



Generate CDISC Define.xml & Define.pdf from Metadata Repository

Rohit Banga

Topics

- Introduction
- Generate Define.xml
- Conclusion

Introduction

INTRODUCTION

Define.xml

- Describes the content and structure of the included data in a machine readable format
- Based on the CDISC Operational Data Model (ODM) making it vendor neutral & platform independent
- Metadata needed for **both** clinical data and analysis data
 - Uses xsl stylesheets and hyperlinks
- Needed for all electronic submissions
- Referenced by FDA in the eCTD (electronic submissions) guidance

(Source : ECTD Guidance - http://www.cdisc.org/models/def/v1.0/CRT_DDSpecification1_0_0.pdf)

Define.xml - Metadata

Table Metadata

Dataset Name
Dataset Label
Class
Structure
Purpose
Keys

Variable Metadata

Variable Name
Variable Label
Type
Controlled Terminology
Origin
Role
Comment

Value Level Metadata

Source Variable
Value
Label
Type
Controlled Terminology
Origin
Role
Comment

Controlled Terminology

Code Value
Code Text

Computational Algorithms

Reference Name
Computation Method

Annotated Case Report Form

Annotated Case Report FormAnnotated Case Report FormAnnotated Case Report Form

Annotated Case Report Form

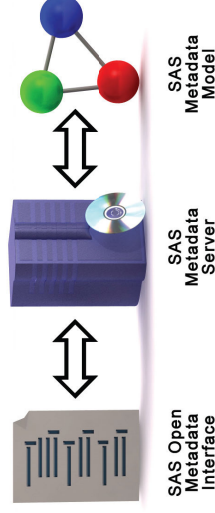
Annotated Case Report Form

SAS Open Metadata Architecture

The SAS Open Metadata Architecture is a general-purpose metadata management facility that provides common metadata services to SAS applications.

The Metadata Architecture consists of :

- ***SAS Metadata Server***
- ***SAS Open Metadata Interface***
- ***SAS Metadata Model***
- ***XML transport format and XML representation of metadata***

