The Path to Data Standards

Sy Truong, Meta-Xceed, Inc, Fremont, CA

ABSTRACT

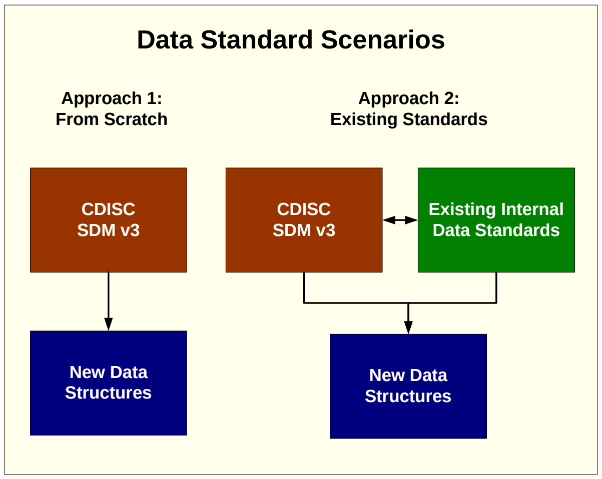
Data standards can make data and its associated programs more portable. Team members who work with the data also become more portable since they can understand and use data with more ease. This makes the development and validation of SAS® programs within a regulated environment much more efficient. This paper will present strategies on working with the new CDISC Submission Data Model version 3.0 along with other data standard strategies that are independent of CDISC. Some of the approaches include:

1. Automated evaluation of existing data structures against CDISC standards
2. Automated evaluation of existing data structures and formats among each other
3. Designing new data standards from existing data structures

The practical concepts of these techniques are demonstrated through both a manual process and tools such as *%cdisc* and *%difftest*. The meticulous review of data attributes among all the data is an important step towards achieving data standards. Automating this process makes the task less mundane, as well as catches non-standard differences that are not easily caught through manual verification.

INTRODUCTION

There are two general approaches towards achieving data standards. If you are starting from scratch, it makes sense to use a suggested standard such as CDISC. In this case, the effort will be in ensuring that new data created adheres to this standard. A second approach is when you already have existing data that is structured very different from CDISC standards. In this case, the task is to make sure that all existing data structures follow an internal standard. Any new data created would then need to adhere to this new standard. It is more common for the second scenario to occur. This paper will examine both scenarios and suggest techniques along with tools to assist you in these approaches.



PORTABILITY OF DATA

Establishing data standards and applying the standards across all studies and projects can be resource intensive. It is reasonable to ask the question whether it is worth all the effort. One of the key benefits is that the programs associated with this data become more portable. They can be moved from one study to the next with minor modifications. Not only are the programs more portable, the programmer and statistician working on one study can understand a new study with the same structure relatively quickly compared to learning a new set of programs, macros and data structures. The productivity gain is sometimes difficult to measure but, in the long run, it will outweigh the efforts invested in standardizing.

IMPLEMENTING CDISC

The version of CDISC Submission Data Model (SDM) presented in this paper is version 3.0 which is radically different from version 2.0. This means that if you were implementing the standards set by version 2, you almost have to start over. However, if you are just beginning to set up standards, and there is no history from any legacy systems, adapting to CDISC SDM is a smoother transition. In this case, your main goal is to ensure that when you create a new dataset, you have a method of ensuring that the new variables and associated attributes adhere to the CDISC standards. The standards referenced in this paper are found at: http://www.cdisc.org/pdf/V3CRTStandardV1_2.pdf. You can get the latest standards at: http://www.cdisc.org/standards/index.html. There are two aspects to the strategy presented in this case. The first is a non-technical procedural component which includes methodologies and the second is comprised of tools that can help the process.

If there is more than one team member in your group, it is essential that you establish Standard Operating Procedures (SOPs) for data standards. Not only are these required by regulations, they can help the group work together. A common approach is to have someone on your team be assigned the role of the Metadata Administrator who acts as the point person pertaining to all data submission standards. This narrows down the scope from other data sources such as operational source data or exploratory analysis data that is not intended for submission. The Metadata Administrator is the gate keeper to all new attributes of variables and datasets. Their responsibility is to ensure that new proposed names adhere to standards while also maintaining existing attributes and retiring old attributes that are no longer valid. This centralized approach ensures that standards are established consistently and that they are enforced. The Metadata Administrator can work with the rest of the team to establish SOPs and tasks pertaining to:

New Data	New datasets defined for a particular study
New Variables	New variables and associated attributes defined within a particular data domain
Reconciliation	Reconciling differences among metadata between studies
Retirement	Retiring old dataset attributes and/or associated variables that are no longer used

It is useful that these SOPs be documented. In the event that the primary Metadata Administrator is not available, a backup person can take over. It is also important from a regulatory standpoint, since if SOPs are not documented, it is perceived that the effort was done haphazardly or that it was never done at all.

