Harnessing the power of JSON and SAS

PHUSE SDE, Tarrytown, NY

Pritesh Desai, SAS Institute Inc.
Samiul Haque, SAS Institute Inc.

Agenda

- State of the industry and FDA
- Introduction to JSON
- How SAS handles JSON
- SAS Support for dataset-JSON and beyond
- Q&A

State of the industry

Technical Limitation of previous transport format

- Limited variable type
- No multibyte characters
- Restricted variable width and format
- Storage limitations
- Structural limitations
- Larger file sizes (Dataset-XML)
- Metadata is separate (Dataset-XML)

Timeline

SAS V5 Transport format XPT 1989

FDA Regulation of Electronic Submission 1997

FDA standardized submission using SAS XPT 1999

FDA recommendation to include Define.XML 2005

CDER, CBER of FDA evaluated Dataset-XML 2013

PHUSE Transport for Next Generation 2017

Dataset-JSON

Introduction to JSON

As per https://www.w3schools.com/js/js_json_intro.asp

- JSON stands for JavaScript Object Notation
- It is a lightweight data-interchange format
- It is plain text written in JavaScript object notation
- It is used to send data between computers
- It is language independent

As per Working with Dataset-JSON using SAS ©

- JSON value can be an *object, array, number, string, true, false or null.*

- Example:

```
{
   "clinicalData": {
    "studyOID": " cdisc.com/CDISCPILOT01",
     "metaDataVersionOID": "MDV.MSGv2.0.SDTMIG.3.3.SDTM.1.7",
      "itemGroupData": {
       "IG.DM": { ... },
          ... }
        }
}
```

Why JSON ?

JSON stands for JavaScript Object Notation. JSON is a lightweight format for storing and transporting data.

Interoperability

- Files are compact
- Easier to read for humans

String Limit

- No inherent limit
- Default max is 2097152 characters

Dataset Size

- Files are smaller
- Makes data transfer faster

Column Limit

- Max allowed packet bytes
- Default is 104,857,600 bytes

Unicode

- Language-independent
- Parsers available for most programming languages

Metadata

- Describes data, metadata and location
- File generation and data producer

JSON in SAS

Read JSON

- LIBNAME JSON

```
filename mydata "/path/my-json-file.json";
libname myjson JSON fileref=mydata;

proc datasets lib=myjson; quit;
```

Write JSON

- PROC JSON

```
proc json out="path-to-DefaultOutput.json" pretty;
   export sashelp.class (where=(age=11));
run;
```

How SAS Handles JSON – Libname Engine

As per SAS community post on How to read JSON data in SAS Chris Hemedinger has some good notes as following for JSON libname engine

Notes:

- The JSON libname engine is a read-only engine. To convert SAS data to JSON, use PROC JSON.
- The JSON engine can read data only from a fileref, not from literal string values or from variables in a table. If you have JSON data from another source (such as a field in a database), you must first write the content to a file so the JSON engine can process it.
- The JSON engine requires complete, valid JSON in order to map the data into SAS. JSON snippets, or variations such as JSONL (multiple JSON objects in a single file, one per line), require preprocessing to build valid JSON before the engine can read them.
- JSON is almost always UTF-8 encoded text (so it supports national characters and even extended characters such as emojis). Therefore, you'll have the best success when using SAS with Unicode support, or ENCODING=UTF8 enabled. This is the default in SAS Viya

```
filename mydata "/path/my-json-file.json";
libname myjson JSON fileref=mydata;

proc datasets lib=myjson; quit;
```

How SAS Handles JSON-Procedure

As per SAS Documentation

What Does the JSON Procedure Do?

