



SAS2P21CLI

Pinnacle 21 3.1.0

Gustav Bernard, AD

Gustav Bernard

Gustav Bernard is an Associate Director at IQVIA who has been with the company for 17 years. His work focuses on the implementation of CDISC Standards (SDTM, ADaM and Define-XML) within the IQVIA Global Biostatistics department.

He is currently Leading the ADaM Innovation Initiatives. He has also created the Define-XML 2.0 automation process within IQVIA and the ADaM Spec Generator application. Gustav earned a Bachelor of Business in computer science in 2004 from the University of the Orange Free State in South Africa.





Agenda

- + Introduction
- + Package
- + User Manual
- + SAS Macro
- + IQ Conform

Introduction

Introduction

- ✓ SAS Macro to Run any of the Pinnacle 21 Community 3.1.0 functionality using SAS.
- ✓ Setup once for a study and reran when new data is created.
- ✓ Multiple Calls can be saved \ ran from the macro
 - ✓ Call to Validate Data Only
 - ✓ Call to Validate Define-XML Only
 - ✓ Call to Validate Data and Define
- ✓ SAS Macro should be run Interactively in Enterprise Guide and Not Batch Submitted.

SAS Macro

- Validate Macro Parameters
- Create Parameters for P21

Pinnacle
21

- Creates P21 output
 - xlsx or xml

SAS Macro

- Verifies P21 output was created

Why? - Speed

Pinnacle 21 Community 3.1.0

- Configuration Changes
- Require a min of 300 mb of server space per user
- Users would be responsible for maintaining metadata and creating + maintaining p21 account

SAS Macro

- No Server Space Requirement
- Central Location for maintaining Metadata and single p21 account
- Improved Run time for Validation
 - Using both the SAS Server + SAS GRID Nodes when processing
 - Have seen Validation time reduced from 6 hours to 30 minutes

Why not an application to run the P21 CLI commands in Batch

- Would not have benefitted from SAS GRID Nodes processing speed

Why? – Process Enhancements



Define.xml Automation

Call sas2p21cli during Define automation process to Create Initial specification from xpt files to have data representation in P21 format.

Use sas to read in dataset specifications and data to complete the p21 specification.

Call sas2p21cli to Create Define.xml.

Call sas2p21cli to Validate Define.xml.

Call sas2p21cli to Validate CDISC data and Define.xml.

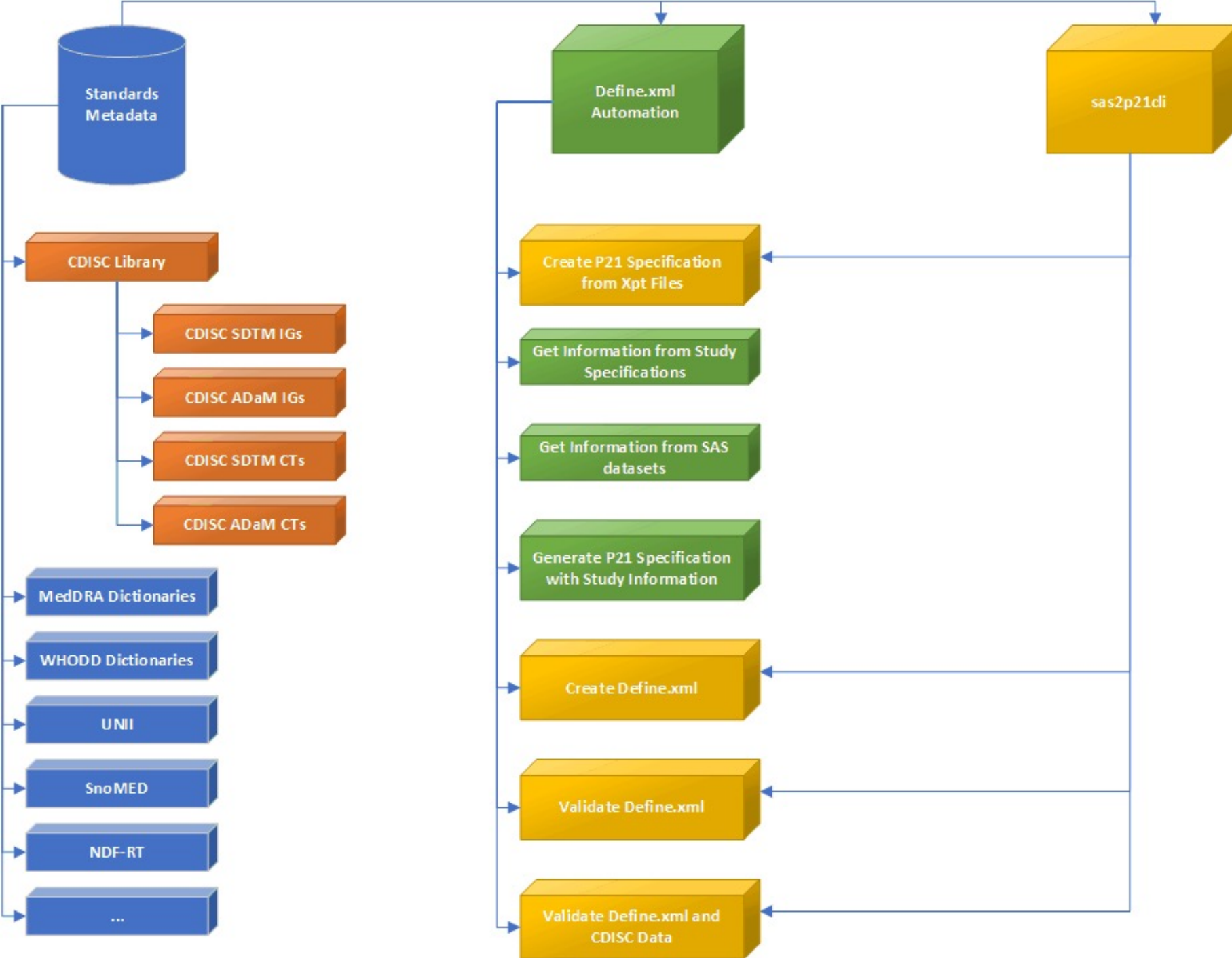
Can create the define from within SAS environment without having to switch environments.