Once the procedure is put into place, tools can be implemented to help the Metadata Administrator be more effective. There can be separate tools for each of the tasks mentioned above in the SOPs. The one discussed in this paper is named *%cdisc* which assists in the reconciliation of differences in metadata between studies against CDISC standards.

MANUAL STANDARDIZATION STEPS

Before automated tools were created, the steps taken to ensure standards were done manually. These steps can be used to work with CDISC standards and also with existing data standards.

STEP 1: Capture all metadata from the complete set of datasets included in the submission.

```
*** Capture metadata of the Adverse
Event data ***;
proc contents data = datalib.ae
   out=m_ae;
run;
```

This is done by applying a PROC CONTENTS on each dataset. A separate dataset is generated to contain the metadata of each dataset. This can be stored in the work area for evaluation.

STEP 2: Review and identify the variables that are found in more than one dataset.

```
*** Compare the metadata between
datasets ***;
proc compare data=m_ae
   compare=m_demog;
run;
```

This can be accomplished by reviewing the output visually and/or using PROC COMPARE to help identify differences and similarities.

STEP 3: Review and identify the variables and attributes that match CDISC standards. The following tasks can be identified from the standards. The list consists of a short name for the task followed by a reference number to the CDISC document in parenthesis, followed by a description of the task.

1. **Required Fields**: (2.4.5) Required identifier variables including: DOMAIN, USUBJID, STUDYID and --SEQ.
2. **Subject Variable**: (3.5.1.2.8) For variable names, labels and comments, use the word "Subject" when referring to "patients" or "healthy volunteer".
3. **Variable Length**: (3.5.1.2.6) Variable names are limited to 8 characters with labels up to 40 characters.
4. **Yes/No**: (3.5.1.3.18) Variables where the response is Yes or No (Y/N) should normally be populated for both Yes and No responses.
5. **Date Time Format**: (3.5.1.4.19) Use yymmdd10. but yymmdd8. is acceptable.
6. **Study Day Variable**: (3.5.1.4.22) Study day variable has the name ---DY.
7. **Variable Names**: (3.5.2) If any variable names used match CDISC variables, the associated label has to match.
8. **Variable Label**: (3.5.2) If any variable labels used match CDISC labels, the associated variable has to match.
9. **Variable Type**: (3.5.2) If any variables match that of CDISC variables, the associated type has to match.
10. **Dataset Names**: (3.5.2) If any of the dataset names match CDISC, the associated data label has to match.
11. **Dataset Labels**: (3.5.2) If any of the dataset labels match CDISC, the associated dataset name has to match.
12. **Abbreviations**: (3.5.2) The following abbreviations are suggested for variable names and data sets.
    - DM Demographics
    - CM Concomitant Medications
    - EX Exposure
    - AE Adverse Events
    - DS Disposition
    - MH Medical History
    - EG ECG
    - IE Inclusion/Exclusion Exceptions
    - LB Labs
    - PE Physical Exam
    - SC Subject Characteristics
    - SU Substance Use
    - VS Vital Signs
13. **SEQ Values**: (4.3.2.1) When the --SEQ variable is used, it must have unique values for each USUBJID within each domain.

STEP 4: Ensure that all other attributes of the variables that matched with CDISC standards also adhere to the standards. This is a manual visual process of comparing the results from the PROC CONTENTS with the list of tasks to ensure that standards are being applied.