The JSON procedure reads data from a SAS data set and writes it to an external file in JSON representation. You can control the exported data with several options that remove content and affect the format. In addition to exporting data from a SAS data set, PROC JSON provides statements that enable you to write additional data to the external file and control JSON containers. SAS provides the JSON engine to read a JSON file. For more information, see LIBNAME Statement: JSON Engine in SAS Global Statements: Reference.

```
proc json out="path-to-DefaultOutput.json" pretty;
   export sashelp.class (where=(age=11));
run;
```

RAW JSON FILE

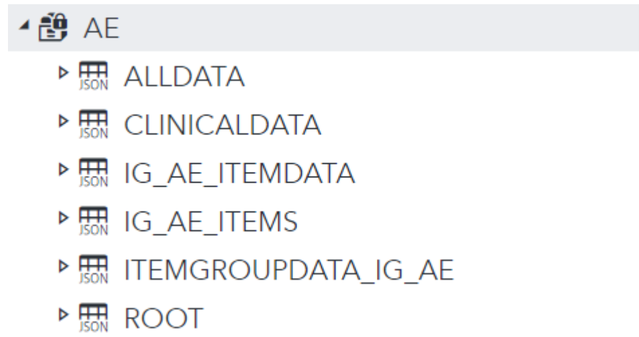
File Edit Format View Help


```
{"creationDateTime":"2023-06-28T15:38:42","datasetJSONVersion":"1.0.0","fileOID":"www.cdisc.org/StudyMSGv2/1/Def
type":"integer","length":3},{"OID":"IT.AE.AELNKID","name":"AELNKID","label":"Link ID","type":"string","length":5
:"string","length":1},{"OID":"IT.AE.AEBODSYS","name":"AEBODSYS","label":"Body System or Organ Class","type":"str
":"IT.AE.AESCONG","name":"AESCONG","label":"Congenital Anomaly or Birth Defect","type":"string","length":1},{"OI
","length":8},{"OID":"IT.AE.AEENDY","name":"AEENDY","label":"Study Day of End of Adverse Event","type":"integer"
ERED/NOT RESOLVED","N","N","N","N","N","N","N","TREATMENT","2012-11-16","",2,null,"ONGOING","2013-01-14"],[4,"CD
OLVED","N","N","N","N","N","N","N","TREATMENT","2012-11-29","",15,null,"ONGOING","2013-01-14"],[8,"CDISCPILOT01"
"N","Y","N","TREATMENT","2013-01-14","2013-01-14",61,61,"",""],[12,"CDISCPILOT01","AE","CDISC003",1,"1","HYPERHI
","N","TREATMENT","2013-09-01","",4,null,"ONGOING","2013-02-13"],[16,"CDISCPILOT01","AE","CDISC003",5,"5","MALAI
"N","N","TREATMENT","2013-09-02","",5,null,"ONGOING","2013-02-13"],[20,"CDISCPILOT01","AE","CDISC003",9,"9","DIZ
null,"ONGOING","2013-02-13"],[24,"CDISCPILOT01","AE","CDISC003",13,"13","EPISTAXIS","","","","","","","","","","
013-10-23","",56,null,"ONGOING","2013-02-13"],[28,"CDISCPILOT01","AE","CDISC003",17,"18","FEELING ABNORMAL","","
ATMENT","2013-02-07","2013-02-07",4,4,"",""],[32,"CDISCPILOT01","AE","CDISC005",2,"2","INJECTION SITE REACTION",
"TREATMENT","2013-06-24","2013-06-24",141,141,"",""],[36,"CDISCPILOT01","AE","CDISC005",6,"8","HEADACHE","","","
9","",15,null,"ONGOING","2013-06-20"],[40,"CDISCPILOT01","AE","CDISC007",3,"6","PARAESTHESIA","","","","","","",
09",3,"3","HEADACHE","","","","","","","","","","","","","MILD","N","DOSE NOT CHANGED","NOT RELATED","NOT RECOVE
,"","","","","","MILD","N","DOSE NOT CHANGED","POSSIBLY RELATED","RECOVERED/RESOLVED","N","N","N","N","N","N","N
T RELATED","NOT RECOVERED/NOT RESOLVED","N","N","N","N","N","N","N","TREATMENT","2013-04-27","",25,null,"ONGOING
D","RECOVERED/RESOLVED","N","N","N","N","N","N","N","TREATMENT","2013-05-27","2013-05-31",55,59,"",""],[57,"CDIS
D/RESOLVED","N","N","N","N","N","N","N","TREATMENT","2013-10-09","2013-10-09",19,19,"",""],[61,"CDISCPILOT01","A
","TREATMENT","2013-12-29","2014-02-19",100,152,"",""],[65,"CDISCPILOT01","AE","CDISC018",1,"1","BACK PAIN","","
C018",5,"5","LETHARGY","","","","","","","","","","","","","","MODERATE","N","DOSE NOT CHANGED","POSSIBLY RELATED",
```