Central Area with Metadata Standards

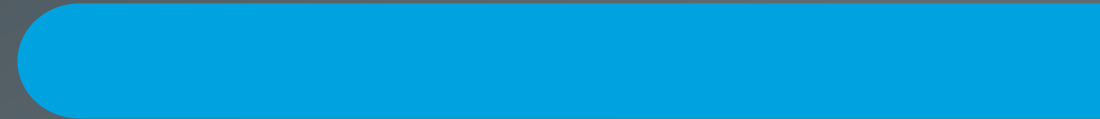
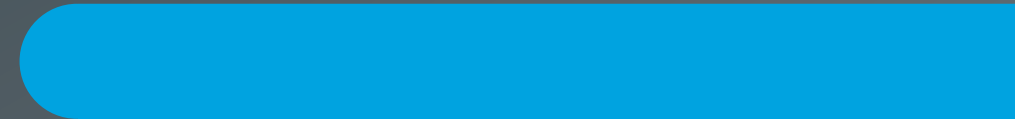
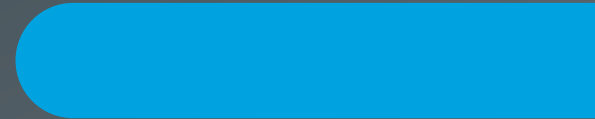
Metadata Standards Downloaded from CDISC Library and other Sources and maintained in central area.

Multiple processes can use Metadata



Sas2p21cli macro implementation into the define automation process enables to create define.xml from an single environment.

Package



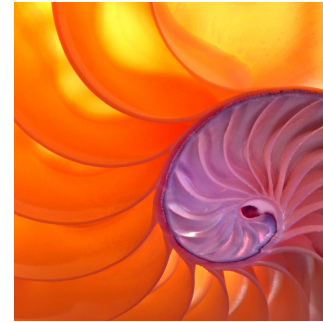
Package



Shared with Teams

Sas2p21cli.7z (zip file)

- Call_sas2p21cli.sas (sas macro)
- Sas2p21cli_um.pdf (user manual)



Not shared with Teams

Sas2p21cli.sas

Sas2p21cli.sas

Function 1

```
*****  
*Function 1 --> Validate Data Only SDTM or ADaM;  
*****  
%if &Function=1 %then %do;  
  %let OutputName = %sysfunc(catx(-,p21-cli-Validate(DATA), &StdVer, &TOD));  
  %put NOTE: Output Name: &OutputName;  
  x  
  java -jar "&jarpath.\&jarfile." ^  
    --engine.version=&EngineFullName ^  
    --standard=&Standard ^  
    --standard.version=&StandardVersion ^  
    --source.&Standard=&xptpath." ^  
    --config=&configLoc." ^  
    --meddra.version=&MedDRAv ^  
    %if %isblank(&NDF_RT.) = 0 %then %do;  
      --ndf-rt.version=&NDF_RT. ^  
    %end;  
    %if %isblank(&SnoMed.) = 0 %then %do;  
      --snomed.version=&SnoMed. ^  
    %end;  
    %if %isblank(&Unii.) = 0 %then %do;  
      --unii.version=&Unii. ^  
    %end;  
    %if %isblank(&SDTMct.) = 0 %then %do;  
      --cdisc.ct.sdtm.version=&SDTMct ^  
    %end;  
    %if %isblank(&ADaMct.) = 0 %then %do;  
      --cdisc.ct.adam.version=&ADaMct ^  
    %end;  
    --report=&SavePath.\&OutputName..xlsx" ^  
  ;  
%end;
```

Validate Data Only

Optional Parameters will only be parsed if available.