AUTOMATING VERIFICATION TASKS

The same tasks performed in the previous section can be automated through a macro named *%cdisc*. This macro is made available as a free download at http://www.meta-x.com/sydata/cdisc_download/. The *%cdisc* macro is part of a larger toolset named Sy/Data™. This macro simplifies the previous steps since all that is needed is that the user specifies the *libname* and the dataset in which to perform the standards evaluation.

```
%cdisc (datlib = data library,
       datname = dataset name);
```

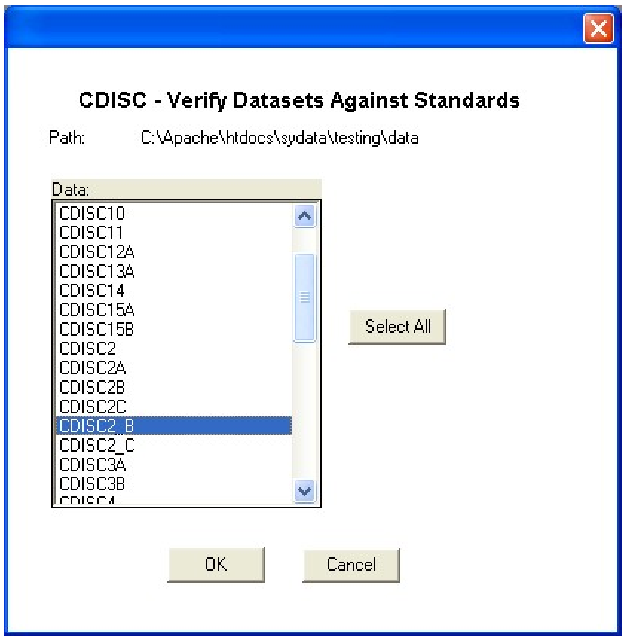
Where	Is Type...	And represents...
datalib	C (200)	Library name referencing the location where the dataset resides.
datname	C (200)	Name of the dataset to be verified. Wild cards can be specified such as ae*.

This macro systematically goes down the list of verification tasks to verify all attributes of your data against that set by the CDISC guidelines. For example, if you use a standard variable *SEX* in your data that was defined in the guidelines, but you have defined a different label or length, this will be highlighted in a resulting report.

```
          --- Findings from CDISC Evaluation ---

                      Data
     Library          Table     Variable   Variable      Case
Obs   Name            Name        Name       Label      Number

 1   templib     cdisc9                                    1
 2   templib     cdisc9                                    1
 3   templib     cdisc9                                    1
 4   templib     cdisc9       SEX     WRONG LABEL          7

Obs  Comments

 1   Data Missing Variable USUBJID
 2   Data Missing Variable STUDYID
 3   Data Missing Variable --SEQ
 4   Variable name matches guidelines but not label
```

In this example, the results highlight Test Case Number 1 which includes required fields and also Test Case 7 which it describes as "Variable name matches guidelines but not label". The descriptive comments state what the user should be aware of in the event that his data is diverging from the standard. This report is generated by default but the information is also available in a temporary dataset named WORK.CDISC. This gives users more flexibility by using their own favorite SAS procedure or ODS to extend the reporting possibilities.

This macro can be even easier to use through a graphical user interface implemented in SAS/AF. In this case, the selection of dataset becomes as simple as selecting items from a list.


CDISC - Verify Datasets Against Standards

Path: C:\Apache\htdocs\sydata\testing\data

Data:
CDISC10
CDISC11
CDISC12A
CDISC13A
CDISC14
CDISC15A
CDISC15B
CDISC2
CDISC2A
CDISC2B
CDISC2C
CDISC2_B
CDISC2_C
CDISC3A
CDISC3B

STANDARDS WITHOUT CDISC

The strategy of comparing datasets being created with CDISC guidelines is a great method for catching standard deviations. However, many users decide not to follow CDISC standards. This is due to a number of reasons. It could be that their studies started prior to the current CDISC guidelines and therefore it is too much effort to convert. It could also be that the data were standardized to version 2 of the CDISC guidelines or to an internal standard that is not compatible with version 3. In many circumstances, the standards being implemented are not the latest CDISC guidelines and therefore the previous approach is not applicable.

Standards are still essential even if your approach is not in line with CDISC. In this event, you can follow a similar series of steps. However, you are no longer just comparing your data with the prescribed CDISC guidelines, but rather performing the verification against other SAS datasets that contain your own standards.

STEP 1: Capture all metadata from the complete set of datasets included in the submission. This can be accomplished with PROC CONTENTS.

STEP 2: Review and identify the variables that match any attributes found in comparison with the standard dataset along with other data submitted. This can be implemented by using PROC COMPARE.

STEP 3: Ensure that all attributes of the variables which are found in more than one location match up and align with each other. This can be accomplished by reviewing the output of the PROC CONTENTS to ensure that all other attributes are the same. The following test criteria are used to perform your comparisons.