Reading JSON data in SAS

JSON engine reads JSON data and convert It into sets of related tables

```
%let fileloc ='/nfsshare/dataset_json/json_sas/json/sdtm/ae.json';
libname ae JSON &fileloc;
```



- AE
  - ALLDATA
  - CLINICALDATA
  - IG_AE_ITEMDATA
  - IG_AE_ITEMS
  - ITEMGROUPDATA_IG_AE
  - ROOT

```
proc sql noprint;
select name into :Col_name separated by ' '
from AE.IG_AE_ITEMS
order by ordinal_items;
quit;
%put &Col_name;
```

```
proc sql noprint;
select count(*) into:Num_elements from AE.IG_AE_ITEMS;
quit;

%put &Num_elements;
```

```
%macro doit;
  data AE;
      set AE.ig_ae_itemdata;
      %do i = 1 %to Num_elements;
          rename element&i.= %scan(&Col_name.,&i.);
      %end;
  run;
 %mend doit;
```

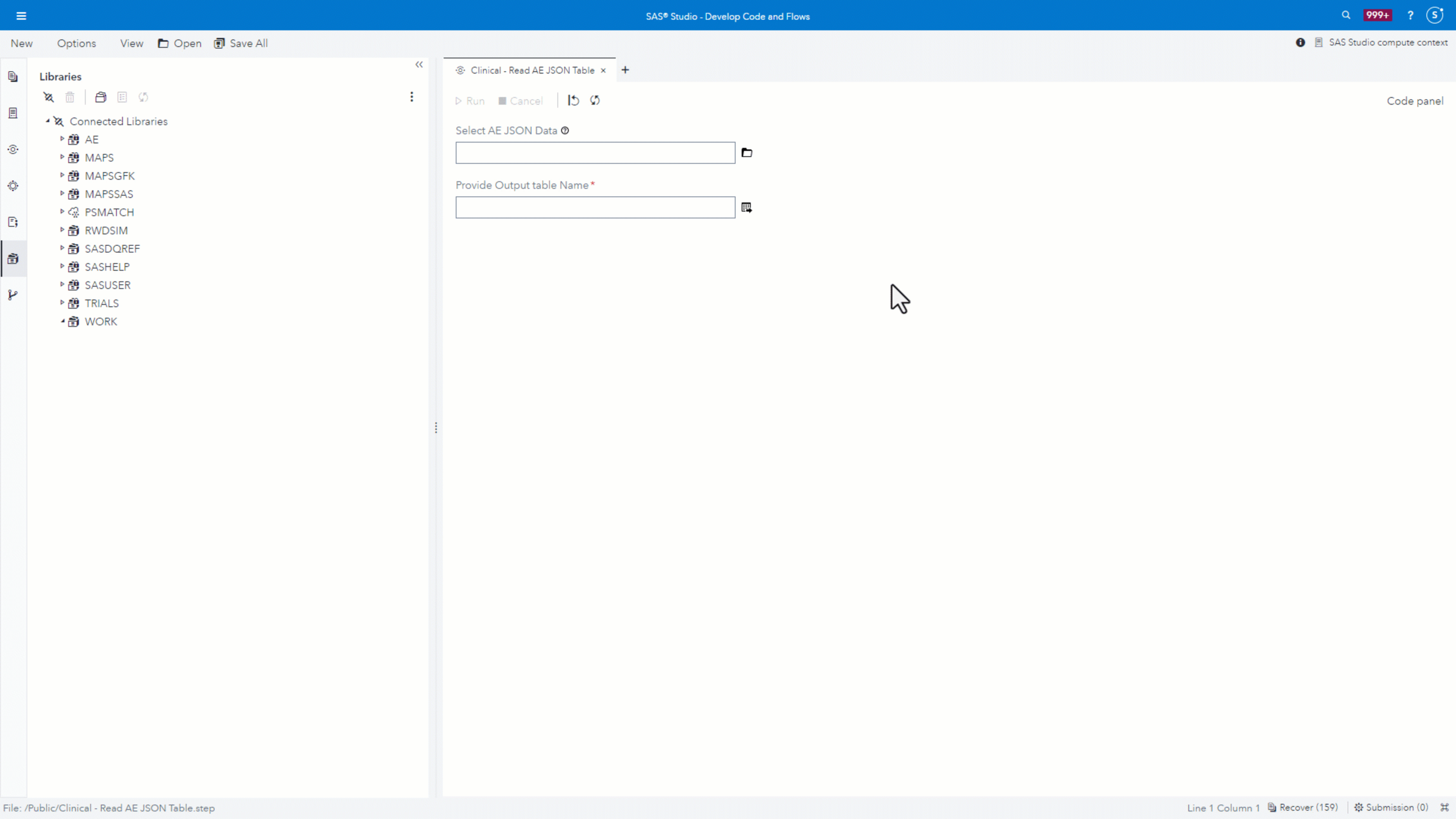
At this point it becomes a simple programming Exercise to get the AE table in desired format

