

Taking a Walk on the Wildside: Use of the PROC CDISC-SDTM 3.1 Format

William C. Csont, Howard M. Proskin & Associates, Inc., Rochester, N.Y.

ABSTRACT

Many people are anxiously awaiting SAS's ® interpretation of the CDISC standards with regard to clinical pharmaceutical trials. After hearing about the PROC CDISC procedure at a previous conference, I decided to investigate the use of PROC CDISC with a particular emphasis on the SDTM (Standard Data Tabulation Model) format upon my return to the office. When I tried running PROC CDISC, I was met with log error messages. This is where my odyssey began with adding PROC CDISC to my Base SAS system and the exploration of the procedure. This paper will describe the steps to add PROC CDISC on a Windows based system and explain what information can be derived from the PROC CDISC SDTM 3.1 Format.

PROC CDISC: WHERE ARE YOU!

I went to PharmaSUG 2005 in Phoenix, AZ and began to hear about the wonders of CDISC and some rumblings about a SAS procedure named: PROC CDISC. I returned to work and filed it in the back of my mind for further investigation. A year went by and I laterally moved from a position as a SAS programmer to the Data Manager for our small CRO-type company. I was unable to attend PharmaSUG 2006, but several of my colleagues attended it and wanted to know more about CDISC. Since I was the person who knew the most about CDISC, I was asked to present my colleagues with more information about CDISC. Upon my journey of reviewing CDISC materials, I stumbled across information about the PROC CDISC procedure. This led me to the start of my epic odyssey. I was using Windows SAS System 9.1.3 Service Pack 2 and I looked through the locally installed SAS Help and Documentation for descriptions of the PROC CDISC procedure. Also, I couldn't find any descriptions about PROC CDISC under the SAS Procedures folder for Base SAS. Next, I tried a global search of the term CDISC, but came up with a few citations relating to CDISC and XML, but nothing on PROC CDISC. Next, I thought that even though I did not know the syntax, I would try to run PROC CDISC in my Enhanced Editor window. After running the procedure, the log stated it was an unrecognized procedure error. Now what could I do?

WALKING ON THE WILDSIDE

I decided that the Internet might hold some answers to my dilemma. I went to the SAS Support Website (support.sas.com) and typed PROC CDISC into the site's search engine. I was eventually led to a SAS page about PROC CDISC - <http://support.sas.com/rnd/base/index-xml-resources.html>. This page stated that PROC CDISC was not installed by default, but that I could download a zip file (currently named `cdisc55_v9_win.zip`) containing the PROC CDISC engine components. Upon unzipping the files contents, I noticed a `readme.txt` file. I opened that file in a text editor and read it. After reading the text, I knew that I would be "Walking on the Wildside" if I wanted to use PROC CDISC. The text file contained the words, "Update the SAS configuration file", "Create a new directory in the <SASROOT>" and "may invalidate the SAS Installation Qualification (IQ) checksum". These words should serve as red flags to the novice SAS user, as well as that the more experienced SAS user should be very cautious. Undeterred and being an experienced SAS User who was familiar with this terminology, I was eager to try my fortune in updating my system to allow for the processing of PROC CDISC. I will note that subsequent to my initial PROC CDISC install, SAS is currently releasing the procedure as a Field Response Module, which has different versions and contains different system files than those included in Service Packs, but generally contains the latest and greatest enhancements to the procedure. More will be discussed about this later in this paper.

IT'S ONLY A FOUR STEP PROCESS

STEP 1

I followed the instructions as written in the `readme.txt` file. It is a four step process. For the first step, it was recommended that I create a CDISC subdirectory in the <SASROOT> area. <SASROOT> is defined as the location of your core SAS executable files and for most Windows users is found at `C:\Program Files\SAS\SAS 9.1`.

STEP 2

Step two is you need to extract the contents of the zip file (currently named: cdisc55_v9_win.zip) into the newly created CDISC folder found under <SASROOT>. There are several fine commercial and freeware (such as: WinZip and PowerDesk) or your operating system (Windows XP) unzipping utilities available to perform this unzipping action.