Sas2p21cli.sas

Function 2

```
*****
*Function 2 --> Validate Data and Define SDTM or ADaM;
*****
%if &Function=2 %then %do;
  %let OutputName = %sysfunc(catx(.,p21-cli-Validate(FULL), &StdVer, &TOD));
  %put NOTE: Output Name: &OutputName;
  x
  java -jar "&jarpath.\&jarfile." ^
    --engine.version="&EngineFullName" ^
    --standard=&Standard ^
    --standard.version=&StandardVersion ^
    --source.&Standard="&xptpath." ^
    --source.define="&InputFilePath.\&InputFileName..xml" ^
    --config="&configLoc." ^
    --meddra.version=&MedDRAv ^
    %if %isblank(&NDF_RT.) = 0 %then %do;
      --ndf-rt.version=&NDF_RT. ^
    %end;
    %if %isblank(&SnoMed.) = 0 %then %do;
      --snomed.version=&SnoMed. ^
    %end;
    %if %isblank(&Unii.) = 0 %then %do;
      --unii.version=&Unii. ^
    %end;
    %if %isblank(&SDTMct.) = 0 %then %do;
      --cdisc.ct.sdtm.version=&SDTMct ^
    %end;
    %if %isblank(&ADaMct.) = 0 %then %do;
      --cdisc.ct.adam.version=&ADaMct ^
    %end;
    --report="&SavePath.\&OutputName..xlsx" ^
  ;
%end;
```

Validate Data and Define.xml

Sas2p21cli.sas

Function 3

```
*****
*Function 3 --> Validate Define;
*****

%if &Function=3 %then %do;
    %let OutputName = %sysfunc(catx(-,p21-cli-Validate(DEFINE), &StdVer, &TOD));
    %put NOTE: Output Name: &OutputName;
    x
    java -jar "&jarpath.\&jarfile." ^
        --engine.version="&EngineFullName" ^
        --config="&configLoc." ^
        --standard=&Standard ^
        --standard.version=&StandardVersion ^
        --source.define="&InputFilePath.\&InputFileName..xml" ^
    %if %isblank(&SDTMct.) = 0 %then %do;
        --cdisc.ct.sdtm.version=&SDTMct ^
    %end;
    %if %isblank(&ADaMct.) = 0 %then %do;
        --cdisc.ct.adam.version=&ADaMct ^
    %end;
        --report="&SavePath.\&OutputName..xlsx" ^
    ;
%end;
```

Validate Define.xml

Sas2p21cli.sas

Function 4

```
*****
*Function 4 --> Create Specification;
*****
%if &Function=4 %then %do;
  %let OutputName = %sysfunc(catx(-,p21-cli-Spec, &StdVer, &TOD));
  %put NOTE: Output Name: &OutputName;
  x
  java -jar "&jarpath.\&jarfile." ^
    --standard=&Standard ^
    --standard.version=&StandardVersion ^
    --source.&Standard="&xptpath." ^
    --meddra.version=&MedDRAv ^
    --config="&configLoc.\&EngineShortName.\&config..xml" ^
  %if %isblank(&SDTMct.) = 0 %then %do;
    --cdisc.ct.sdtm.version=&SDTMct ^
  %end;
  %if %isblank(&ADaMct.) = 0 %then %do;
    --cdisc.ct.adam.version=&ADaMct ^
  %end;
  --output.format=define-excel ^
  --output="&SavePath.\&OutputName..xlsx" ^
;
%end;
```

Create P21 Specification
from xpt files

Sas2p21cli.sas

Function 5

```
*****  
*Function 5 --> Create Define.XML;  
*****  
%if &Function=5 %then %do;  
    %let OutputName = %sysfunc(catx(-,p21-cli-DefineXML, &StdVer, &TOD));  
    %put NOTE: Output Name: &OutputName;  
    x  
    java -jar "&jarpath.\&jarfile." ^  
        --config="&configLoc" ^  
        --standard=&Standard ^  
        --standard.version=&StandardVersion ^  
        --source="&InputFilePath.\&InputFileName..xlsx" ^  
        --output="&SavePath.\&OutputName..xml" ^  
    ;  
%end;
```