WORK.AE

AE — Table rows: 74 | Columns

Enter expression

	STUDYID	DO...	USUBJID	AESEQ	AELN...	AETERM
1	CDISCPILOT01	AE	CDISC001	1	1	INJECTION SITE REACTION
2	CDISCPILOT01	AE	CDISC001	2	2	FATIGUE
3	CDISCPILOT01	AE	CDISC002	1	1	INJECTION SITE REACTION
4	CDISCPILOT01	AE	CDISC002	2	2	SHOULDER PAIN
5	CDISCPILOT01	AE	CDISC002	3	3	JOINT DISLOCATION
6	CDISCPILOT01	AE	CDISC002	4	4	INCONTINENCE
7	CDISCPILOT01	AE	CDISC002	5	5	INJECTION SITE REACTION
8	CDISCPILOT01	AE	CDISC002	6	6	SKIN LACERATION
9	CDISCPILOT01	AE	CDISC002	7	7	CONFUSIONAL STATE
10	CDISCPILOT01	AE	CDISC002	8	8	DYSPNOEA
11	CDISCPILOT01	AE	CDISC002	9	9	SUDDEN DEATH
12	CDISCPILOT01	AE	CDISC003	1	1	HYPERHIDROSIS
13	CDISCPILOT01	AE	CDISC003	2	2	STOMACH DISCOMFORT
14	CDISCPILOT01	AE	CDISC003	3	3	PAIN
15	CDISCPILOT01	AE	CDISC003	4	4	NASAL CONGESTION
16	CDISCPILOT01	AE	CDISC003	5	5	MALAISE
17	CDISCPILOT01	AE	CDISC003	6	6	MYALGIA
18	CDISCPILOT01	AE	CDISC003	7	7	PHARYNGOLARYNGEAL PAIN