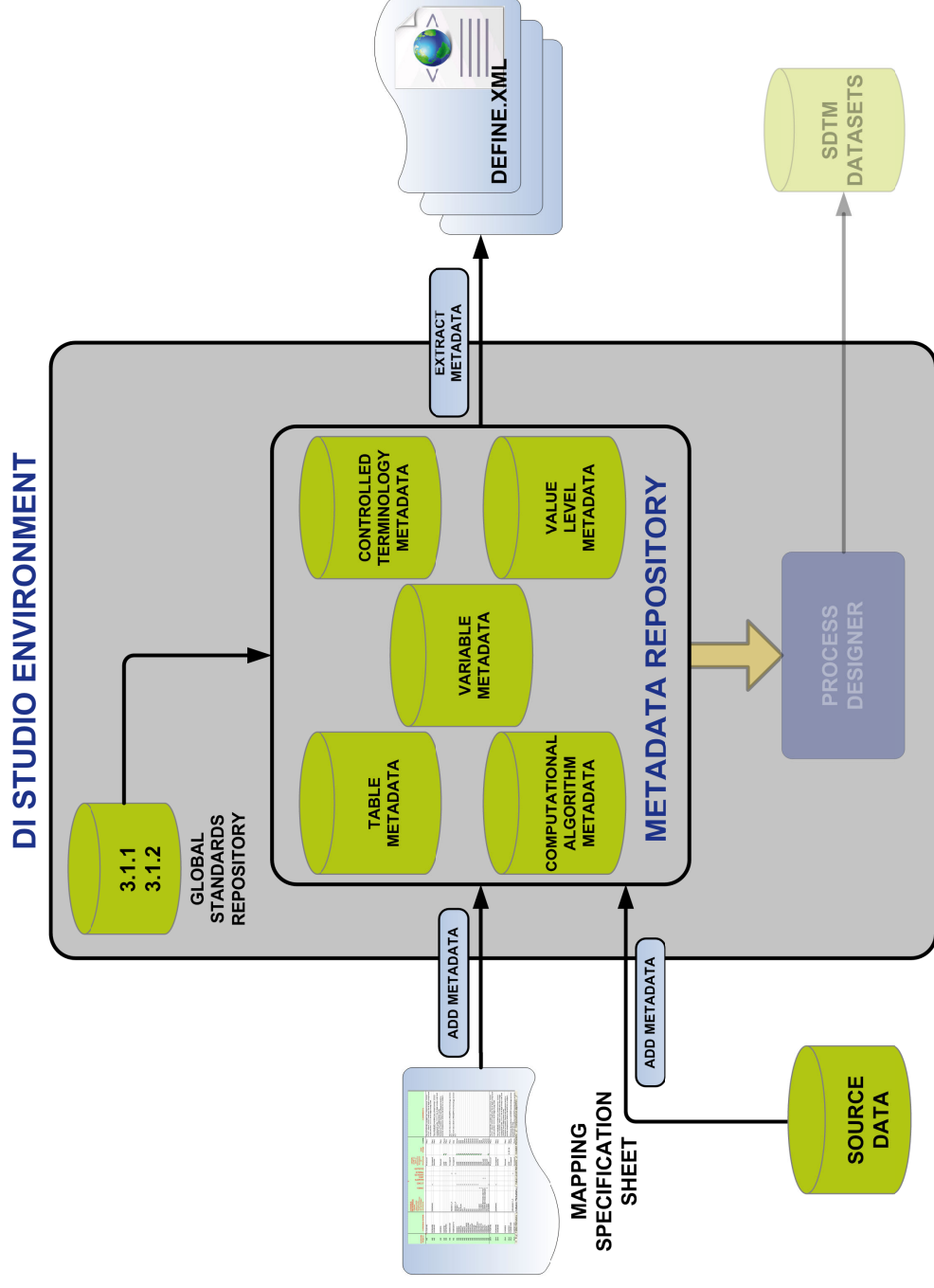


*SAS® 9.1.3 Open Metadata Interface Reference Second Edition

Generating Define.xml

GENERATING DEFINE.XML

Metadata Flow



Metadata Specification Table

Dataset	Description	Class	Structure	Purpose	Keys
AE	Adverse Events	Findings	One record per adverse event per subject	Tabulation	STUDYID, USUBJID, AETERM, AESTDTC
BA	Baseline Event Assessment	Events	One record bleeding event assessment per subject	Tabulation	STUDYID, USUBJID, BATESTCD, VISITNUM, BASPID
BE	Bleeding Event	Events	One record per bleeding event per subject	Tabulation	STUDYID, USUBJID, BESPID
BP	Diagnostic Assessments	Findings	One record per test code	Tabulation	STUDYID, USUBJID, BPTESTCD, VISITNUM, BPSPID, BPCAT, BPTCAT
BT	Blood Transfusion	Interventions	One record per blood transfusion per subject	Tabulation	STUDYID, USUBJID, BITRT, BTSTDTC, BTSPID
CA	Central Assessments	Findings	One record per assessment per visit per subject	Tabulation	STUDYID, USUBJID, CATESTCD, VISITNUM, CACAT, CASCAT, CALOC, CASPID
CM	Concomitant Medications	Interventions	One record per medication intervention episode per subject	Tabulation	STUDYID, USUBJID, CMTRT, CMSTDTC, CMCAT, CMINDC
CO	Comments	Special Purpose	One record per comment per subject	Tabulation	STUDYID, USUBJID, ROOMAIN, IDVARVAL
CP	Compliance	Findings	One record per test code	Tabulation	STUDYID, USUBJID, CPTSTCD, VISITNUM
DM	Demographics	Special Purpose	One record per subject	Tabulation	STUDYID, USUBJID
DS	Disposition	Events	One record per disposition status or protocol milestone per subject	Tabulation	STUDYID, USUBJID, DSSTDTC, EPDCH
EX	Exposure	Interventions	One record per constant dosing interval per subject	Tabulation	STUDYID, USUBJID, EXSTDTC, EXTRT
HO	Hospitalizations	Events	One record per hospitalization event per subject	Tabulation	STUDYID, USUBJID, HOTERM
IE	Inclusion/Exclusion Exceptions	Findings	One record per inclusion/exclusion criteria exception per subject	Tabulation	STUDYID, USUBJID, IESTDTC
IM	Intake of Investigational Medication	Interventions	One record per intake of investigational medication per subject	Tabulation	STUDYID, USUBJID, IMTRT, IMSPID
LB	Laboratory Test Results	Findings	One record per lab test per time point per visit per subject	Tabulation	STUDYID, USUBJID, LBTESTCD, VISITNUM, LBTRC
LBH	Medical History	Events			
MH	Related Records	Special Purpose R			
PELREC	Pharmacokinetic/Pharmacodynamic Sampling	Findings			
PS	Subject Elements	Trial Design			
SE	Subject Elements	Events			
SR	Surgery	Interventions			
SU	Substance Use	Interventions			
SUPPAE	Supplemental Qualifiers	Special Purpose R			
SUPPBA	Supplemental Qualifiers	Special Purpose R			
SUPPBE	Supplemental Qualifiers	Special Purpose R			
SUPPBP	Supplemental Qualifiers	Special Purpose R			
SUPPBT	Supplemental Qualifiers	Special Purpose R			
SUPPCA	Supplemental Qualifiers	Special Purpose R			
SUPPDM	Supplemental Qualifiers	Special Purpose R			
SUPPDS	Supplemental Qualifiers	Special Purpose R			
SUPPEX	Supplemental Qualifiers	Special Purpose R			
SUPPIM	Supplemental Qualifiers	Special Purpose R			
SUPPSR	Supplemental Qualifiers	Special Purpose R			
SUPPMH	Supplemental Qualifiers	Special Purpose R			
SV	Subject Visits	Trial Design			
TA	Trial Arms	Trial Design			
TE	Trial Elements	Trial Design			
TI	Trial Inclusion/Exclusion Criteria	Trial Design			
TS	Trial Summary	Trial Design			
TV	Trial Visits	Trial Design			
VS	Vital Signs	Findings			
WA	Wound Assessment	Findings			

