Paper 247-25

Preparing the SAS® Software Programming Environment for Regulatory Submission

Sunil K. Gupta, Gupta Programming, Simi Valley, CA

ABSTRACT

For pharmaceutical companies, there is a push to prepare SAS® programs for clinical study reporting and analysis. SAS® programs are set up in advance of the study completion to facilitate the generation of the final study analysis, tables, listings, and graphs. Throughout the study project, SAS software provides the key elements for successful regulatory submission.

This paper reviews the process of establishing a standard method to prepare the SAS® software programming environment (SAS/BASE®, SAS/ACCESS®, SAS/AF®, SAS/FSP®, SAS/STAT®, SAS/GRAPH®) for regulatory submissions. By incorporating best practices in the data collection and entry, data cleaning and data reporting of clinical studies, the generation of the final end result will be more efficient. In addition, because quality control is a key component throughout the system, pharmaceutical companies have a higher comfort level of the quality and completeness of the regulatory submission. The intended audience for this presentation is the intermediate to advanced SAS® user.

The paper is divided into the following sections for review: project objectives and milestones, clinical data management, system documentation and standards, the statistical analysis plan, and the regulatory submission.

PROJECT OBJECTIVES & MILESTONES

The purpose of a clinical study needs to be understood by everyone involved. The research, analysis and reporting tasks will require total involvement and understanding of everyone on the team. The team members will need to communicate their requests to each other in terms of understanding each other’s strengths and responsibilities. Typical team members include data experts such as physicians and medical writers, and programming experts such as biostatisticians and SAS programmers. The data experts know what they want and rely on the programmer to create the tables and analysis for review.

SAS programmers often have the responsibility of supporting the reporting requirements of the Clinical Affairs Department, and other departments of pharmaceutical companies. This involves interacting with department members to define the report. The SAS programmer's task is to understand the request and to design the program to achieve the desired outcome. This is often the most difficult part of the request.

A SAS Service Request Form should be developed to meet the needs of the department and reflect the type of information available for reporting. The form should be organized to guide the customer through a series of questions. Sections on the form should include who the request is from and the required date, the description of the SAS request with the purpose listed, the selection criteria to identify the population and time period of the report, and the format and organization of the report along with the method of output desired. Finally, the SAS programmer should log the completion date and time.

Proper tools such as a SAS Service Request form should be in place to facilitate the communication and documentation of the customer's requirement and the customer's expectations of the programmer. Establishing standards in service requests will increase both efficiency and customer satisfaction.

Typically, the SAS programmer will need to generate a variety of reports to fulfill the regulatory requirements of the clinical investigation. By establishing clinical reporting templates for each of the functional aspects of Clinical Data Management, the SAS programmer can dramatically improve the efficiency of developing and generating reports.

With appropriate meetings and project status updates, significant milestones can be achieved and monitored. Working closely with the FDA reviewer from the beginning to identify and plan the course of action will facilitate the project schedule and completion.

CLINICAL DATA MANAGEMENT

For large companies, another department may be responsible for the data collection and entry, data editing and data cleaning of clinical studies. Where possible, additional methods and programs should be developed to confirm the quality of the data received and analyzed. It is important to discover as soon as possible if any invalid values have been entered into the data set. In addition, statistical and clinical study assumptions may not be correct and need to be verified. At a minimum the format of coded values must be confirmed to assure correct reading of coded values. Ideally, a data validation manual should be prepared to define all data checks to be performed.

For accessing Informix or Oracle views, the SAS/ACCESS module can be used to create standard SAS views. You may also want to consider SAS/IntrNet capabilities. For using a SAS based data entry system, the following SAS modules can be utilized to create a quality control entry system: SAS/AF, SAS/FSP. Screen Control Language (SCL) allows you to add any logic and field validation for data entry.

Data Validation Plan

- Logical checks - variable level, date consistency
- Check for duplicate records
- Check for required variables
- Check for unique key variables

SYSTEM DOCUMENTATION AND STANDARDS

For each clinical study, system documentation and programming standards should be a requirement. A naming convention should be utilized for all data sets, variables, formats, macro variables and macro programs. By defining a naming system from a global perspective at the start, all documentation and program development will need minimum update at a later stage. Each programmer should have available a code book containing data set contents, sample proc prints and key to all formats.

Good directory structure to store and access data sets and programs is essential for good communication and understanding in a multi-user environment. Considerations should be made for separating raw data from SAS data sets and SAS programs from format libraries. This is a good stage to get input from all programmers and statisticians. Depending on what the FDA reviewer requests, the preferred data file structure for many data sets may be a horizontal file structure to facilitate analysis and review.

