalib=metasdtm,'
            , "fileOID=", fileoid, ","
            , "asOfDateTime=2023-05-31T00:00:00, "
            , "originator=CDISC SDTM MSG Team,"
            , "sourceSystem=Sponsor System,"
            , "sourceSystemVersion=1.0,"
            /* , "studyOID=&_studyOID," */
            /* , "metaDataVersionOID=&_metaDataVersionOID," */
            , "metaDataRef=define.xml"
            ,'))');
  call execute(code);
run;
```



# Reading Dataset-JSON with SAS



# Reading Dataset-JSON with SAS

- Starting in SAS<sup>®</sup> 9.4TS1M4, you can use the **JSON engine** to read JSON files into SAS data sets
- A **JSON map** is a file that the JSON engine uses to define the data set structures that are created when reading JSON

# Reading Dataset-JSON with SAS

```
FILENAME jsonfile "<path to the JSON file>";  
FILENAME mapfile "<path to the MAP file to be created>";  
LIBNAME data_out "<path to the output folder>";  
  
LIBNAME jsonfile JSON FILEREF=jsonfile MAP=mapfile  
        AUTOMAP=CREATE NOALLDATA ORDINALCOUNT=NONE;  
PROC COPY IN=jsonfile OUT=data_out;  
RUN;
```

# Reading Dataset-JSON with SAS

```
{
  "creationDateTime": "<yyyy-mm-ddThh:mm:ss>",
  "datasetJSONVersion": "1.0.0",
  "clinicalData": {
    "studyOID": "<study OID>",
    "metaDataVersionOID": "<MDV OID>",
    "itemGroupData": {
      "IG.DM": {
        "records": 18,
        "name": "DM",
        "label": "Demographics",
        "items": [
          ]
        ]
      "itemData": [
        ]
      }
    }
  }
}
```

column metadata

column data

Creates 6 data sets:

- ALLDATA
- CLINICALDATA
  - studyOID, metaDataVersionOID
- ITEMGROUPDATA\_IG\_DM
  - records, name, label
- IG\_DM\_ITEMS
  - contains column metadata (OID, name, label, type, length, displayFormat, keySequence)
- IG\_DM\_ITEMDATA
  - contains column data (element1, element2, element3, ...)
- ROOT
  - contains creationDateTime, datasetJSONVersion



# Reading Dataset-JSON with SAS

ITEMGROUPDATA\_IG\_DM

	records	name	label
1	18	DM	Demographics

IG\_DM\_ITEMS

	OID	name	label	type	length	keySequence
1	ITEMGROUPDATASEQ	ITEMGROUPDATASEQ	Record Identifier	integer	.	.
2	IT.DM.STUDYID	STUDYID	Study Identifier	string	12	1
3	IT.DM.DOMAIN	DOMAIN	Domain Abbreviation	string	2	.
4	IT.DM.USUBJID	USUBJID	Unique Subject Identifier	string	8	2
5	IT.DM.SUBJID	SUBJID	Subject Identifier for the Study	string	4	.
6	IT.DM.RFSTDTC	RFSTDTC	Subject Reference Start Date/Time	string	.	.
7	IT.DM.RFENDTC	RFENDTC	Subject Reference End Date/Time	string	.	.
8	IT.DM.RFXSTDTC	RFXSTDTC	Date/Time of First Study Treatment	string	.	.
9	IT.DM.RFXENDTC	RFXENDTC	Date/Time of Last Study Treatment	string	.	.
10	IT.DM.RFICDTC	RFICDTC	Date/Time of Informed Consent	string	.	.

IG\_DM\_ITEMDATA

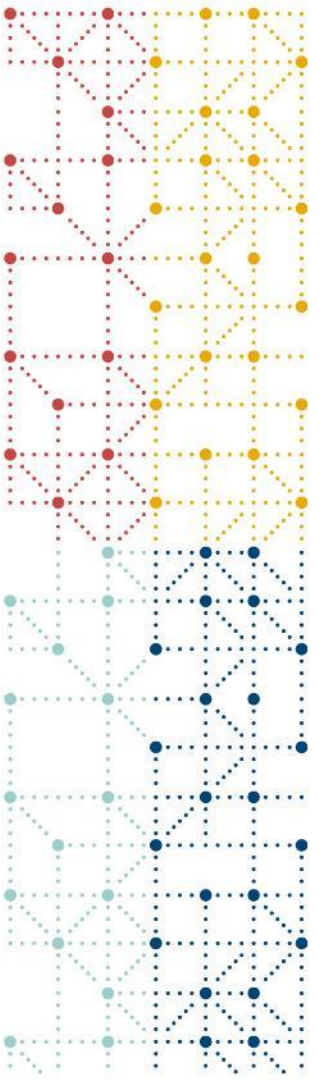
	element1	element2	element3	element4	element5	element6	element7	element8
1	1	CDISCPILOT01	DM	CDISC001	1115	2012-11-30	2013-01-23	2012-11-30
2	2	CDISCPILOT01	DM	CDISC002	1211	2012-11-15	2013-01-14	2012-11-15
3	3	CDISCPILOT01	DM	CDISC003	1302	2013-08-29	2013-11-05	2013-08-29
4	4	CDISCPILOT01	DM	CDISC004	1345	2013-10-08	2014-03-18	2013-10-08
5	5	CDISCPILOT01	DM	CDISC005	1383	2013-02-04	2013-08-06	2013-02-04
6	6	CDISCPILOT01	DM	CDISC006	1429	2013-03-19	2013-04-30	2013-03-19
7	7	CDISCPILOT01	DM	CDISC007	1444	2013-01-05	2013-02-13	2013-01-05
8	8	CDISCPILOT01	DM	CDISC008	1445	2014-05-11	2014-11-01	2014-05-11
9	9	CDISCPILOT01	DM	CDISC009	1087	2012-10-22	2013-04-28	2012-10-22

# Reading Dataset-JSON - %read\_datasetjson()