1. For variables with the same name across different datasets, verify that the following attributes are the same:
   a. Type
   b. Length
   c. Label
   d. Format Name
   e. Informat Name
2. For variable labels that are the same, verify if the corresponding variable names are the same.
3. For format names that are the same, verify if the coded values of the formats are the same.
4. For format codes that are the same, verify if the format names are the same.

5. For dataset names that are the same, verify if the dataset labels are the same.
6. For dataset labels that are the same, verify if the dataset names are the same.
7. For variables with coded formats, verify if the values in the data match up with the specified format codes.

The above steps no longer rely on CDISC as a comparison benchmark, but rather your own internal standards. This standard is comprised of a set of datasets and format catalogs that you maintain for each domain, such as demographic data, adverse events, concomitant drugs, etc… By applying the comparisons between your data against this set of standard datasets, you can quickly identify deviations and make the changes to adhere to the set standards. This step can be revealing in that, for some instances, it may lead you to change your standards if you find that it make more sense to follow the example of the new data.

AUTOMATING COMPARISONS

In the same way that *%cdisc* can be applied to CDISC standards, the *%difftest* macro is used to compare differences among datasets for determining internal standards. In this case, rather than comparing all the target datasets to one base model, the *%difftest* compares all the permutations between the datasets selected. For example, if you have three datasets: DEMOG, AE and CONMED, the comparisons made to each other are similar to performing a Cartesian join between the base and target datasets, with the exception of making comparisons to itself.

```
Obs    base      target

  1     CONMED    CONMED*
  2     CONMED    DEMOG
  3     CONMED    AE
  4     DEMOG     CONMED
  5     DEMOG     DEMOG*
  6     DEMOG     AE
  7     AE        CONMED
  8     AE        DEMOG
  9     AE        AE*

* Excluding comparisons to itself
```

This approach ensures that every variable is compared. The above Cartesian join example is created by the following program.

```
data one;
   base = "CONMED"; output;
   base = "DEMOG"; output;
   base = "AE"; output;
run;

data two;
   target = "CONMED"; output;
   target = "DEMOG"; output;
   target = "AE"; output;
run;

proc sql;
   create table compare as
   select * from one, two;
quit;
```

This illustrates how the number of comparisons can multiply exponentially as the number of datasets increases. It is therefore more efficient to automate this task rather than doing it manually.

```
%difftest (fpath_a ... fpath_z = format path,
       path_a ... path_z = path to the SAS data,
       dat_a1...dat_z100 = dataset name);
```

Where	Is Type...	And represents...
fpath_a - fpath_z	C (200)	Path location to the format catalog which contains user defined formats for the data.
path_a - path_z	C (200)	Physical path location to SAS data used for verification.
dat_a1 - dat_a100 ... *dat_z1 - dat_z100*	C (8)	SAS dataset name which will be used for verification. Note that the alpha variable following the "dat_" matches with the name of the path parameter. If an asterisk is specified as a wildcard, all datasets within the specified path will be selected.

The macro will first perform a Cartesian join between the data sets specified by the *dat_n* parameters and then loop through the combinations to perform the seven tests defined in step 3. The resulting differences are presented in a report organized by each test condition. An example for test condition 1 looks like:

DIFFTEST
Data, Variable and Format Difference Test

Test 1: Variables with the same name containing different attributes across different datasets