Create Define.xml

Sas2p21cli.sas

Function 6

```
*****
*Function 6 --> Create Specification from Define-XML;
*****
%if &Function=6 %then %do;
  %let OutputName = %sysfunc(catx(-,p21-cli-Spec, &StdVer, &TOD));
  %put NOTE: Output Name: &OutputName;
  x
  java -jar "&jarpath.\&jarfile." ^
    --config="&configLoc" ^
    --standard=&Standard ^
    --standard.version=&StandardVersion ^
    --source.define="&InputFilePath.\&InputFileName..xml" ^
    --meddra.version=&MedDRAv ^
    %if %isblank(&SDTMct.) = 0 %then %do;
      --cdisc.ct.sdtm.version=&SDTMct ^
    %end;
    %if %isblank(&ADaMct.) = 0 %then %do;
      --cdisc.ct.adam.version=&ADaMct ^
    %end;
    --output.format=define-excel ^
    --output="&SavePath." ^
  ;
%end;
```

Create P21 Specification from Define.xml

Sas2p21cli.sas

Macro Parameters

```
%Macro SAS2P21CLI
(
  Function=,
  Engine=FDA,
  Standard=adam,
  StandardVersion=1.1,
  xptPath=&root.\Data\Transport\&Standard,
  SDTMct=2020-09-25,
  ADaMCt=2020-09-25,
  MedDRAv=22.1,
  NDF_RT=,
  SNOMED=,
  UNII=,
  InputFilePath=&root.\Production\Define\&Standard\OUTPUT,
  InputFileName=define,
  SavePath=&root.\Documentation\CDISC_Compliance
)
/ minoperator;
```

Each Macro Parameter is validated.