Often multi-users will be required to complete all the necessary programming in the time allocated. A central location of programs and method of access allows for a shared environment. Utility macro programs can be used to create data sets from views, provide Proc CONTENTS and sample listing, and Proc FREQ of key categorical variables. A macro library should be established. Having a good and sensible naming convention through out the process will go a long way to improve the development and maintenance of programs. By designing a modular system, code can be reused by other clinical studies with minimum effort. Time efficiency can be realized. The concept of best practices should be exercised where possible.

A single statistical analysis file can be created from all significant data sets for the study. This data set will contain all the primary and secondary measurements along with demographics and safety information. When doing integrated summary analysis, a similar setup can be utilized. By standardizing at the study level, the integration process becomes much easier. The alternative is to compare each variable for each data set across all studies to assure consistent variable name and range of values before combining all studies. By taking a systematic approach, macros can be written to standardize individual studies into common data sets to be combined into a single set of data sets. The function of these standardization macros would be to recode, rename, keep, drop and assign variables as required for each study.

Several good programming methodology and strategies include having the relative path in the libname statement to facilitate upward scalability and portability, to archive data sets as backups, and to execute SAS in batch mode to save listing and log files by the same name. Proc DATASETS with the AGE statement can be used to archive data sets as backups. Defining macro variables to be used in footnotes facilitates the program identification, execution date and the path of SAS program. Many of these things can be established in the initialization program.

Directory Structure	
Raw	SAS Data from Informix
Source	Data from other source
Data sets	Clinical SAS Data sets
Programs	Analysis Programs
Catalogs	Format Library
Dev	Program Development Directory

Macro Variables	
%let prot = AV_IS	Protocol number
%let pgm = demog	Program name
%let in_dsn = demog	Input data set
%let dsn = demog	Data set name
%let dscode = dm	Data set code
&sysdate	System date

Setup macros	
%init	Initialize program
%c_view	Create view
%c_dsn	Create data set
%c_frm	Create format library
%df_ae	Define AE evaluable population
%df_eff	Define Efficacy population

Initialize program - init.sas

```
%macro init
options pagesize = 59 linesize=130;

%let datloc = ‘c:\projectA01\data’;
%let libloc = ‘c:\projectA01\catalogs’;

libname vdat        &datloc;
libname library     &libloc;

Proc FORMAT library = library;
quit;

%include ‘df_ae.sas’;
%include ‘header.sas’;
%include ‘footer.sas’;
%include ‘line.sas’;

%mend init
```

Define AE evaluable population macro - df_ae.sas

```
%macro df_ae;

Proc SORT data=vdat.eval out=ae (keep = inv pat trtgroup);
 by pat;
 where ae = 1;
run;

%mend df_ae;
```

Standardization macros and program	
c_demog.sas	Create demog data set
%dm_atr	Define variables with attribute statements for demog data set
%dm_keep	List of variables to keep in data set
%dm_renm	List of variables to rename into

	standard variables in demog
%dm_rcod	Recode variable in demog
%ddl	Data Dictionary Listing

Create demog data set - c_demog.sas

```
%include 'init.sas';

%let pgm=demog;
%let in_dsn = demog;
%let dsn=demog;
%let dscode=dm;

data &dsn;
 %&dscode._atr;
 set vdat.&in_dsn.(rename=(%&dscode._renm));
  pat = input(patno, 4.);
 keep %&dscode._keep;
run;
```

Define variables with attribute - dm_atr.sas

```
%macro dm_atr;
 attrib pat length = 8           label='Participant Number'
 attrib site length = 8          label='Site Identifer'
 attrib aeany length=8 format=yn. label='Any Aes Occur'
%mend dm_atr;
```

List of variables to keep in data set - dm_keep.sas

```
%macro dm_keep;
 ptid prot site ptnum sex race
%mend dm_keep;
```

Validation macros

%dsn_wk	Working data set
%dsn_cn	Proc CONTENTS
%dsn_pt	Proc PRINT - sample listing
%dsn_uni	Proc UNIVARIABTE
%dsn_tab	Proc TABULATE
%dsn_frq	Proc FREQ
%dsn_dup	Check for duplicate records

Working data set - dsn_wk.sas

```
%macro dsn_wk(ds);
 Proc SORT data=vdat.&ds out=&ds;
  by pat;
 run;
%mend dsn_wk;
```