```
%read_datasetjson(  
    jsonpath=,          /* Path to Dataset-JSON file          */  
    datalib=           /* Library to save SAS dataset          */  
    dropseqvar=N,     /* Drop ITEMGROUPDATASEQ variable? (Y/N) */  
    savemetadata=N,  /* Save Dataset-JSON metadata? (Y/N)    */  
    metadatalib=     /* Library to save Dataset-JSON metadata */  
);
```

# Reading Dataset-JSON - %read\_datasetjson() - loop

```
%util_gettree(  
  dir=&project_folder/json_out/adam,  
  outds=work.dirtree_adam,  
  where=%str(ext="json" and dir=0)  
);  
  
data _null_;  
  length code $2048;  
  set work.dirtree_adam;  
  code=cats('%nrstr(%read_datasetjson(',  
            'jsonpath=', fullpath, ', ', '  
            'datalib=outadam, ', '  
            'dropseqvar=Y, ', '  
            'savemetadata=Y, ', '  
            'metadatalib=metasvad',  
            ');)');  
  call execute(code);  
run;
```



## Comparing SAS datasets / Dataset-JSON files



# Comparing SAS datasets

- Compare results in a summary
- Details for datasets that had differences

```
%macro util_comparedata(  
    baselib=,  
    complib=,  
    dsname=,  
    compareoptions=%str(listall criterion=0.00000001 method=absolute),  
    resultds=,  
    detailall=N  
);
```

## Compare results for dataset ADADAS

The COMPARE Procedure  
 Comparison of DATAADAM.ADADAS with OUTADAM.ADADAS  
 (Method=ABSOLUTE, Criterion=1.0E-08)

### Data Set Summary

Dataset	Created	Modified	NVar	NObs	Label
DATAADAM.ADADAS	02OCT23:22:29:17	02OCT23:22:29:17	40	12463	ADAS-COG Analysis Dataset
OUTADAM.ADADAS	03OCT23:23:04:27	03OCT23:23:04:27	40	12463	ADAS-Cog Analysis

