Leveraging Metadata for SDTM QC Automation

Sunil Gupta, TalentMine, Raleigh, NC

ABSTRACT

Seasonal SAS programmers know how to leverage SAS for automation by taking advantage of metadata and SAS macro programming. Accessing metadata of datasets and SAS programs to automate and standardize processing of multiple files can be achieved without manual efforts. Combined with SAS macro programming, automation allows for more robust and advanced programming techniques to handle messy data as well as unexpected changes in data input. In a dynamic environment, this helps to assure confidence in processing large amount of data to convert into meaningful information and action items.

INTRODUCTION

Metadata exists in most all files and consists of standard and custom metadata information. Standard metadata information includes physical attributes such as file size, date time stamp and number of variables. Custom metadata information provides more specific and vital information about data integrity, patient safety and efficacy. This type of insight is very important to monitor primary and secondary study endpoints.

This paper shows several examples of applying metadata to automatically check date time sequence of source, qc, source, log files and SDTMs and ADaMs across multiple libraries. In addition, another example shows how to automatically compare SDTMs and ADaMs across two libraries. Many SAS programming techniques including Proc SQL's dictionary tables, SAS and Dataset functions and Proc Compare are introduced.

METADATA OUTLINE

Below is the outline for this paper. The applications are examples only but represent a variety of applications.

Outline

- Metadata
  - Extracting intelligence Information
  - Standard Metadata
  - Custom Metadata
- Applications
  - Check DateTime Sequence of Source, QC, Log and List files
  - Check DateTime Sequence of SDTMs and ADaMs across multiple libraries
  - Compare SDTMs and ADaMs across two libraries

In general, the process is to first create an input file of directory of files such as excel files, SDTMs, ADaMs, SAS programs, log or Proc Compare.lst files. These files serve as input to the metadata processing of each file.

The common SAS tools used to process metadata for automation are below:

- Libnames
- Data Step
- SAS Macro Programming
- SAS & Dataset Functions
- Proc SQL
- Proc Compare
- Proc Means

Examples of output for metadata attributes are below:

- File pathname and names
- # of Files, Datetime stamps
- # and Type of variables
- Macro loop through all files
- Required datasets, variables, etc.
- Codelist dictionary, SDTM/ADaM attributes
- Maximum variable lengths
- Data cleaning & monitoring of valid variables and special characters

- Compare and contrast previous file
- Descriptive Statistics on categorical and continuous variables
- Search for ERRORs, WARNINGs or Notes in SAS Logs
- Search for QC differences in # of VARs, OBS, attributes and dups

STANDARD METADATA

Common standard metadata for all SDTMs and ADaMs domains and variables access the following dictionary tables.

1. VTABLE

a. Confirm # of datasets, observations and datetime stamps
b. Confirm required datasets and variables

2. VCOLUMN

a. Confirm # and type of variables and attributes
b. Confirm order of variables

CUSTOMIZED METADATA

While standard metadata is useful, customized metadata is focused to track primary goals. Customized metadata can be created for any SDTM or ADaM using Proc SQL. Examples are below.

1. Confirm # of patients
2. Confirm required non-missing variables
3. Confirm not greater than Maximum variable lengths
4. Confirm descriptive statistics on key categorical and continuous variables
5. Confirm compare and contrast previous datasets, attributes (more/less datasets/records/variables)
6. Confirm no duplicate records in each dataset
7. Include codelist dictionary from each dataset, ex. lab units

SDTM QC AUTOMATION

This example provides automatic comparison between SDTMs from one library with similar SDTMs to a second library. This technique can also be applied to compare ADaMs.

In the code below, Proc SQL accesses the VCOLUMN dataset from the WORK library and CLASS dataset to create DS_DIFF. For testing purposes the CLASS dataset is created. The second DATA step will then loop through each record in DS_DIFF to create Proc COMPARE code for CLASS.

This same process can loop through each SDTM in a library without having to hard code each SDTM name. CALL EXECUTE contains the Proc Compare syntax which compares same SDTMs from two different libraries. Other Proc COMPARE options such as NOVALUES, WARNING and NOPRINT convert dataset differences into warnings in SAS logs which make it easier to identify.

```
data class;
 name = 'Alfred';
 age=14;
 sex='F';
 height=75;
 weight=100;
run;

proc sql;
create table ds_diff as
 select unique memname
 from sashelp.vcolumn where upcase(libname)='WORK' and upcase(memname) = 'CLASS';
quit;

data ds_diff1;
 set ds_diff;

   call execute('proc compare data=work.'
|| memname
|| ' compare= sashelp.' || memname || ' listall out= ' || memname_ot || ' outnoequal; run;' );
```

```
run;
```

The first output shows the SAS general Proc Compare syntax for CLASS. The second output shows the partial results of Proc Compare. As expected, there are differences between the two CLASS datasets.

