A Simplified Approach Using Macro Techniques to Create a Clinical Data Reporting System

Mark Parra
Organon Inc., West Orange, N.J.

Abstract

In the pharmaceutical industry today expressions like "Timeline Compression" and "Process Facilitation" are common and reflect the industry goals of producing faster New Drug Applications (NDAs). Reporting of clinical trial data is essential and the ever increasing demands for faster report generation remains a high priority in the NDA process.

This paper describes a simplified approach to creating a clinical data reporting system using macro techniques which not only function in producing different reports, but by using %Window also menu drives the system. Preparation of the SAS data sets as well as the relationship between clinical trial structures and the report format types are explored. The flexibility of report modules being processed through a single Data_Null_step is demonstrated, and finally the macro %Window code which produces the menu-driven front-end is examined.

The basic concepts discussed in this paper provide a framework from which elaborate systems can be developed. This simplified approach to creating a clinical data reporting system utilized SAS macro techniques producing benefits in the reduction of effort, error, code, and most importantly time in the processing of similar clinical trial data in order to meet the future demand for streamlining report generation.

Introduction

The process by which a new drug is created, studied and finally marketed is a complex one. From the time an Investigational New Drug(IND) application is submitted to the Food and Drug Administration(FDA), all phases of the clinical trial are complete, and an NDA is prepared for submission a large portion of a drug's patent life span is over. Usually most of a company's profits from drug sales are made in the period between FDA approval and the expiration of a drug's patent. In many cases generics diminish a drug's market share thereby diminishing profits. In order to reap the greatest profits companies are looking for any method to shorten the NDA process of which clinical data reporting constitutes a large part.

The power and flexibility of SAS software offers a variety of reporting procedures such as Proc Print, Proc Tabulate, and most recently Proc Report which can be used in report generation. The Clinical Data Reporting System utilized the Data_Null_step to produce reports due to the greater control which could be attained over report production and the ability to use special variables to control what happened during execution. The macro language provided a menu-driven front-end which allowed for a selection of reports and variability in the size of the reports. The construction of this system will be discussed in the following sections.

The Clinical Data Reporting System Construction

The following phases chronicled the steps involved in creating this system. Keep in mind that the person invoking this system had prior knowledge of the study, center, and patient numbers.

Phase 1: The Clinical System SAS Data Sets

The data collection form (e.g. Case Report Form) was examined for data groupings and their corresponding schedules (see Figure 1).

![Figure 1](image-url)

Using this information data set structures were identified. This system presents three data set structures and can be described as single time point, multiple time point and data lists. An example of each can be seen in the cards statement of the Data steps called CRF1, CRF2, and CRF3, respectively, from the program called MAIN.SAS. For the sake of demonstration these simplified and fictitious data sets were created at the time of execution; however they could have been created and stored prior to execution. It was necessary at this point to design report layouts and to decide in what order the reports would be displayed in a report package. Each report was assigned a number from 1 to N number of reports sequentially. The Demographic data would comprise the first report followed by the Vital Signs data for the second report and finally the...
Adverse Experience data for the third report. Two control variables were added to each data set. The first control variable identified the number of the report module which the data set supported and was called RMOD. This was a numeric variable which was assigned the value of the report number previously determined. The second control variable provided an increment reflecting a measure of time and was called CTR. This was a numeric variable which was assigned a value of one for single time point and data list assessments. For the multiple time point assessment this variable was assigned a value of one for the first occurrence with an increment of one for every occurrence thereafter.

It was necessary to establish a naming convention for these data sets consisting of the same name with a numeric suffix corresponding to the report number the data set supported. At the completion of this phase the same number of data sets would be created as the number of reports in the display package.

Phase 2: The Clinical System Report Modules

This system presents three simplified report layouts comprising the report package and can be described as free format, multiple column, and multiple row reports (see Figures 2-4).