### Variables Summary

Number of Variables in Common: 40.  
 Number of Variables with Differing Attributes: 8.

### Listing of Common Variables with Differing Attributes

Variable	Dataset	Type	Length	Label
ITTFL	DATAADAM.ADADAS	Char	1	Intent-To-Treat Population Flag
	OUTADAM.ADADAS	Char	1	Intent-to-Treat Population Flag
AVISIT	DATAADAM.ADADAS	Char	8	Analysis Visit
	OUTADAM.ADADAS	Char	16	Analysis Visit
VISIT	DATAADAM.ADADAS	Char	17	Visit Name
	OUTADAM.ADADAS	Char	10	Visit Name

resultc
DSLABEL/LENGTH/LABEL
DSLABEL/LABEL
DSLABEL/FORMAT/LABEL
DSLABEL/LENGTH
DSLABEL/LENGTH
DSLABEL/LENGTH
DSLABEL/LENGTH
DSLABEL/LENGTH
DSLABEL/LENGTH/LABEL

# Comparing JSON files

WinMerge (<https://winmerge.org>) can compare JSON files in 'pretty' mode

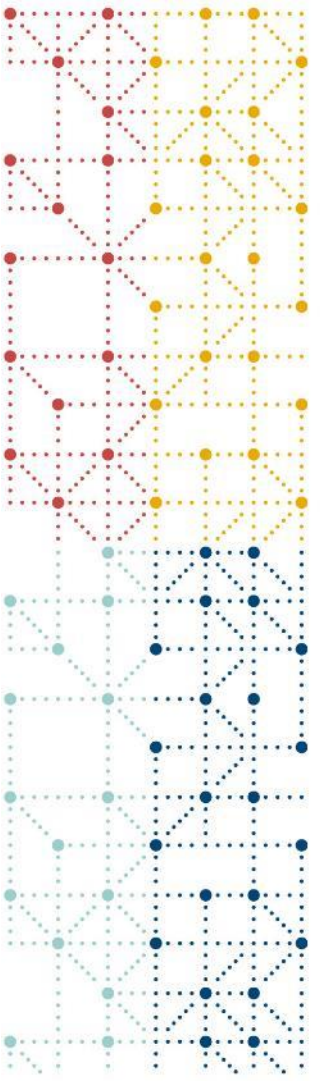
The screenshot shows the WinMerge application window with the following details:

- Window Title: WinMerge - [json\ - json\_out\]
- Menu Bar: File Edit View Merge Tools Plugins Window Help
- Address Bar: C:\\_github\lexjansen\dataset-json-sas\json\ (Left) and C:\\_github\lexjansen\dataset-json-sas\json\_out\ (Right)
- Table of Comparison Results:

Filename	Folder	Comparison result	Left Date	Right Date	Extension
> adam		Folders are different	2023-07-30 08:11:53	* 2023-10-02 21:01:19	
▼ sdtm		Folders are different	2023-07-30 08:11:53	* 2023-10-02 21:01:22	
ae.json	sdtm	Text files are different	2023-07-30 08:11:53	* 2023-10-03 23:09:56	json
cm.json	sdtm	Text files are different	2023-07-30 08:11:53	* 2023-10-03 23:09:57	json
dd.json	sdtm	Text files are different	2023-07-30 08:11:53	* 2023-10-03 23:09:57	json