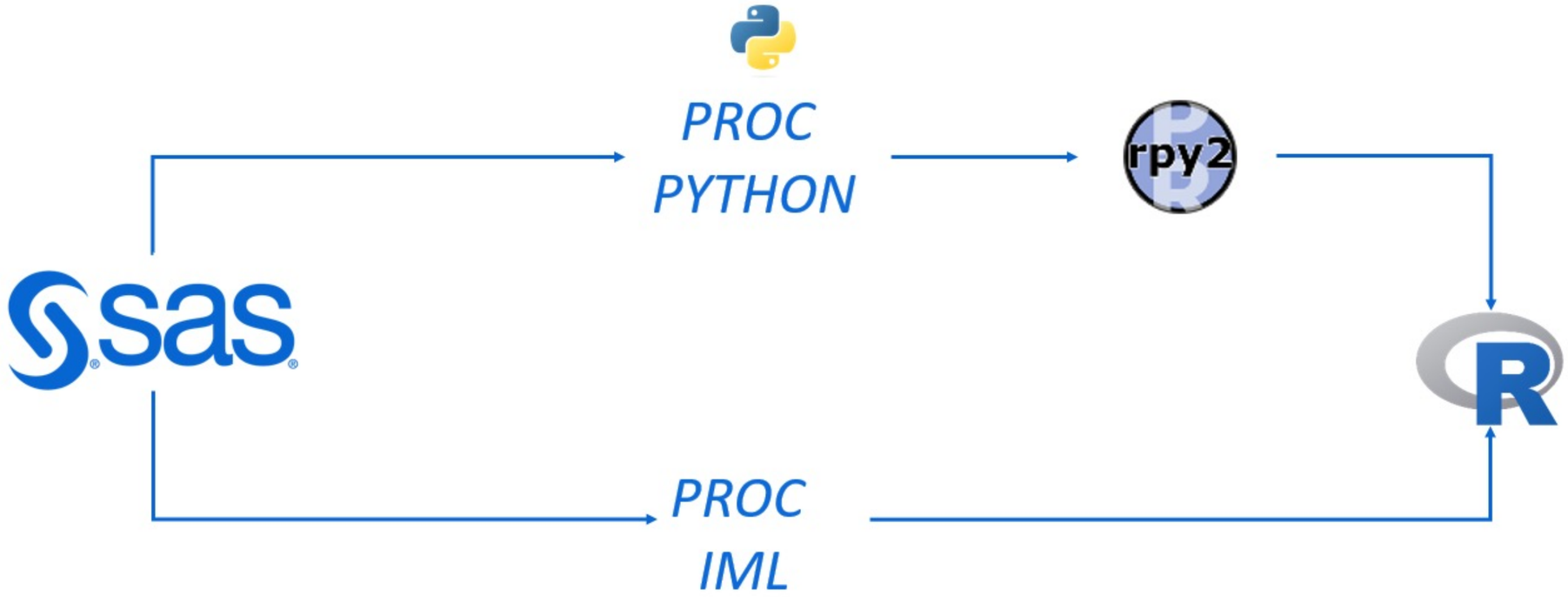

SAS® Studio - Develop Code and Flows
999+
New
Options
View
Open
Save All
SAS Studio compute context
Libraries
Connected Libraries
AE
MAPS
MAPSGFK
MAPSSAS
PSMATCH
RWDSIM
SASDQREF
SASHELP
SASUSER
TRIALS
WORK
Clinical - Read AE JSON Table
Run
Cancel
Code panel
Select AE JSON Data
Provide Output table Name*
File: /Public/Clinical - Read AE JSON Table.step
Line 1 Column 1
Recover (159)
Submission (0)

SAS, JSON, and Opensource

- JSON has been extensively used by web developer community
- Resources available in Python
- SAS can execute Python (and R) through PROC Python
- SAS Platform supports SAS, Python, and R

```
proc python;
submit ;

import pandas as pd
import json

#your python code here

# Closing file

endsubmit;
run;
```

SAS, JSON, and Opensource



To Learn More

- An excellent paper by Lex Jansen Working with Dataset-JSON using SAS©
- Webinar How to Use JSON Data in SAS
- SAS Community Forum

SAS support for JSON and beyond

As per 2022 PharmaSUG paper Working with Dataset-JSON using SAS© , Lex Jansen from CDISC added a nice conclusion.

CONCLUSION

SAS fully supports reading and writing Dataset-JSON files in an efficient way.

VALIDATION

When reading or writing JSON files, it is important to validate the process. The published JSON specification also comes with a JSON schema that can be used to validate JSON files [15].

With a simple Python program, the JSON file can be validated against the schema (Appendix 2).

Also, the read/write process can be validated by doing a roundtrip and comparing SAS datasets or JSON-files:

- Dataset-JSON → SAS dataset → Dataset-JSON
- SAS dataset → Dataset-JSON → SAS dataset

SAS datasets can be compared with PROC COMPARE. It can be expected that there will differences for numeric floating-point variables in the order of machine precision.

Thank You!

pritesh.desai@sas.com
samiul.haque@sas.com