Using the report package as a reference the appropriate Put statements were written for each module to produce the desired reports. These modules had their own headers with unique header labels. The first two statements in all the modules used the control variable RMOD to advance the page and print the report header. The multiple column report used the control variable CTR to establish the place on the page where each data column would be written out to. It was necessary to establish a naming convention for these external report modules consisting of the same name with a numeric suffix corresponding to its report number. At the completion of this phase the same number of report modules were created as the number of reports in the display package.

Phase 3: The Clinical System Main Program

The following describes the Data steps and statements making up the program which the menu-driven front-end accessed and executed to produce the reports. For the purpose of demonstration the Clinical SAS data sets were created within this program. If the data sets were created externally they would have to be made available (e.g. Libname statement with a Data step and Set statement or Proc Sort statement with OUT= option). The first three Data steps to this program created the data sets CRF1, CRF2, and CRF3 using Input and Cards statements. At this point a macro called CRF was invoked. The purpose of this macro was to determine which report(s) were requested based on the menu response and to create a macro variable called CRF which would have the value of the name(s) of the SAS data set(s) that would support the report(s) requested. This was accomplished by iterative and Conditional Do statements both using the macro variable RESP. The macro variable RESP takes its value from the menu response and will be discussed further in Phase 4. Additionally in this macro if no reports were selected from the menu the program execution would end and a message in the WINDOWS LOG file would reflect this choice. If at least one report was selected from the menu then the next Data step would execute creating the data set CRFCHK. This data set would include the data set(s) assigned in the macro variable CRF. At this point the macro called CHECK would be invoked. The purpose of this macro was to check the data entry response of study and patient or center numbers as requested from the menu against what study and patient or center numbers were
part of the data set(s) available to the program based on the report(s) requested. This was accomplished by creating three macro variables called STIDLIST, PATLIST, CTRLIST which would have the value of a string list of all study, patient, and center numbers, respectively, from the data set(s) included. If the study, patient or center numbers provided were not valid the execution of the program would end and a message in the WINDOWS.LOG file would reflect that. If the study, patient, or center numbers provided were valid the data set CRFCHK was sorted by the variable RMOD. The next Data step created a data set CRFDIS which would hold a subset of the data set CRFCHK. This subset was dependent upon whether a report was to be run by center or patient and was accomplished utilizing a Where statement. One Where statement would be executed and the other Where statement would be commented out by the value assigned to the macro variables QPAT and QCTR. Included in this Data step was an increment of the number of reports requested and the total was placed in a macro variable called N. Additional macro variables were created dependent upon the number of reports requested (one per report). These macro variables had the naming convention of a prefix RP concatenated with the increment of the number of the report requested (e.g. the first report requested = RP1). The value of these macro variables would equal the value of RMOD. For example if the first and only report requested was Adverse Experiences (RMOD=3) the macro variable created would be RP1 with a value of 3. The CRFDIS data set was sorted by study number, center number, patient number, RMOD, and CTR. The next Data step was the Data _Null_ and it was set with the data set CRFDIS. The File statement with the fileref of Print and the Page number, center number, patient number, RMOD, and CTR would occur. The active Data set was dependent upon whether a study, patient, or center numbers provided were valid that number, center number, patient number, RMOD, and CTR, The next Data step was the Data _Null_ and it was set with the data set CRFDIS. The File statement with the fileref of Print and the Page number, center number, patient number, RMOD, and CTR would occur. The active Data set was dependent upon whether a study, patient, or center numbers provided were valid the value assigned to the macro variable QPAT was assigned the value of an asterisk. After either choice 1 or 2 completed its execution and dependent upon valid responses the main menu would be displayed again. If the choice was 3 (leave the menu) the program would end execution. If any other response to Choice was provided a system message would display the appropriate responses available. This program generated a WINDOWS.LOG file with the program notes as usual. The WINDOWS.USR file contained the reports requested. At the completion of this phase the program was written and placed in a file called MAIN.SAS.

