



Building Your Own CDISC Application with SAS® Clinical Standards Toolkit

Jing Gao, SAS R&D, Beijing

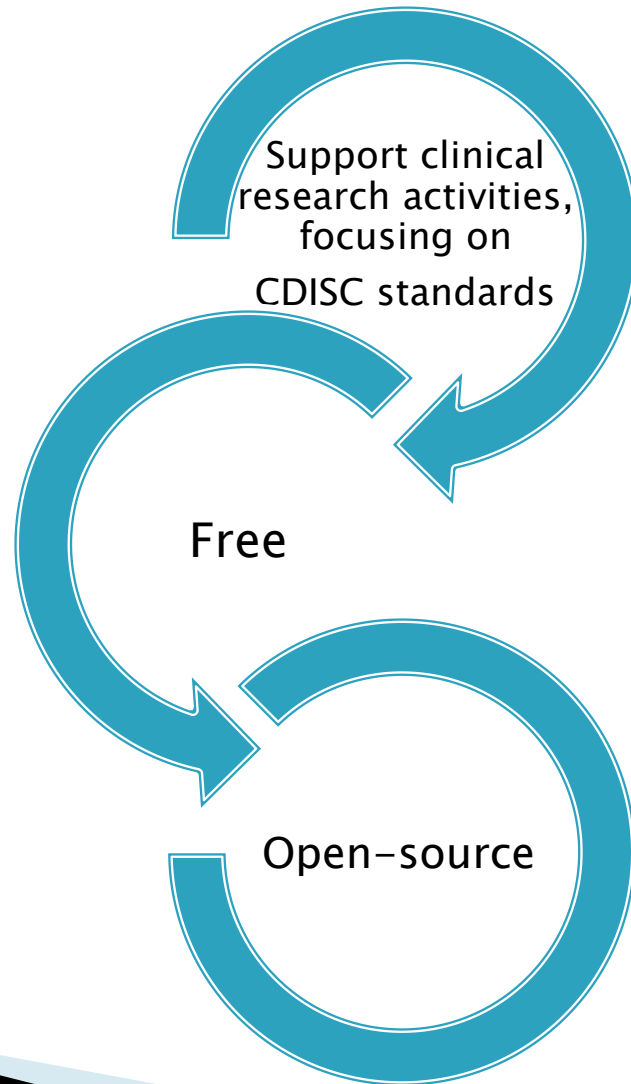
Agenda

- ▶ Introduction to SAS Clinical Standards Toolkit
- ▶ A Sample of CDISC Application based on SAS Clinical Standards Toolkit
- ▶ How to implement the sample application
- ▶ Extensibility of SAS Clinical Standards Toolkit



Introduction to SAS Clinical Standards Toolkit

What is SAS Clinical Standards Toolkit?



- SDTM 3.1.2, 3.1.3, and 3.2
- ADaM 2.1
- CRT-DDS 1.0 & Define-XML 2.0
- Dataset-XML 1.0
- ODM 1.3.0 and 1.3.1
- CDASH 1.1
- SEND 3.0
- Controlled Terminology

SAS Clinical Standards Toolkit Functions

Major
functions of
the CST:

Checking compliance with SDTM, SEND, ADaM and Define-XML

Reading and creating **Define-XML** files for SDTM, SEND and ADaM

Converting Data between **Dataset-XML** and SAS dataset

Validating XML-Based Standards files (such as ODM, CT, CRT-DDS 1.0, and Define-XML 2.0)

SAS Clinical Standards Toolkit Structure

Sample Library – C:\cstSampleLibrary

Global Standards Library – C:\cstGlobalLibrary

Framework Macros – !SASROOT\cstframework\sasmacro



A Sample of CDISC Application based on SAS Clinical Standards Toolkit

Sample Application GUI

Powered by SAS® Clinical Standards Toolkits

File Settings Help

Powered by SAS® Clinical Standards Toolkits

 Compliance Checks  Dataset-XML Creation  Configure SAS Paths  Help

Compliance Checks

Create CDISC Dataset-XML from SAS Data Set

Source Data:

A dataset-xml will be created for each of data sets in the selected folder

Define-XML file:

Dataset-XML output folder:

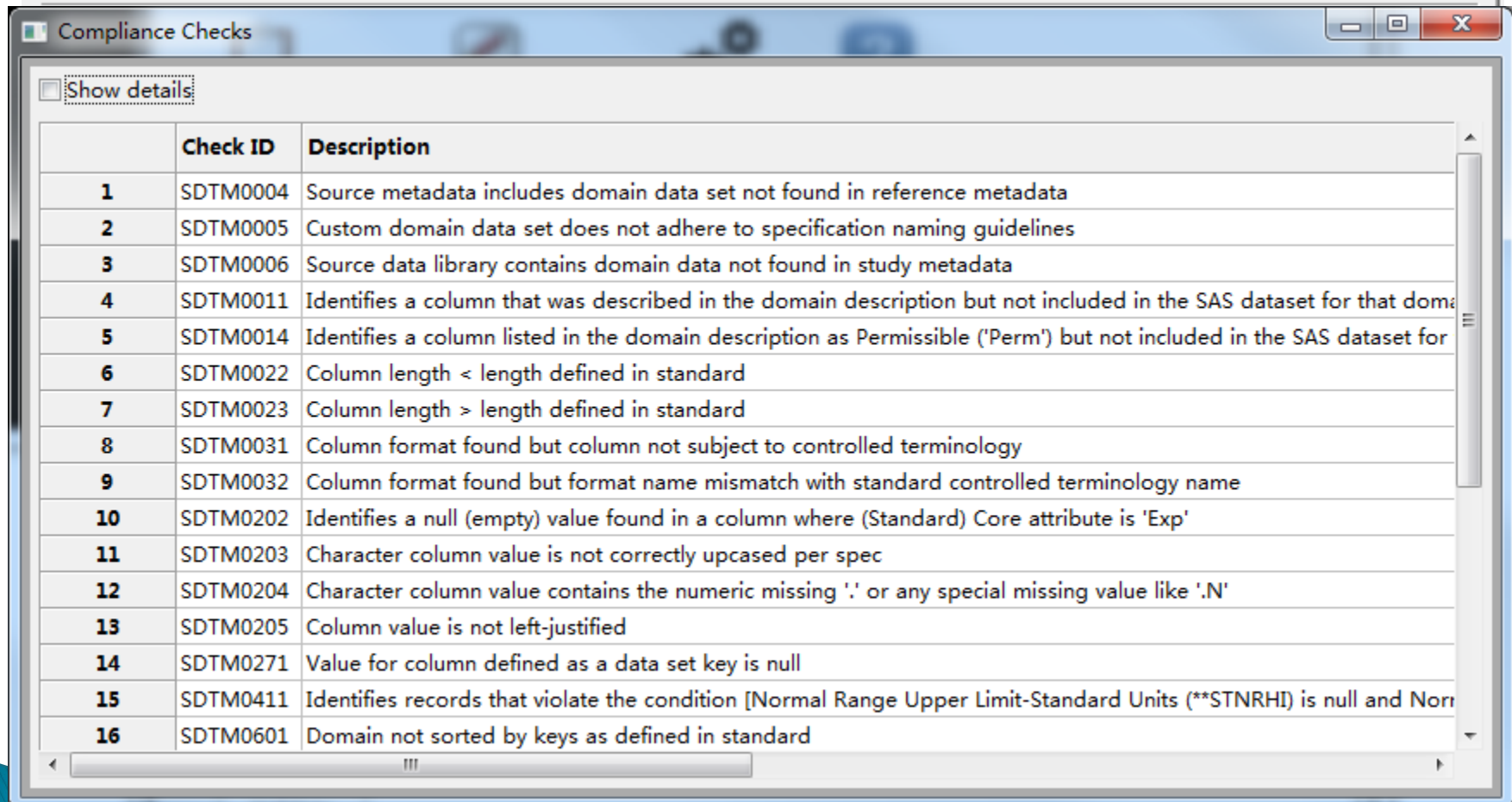
File encoding:

☐ Check text variable data length against metadata

CDISC-Dataset-XML Creation

SDTM compliance checks

Compliance checks against SDTM, SEND, ADaM, and Define.xml



☐ Show details

	Check ID	Description
1	SDTM0004	Source metadata includes domain data set not found in reference metadata
2	SDTM0005	Custom domain data set does not adhere to specification naming guidelines
3	SDTM0006	Source data library contains domain data not found in study metadata
4	SDTM0011	Identifies a column that was described in the domain description but not included in the SAS dataset for that domain
5	SDTM0014	Identifies a column listed in the domain description as Permissible ('Perm') but not included in the SAS dataset for that domain
6	SDTM0022	Column length < length defined in standard
7	SDTM0023	Column length > length defined in standard
8	SDTM0031	Column format found but column not subject to controlled terminology
9	SDTM0032	Column format found but format name mismatch with standard controlled terminology name
10	SDTM0202	Identifies a null (empty) value found in a column where (Standard) Core attribute is 'Exp'
11	SDTM0203	Character column value is not correctly upcased per spec
12	SDTM0204	Character column value contains the numeric missing '.' or any special missing value like '.N'
13	SDTM0205	Column value is not left-justified
14	SDTM0271	Value for column defined as a data set key is null
15	SDTM0411	Identifies records that violate the condition [Normal Range Upper Limit-Standard Units (**STNRHI) is null and Normal Range Upper Limit-Standard Units (**STNRHI) is not null]
16	SDTM0601	Domain not sorted by keys as defined in standard

Report by Check

SAS Clinical Standards Toolkit 1.7 CDISC-SDTM 3.1.2 VALIDATION

Process Results, CheckID: SDTM0202

Description: Identifies a null (empty) value found in a column where (Standard) Core attribute is 'Exp'

Check scope: (Tables) _ALL_, (Columns) _ALL_

Source: SAS (SAS0015)

Validation check macro: cstcheck_violatesstd, using source metadata

Check Invocation	Seq #	Source Data	Result Identifier	Message	Severity	Problem Detected?	Actual Value	Keys
1	23	lib2016.DM	SDTM0202	Null value in column	Note	Yes	RACE=	STUDYID=SASCSTDEMODATA, USUBJID=S001P010
1	24	lib2016.DM	SDTM0202	Null value in column	Note	Yes	RACE=	STUDYID=SASCSTDEMODATA, USUBJID=S001P014

SAS Clinical Standards Toolkit 1.7 CDISC-SDTM 3.1.2 VALIDATION

Process Results, CheckID: SDTM0642

Description: Inconsistency between Description of Arm (ARM) and Arm Code (ARMCD) values across all records

Check scope: (Tables) DM, (Columns) [ARM][ARMCD]

Source: SAS (SAS0028)

Validation check macro: cstcheck_notunique, using source metadata

Check Invocation	Seq #	Source Data	Result Identifier	Message	Severity	Problem Detected?	Actual Value	Keys
1	1	lib2016.DM	SDTM0642	ARM and ARMCD must map 1:1	Warning	Yes	ARM=SCREEN FAILURE,ARMCD=SCRNFAIL	STUDYID=SASCSTDEMODATA, USUBJID=S002P034
1	2	lib2016.DM	SDTM0642	ARM and ARMCD must map 1:1	Warning	Yes	ARM=SCREEN FAILURE,ARMCD=XYZ	STUDYID=SASCSTDEMODATA, USUBJID=S002P029

Report by Domain

SAS Clinical Standards Toolkit 1.7 CDISC-SDTM 3.1.2 VALIDATION

Process Results, Table: AE

Check ID	Check Invocation	Seq #	Source Data	Result Identifier	Message	Severity	Problem Detected?	Actual Value	Keys
SDTM0202	1	8	lib2016.AE	SDTM0202	Null value in column	Note	Yes	AEREL=	STUDYID= SASCSTDEMODATA, USUBJID=S002P031, AEDECOD=Headache, AESTDTC=2008-05-15T01:00
SDTM0202	1	9	lib2016.AE	SDTM0202	Null value in column	Note	Yes	AEREL=	STUDYID= SASCSTDEMODATA, USUBJID=S003P010, AEDECOD=Rash pruritic, AESTDTC=2008-05-30T17:30

SAS Clinical Standards Toolkit 1.7 CDISC-SDTM 3.1.2 VALIDATION

Process Results, Table: DM

Check ID	Check Invocation	Seq #	Source Data	Result Identifier	Message	Severity	Problem Detected?	Actual Value	Keys
SDTM0202	1	11	lib2016.DM	SDTM0202	Null value in column	Note	Yes	AGE=.	STUDYID= SASCSTDEMODATA, USUBJID=S001P001
SDTM0202	1	12	lib2016.DM	SDTM0202	Null value in column	Note	Yes	AGEU=	STUDYID= SASCSTDEMODATA, USUBJID=S001P001

Dataset-XML Creation

Create CDISC Dataset-XML from SAS Data Set

Source Data:

Browse

A dataset-xml will be created for each of data sets in the selected folder

Define-XML file:

Browse

Dataset-XML output folder:

Browse

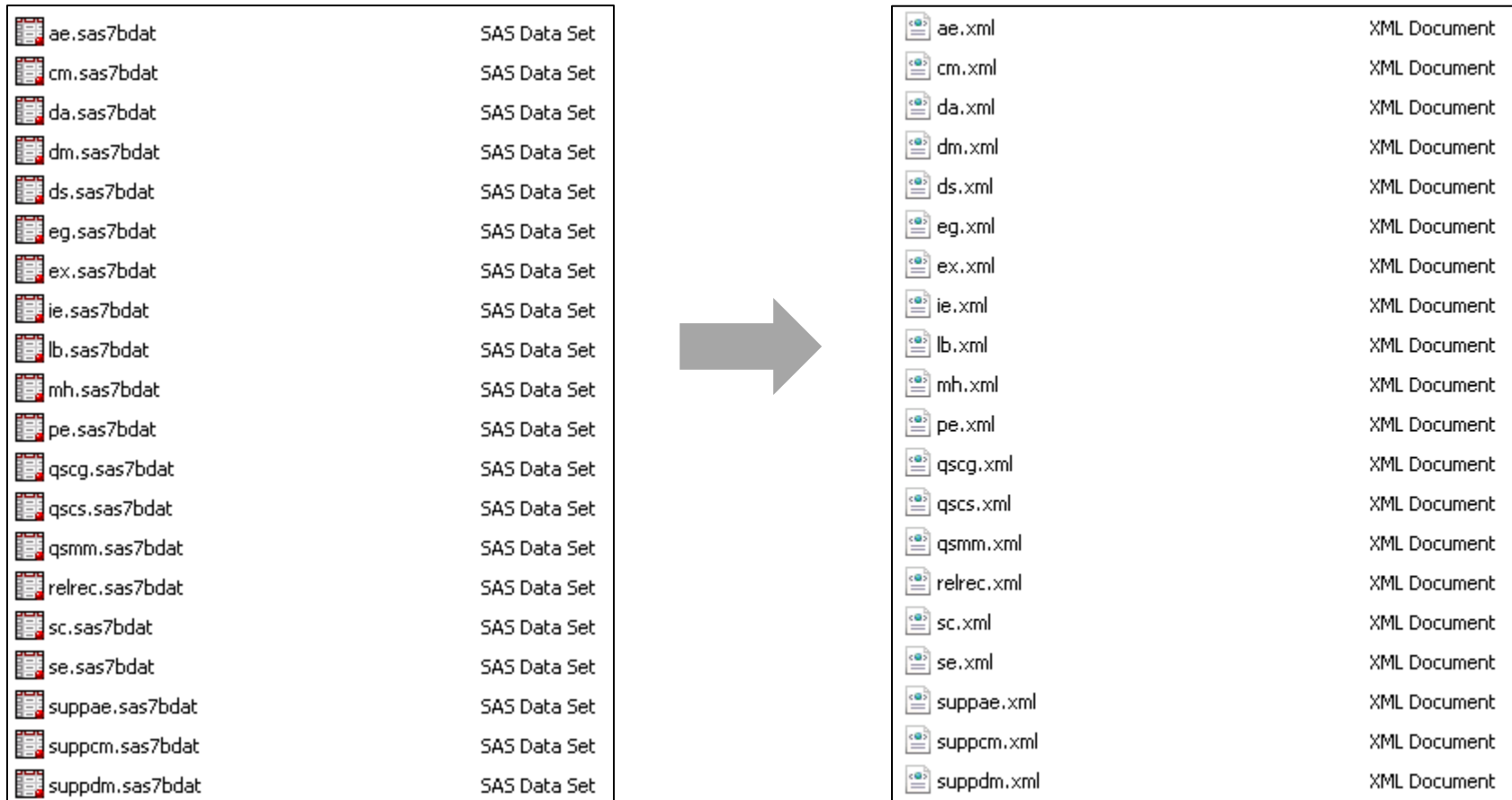
File encoding:

UTF-8

☐ Check text variable data length against metadata

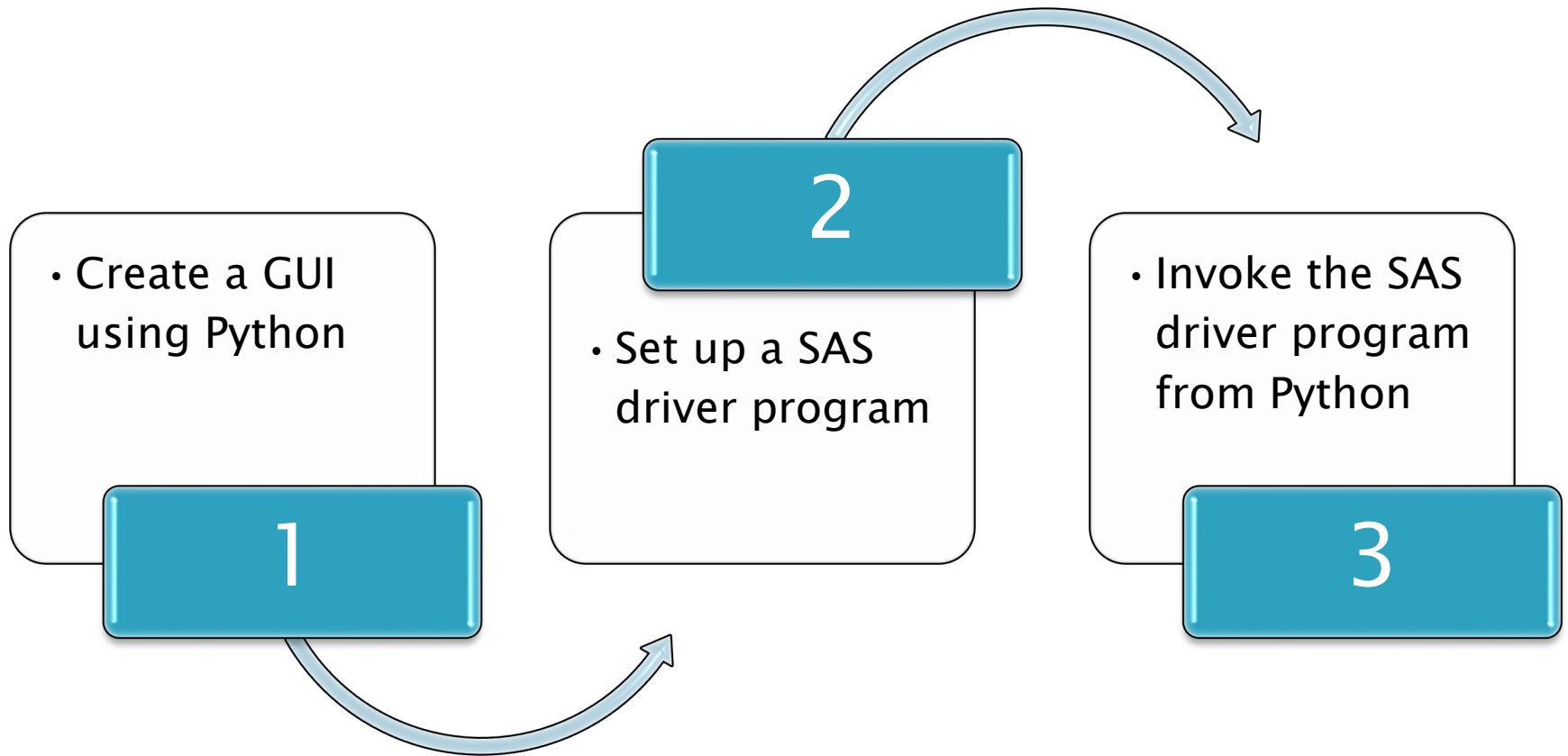
Create

Dataset-XML Creation

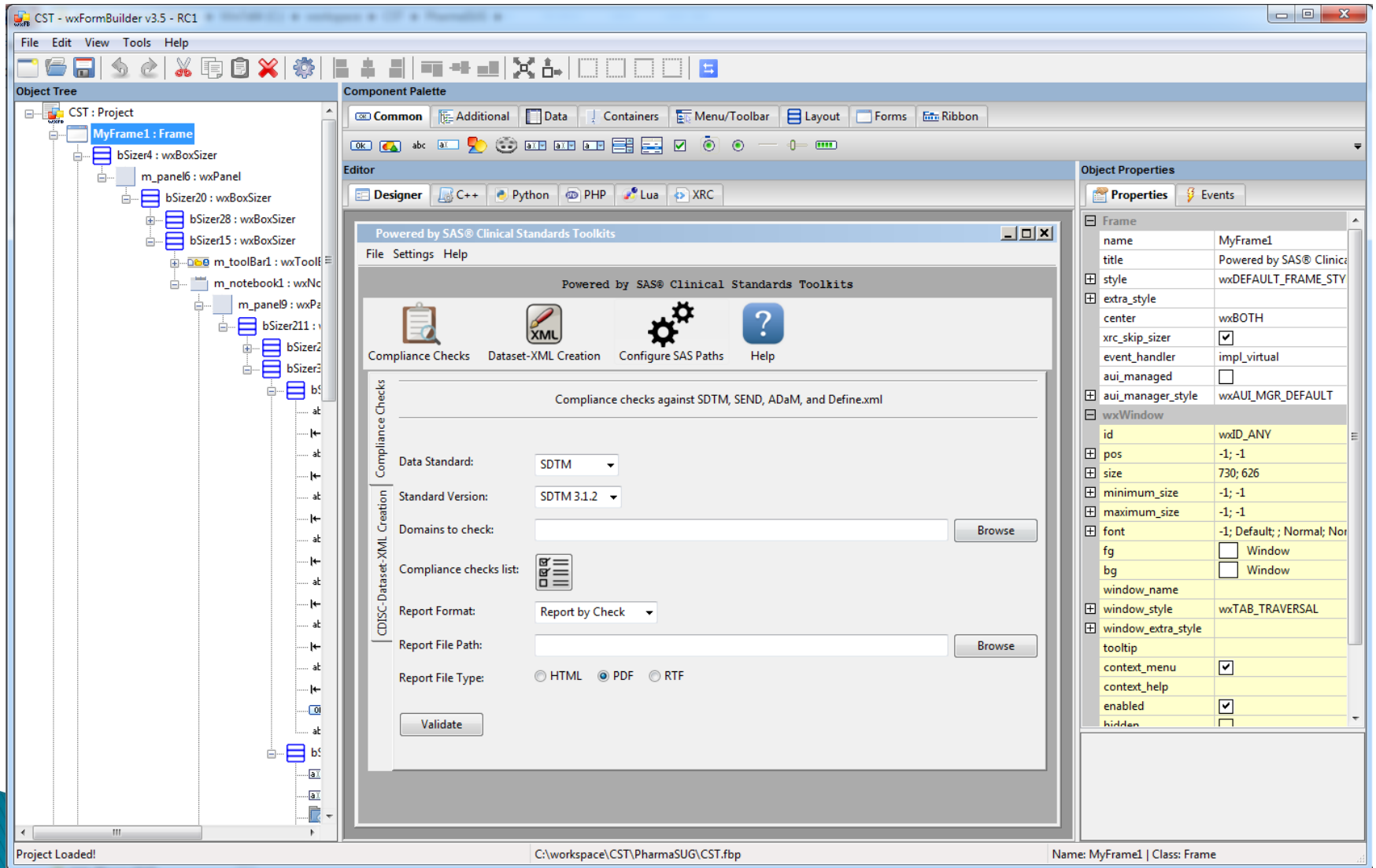


How to implement the sample application

Steps to implement the sample application



1 – Create a GUI using Python



2 – Set up a SAS driver program (validate_data.sas)

Step 1: Define macro variables required by the validation process

```
%let _cstStandard=%sysget(cstStandard);  
%let _cstStandardVersion=%sysget(cstStandVersion);
```

Step 2: Build and populate the SASReferences data set

```
proc sql;  
  insert into work.sasreferences  
Quit;
```

Step 3: Call the cstutil_processsetup macro

```
%cstutil_processsetup();
```

Step 4: Call the validation macro

```
%sdtm_validate;
```

Step 5: Call the report macro

```
%cstutil_createreport();
```

2 – Set up a SAS driver program (validate_data.sas)

Sample Library

```
%let _cstStandard=%sysget(cstStandard);  
%let _cstStandardVersion=%sysget(cstStandVersion);
```

```
proc sql;  
  insert into work.sasreferences  
Quit;
```

```
%cstutil_processsetup();
```

 Framework Macros

```
%sdtm_validate;
```

 Global Standards Library

```
%cstutil_createreport();
```

 Framework Macros

3 – Invoke SAS driver program from Python

Step 1: import the subprocess module for spawning new process

```
import subprocess
```