Add Metadata

```
%Macro Addtablemetadata( Extension = , Value= , TableId = );  
data _null_;  
file addmeta;
```

```
PUT '<AddMetadata>';  
PUT '<Metadata>';  
PUT "<Extension Id = " &ExtensionId ">";  
PUT "      Name = " &Name "&ExtensionId";  
PUT "      Value = " &Value ">";  
PUT '<owningobject>';  
PUT " <PhysicalTable objref= " &TableId ">";  
PUT '</owningobject>';  
PUT '</Extension>';  
PUT '</Metadata>';  
PUT '<Reposid>$METAREPOSITORY</Reposid>';  
PUT '      <NS>SAS</NS>';  
PUT '<!-- OMI_TRUSTED_CLIENT FLAG -->';  
PUT '<Flags>268435456</Flags>';  
PUT '<Options>';  
PUT '</AddMetadata>';  
run;
```

```
Proc Metadata Repos = "&REP_ID." in = addmeta ;  
run;
```

```
%Mend Addtablemetadata;
```

SAS Code to Add Metadata

- Uses <AddMetadata> method call
- Uses PROC METADATA

Add Metadata – Results

Business & Decision Management Console View

The screenshot displays the 'Adverse Events - AE Properties' window in the Business & Decision Management Console. The 'Extended Attributes' tab is selected, showing a list of attributes for the 'STUDYID' variable. A red box highlights the 'Extended Attributes for STUDYID' dialog box, which is open and shows the 'Notes' tab. The 'Notes' tab contains a text area with the following text: 'The STUDYID variable has a fixed format: 'XXXX-YYYY', where 'XXXX' indicates the 4-digit compound code and the 'YYYY' the 4-digit study code'. The 'References' tab is also visible, showing a list of references. The 'Adverse Events - AE Properties' window has tabs for General, Columns, Indexes, Keys, Physical Storage, Parameters, Notes, Extended Attributes, and Advanced. The 'Extended Attributes' tab is currently active, showing a table with columns for #, Field Name, Req, Value, and Description. The table lists various attributes for the 'STUDYID' variable, including 'CLASS', 'STRUCTURE', 'PURPOSE', 'KEYS', 'DESCRIPTION', 'DOMAIN', and 'EXPORT'. The 'Notes' dialog box is open, showing the 'Notes' tab with a text area containing the text: 'The STUDYID variable has a fixed format: 'XXXX-YYYY', where 'XXXX' indicates the 4-digit compound code and the 'YYYY' the 4-digit study code'. The 'References' tab is also visible, showing a list of references. The 'Adverse Events - AE Properties' window has tabs for General, Columns, Indexes, Keys, Physical Storage, Parameters, Notes, Extended Attributes, and Advanced. The 'Extended Attributes' tab is currently active, showing a table with columns for #, Field Name, Req, Value, and Description. The table lists various attributes for the 'STUDYID' variable, including 'CLASS', 'STRUCTURE', 'PURPOSE', 'KEYS', 'DESCRIPTION', 'DOMAIN', and 'EXPORT'. The 'Notes' dialog box is open, showing the 'Notes' tab with a text area containing the text: 'The STUDYID variable has a fixed format: 'XXXX-YYYY', where 'XXXX' indicates the 4-digit compound code and the 'YYYY' the 4-digit study code'. The 'References' tab is also visible, showing a list of references.