**Phase 4: The Clinical System Menu-Driven Front-End**

This is the program that invoked the system. The following describes the program statements making up the menu-driven front-end. They consisted of a set of %Window statements and other macro statements which displayed the menus which solicited the necessary information for the generation of the reports. It also accessed and executed the main program which produced the reports. The main menu consisted of a %Window statement soliciting study, patient, and page numbers. Those responses were assigned in the macro variables MSTD, MPAT, and PG, respectively. The center sub-menu consisted of a %Window statement soliciting study, patient, and page numbers. Those responses were assigned in the macro variables MSTD, MPAT, and PG, respectively. The report sub-menu consisted of a %Window statement and solicited a Y or N (YES or NO) response to each of the reports (Demographics, Vital Signs, Adverse Experiences). Those responses were assigned in the macro variables MODEM, MVT, and MAE, respectively. The process display consisted of a %Window statement and was active during processing. When this program was executed the macro Generate was invoked. The purpose of this macro was to display the menus, process the menu responses and execute the appropriate statements. A Do Until statement checked the value of the Choice variable and allowed looping until the value of Choice was to leave the menu (3). Initially the Choice variable had a null value and so the main menu was displayed. If the choice was 1 (run by patient) then the following statements executed. The macro variable QCTR was assigned the value of an asterisk. Several other macro variables were initialized to null. The sub-menus for patient and report were displayed. A macro variable RESP was created which had the value of the responses from which reports should be generated. That value was the concatenated values of the macro variables MODEM, MVT, and MAE, in that order. For example if all reports were requested then the value of RESP would be YYY. At this point the process window would be displayed and at the hitting of the enter/return key the program accessed and executed the main program producing the requested patient report(s). If the choice was 2 (run by center) similar statements would execute with one difference. The macro variable QPAT was assigned the value of an asterisk. After either choice 1 or 2 completed its execution and dependent upon valid responses the main menu would be displayed again. If the choice was 3 (leave the menu) the program would end execution. If any other response to Choice was provided a system message would display the appropriate responses available. This program generated a WINDOWS.LOG file with the program notes as usual. The WINDOWS.USR file contained the reports requested. At the completion of this phase the program was written and placed in a file called MAIN.SAS.

**Conclusion**

This paper has presented a simplified approach to the construction of a clinical data reporting system using macro techniques. Although the four phases were presented sequentially the actual process did overlap. Through the use of a consistent naming convention and sequential numeric ordering, data sets and reports were linked throughout this system. The data sets and reports were simplified for the purpose of demonstration, but it can be seen that much more sophisticated data sets and reports could have been incorporated using this approach. This system eliminated the redundancy of individual programs that would have been needed to produce the same type of reports across similar studies. It also permitted the programmer to step away from the process by allowing the data to drive the report generation. It is hoped that this system and others like it will help meet the future demand for streamlining report generation.

SAS is a registered trademark or trademark 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.

**PROGRAM NAME: WINDOWS.SAS**
**PURPOSE: TO PROVIDE A CLINICAL DATA REPORTING SYSTEM**
**AUTHOR: MARK PARRA**
**DATE: SUMMER 1995**
**NECESSARY MODULES: MAIN.SAS, MOD1.SAS, MOD2.SAS, MOD3.SAS**

FILENAME CODE "YOUR DIRECTORY";

%* MAIN MENU WINDOW ;
%WINDOW MENU
#1 @<address> SYSDATE ? PROTECT=NOT
#2 @<address> SYSDAY 9 PROTECT=NOT
#3 @<address> SYSTIME 5 PROTECT=NOT
#6 @26 "CLINICAL DATA REPORTING SYSTEM"
#7 @26 'MAIN MENU'
#11 @26 '1' +1 'RUN BY PATIENT'
#13 @26 '2' +1 'RUN BY CENTER'
#15 @26 '3' +1 'LEAVE MENU'
#19 @26 'CHOICE 1 ATTR=UNDERLINE REQUIRED=YES AUTOSKIP=NO' +1 ' ' <= == PLEASE ENTER YOUR CHOICE' +1 ' ' AND HIT THE ENTER/RETURN KEY';
#20 @28 'AND HIT THE ENTER/RETURN KEY';