di.json	sdtm	Text files are different	2023-07-30 08:11:53	* 2023-10-03 23:09:57	json
dm.json	sdtm	Text files are different	2023-07-30 08:11:53	* 2023-10-03 23:09:58	json
ds.json	sdtm	Text files are different	2023-07-30 08:11:53	* 2023-10-03 23:09:58	json
ec.json	sdtm	Text files are different	2023-07-30 08:11:53	* 2023-10-03 23:09:58	json
ex.json	sdtm	Text files are different	2023-07-30 08:11:53	* 2023-10-03 23:09:59	json
fajson	sdtm	Text files are different	2023-07-30 08:11:53	* 2023-10-03 23:09:59	json
ft.json	sdtm	Text files are different	2023-07-30 08:11:53	* 2023-10-03 23:09:59	json
ie.json	sdtm	Text files are different	2023-07-30 08:11:53	* 2023-10-03 23:09:59	json
lb.json	sdtm	Text files are different	2023-07-30 08:11:53	* 2023-10-03 23:10:00	json
mh.json	sdtm	Text files are different	2023-07-30 08:11:53	* 2023-10-03 23:10:00	json
oe.json	sdtm	Text files are different	2023-07-30 08:11:53	* 2023-10-03 23:10:00	json
qsph.json	sdtm	Text files are different	2023-07-30 08:11:53	* 2023-10-03 23:10:01	json
qssl.json	sdtm	Text files are different	2023-07-30 08:11:53	* 2023-10-03 23:10:01	json
relrec.json	sdtm	Text files are different	2023-07-30 08:11:53	* 2023-10-03 23:10:01	json

The context menu for 'dm.json' includes the following options:

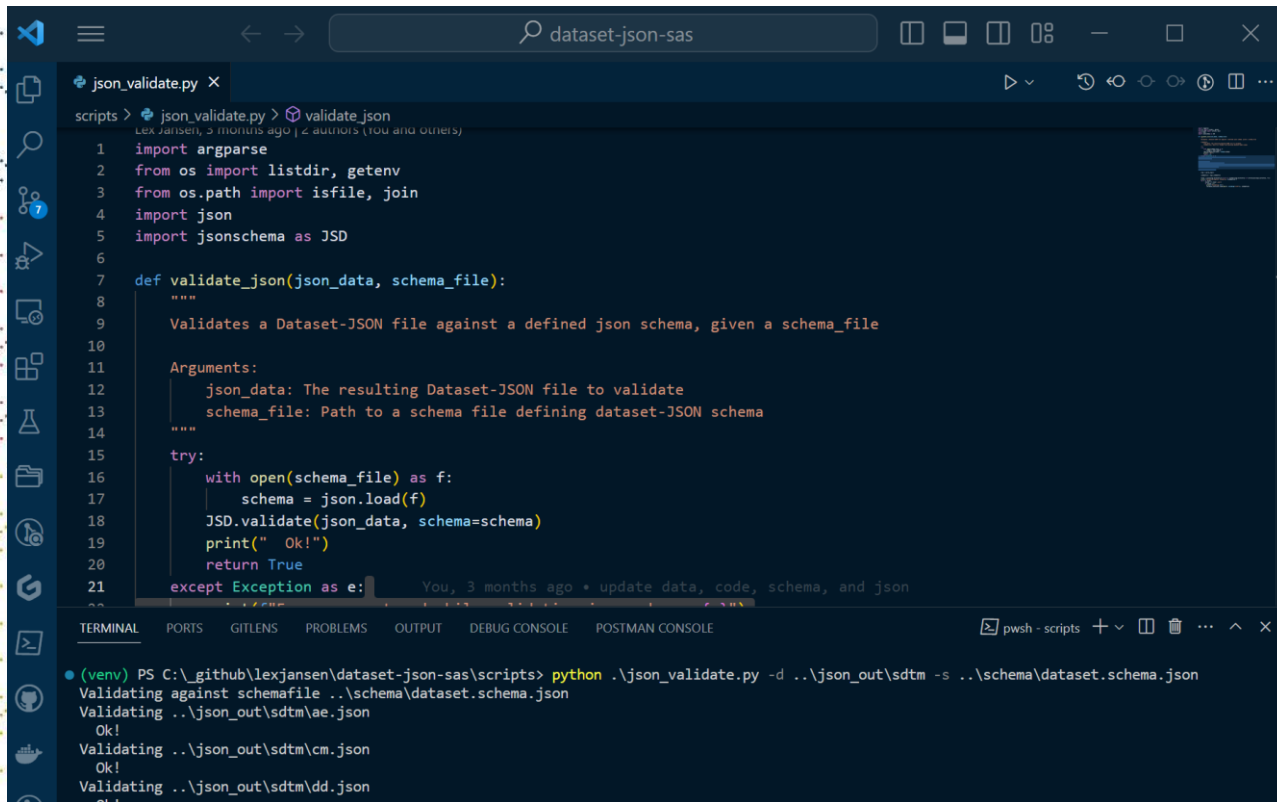
- Compare
- Compare Non-horizontally
- Compare As
  - Text
  - Table
  - Binary
  - Image
  - Webpage
  - Prettify JSON (highlighted)
  - Query JSON Data...
- Copy
- Move
- Delete
- Rename
- Hide Items
- Open Left
- Open Right



# Validating Dataset-JSON files



# Validating Dataset-JSON files (Python)

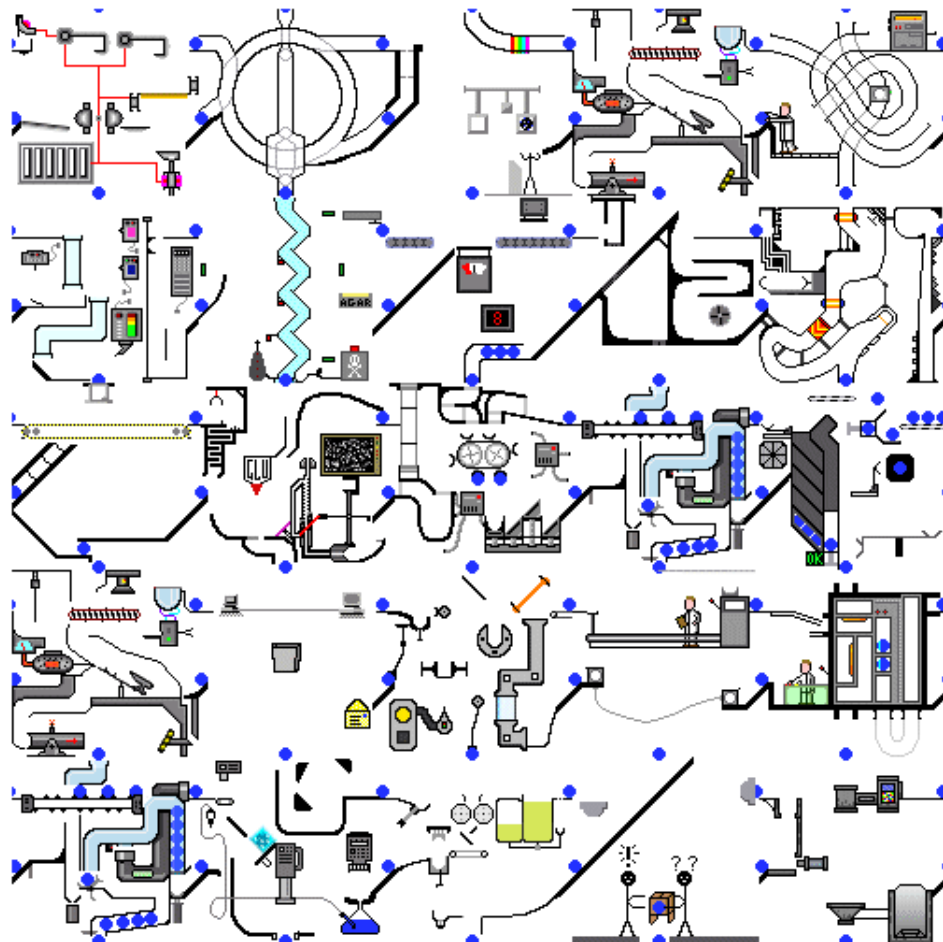