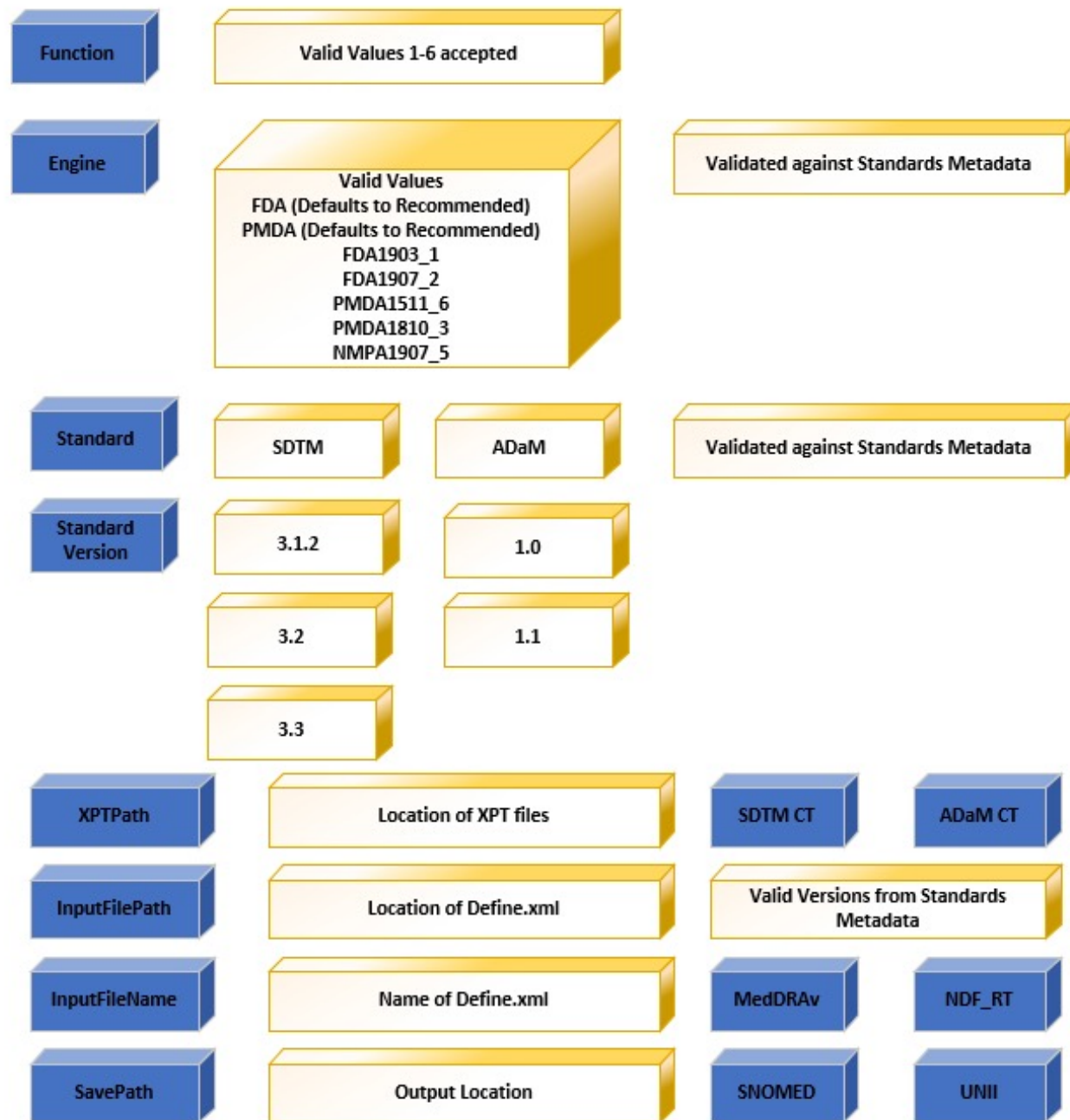
Function 1-6 Allowed

Validated vs Metadata

- Engine
- Standard
- Standard Version
- Controlled Terminology
- Dictionaries

Define Version in Define file validated
against StandardVersion

Folder and File Existence Checks



User Manual

User Manual - Bookmarks

SAS2P21CLI USER MANUAL

- Introduction
- Macro Parameters
 - Function=
 - Engine=
 - Standard=
 - StandardVersion=
 - xptPath=
 - SDTMct=
 - ADaMct=
 - MedDRAv=
 - Ndf_rt
 - SNOMED
 - UNII
 - InputFilePath=
 - InputFileName=
 - SavePath=

- Functions
 - Function = 1 - Validate SDTM or ADaM Data only
 - Function = 2 - Validate SDTM or ADaM DATA and Define-XML
 - Function = 3 - Validate Define-XML
 - Function = 4 - Create P21 Specification
 - Function = 5 - Create Define-XML
- Recommended folder locations
 - SAS Macro [call_sas2p21cli.sas]
 - Xpt files
 - Recommended Macro Output (save) Locations
 - Validation Reports
 - P21 Specification
 - Define-XML 2.0
- SAS Log messages

- Function = 1 - Validate SDTM or ADaM Data only
 - Required and Optional Macro Parameters
 - Standard output name - DATA
 - Examples
 - Validate Data FDA SDTM 3.2
 - Validate Data PMDA SDTM 3.1.2
 - Validate Data FDA ADaM 1.1
 - Validate Data PMDA ADaM 1.0

User Manual – Macro Parameters Section

- Introduction
- Macro Parameters
 - Function=
 - Engine=
 - Standard=
 - StandardVersion=
 - xptPath=
 - SDTMct=
 - ADaMct=
 - MedDRAv=
 - Ndf_rt
 - SNOMED
 - UNII
 - InputFilePath=
 - InputFileName=
 - SavePath=

- List each SAS Macro Parameters
- Gives List of Valid or expected values

MACRO PARAMETERS

FUNCTION=

- ✓ Valid Values = 1,2,3,4 or 5
 1. Function = 1 - Validate SDTM or ADaM Data only
 2. Function = 2 - Validate SDTM or ADaM and Define-XML
 3. Function = 3 - Validate Define-XML
 4. Function = 4 - Create P21 Specification
 5. Function = 5 - Create Define-XML