#	Field Name	Req	Value	Description
1	CORE			
2	ORIGIN			
3	TERM			
4	ROLE			

Notes for STUDYID

Note text:

The STUDYID variable has a fixed format: 'XXXX-YYYY', where 'XXXX' indicates the 4-digit compound code and the 'YYYY' the 4-digit study code

References

Documents

Assigned:

Name

Path

New Delete Attach...

OK Cancel Help

Get Metadata

```
filename myxmlmap "&define.\tempdata\tablemetadata.map";
libname tempdata xml xmlmap=myxmlmap xmlfileref=results;
```

```
%Macro ExtractTableMetadata;
```

```
data _null_;
```

```
file Input;
```

```
put '<GetMetadataObjects>';
```

```
put '<Reposid> $METAREPOSITORY </Reposid>';
```

```
put '<Type>PhysicalTable</Type>';
```

```
put '<Objects>';
```

```
put '<NS> SAS </NS>';
```

```
put '<Flags> 388 </Flags>';
```

```
put '<Options>';
```

```
put '<XMLSelect search="PhysicalTable[@SASTableName = '@';
```

```
PUT " " "&NAME" ""]/TablePackage/SASLibrary[@Libref = " @;
```

```
PUT " " "&libref" " " " @;
```

```
PUT ']]' />';
```

```
put '<Templates>';
```

```
put '<PhysicalTable Id=" " Name=" " Desc=" " SASTableName=" " >';
```

```
put '<Columns>';
```

```
put '<Extensions>';
```

```
put '<Notes>';
```

```
put '</PhysicalTable>';
```

```
put '<Column ID=" " Name=" " Desc=" " SASColumnLength=" " SASColumnType=" "';
```

```
put ' ' SASColumnName=" " SASFormat=" " SASInformat=" " >';
```

```
put ' ' <Extensions/> <Notes/> </Column>';
```

```
put '<Extension ID=" " Name=" " Desc=" " Value=" " />';
```

```
put '<TextStore ID=" " Name=" " TextType=" " StoredText=" " />';
```

```
put ' ' </Templates>';
```

```
put ' ' </Options>';
```

```
put '</GetMetadataObjects>';run;
```

```
Proc Metadata Repos = "&Rep_ID." in=input out=results;
```

```
run;
```

```
%Mend ExtractTableMetadata;
```

SAS Code to Extract Metadata

- Uses
<GetMetadataObjects>
method call
- Uses <XML Select> element
to specify search string
- Uses PROC METADATA