Proc Contents - dsn_cn.sas

```
%macro dsn_cn(ds);
 Proc CONTENTS data=&ds;
 run;
%mend dsn_cn;
```

Report Output Layout macros

%header	Header information
%footer	Footer information
%line	Line output
%colcnt	Column counter for put statements

Footer information - footer.sas

```
%macro footer;
 tday = date();
 put @&c1 '______________________________________________';
 put / @&c1 "Directory: &libloc. File: &pgm.  " tday
worddate.;
%mend footer;
```

STATISTICAL ANALYSIS PLAN

Creation of the statistical analysis file is a good idea to use as a single resource data set for clinical reporting and analysis. This single data set will contain all key demographic information along with all significant efficacy and safety parameters. Variables from demographic and follow-up data sets can be merged by patient and visitdate to create the single statistical analysis file.

Utilizing a consist method for generating reports methods makes good sense. Typical methods to consider include Data *null*, Proc REPORT, Proc TABULATE, and Proc SQL. The table below lists some benefits and features for each approach. A complete list and review of these reporting methods can be found in the Observations - The Technical Journal for SAS Software Users - First Quarter 1994 - Writing Reports with SAS Software. What are your options? page 10 -42.

Reporting Approach

Data _Null_ - For complete control and customized repo rts
Proc REPORT - Produces organized output for review
Proc TABULATE - Produces multi-dimensional tables with descriptive statistics in a finished tabular format
Proc SQL - Can combine data from several data sets to build the report but does not have much control over the output format

In the final step of program completion, program validation and testing should be performed for quality assurance. Test cases should be identified and tested to assure complete accountability. It is very important to first define what the expected results before the tests are performed. A key objective is to check for consistency of numbers throughout all tables. For optimal performance, it is often best for anothe r programmer to verify the program of the original programmer. A system should be defined to migrate tested and quality assured programs to the production library.

Migration Process

- Development and Testing
- Quality Assurance
- Production Library

Edit Check Listings

- Review baseline variables
- Identify any missing key variable
- Confirm dates are logical
- Descriptive statistics on all continuous variables Frequency Counts on key categorical variables

Summary Level Analysis macros

Descriptive Statistics - n, mean, sd, min, max, sum	
%prop	General Proportions macro
%cont	Continuous Variables
Statistical Analysis	
%ttest	T-test
%chist	Chi-Square test

Detail Level Analysis macros

%patlst	Patient Listings

General Proportions macro - prop.sas

```
%macro prop(dsn, labl, xvar);

 Proc MEANS data=&dsn noprint;
  by tx;
  var &xvar;
  output out=tmp1 n=n mean=mean sum=sum;
 run;

 Data _null_;
  file &outpt notitles mod ls=180;
  obsnum=1; set tmp1 point=obsnum; n1 = n; pct1 = mean*100;
            sum1=sum;
  obsnum=2; set tmp1 point=obsnum; n2 = n; pct2 = mean*100;
            sum2=sum;
  put @1 &label @17 sum1 5.  ‘ / ‘ n1 2. pct1 6.1 ‘%’
                   sum2 12. ‘ / ‘ n2 2. pct2. 6.1 ‘%’ @;
  stop;
 run;

 Proc DATASETS nolist;
  delete tmp1;
 run;

%mend prop;
```

Data Management Analysis macros

%trans	Transpose data set

Test Plan and Log

Test	Test	Expected	Observed
Item	Performed	Result	Results

Error Code	**Specification**
Demo_i005	Sex not M or F
Demo_i006	Race not W, B, O, H or X

With configuration management, system and program updates can be performed in the production environment. A good reference for documentation and software development is the “Taming the Chaos: A Primer on the Software Life Cycle and Programming Standards”. A good reference for software validation is the “Software validation for the rest of us”.

REGULATORY SUBMISSION

Ultimately, SAS data sets and files will be prepared for the FDA submission. FDA has provided updated guidelines to follow outlining the directory structure, general rules, SAS programs, SAS data sets, list and log files. In addition, tables and listings may need to be saved as Rich Text Formatting (RTF) files and html files for ease of review by Microsoft Word and a Internet Browser. For example, the general considerations for data sets include a unique identifier for each patient in the study to be included in all data sets, the variable names and codes being consistent across all studies, and the format of variables for similar types of data should also be consistent within and across studies. In addition, several key grouping variables such as treatment group and sex may be required in each data set to facilitate analysis. Where possible, macros that generate RTF files should be utilized to automate the process.