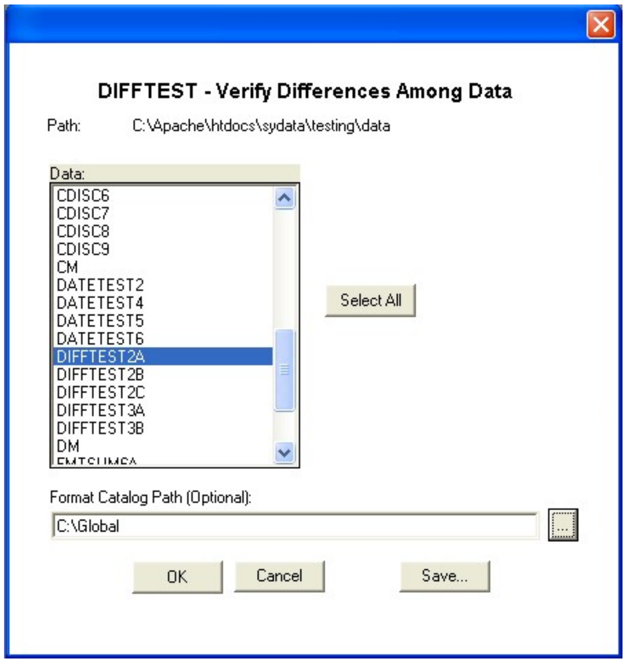
		Data Path	Data name	Variable	Type	Length	Label	Format	Informat
1	**Base**	C:\temp	DEATH	visitno	C	17	VISIT NUMBER		$CHAR17.*
	Compare	C:\temp	DEMOG	visitno	C	17	VISIT NUMBER		$CHAR9.*
2	**Base**	C:\temp	DEATH	specsite	C	80*	PROG DISEASE SITE SPECIFIED*		$CHAR80.*
	Compare	C:\temp	METASTA	specsite	C	40*	ULTRASOUND SPECIFIED SITE*		$CHAR40.*
3	**Base**	C:\temp	DEATH	visitno	C	17	VISIT NUMBER		$CHAR17.*
	Compare	C:\temp	METASTA	visitno	C	17	VISIT NUMBER		$CHAR9.*
4	**Base**	C:\temp	DEATH	othspec	C	100	OTHER EVENT SPECIFIED*		$CHAR100.

	Compare	C:\temp	TERM	othspec	C	100	OTHER SPECIFY*		$CHAR100.
5	Base	C:\temp	DEATH	ptstatus	C	4*	*		
	Compare	C:\temp	TERM	ptstatus	C	5*	PATIENT STATUS*		

* Found Difference

In this example, the datasets DEMOG, DEATH, METASTA and TERM were compared. This report shows how the variables have the same name across different datasets, but their attributes differ. The differences are highlighted with red asterisks to help you quickly identify possible standard deviations. Differences in length or variable type can cause significant problems if this is a key variable used in a merge. SAS label differences are sometimes intentional but many times it makes more sense to keep variables consistently labeled. There are many subtle differences that can occur which are challenging to catch with visual inspection.

Similar to the *%cdisc* tool, there is an accompanying graphical user interface for *%difftest* to make the selection of datasets and format catalog easier.



DIFFTEST - Verify Differences Among Data

Path: C:\Apache\htdocs\sydata\testing\data

Data:
CDISC6
CDISC7
CDISC8
CDISC9
CM
DATETEST2
DATETEST4
DATETEST5
DATETEST6
DIFFTEST2A
DIFFTEST2B
DIFFTEST2C
DIFFTEST3A
DIFFTEST3B
DM

Format Catalog Path (Optional): C:\Global

Applying a difference test comparison on all your data is a good way of maintaining a standard but it can also be used to set up new standards. In the case where you have a lot of legacy data and you are just starting out with setting up new standards, an analysis of the similarities and differences is a good way to identify new standards. Even if you have standards already established, this method helps in refining the standards once you see how they compare to data that is being used. A Metadata Administrator implements data standards in a similar way to how software is released. For example, the first set of data standards version 1.0 would be considered. Once it has been compared to real data being used, you may see that it makes more sense to alter and update your standards. Changing variable attributes standards incrementally can cause change control problems among users trying to keep up and adhere to the standards. You can choose to group the changes in logical steps, such as those dealing with a particular domain, and update the standards to version 2.0. A practical approach is to collect all the enhancements and wait until a pivotal point in a project is completed before implementing the next version of data standards. An example of this is when a series of studies are completed for a FDA submission; you would then roll out the data standards for the next set of new studies.

PUBLISHING AND ENFORCEMENT

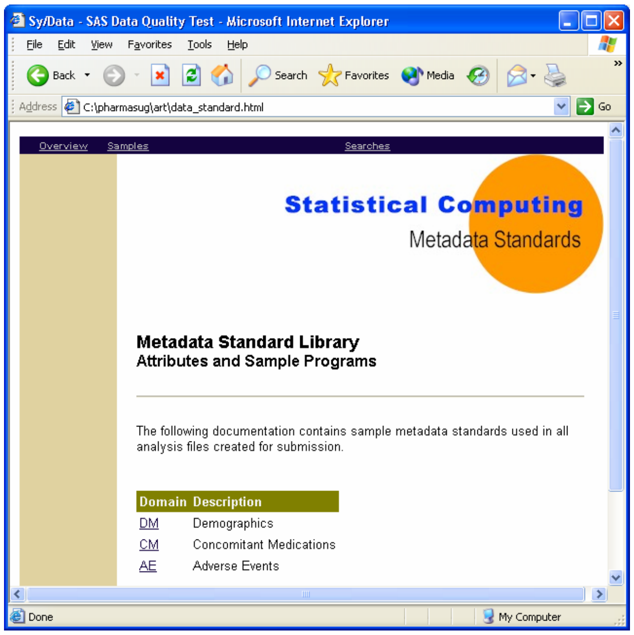
Once standards are established, one of the main challenges is to enforce the standards. This is most effectively done through awareness, training along with clear communication between the Metadata Administrator and the end users. There are many forms in which this communication can take shape. The methods described in this section are intended to minimize resource drain on the administrator, while optimizing the "bandwidth" of communication to as many users as possible. The approaches are shown in steps starting with the easiest methods and moving to higher levels of sophistication.