Map XML to SAS Datasets

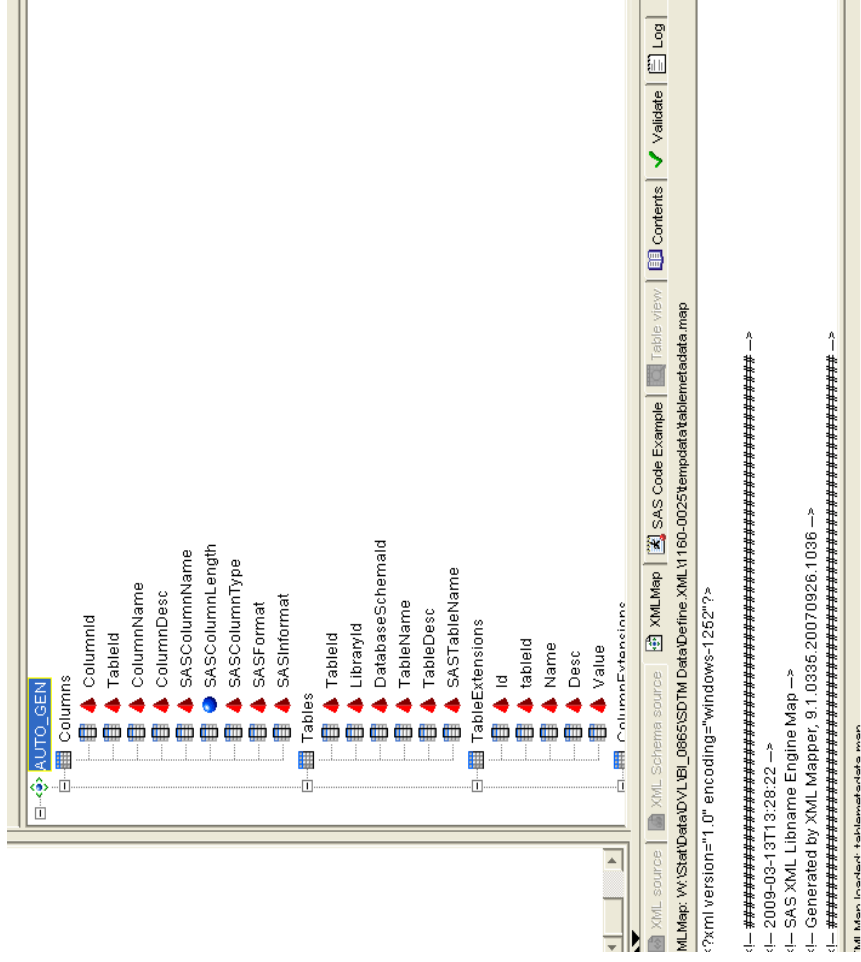
```
- <PhysicalTable Id="A50WT7QU.AA0000Z" Name="Adverse Events" AE" Desc="SASTableName="AE" />
- <Columns>
  - <Column Id="A50WT7QU.AB0002F5" Name="STUDYID" Desc="Study Identifier" SASColumnLength="20" SASColumnType="C" SASColumnName="STUDYID" />
  - <Extensions>
    <Extension Id="A50WT7QU.AC000535" Name="TERM" Desc="Controlled Term or Format" Value="" />
    <Extension Id="A50WT7QU.AC000536" Name="ORIGIN" Desc="Origin" Value="CRF" />
    <Extension Id="A50WT7QU.AC000537" Name="ROLE" Desc="Role" Value="Identifier" />
    <Extension Id="A50WT7QU.AC000538" Name="CORE" Desc="Core" Value="Req" />
    <Extension Id="A50WT7QU.AC000539" Name="References" Desc="References" Value="SDTM 2.2.4" />
  </Extensions>
</Column>
- <Column Id="A50WT7QU.AB0002F6" Name="DOMAIN" Desc="Domain Abbreviation" SASColumnLength="2" SASColumnType="C" SASColumnName="DOMAIN" />
- <Extensions>
  <Extension Id="A50WT7QU.AC00053A" Name="TERM" Desc="Controlled Term or Format" Value="" AE"" />
  <Extension Id="A50WT7QU.AC00053B" Name="ORIGIN" Desc="Origin" Value="Derived" />
  <Extension Id="A50WT7QU.AC00053C" Name="ROLE" Desc="Role" Value="Identifier" />
  <Extension Id="A50WT7QU.AC00053D" Name="CORE" Desc="Core" Value="Req" />
  <Extension Id="A50WT7QU.AC00053E" Name="References" Desc="References" Value="SDTM 2.2.4" />
</Extensions>
</Column>
- <Column Id="A50WT7QU.AB0002F7" Name="USUBJID" Desc="Unique Subject Identifier" SASColumnLength="20" SASColumnType="C" SASColumnName="USUBJID" />
- <Extensions>
  <Extension Id="A50WT7QU.AC00053F" Name="TERM" Desc="Controlled Term or Format" Value="" />
  <Extension Id="A50WT7QU.AC00053G" Name="ORIGIN" Desc="Origin" Value="Sponsor Defined" />
  <Extension Id="A50WT7QU.AC00053H" Name="ROLE" Desc="Role" Value="Identifier" />
  <Extension Id="A50WT7QU.AC00053I" Name="CORE" Desc="Core" Value="Req" />
  <Extension Id="A50WT7QU.AC00053J" Name="References" Desc="References" Value="SDTM 2.2.4" />
</Extensions>
</Column>
- <Column Id="A50WT7QU.AB0002F8" Name="AESEQ" Desc="Sequence Number" SASColumnLength="8" SASColumnType="N" SASColumnName="AESEQ" />
- <Extensions>
  <Extension Id="A50WT7QU.AC00053K" Name="TERM" Desc="Controlled Term or Format" Value="" />
  <Extension Id="A50WT7QU.AC00053L" Name="ORIGIN" Desc="Origin" Value="CRF or Derived" />
  <Extension Id="A50WT7QU.AC00053M" Name="ROLE" Desc="Role" Value="Identifier" />
  <Extension Id="A50WT7QU.AC00053N" Name="CORE" Desc="Core" Value="Req" />
  <Extension Id="A50WT7QU.AC00053O" Name="References" Desc="References" Value="SDTM 2.2.4" />
</Extensions>
</Column>
- <Column Id="A50WT7QU.AB0002F9" Name="AEGRPID" Desc="Group ID" SASColumnLength="20" SASColumnType="C" SASColumnName="AEGRPID" />
- <Extensions>
  <Extension Id="A50WT7QU.AC00053P" Name="TERM" Desc="Controlled Term or Format" Value="" />
  <Extension Id="A50WT7QU.AC00053Q" Name="ORIGIN" Desc="Origin" Value="Sponsor Defined" />
  <Extension Id="A50WT7QU.AC00053R" Name="ROLE" Desc="Role" Value="Identifier" />
  <Extension Id="A50WT7QU.AC00053S" Name="CORE" Desc="Core" Value="Perm" />
  <Extension Id="A50WT7QU.AC00053T" Name="References" Desc="References" Value="SDTM 2.2.4" />
</Extensions>
</Column>
```