The significance of empowering the FDA reviewer with PDF files that allow for drill-down by hyper-text features and navigation tools with user instructions will greatly improve the review process. In fact, in the guidelines, it states to provide a hypertext link from the listing of the file to the SAS transport file. Study definitions, data set definitions: descriptions of data set and variable definition, program index, and format definition. If possible, write SAS programs that automatically create and index PDF files for faster processing.

As part of the FDA move toward a paperless regulatory submission, the FDA has proposed using Version 5 SAS Transport file format as a standard for electronic data submission and archival. The SAS Version 5 Transport file format provides a mechanism for movement of data between different computer types and operating systems. This ensures the long-term availability of submission data. The data set files should not exceed 25 MB per file. Each transport file should be saved as an individual file representing the SAS data set.

The following is a list of all items required for on -line documentation:

- Study (protocol) Definitions
- User Instructions
- File Definitions
- Variable Definitions
- Data set List
- Program List
- Listings List
- Tables List
- Outputs - Listings and Tables

Study Definition

Protocol	**Description of Study**
DRUG001	Safety
DRUG002	Phase 2 double blind, randomized, placebo-controlled, dose escalation study

File Definition

Data set Name	**#of obs**	**Filename Description**
AE2 non-	5833	The AE2 data set contains all missing adverse event records with selected demographic and

treatment group variables. It has one record per participate per adverse event.

Data set AE2

Description:
The AE2 data set contains all non-missing adverse event records with selected demographic and treatment group variables. It has one record per participate per adverse event.

Variable Definition

Variable	Sort	Length	Format	Label Description
Ptid	1	$ 13	.	Participant ID
Aeact	2	8	Aeact.	Action Taken
Aeany		8	Yn.	Any AE Occur

Formats

AEACT	YN
1='None'	1='Yes'
2='Discontinued'	2='No'

Notes
1. Sort Keys: Ptid, Aeact.
2. The COSTART dictionary was used for the AE preferred terms and the AE body system terms.

SUMMARY

Before starting to write individual programs to produce output for regulatory submission, it is wise to take a systematic global perspective of all the resources available and the best strategy to achieve the outcome. Advance preparation of SAS® programs in Clinical Data Management, statistical analysis and quality control will facilitate the generation of all required output files. By including everyone from the beginning, a greater understanding of the submission objectives can be achieved.

TRADEMARK INFORMATION

SAS® is a registered trademark of the SAS Institute Inc., Cary, NC, USA.

http:/www.sas/service/techsup/sas_xport.html

Gupta, Sunil K., Gupta Programming (1995), "Designing Clinical SAS Service Request Forms", WUSS.

Gupta, Sunil K., Gupta Programming (1995), "Utilizing Clinical SAS Report Templates", WUSS.

Gupta, Sunil K., Gupta Programming (1996), "Database Design Strategies in CANDAs", PharmaSUG.

Carol Linden and John E. Green III, (1994), "Writing Reports with SAS Software. What are your options?" Observations: The Technical Journal for SAS Software Users - First Quarter, 10-42.

C. Michael Whitney (1996) "Taming the Chaos: A Primer on the Software Life Cycle and Programming Standards", Observations: The Technical Journal for SAS Software Users - Fourth Quarter, 15-21.

Harris, Michael, Amgen Inc. (1998), "Software validation for the rest of Us", WUSS.

Iza Peszek, Cindy Song, Olga Kuznetsova, Merck & Co (1999), "Producing Tabular Reports in SAS® Systems in the Form of MS Word® Tables", PharmaSUG.

Guidance for Industry - Providing Regulatory Submissions in Electronic Format - NDA, U.S. Department of Health and Human Services, Food and Drug Administration, Center for Drug Evaluation and Research (CDER), IT 3 January 1999

ABOUT THE AUTHOR

The author welcomes your comments & suggestions.

Sunil K. Gupta
Gupta Programming
SAS Institute Quality Partner™
http://www.GuptaProgramming.com
Phone: (805)-577-8877
E-mail: Sunil@GuptaProgramming.com

Sunil is a senior consultant at Gupta Programming. He specializes in SAS/BASE®, SAS/AF®, SAS/FSP®, SAS/STAT® and SAS/GRAPH®. His consulting projects with pharmaceutical companies include the development of a Clinical Study Data Entry System, a Macro-Based Application for Report Generation, and customized plots and charts with SAS/GRAPH®. He has been using SAS® software for over 9 years and is a SAS Institute Quality Partner™.