Step 2: invoke the SAS driver program and pass values to it

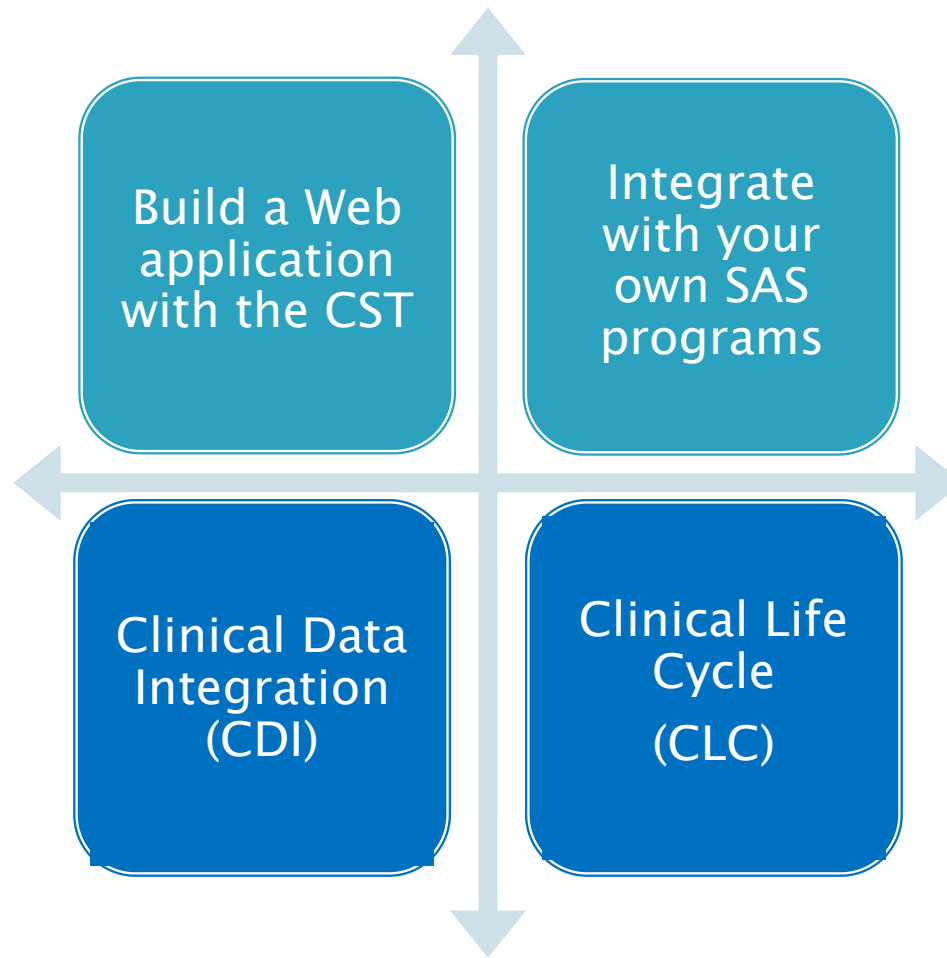
```
subprocess.Popen(  
args='  
"C:\\Program Files\\SASHome\\SASFoundation\\9.4\\sas.exe"  
-sysin "C:\\sample\\validate_data.sas"  
-set    cstStandard      "CDISC-SDTM"  
-set    cstStandVersion  "3.1.2"  
,  
)
```

Validate_data.sas

```
%let _cstStandard=%sysget(cstStandard);  
%let _cstStandardVersion=%sysget(cstStandVersion);
```

Extensibility of SAS Clinical Standards Toolkit

Extensibility of SAS Clinical Standards Toolkit



CONCLUSION

- ▶ The CST is a powerful and flexible toolkit supporting multiple CDISC standards.
- ▶ There is no additional charge to licensed SAS users.
- ▶ The CST macros are provided as open source and are accessible to the users, which allows users to customize the CST to meet their needs.
- ▶ As introduced in this presentation, you can build a CDISC application based on the CST for your own needs. Maybe there are a lot of work to do at the beginning. After having done this, it is more intuitive and easy to use the CST.

Name: Jing Gao

Organization: SAS R&D (Beijing) Co., Ltd.

Address: Motorola Plaza, No. 1 Wang Jing East Road

City, State ZIP: 100102

Work Phone: (8610) 83193355-3462

Fax: (8610) 6310-9130

E-mail: Jing.gao@sas.com

Web: www.sas.com