* MENU CHOICE 1 WINDOW;
* WINDOW PATIENT
#1 @28 'SYSDATE 7 PROTECT=YES' +1 'E.G. 7'
#2 @28 'SYSDAY 9 PROTECT=YES'
#3 @28 'SYSTIME 5 PROTECT=YES'
#5 @28 'CLINICAL DATA REPORTING SYSTEM'
#7 @26 'PATIENT SUB-MENU'
#9 @16 'PLEASE SUPPLY THE STUDY NUMBER:'
#11 @16 'MSTUDY 1 ATTR=UNDERLINE REQUIRED=YES (E.G. 7)' +1 '+1 (E.G. 7)'
#13 @16 'PLEASE SUPPLY THE PATIENT NUMBER:'
#15 @16 'MPAT 1 ATTR=UNDERLINE REQUIRED=YES (E.G. 1)'
#1 @28 'PLEASE SUPPLY THE STARTING PAGE NUMBER:'
#3 @28 'PG 5 ATTR=UNDERLINE REQUIRED=YES (E.G. 0)'
#5 @16 'PLEASE HIT THE ENTER/RETURN KEY';

* MENU CHOICE 2 WINDOW;
* WINDOW CENTER
#1 @28 'SYSDATE 7 PROTECT=YES'
#2 @28 'SYSDAY 9 PROTECT=YES'
#3 @28 'SYSTIME 5 PROTECT=YES'
#5 @28 'CLINICAL DATA REPORTING SYSTEM'
#7 @26 'CENTER SUB-MENU'
#9 @16 'PLEASE SUPPLY THE STUDY NUMBER:'
#11 @16 'MSTUDY 1 ATTR=UNDERLINE REQUIRED=YES (E.G. 7)' +1 '+1 (E.G. 7)'
#13 @16 'PLEASE SUPPLY THE CENTER NUMBER:'
#15 @16 'MCENTER 1 ATTR=UNDERLINE REQUIRED=YES (E.G. 1)'
#1 @28 'PLEASE SUPPLY THE STARTING PAGE NUMBER:'
#3 @28 'PG 5 ATTR=UNDERLINE REQUIRED=YES (E.G. 0)'
#5 @16 'PLEASE HIT THE ENTER/RETURN KEY';

* SUB MENU REPORT;
* WINDOW REPORT
#1 @28 'SYSDATE 7 PROTECT=YES'
#2 @28 'SYSDAY 9 PROTECT=YES'
#3 @28 'SYSTIME 5 PROTECT=YES'
#5 @28 'CLINICAL DATA REPORTING SYSTEM'
#7 @26 'REPORT SUB-MENU'
#9 @16 'PLEASE RESPOND YES OR NO TO SELECT EACH REPORT'
#11 @16 'DEMOGRAPHICS:'
#13 @16 'VITAL SIGNS:'
#15 @16 'ADVERSE EXPERIENCES:'
#17 @16 'PLEASE HIT THE ENTER/RETURN KEY';

* WINDOW PROCESS
#1 @28 'SYSDATE 7 PROTECT=YES'
#2 @28 'SYSDAY 9 PROTECT=YES'
#3 @28 'SYSTIME 5 PROTECT=YES'
#5 @28 'CLINICAL DATA REPORTING SYSTEM'
#7 @16 'PRESS THE ENTER/RETURN KEY TO CONTINUE'
#11 @16 'YOUR REQUEST IS BEING PROCESSED' ATTR=BLINK;