```
dataset-json-sas
json_validate.py x
scripts > json_validate.py > validate_json
Lex Jansen, 3 months ago | 2 authors (you and others)
1 import argparse
2 from os import listdir, getenv
3 from os.path import isfile, join
4 import json
5 import jsonschema as JSD
6
7 def validate_json(json_data, schema_file):
8     """
9     Validates a Dataset-JSON file against a defined json schema, given a schema_file
10
11     Arguments:
12     json_data: The resulting Dataset-JSON file to validate
13     schema_file: Path to a schema file defining dataset-JSON schema
14     """
15     try:
16         with open(schema_file) as f:
17             schema = json.load(f)
18             JSD.validate(json_data, schema=schema)
19             print(" Ok!")
20             return True
21     except Exception as e:
22         print(f"Error: {e}")
23
24 if __name__ == '__main__':
25     parser = argparse.ArgumentParser()
26     parser.add_argument('-d', '--dir', type=str, required=True, help='Directory containing Dataset-JSON files')
27     parser.add_argument('-s', '--schema', type=str, required=True, help='Path to the schema file')
28     args = parser.parse_args()
29     dir_path = args.dir
30     schema_file = args.schema
31
32     for file in listdir(dir_path):
33         if isfile(join(dir_path, file)):
34             with open(join(dir_path, file)) as f:
35                 json_data = json.load(f)
36                 validate_json(json_data, schema_file)
```

TERMINAL PORTS GITLENS PROBLEMS OUTPUT DEBUG CONSOLE POSTMAN CONSOLE pwsh - scripts + v [] [] ... ^ x

```
(venv) PS C:\github\lexjansen\dataset-json-sas\scripts> python .\json_validate.py -d ..\json_out\sdtm -s ..\schema\dataset.schema.json
Validating against schemafile ..\schema\dataset.schema.json
Validating ..\json_out\sdtm\ae.json
Ok!
Validating ..\json_out\sdtm\cm.json
Ok!
Validating ..\json_out\sdtm\dd.json
Ok!
```

# Demo



спасибо 谢谢  
GRACIAS 谢谢  
**THANK YOU**  
ありがとうございました MERCI  
DANKE धन्यवाद  
شُكراً **OBRIGADO**

**GitHub:** <https://github.com/lexjansen/dataset-json-sas>

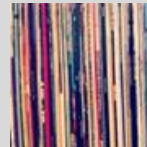
**open issues at:** <https://github.com/lexjansen/dataset-json-sas/issues>

**Email:** [ljansen@cdisc.org](mailto:ljansen@cdisc.org)

[lexjansen@gmail.com](mailto:lexjansen@gmail.com)

**Web:** <https://www.lexjansen.com>

**LinkedIn:** <https://www.linkedin.com/in/lexjansen/>



**cdisc**