Result.xml

Result of
<GetMetadataObjects>
method call in native SAS
XML Format

Map XML to SAS Datasets

- SAS XML Mapper used to make Map file
- Map file converts Result.xml into SAS Datasets
- Resulting SAS Datasets manipulated to make Define.xml



SAS XML Mapper

Proc Template

```
proc template;
define tagset def_tmpl.ComputationMethod;

parent = tagsets.sasxmog;
notes "SAS XML Engine tagsets for SDTM ComputationMethod";

mvar markup
    OID
    ItemOID;
define event initialize;
unset $markup;
unset $OID;
.....
define event row;
    start;
    break;
finish:
do /if cmp($markup, "First");
    trigger markupFirst;
.....
define event SASColumn;
do /if cmp(NAME, "markup");
    set $markup VALUE;
break;
else /if cmp(NAME, "OID");
    set $OID VALUE;
.....
define event writeStart;
    putq "<def:ComputationMethod OID=" $OID ">" CR;
break;
end;
define event writeEnd;
.....
define event writeMid;
define event markupFirst;
    trigger writeStart;
    trigger writeMid;
    trigger writeEnd;
.....
```

Proc Template program to
convert SAS Datasets into
Define.xml

- Uses Events to make
dynamic templates

Proc Report

```
%MACRO REP_Domain(domain, label);

ODS PDF ANCHOR="&Domain";
ODS PROCLABEL = "Datasets" ;

PROC REPORT DATA = Column_metadata nowd noheader nocenter SPLIT=" CONTENTS="&Domain." style(report) = [OUTPUTWIDTH = 100%];

columns SASTableName SASColumnName ColumnLabel ColumnType TERM Origin Role ColumnComments CODELIST FORMAT EXT_DIC;

define SASTableName /noprint;
define CODELIST /noprint;
define FORMAT /noprint;
define EXT_DIC /noprint;
.....

Where SASTableName eq "&Domain." ;

COMPUTE SASColumnName;
URLSTRING = "#Valuelist.&DOMAIN.." || STRIP(SASColumnName);
CALL DEFINE (_COL_, 'URL', URLSTRING);
ENDCOMP;

COMPUTE TERM;
.....

COMPUTE ColumnComments;
.....

COMPUTE ORIGIN;
.....
RUN;

%MEND REP_Domain;

DATA _null_;
SET Table_metadata;
CALL EXECUTE ('%REP_Domain(' || STRIP(SASTABlename) || " , " || STRIP(DESCRIPTION) || "');');
RUN;
```

Proc Report Program to
convert SAS Datasets into
Define.pdf

- Uses ODS option
PROCLABEL, PROC REPORT
CONTENTS option to control
bookmarks
- Uses COMPUTE block to put
Hyperlinks

Conclusion

CONCLUSION

Conclusion

Centralized Metadata Repository

- Integrates all the metadata and stores it in one meta model schema that can be accessed easily.
- Compare metadata across several studies which ensure consistency throughout the project.
- Metadata checks can run on this repository to help better quality control.
- Repository can be populated before the actual programming begins

Questions?