STEP 3

Step three is you need to update your SAS configuration file. Take care, this is where it gets on the wildside, as if you do not properly update your SAS configuration file, SAS may not run at all or be able to properly operate all system functions. To find your SAS configuration file look at your <SASROOT>/nls/en for a file with a .cfg extension. By default, for most user's, it should be named sasv9.cfg. Next, you need to open your SAS configuration file with a text editor (such as Word Pad, SAS Viewer, or SAS itself). When the SAS configuration file is open, search for the text string -PATH. When it finds this text string, this section of the SAS configuration file should look like this:

```
-PATH      (
           " !sasroot\core\sasexe"
           " !sasext0\sview\sasexe"
           ...
           )
```

You will need to add one line of code to have the SAS system activate the CDISC procedure. After you add the code, it should look as follows below (see bolded line):

```
-PATH      (
           " !sasroot\cdisc"
           " !sasroot\core\sasexe"
           " !sasext0\sview\sasexe"
           ... )
```

STEP 4

Step four requires you to open your SAS system and execute three lines of code. The code permanently updates the SAS help files with some template information about XML and CDISC. You can either copy the code from the readme.txt file found in your newly created CDISC folder or use the code listed here below:

```
ods path sashelp.tmplmst(update);
%include 'XMLcdisc.tpl';
%include 'XMLodm.tpl';
```

You need to make certain that before this code is run that your current folder is <SASROOT>/CDISC. If it is not, you will receive an error message and the %include lines will not run. Another potential error with this procedure is if you do not have write authorization to the sashelp catalog. In this instance you may wish to create a local copy of the templates in your autoexec.sas. The autoexec.sas is found in the same location as the SAS configuration file: <SASROOT>/nls/en. After you find this file, open it with your favorite text editor and add the following lines:

```
%include 'XMLcdisc.tpl';
%include 'XMLodm.tpl';
```

The %include lines must have a fully qualified pathname to each of the two (tpl) template files.

Once step four is complete, you are now ready to PROC CDISC!

A word of caution for those of us working in pharmaceutical clinical trials; this process may cause the checksums on your SAS Installation Qualification Tool to report an error after this update is applied. A description of what to do to correct this situation will follow in the next section of this paper.

DANGER, WILL ROBINSON

If you are not in a pharmaceutical clinical trial production environment that is susceptible to FDA audits, feel free to ignore this section. Otherwise, you need to **PAY ATTENTION**. GCP (Good Clinical Practice) and SOPs (Standard Operating Procedures) generally state that any changes to software must be verified for proper installation through qualifications. SAS supplies two very useful tools for fulfilling the Installation Qualification (Were the proper files copied and updated) and the Operational Qualification (Do the SAS Procedures work as expected). There is an issue that can occur with regard to the Installation Qualification (IQ). Apparently, the current Field Release may invalidate your IQ because checksums (what SAS uses to validate the install) are only updated at the hotfix or full new version release level. If you run your IQ and it fails, have no fear, SAS has documented methods of allowing the IQ to pass. There are at least 2 support articles that I found which address how to fix this problem: (SAS Notes: SN-V9-018738 and SN-V9-017442). Make certain you document what you did if you need to make modifications to files used by the SAS system and you should be good to go.

A BRIEF HISTORY AND THE FUTURE OF PROC CDISC

Before we move on to describing PROC CDISC relating to SDTM, I thought it might be helpful to get some historical perspective on this procedure and its future. I decided the best route would be to go contact the SAS Institute. I sent an e-mail to SAS and promptly received a response from Anthony 'T' Friebel, the actual code writer and maintainer of PROC CDISC. He stated that PROC CDISC first was made widely available in SAS Version 9.1.3 Service Pack 3, but was available prior to that as a Field Response Release. He added that Field Response Releases generally are used in the next production candidates and that they are made available for early adopters and to solicit feedback. Currently, PROC CDISC Field Response Releases are not available for all operating systems. These Field Response Releases can be found on the Internet at: <http://support.sas.com/rnd/base/topics/sxle913/fieldresponse.htm>.