ENGINE=

- ✓ Valid Values = FDA or PMDA

STANDARD=

- ✓ Valid Values = sdtn or adam

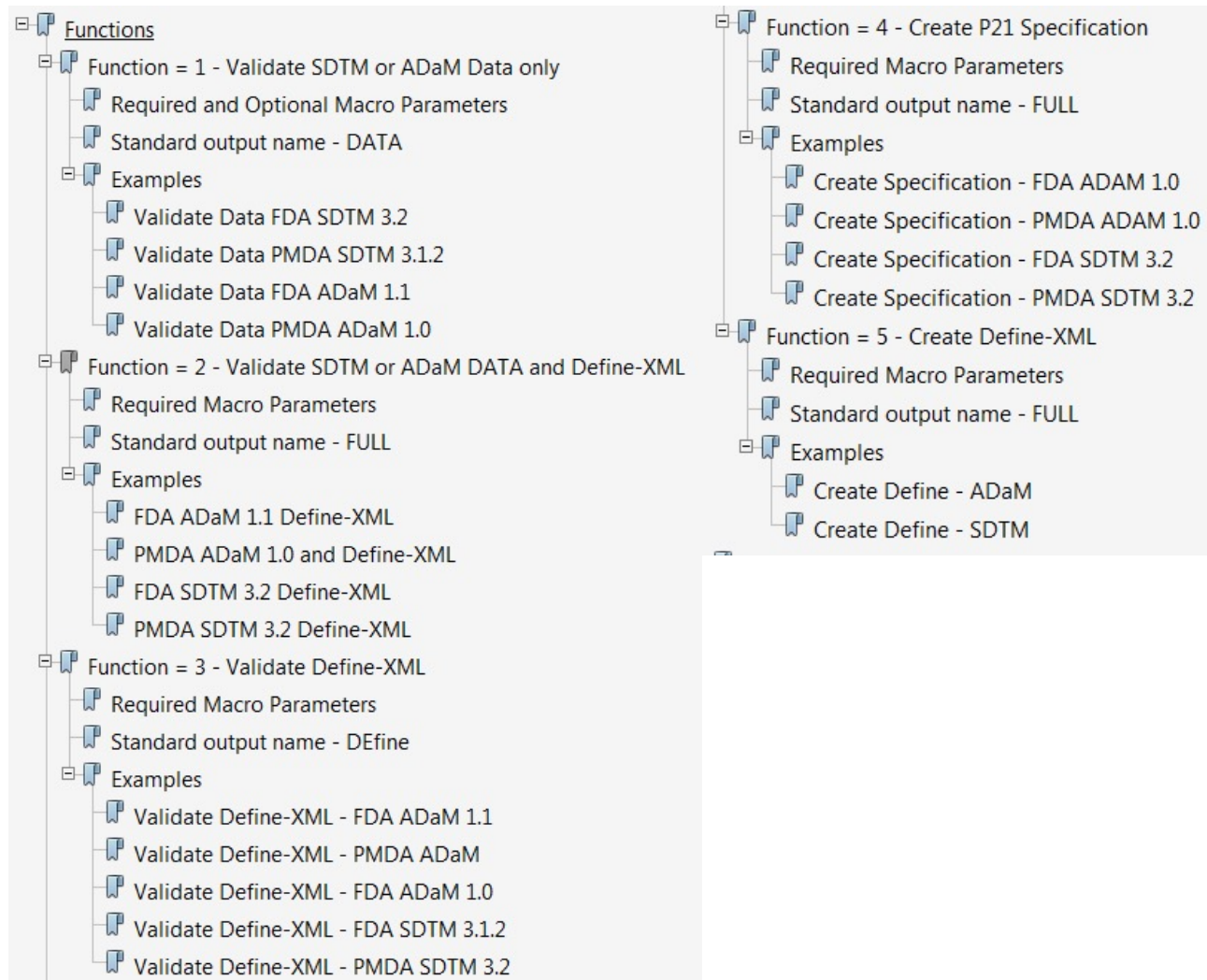
STANDARDVERSION=

- ✓ Valid Values = 1.0, 1.1, 3.1.2, 3.1.3 or 3.2
 - For ADaM
 - FDA ADaM 1.0 or 1.1
 - PMDA ADaM 1.0
 - For SDTM
 - FDA and PMDA 3.1.2, 3.1.3 or 3.2

XPTPATH=

- ✓ Ensure there are xpt files in this folder
- ✓ xpt folder path
- ✓ Default = &root.\Data\Transport\Standard

User Manual - Functions



Within Each Function:

REQUIRED AND OPTIONAL
MACRO PARAMETERS

STANDARD OUTPUT NAME

EXAMPLES CALLS

User Manual - SAS LOG MESSAGES

- List of Possible Log Messages and explanation for each

SAS LOG MESSAGES

ERROR: Please run Interactively(Enterprise Guide);
Do not Batch Submit run in EG.

ERROR: Standard Value [&Standard] is not a valid.
Valid values are: sdtm or adam.
Will only allow if SDTM or ADaM is specified not case sensitive.

ERROR: Engine Value [&Engine] is not a valid. Valid values are: FDA or PMDA.
Will only allow FDA or PMDA not case sensitive.

ERROR: StandardVersion Value [&StandardVersion] is not a valid.
Valid values are: 3.2, 3.1.3 or 3.1.2.;
Will only allow 3.2, 3.1.3 or 3.1.2.

ERROR: StandardVersion Value [&StandardVersion] is not a valid. Valid values are: 1.1. or 1.0.;
Will only allow 1.0 or 1.1.

ERROR: Invalid MedDRA Version Specified: &MedDRAv;
%Put Valid MedDRA Versions:
List of Valid Versions will be given select appropriate version from list.