% MACRO GENERATE;
% LOCAL CHOICE MDEM MVIT MAE:
% DD %UNTIL [(@CHOICE=3)]
% LET CHOICE=
% DISPLAY MENU BLANK BELL
% IF @CHOICE=1 THEN %DO
% LET MSTDY=
% LET PG=
% LET MDEM=
% LET MVIT=
% LET MAE=
% LET MPAT=
% LET QPAT=
% LET QCENTER=
% DISPLAY PATIENT BLANK BELL
% DISPLAY REPORT BLANK BELL
% LET RESP=%UPCASE(MDEM&MVIT&MAE):
% DISPLAY PROCESS BELL
% IF @MPAT NE %THEN %STR(INC CODE(MAIN));
% END;
% ELSE %IF @CHOICE=2 THEN %DO
% LET MSTDY=
% LET PG=
% LET MDEM=
% LET MVIT=
% LET MAE=
% LET QPAT=
% END;
% ELSE %IF @CHOICE=3 THEN %DO
% END;
% ELSE %LET SYSMSG=PLEASE ENTER 1, 2, OR 3:
%END;
%END GENERATE;

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
** PROGRAM NAME: MAIN.SAS
** PURPOSE: TO PROVIDE A CHECK OF MENU RESPONSES AND PRODUCE THE REQUESTED REPORTS
** AUTHOR: MARK PARRA
** DATE: SUMMER 1995
** NECESSARY MODULES: MOD1.SAS, MOD2.SAS, MOD3.SAS
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

FILENAME CODE YOUR DIRECTORY:

%MACRO CHECK(DELIM=);%LET PATLIST=;%LET CTRLLIST=;%LET STUDYLIST=
PROC SORT DATA=CRFCHK(KEEP=PAT CENTER STUDY) OUT=UNIQ NODUPKEY;
BY STUDY CENTER PAT;
DATA_NULL;
SET UNIQ;

995
DATA CRF2:
  INPUT STUDY CENTER PAT TX $ CTR VISIT VDATE MMDDYYV6.
  BPSYS BP Dia PULSE HGT WGT RMOD ;
  CARDS ;
  1 1 1 A 1 1 060394 122 82 76 70 165 2
  1 1 1 A 2 2 070694 124 84 79 70 162 2 ;
  RUN ;

DATA CRF3:
  INPUT STUDY CENTER PAT TX $ CTR AE $ SEV $ REL $ ACT $ RMOD ;
  CARDS ;
  1 1 1 A 1 HEADACHE MILD NONE NONE 3
  1 1 1 A 1 NAUSEA MILD POSSIBLE REDUCED 3
  1 1 1 A 1 VOMITING MILD POSSIBLE REDUCED 3 ;
  RUN ;

%CRF
  DATA CRFCHK ;
  SET &CRFL ;
  RUN ;

%CHECK
  PROC SORT DATA=CRFCHK ;
  BY RMOD ;
  DATA CRFDIS ;
  SET CRFCHK END=LAST ;
  BY RMOD ;
  &OPAT WHERE PAT~&MPAT AND STUDY=&MSTUDY ;
  &OCENTER WHERE CENTER~&MCENTER AND STUDY=&MSTUDY ;
  IF FIRST. RMOD THEN DO ;
  RCTR+1 ;
  CALL SYMPUT('RMOD')||LEFT(RCTR),LEFT(TRIM(RMOD)) ;
  END ;
  IF LAST THEN CALL SYMPUT(N',LEFT(RCTR)) ;
  RUN ;

PROC SORT DATA=CRFDIS ;
  BY STUDY CENTER PAT RMOD CTR ;
  DATA _NULL_ ;
  SET CRFDIS ;
  BY STUDY CENTER PAT RMOD CTR ;
  FILE PRINT NOTITLE N=PS PS=64 ;
  RETAIN PAGE &PG ;

%PRRP

--------------------------------------------------------------------------------------------------
** PROGRAM: MOD1.SAS **
** PURPOSE: TO PROVIDE THE **
** DEMOGRAPHICS REPORT MODULE **
** AUTHOR: MARK PARRA **
** DATE: SUMMER 1995 **
--------------------------------------------------------------------------------------------------

IF FIRST. RMOD THEN PUT _PAGE_ ;
IF FIRST. RMOD THEN LINK H1 ;
PUT &14 @1 'DEMOGRAPHICS - FREE FORMAT REPORT,' &15 @1 ;
&16 @2 'DATE OF BIRTH(MM/DD/YY):' DOB MMDDYYV6.