Anthony described the future of PROC CDISC relating to SDTM into two categories: short and long term. The short-term goals are to re-architect the SDTM validation to support SDTM (IG – Implementation Guide) 3.1.1 by around Jan 2007. The longer-term goal is to allow the new validation methodology to support custom defined domains. A domain is defined per the CDISC organization as a collection of logically related observations with a topic-specific commonality about subjects in a clinical trial. As an aside, he added that other future enhancements to PROC CDISC (non-SDTM) would include: the generation of define.xml and an updated ODM support to Version 1.3 (including new ISO date/time/ datetime features).

Lastly, Anthony added that the release is available in SAS version 8.2 and functions identically to that of version 9.13. The update for version 8.2 must apply tech support hot-fix (82BC01) prior to adding any of the current Field Response Releases.

AND NOW FOR THE REST OF THE STORY...

We've discussed how to get the latest and greatest PROC CDISC onto your system, its history and future, but now its time to discuss what the procedure does. In essence, the procedure verifies (for 15 of the 23 SDTM domains) whether your Version 3.1 SDTM dataset meets selected major criteria to enable you to state that you have a valid domain dataset that can be used in an electronic submission to a governing entity (such as the FDA). The 15 SDTM domains the procedure supports are: DM (Demography), CO (Comments), CM (Concomitant Medications), EX (Exposure), SU (Substance Use), AE (Adverse Events), DS (Disposition), MH (Medical History), EG (ECG Test Results), IE (Inclusion/Exclusion Exception), LB (Laboratory Test Results), PE (Physical Examinations), QS (Questionnaires), SC (Subject Characteristics), and VS (Vital Signs).

The major criteria that the procedure verifies can be separated into two groups: overall domain content and per observation. The major criteria that are verified in the overall domain content include: checking that all required variables are present in the dataset, reporting as errors any variables that are not defined per the domain requirements, reporting a warning for expected domain variables that are missing, noting any permitted variables that are not in the dataset, verifying that all domain variables are the correct type and proper length, and detecting any domain variables that have assigned controlled terminology by the domain but are missing an assigned format. The major criteria that are verified in the per observation review include: verifying that required variable fields do not have missing values, detecting missing values in expected variables, detecting the conformance with ISO-8601 specification assigned values (e.g., date, time, datetime, duration, and interval type variables), and verifying variables with yes/no or yes/no/null responses.

Now, let's look at the actual code and what it produces. First, you need to have a SDTM domain dataset that is ready to check. For our purposes, I have created a dummy SDTM DM (Demographics) Domain dataset in the code listed below and added some basic formatting.

```
proc format;

  value $sex
    'M' = 'Male'
    'F' = 'Female';

  value $invid
    '123456' = 'New York City';

run;

data demog;
  length STUDYID $20. DOMAIN $2. USUBJID ARM $14. SUBJID $3. RFSTDTC RFENDTC
    DMDTC BRTHDTC $20.
    SITEID INVID $6. INVNAM $30. AGE DMDY 4. AGEU $10. SEX RACE ETHNIC ARMCD $1.
    COUNTRY $3.;

  format SEX $SEX. INVID $INVID.;

  STUDYID='Study 437';
  DOMAIN='DM';
  USUBJID='437123456789';
  SUBJID='789';
  RFSTDTC='2006-01-15T12:15:26';
  RFENDTC='2006-02-27T14:19:29';
  SITEID='123456';
  INVID='123456';
  INVNAM='John Smith';
  BRTHDTC='1980-05-21';
  AGE=25;
  AGEU='YEARS';
  SEX='M';
  RACE='1';
  ETHNIC='2';
  ARMCD='0';
  ARM='Test Solution';
  COUNTRY='USA';
  DMDTC='2006-01-15T12:15:26';
  DMDY=0;

run;
```