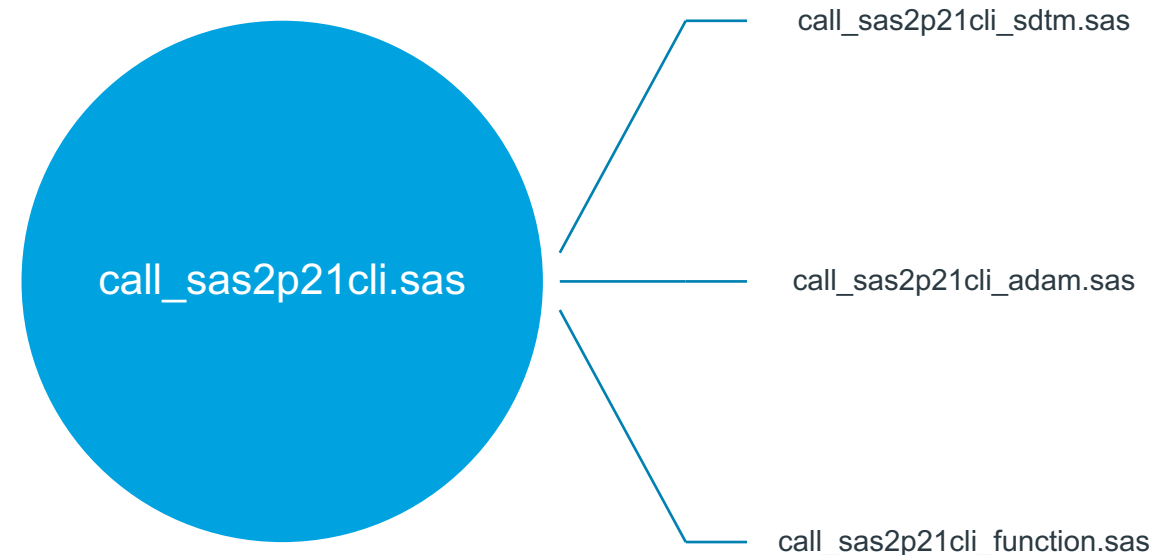
ERROR: Invalid SDTM CT Specified: &SDTMCT;
%Put Valid SDTM CT Versions:
List of Valid Versions will be given select appropriate version date from list.

ERROR: Invalid ADaM CT Specified: &ADaMCT;
%Put Valid ADaM CT Versions:
List of Valid Versions will be given select appropriate version date from list.

SAS Macro

SAS Macro

- Can have a single or multiple calls within the macro program.
- The configuration will be saved in the sas program thus the team are certain that the same selections are being made between different runs.
- SAS program (call_sas2p21cli.sas) can be renamed to reflect what functionary it will be executing.
 - If all SDTM and ADaM Validation done within the same file will keep the original name: call_sas2p21cli.sas
 - If SDTM and ADaM functionality is split between two different call programs, then
 - › call_sas2p21cli_sdtm.sas
 - › call_sas2p21cli_adam.sas
 - If team wants to only have one call in a macro:
 - › call_sas2p21cli_adam_data.sas
 - › call_sas2p21cli_adam_define.sas
 - › call_sas2p21cli_adam_full.sas



SAS Macro - Default

```
*****  
*Do NOT Comment Out or Change;  
%include "\\quintiles.net\Enterprise\Apps\sasdata\StatOpB\CSV\P21v3_Repository\sas_macro\sas2p21cli.sas";  
*****
```

```
%SAS2P21CLI  
(  
  Function=,  
  Engine=FDA,  
  Standard=adam,  
  StandardVersion=1.1,  
  xptPath=&root.\Data\Transport\adam,  
  SDTMct=2019-03-29,  
  ADaMct=2019-03-29,  
  MedDRAv=22.1,  
  NDF_RT=,  
  SNOMED=,  
  UNII=,  
  InputFilePath=&root.\Production\Define\adam\OUTPUT,  
  InputFileName=define,  
  SavePath=&root.\Documentation\CDISC_Compliance  
);
```

Macro Call with
Default Macro
Parameter values

Do Not Comment
out or Change

SAS Macro - Examples

```
*****
*Function 1 - Validate Data;
*****
*Validate Data FDA SDTM 3.2;
*****
%SAS2P21CLI
(
    Function=1,
    Engine=FDA,
    Standard=sdTM,
    StandardVersion=3.2,
    xptPath=&Root.\data\transport\SDTM\Subset,
    SDTMct=2018-12-21,
    MedDRAv=18.1,
    ndf_rt=2018-02-05,
    UNII=2018-10-25,
    SNOMED=2016-01-31,
    SavePath=&root.\Documentation\CDISC_Compliance
);
*****
```

Validate SDTM 3.2
FDA

```
*****
*Validate Data PMDA ADaM 1.0;
*****
%SAS2P21CLI
(
    Function=1,
    Engine=PMDA,
    Standard=adam,
    StandardVersion=1.0,
    xptPath=&Root.\data\transport\adam,
    SDTMct=2018-12-21,
    ADaMct=2019-03-29,
    MedDRAv=18.1,
    SavePath=&root.\Documentation\CDISC_Compliance
);
*****
```