Output

```
NOTE: CALL EXECUTE generated line.
1         + proc compare data=work.CLASS                          compare= sashelp.CLASS
std1   outnoequal; run;

NOTE: There were 1 observations read from the data set WORK.CLASS.
NOTE: There were 19 observations read from the data set SASHELP.CLASS.
NOTE: The data set WORK.STD1 has 1 observations and 7 variables.
NOTE: PROCEDURE COMPARE used (Total process time):
      real time           0.01 seconds
      cpu time            0.00 seconds
```

Variables with Unequal Values

Variable	Type	Len1	Len2	Ndif	MaxDif
sex	CHAR	1	1	1	
height	NUM	8	8	1	6.000
weight	NUM	8	8	1	12.500

Page Break

The COMPARE Procedure
Comparison of WORK.CLASS with SASHELP.CLASS
(Method=EXACT)

Value Comparison Results for Variables

Obs	Base Value sex	Compare Value Sex
_	_	_
1	F	M

SDTM DATA TRANSFER METADATA

This example extracts metadata from each data transfer without having to hard code the name of each domain. In addition to SAS, code can be written to email data transfer summaries to teams for monitoring. While most data transfers are expected to be a complete replacement of the previous version, some data transfers may include only new or changed data. This may require special handling to append to the old data or replace original values.

Proc SQL confirms correct patient counts and creation dates for the three oncology SDTM domains – TU, TR and RS. In general, it is expected that these domains should have similar patient counts and creation dates. This

example can be modified to cross check consistency for any combination of raw, SDTMs and ADaMs. Patient counts and datetime stamp subqueries are created for each domain.

```
proc sql;
create table dc as
select unique
count(unique USUBJID) as tu_ptc label="USUBJID count"
, b.*, c.*, d.*, e.*, f.*, g.*
from sdtm.tu as a,
(select count(unique USUBJID) as tr_ptc from sdtm.tr) as b,
(select count(unique USUBJID) as rs_pct from sdtm.rs)  as c,
(select count(unique USUBJID) as tu_tr_rs_pct from tu_tr_rs)  as d,
(select unique input(scan(put(crdate, DATETIME16.), 1, ':'), date7.) format=date9. as tu_stamp from sashelp.vtable
  where upcase(libname)="SDTM" and upcase(memname)=upcase("TU")) as e,
(select unique input(scan(put(crdate, DATETIME16.), 1, ':'), date7.) format=date9. as tr_stamp from sashelp.vtable
  where upcase(libname)="SDTM" and upcase(memname)=upcase("TR")) as f,
(select unique input(scan(put(crdate, DATETIME16.), 1, ':'), date7.) format=date9. as rs_stamp from sashelp.vtable
  where upcase(libname)="SDTM" and upcase(memname)=upcase("RS")) as g;
quit;
```

DC results show expected patient counts and creation date for each oncology domain. Big differences in data timestamps for TR and RS could indicate that RS needs to be refreshed. The TR_TU_RS_PCT patient count is from the combined datasets from TU, TR and RS.

Output

	tu_ptc	tr_ptc	rs_pct	tu_tr_rs_pct	tu_stamp	tr_stamp	rs_stamp
1	164	164	147	164	09JUL2018	09JUL2018	09JUL2018

SUMMARY

By now, if your organization is not taking advantage of metadata in one form or another then they are missing the boat. Basic standard metadata enables simple yet powerful robust automation processing with minimum manual or custom programming. More advanced customized metadata unleashes the full potential of automation with a focus to address and monitor objectives or issues. This paper introduced methods to apply both standard and custom metadata for practical applications.

REFERENCES

Gupta, Sunil, “Ready To Become Really Productive Using PROC SQL?”, PharmaSUG 2012, SAS Global Forum 2011 and WUSS 2010

CONTACT INFORMATION

The author welcomes your comments and suggestions.
Sunil Gupta
E-mail: Sunil.Gupta@TalentMine.net

Sunil Gupta, MS, is an international speaker, best-selling SAS author, and a global CDISC corporate trainer. Sunil is Executive Director of Data Sciences at TalentMine and has over twenty-five years of experience in the pharmaceutical industry. Most recently, Sunil is a CDISC Oncology ADaM reviewer and has taught his CDISC online class at the University of California at San Diego and SAS Institute India. Sunil also taught his Sharpening Your SAS Skills online class for UCLA Extension. In 2011, Sunil launched his unique SAS resource blog, SASSavvy.com, for smarter SAS searches and teaches Data Science Using SAS. Sunil has MS in Bioengineering from Clemson University and a BS in Applied Mathematics from the College of Charleston.