STEP 1: Generate a report of all the standard metadata for each data domain and deliver this to the user. This can be as simple as a PROC CONTENTS of a sample data standard. Accompanying each metadata report is a sample SAS code segment used to define the structure. For example:

```
*** Define the Subject Demographic data
***;
data demog (label="Subject
Demographics");
   attrib USUBJID  length=$40
   label="Unique Subject Identifier";
   attrib STUDYID  length=$80
   label="Study Identifier"
   attrib SUBJID   length=$20
   label="Subject Identifier";
   attrib BRTHDTM  length=8
   label="Date/Time of Birth"
   format=yymmdd10.;
   ...
run;
```

It is recommended that this information be available in electronic form so that users can cut and paste it into their programs. This can be emailed to users and referenced at a central file location on a server.

STEP 2: Make the data standards accessible via an intranet website. The reports mentioned in step 1 are useful on a central file server. However, navigating to it can be cumbersome compared to a hyperlink. Periodic email updates can include the attached information but a hyperlink which users can bookmark makes it a more useful destination.


Sy/Data - SAS Data Quality Test - Microsoft Internet Explorer
File Edit View Favorites Tools Help
Back Search Favorites Media
Address C:\pharmasug\art\data_standard.html
Go
Overview Samples Searches
Statistical Computing
Metadata Standards
Done
My Computer

Metadata Standard Library
Attributes and Sample Programs

The following documentation contains sample metadata standards used in all analysis files created for submission.

Domain	Description
DM	Demographics
CM	Concomitant Medications
AE	Adverse Events

The use of hyperlinks can be further implemented as a drill down throughout the report with a summary of the domain as the main table of contents.

STEP 3: Have all the attributes of the metadata searchable within the standard library. It is optional to have the search applied to specific attributes such as labels or variable names. However, for simplicity, the default search applies to all attributes.

A search engine can greatly improve the efficiency of how users navigate to a specific data standard. This approach is an extension of step 2 since it accompanies the same website. It can be implemented with standard PROC SQL or SAS data step queries when developed with SAS/IntrNet.

All the steps mentioned above are effective ways of publishing the information. It would be ideal to have all the steps implemented if resources are available. The same information can be "pushed" to users by providing training sessions or emailing or delivering hard copy reports. On the other hand, it can also be useful to have users "pull" the information from a centralized server. Both methods can be employed to increase the chances that users would adhere to an evolving and often changing standard.

CONCLUSION

Data standards are potent tools in becoming more effective and efficient during your analysis programming and reporting. Accomplishing standards leads to portability of data between studies, while also increasing the mobility of team members between projects. This can increase consistency of data for accuracy and decrease validation efforts. Depending on the history and environment of your data, it can be beneficial to use CDISC or develop your own internal data standards. In either case, it is important to stage the release of standards within your group at logical time points to avoid dramatic changes to existing standard software. Once you have your data standards applied, the maintenance can be automated to a degree. There are many attributes to manage, so the use of automated tools will help keep the administrator on top of all the meticulous details, while also maintaining a global view of all data. Structures such as SOPs and assigning an administrator are important in deriving and maintaining standards. The metadata administrator can employ traditional instructor-based training during the roll out of standards. However, taking advantage of tools and techniques of Intranet publishing will increase the success of users' acceptance of standards.

REFERENCES

ABOUT THE AUTHOR

Sy Truong is a Systems Developer for Meta-Xceed, Inc. They may be contacted at:

Sy Truong

48501 Warm Springs Blvd. Ste 117

Fremont, CA 94539

(510) 226-1209

sy.truong@meta-x.com