Validate ADaM 1.0
PMDA

```
*****
*Function 2 - Validate Data and Define;
*****
*FDA ADaM 1.1 DefineXML;
*****
%SAS2P21CLI
(
    Function=2,
    Engine=FDA,
    Standard=adam,
    StandardVersion=1.1,
    xptPath=&Root.\data\transport\ADaM,
    SDTMct=2018-12-21,
    ADaMct=2018-12-21,
    MedDRAv=18.1,
    InputFilePath=&root.\Production\Define\ADaM\OUTPUT,
    InputFileName=define,
    SavePath=&root.\Documentation\CDISC_Compliance
);
```

Validate ADaM 1.1 +
Define for FDA

IQ Conform

IQ Conform

- This tool was developed to help standardise and streamline the completion of the CDISC conformance issue summaries as created by the Pinnacle 21 (P21) tool.
- Used to consolidate a new P21 report with a previously created P21 report to carry forward any Previously added Responses.
- Guides users how to respond to issues and shows Recommended text for more commonly found validation warnings \ errors.

IQ Conform

- Issue Summary Screen
- Output Docx Version of Report ready to be added to xxRG document.

1

Return to Detailed View

Issue (full text):

AE: Missing End Time-Point value

Comment (full text):

Reviewer Note: Known mapping issue, to be corrected in the specifications.

Domain:

Issue Details:

Response Details:

AE

AE start date is after the latest Disposition date

Reviewer Note: Known data issue, to be added to data-issues log.

AE

Missing End Time-Point value

Reviewer Note: Known mapping issue, to be corrected in the specifications.

AE

Value for AENRFR is populated, when RFENDTC is NULL

AE

Model permissible variable added into standard domain

CM

CMDSFRQ value not found in 'Frequency' extensible codelist

CM

CMDSOSU value not found in 'Unit' extensible codelist

CM

CMLOC value not found in 'Anatomical Location' extensible codelist

CM

Value for CMSTRF is populated, when RFSTDTC is NULL

CM

Value for CMENRF is populated, when RFENDTC is NULL

CM

Model permissible variable added into standard domain

DA

USUBJID/VISIT/VISITNUM values do not match SV domain data

DA

Model permissible variable added into standard domain

DA

Duplicate records

DM

RACE value not found in 'Race' extensible codelist

DM

No baseline result in VS for subject

DM

No Exposure record found for subject

DM

RFSTDTC is not provided for a randomized subject

DM

RFENDTC is not provided for a randomized subject

DM

Missing value for DTHDTC, when DTHFL is populated

DM

ACTARMCD does not equal ARMCOD

DS

DSDECD value not found in 'Completion/Reason for Non-Completion' extensible codelist w/

DS

Neither DSSTDTC, DSPTC nor DSSTDY are populated

DV

Model permissible variable added into standard domain

DV

High risk of truncated value for DVTERM variable

EG

EGTEST value not found in 'ECG Test Name' extensible codelist

EG

EGTESTCD value not found in 'ECG Test Code' extensible codelist

EG

Model permissible variable added into standard domain

EX

Exposure end date is after the latest Disposition date

FA

FALOC value not found in 'Anatomical Location' extensible codelist

FA

FATEST value not found in 'Findings About Test Name' extensible codelist

FA

FATESTCD value not found in 'Findings About Test Code' extensible codelist

FA

Model permissible variable added into standard domain

IE

Model permissible variable added into standard domain

IE

Permissible variable with missing value for all records

LB

Missing value for LBSTRESC, when LBORRES is provided

LB

LBMETHOD value not found in 'Method' extensible codelist

LB

LBTEST value not found in 'Laboratory Test Name' extensible codelist

LB

LBTESTCD value not found in 'Laboratory Test Code' extensible codelist

LB

Model permissible variable added into standard domain

LB

Missing value for LBREASND, when LBSTAT is 'NOT DONE'

MH

Model permissible variable added into standard domain

2

Summary Counts

Current:

0

402

147885

0

Previous:

N/A

N/A

N/A

N/A

Back

1

2a

Save Only

Selecting save will first run a spell check on your comment fields. After this is complete the P21 report will be saved as the name of the current P21 file with "_IQVIA Comments_<ISO Date>" as a suffix.

2b

Save & Export (.xlsx/.docx)

'Save & Export' will first perform the same actions as the 'Save Only' and then will generate two items that can be used for the reviewer's guide conformance summary sections.

2c

Close without Saving (cannot be undone)

This will close the Excel spreadsheet without saving any progress. This process cannot be undone, and you will be prompted again before you're able to complete this process.

Q&A

- SAS2P21CLI
- Gustav.Bernard@iqvia.com