The code listed above essentially creates a one record SDTM Demographics file that can be run against PROC CDISC for verification. You need to refer to the SDTM Implementation Guide Version 3.1 dated July 14, 2004 found at www.cdisc.org to determine each domain what are the variables type, name, label, origin, role and any associated formatting or controlled terminology. Next, after the SDTM file is ready you can run PROC CDISC. The next code listed below is a sample of how to run PROC CDISC against the newly created SDTM ready Demographics file.

```
proc cdisc model=sdm;
  SDTM SDTMVersion = "3.1";
  DOMAINDATA DATA = demog
    DOMAIN = DM
    CATEGORY = SPECIAL;
run;
```

After you run the code, I was initially expecting some long output report describing all of its checks and actions, but there was no output, only a brief message in the log that looks like this:

```
/* CDISC SDTM DEMOG DATASET WITH ALL CORRECT VARIABLES */
```

```
proc cdisc model=sdm;  
  SDTM SDTMVersion = "3.1";  
  DOMAINDATA DATA = demog  
             DOMAIN = DM  
             CATEGORY = SPECIAL;  
run;
```

```
NOTE: PROCEDURE CDISC used (Total process time):  
      real time          0.04 seconds  
      cpu time           0.01 seconds
```

However, if you have a dataset that doesn't follow CDISC standards (FYI: I purposely dropped the country variable from the demog data), a log such as this will follow:

```
/* BAD CDISC SDTM DEMOG DATASET WITH DROPPED COUNTRY VARIABLE */
```

```
proc cdisc model=sdm;  
  SDTM SDTMVersion = "3.1";  
  DOMAINDATA DATA = demog  
             DOMAIN = DM  
             CATEGORY = SPECIAL;  
run;
```

```
ERROR: Required parameters not contained on DOMAINDATA(Domain=DM) statement.  
       Required parameter COUNTRY not present.  
NOTE: The SAS System stopped processing this step because of errors.  
NOTE: PROCEDURE CDISC used (Total process time):  
      real time          0.00 seconds  
      cpu time           0.00 seconds
```

Another feature of PROC CDISC is that you can run the following snippet of code and it will generate a message in your SAS Log that states the version that the procedure is executing. This is a good management tool and accounts for potential discrepancies between different field releases or the current production release.

```
proc cdisc version;  
run;
```

CONCLUSION

In conclusion, with a little bit of persistence and research (or by using this paper) you can be on the leading edge of using SAS's PROC CDISC procedure for SDTM. You can also use this as a stepping stone for ODM and define.xml (as it is rolled out in the future). This will be very helpful to those of us in clinical research trials research that will need to prepare and submit CDISC standard data files. My advice is to continue to follow SAS's updates at its XML resources web page.

ACKNOWLEDGEMENTS

I would like to thank Anthony 'T' Friebel for his help on preparing background information and the future uses of PROC CDISC, as well as his comments on this paper. I would also like to acknowledge my colleague, William Murphy, author of many SAS Conference papers for his critique of my paper. Lastly, I would like to acknowledge my boss, Dr. Howard M. Proskin for graciously giving me some time to prepare this paper.

REFERENCES

SAS Institute Inc. 2007, Base SAS: XML (Extensible Markup Language). "PROC CDISC"
<http://support.sas.com/rnd/base/index-xml-resources.html> .

SAS Institute Inc. 2007, SAS 9.1.3 XML LIBNAME Engine: User's Guide. "PROC CDISC Field Response Release"
<http://support.sas.com/rnd/base/topics/sxle913/fieldresponse.htm> .

CONTACT INFORMATION

Your comments and questions are values and encouraged. Contact the author at

William C. Csont
Howard M. Proskin & Associates, Inc.
300 Red Creek Dr., Suite 330
Rochester, NY 14623
Phone 585-359-2420
FAX 585-359-0465
Email wcsont@hmproskin.com
Web www.hmproskin.com

SAS and all other SAS Institute Inc. product or service names are registered trademarks or trademarks of SAS Institute Inc. in the USA and other countries. ® indicates USA registration.

Other brand and product names are registered trademarks or trademarks of their respective companies.