Paper PO-09

Proc Report Data = Subject.Event_Chronology;

Christina Carty and Elizabeth R. Spence
VA Cooperative Studies Program Coordinating Center, Perry Point, Maryland

ABSTRACT

Data management staff at the VA Cooperative Studies Program Coordinating Center located in Perry Point, Maryland was challenged with the task of generating a report to display in chronological order selected data points collected across 10 case report forms. The information was used to determine if myocardial infarction (MI) or other cardiac events occurred for a given subject. Because the classification results play an essential role in the final analysis reporting for the study, it was very important that the data be conveyed accurately in a user-friendly fashion; not an easy task for a large number of variables and two thousand subjects!

Since the chronology lists were produced during the ongoing recruitment phase of the study, a mechanism was employed to screen each subject's data to assure all required records were received, and no queries were left unresolved. Required values were extracted from each form data set and manipulated where necessary, then compiled into a master database containing multiple observations for each subject. Finally, a complete chronology report was created and sent to an outside lab for MI classification as auditing was performed simultaneously to track IDs throughout the process.

This paper will demonstrate how Base SAS® was used to filter and manipulate the data. We will then show how PROC REPORT and the Output Delivery System (ODS) were utilized to format and display the data in an easy-to-read Microsoft Word document.

INTRODUCTION

The Department of Veterans Affairs Cooperative Studies Program Coordinating Center (CSPCC) at the Perry Point, Maryland VA Medical Center is an administrative, data management, and statistical coordinating center for both VA and non-VA medical research clinical trials (studies).

Utilizing SAS version 9.1 for Microsoft Windows members of the Perry Point data management staff designed and developed a detailed event chronology report for use by a core ECG Lab when classifying myocardial infarction (MI) and other cardiac events for 2,013 subjects enrolled in a study. The information used for the report was recorded on ten case report forms and included a variety of relevant time points such as randomization date, surgery date, ECG dates, Troponin draw dates and values, cardiac serum marker results, discharge date, death date, adverse event data and acute coronary event information. Other required data included de-identified personal information such as date of birth, gender and status at discharge.

Prior to generating the report, the coordinating center's existing editing system was utilized to assure data integrity of the individual case report forms. Once records were determined to be cleaned they were released and all required data points were assembled in a master data set. The various steps described below were performed in order, to exclude dirty records, organize needed data points and produce the final report.

STEP 1

The first step required examination of each form data set to determine if any records were flagged for re-edit because of unresolved queries on the quality control report. If a record was awaiting clarification from the site it could not be included in the chronology report. The output from this step was an "errors" work data set containing bad records that would be combined with other exclusion data sets in the final stage to exclude subjects from appearing on the report.

```
data errors (drop=reeddate);
set in.form00 (keep=center subject reeddate where=(reeddate > 0))
    in.form02 (keep=center subject reeddate where=(reeddate > 0))
    in.form06 (keep=center subject reeddate where=(reeddate > 0))
    in.form07 (keep=center subject reeddate where=(reeddate > 0))
    in.form08 (keep=center subject reeddate where=(reeddate > 0))
    in.form81 (keep=center subject reeddate where=(reeddate > 0))
    in.form09 (keep=center subject reeddate where=(reeddate > 0))
    in.form12 (keep=center subject reeddate where=(reeddate > 0))
    in.form13 (keep=center subject reeddate where=(reeddate > 0))
```

```
    in.form15 (keep=center patient reeddate where=(reeddate > 0)) ;
by center subject;
if first.center or first.subject then output;
run;
```

STEP 2

In this step we cycled through the individual form data sets to collect critical data points needed for the chronology such as randomization date and time; surgery date and time; ECG dates; Troponin draw dates and values; cardiac serum marker dates, times and results; discharge date; death date; adverse event data and acute coronary information. Since some forms included records with more than one event per subject it was necessary to manipulate those records so each event was output as a single observation. As new individual work data sets were generated, an event code was assigned to each event. Additional generic variables were added to store existing values for dates, times and assay results. The code below was used to generate the work data sets containing the needed variables.

```
**** Randomization ****;
data random (keep = center subject ran_date ran_time date time event);
set in.random;
format date mmddyy10.;
format time time5.;
date = ran_date;
time = ran_time;
event = 6;

**** Form 00 ****;
data fm00 (keep = center subject pid letcode age gender);
set in.form00 (where=(reeddate = . ));
run;

**** Form 02 ****;
data fm02 (keep = center subject pid letcode surg_dt
                  start_hr startmn end_hr endmn end_time date time event);
set in.form02 (where=(reeddate = . ));

format date mmddyy10.;
date = surg_dt;
time = hms(start_hr,startmn,0);
end_time = hms(end_hr,endmn,0);
event = 3;
run;

**** Form 07 ****;
data count7 (keep=center subject count );
set in.form07;
by center subject;
retain count;

/* Count samples received: 4 required for each subject. */
if first.subject then count=0;
   if period = 1 then count + 1;
   if period = 2 then count + 1;
   if period = 3 then count + 1;
   if period = 4 then count + 1;
if last.subject then output;
run;

data form07;
merge in.form07 count7 ;
by center subject;
run;

data fm07 (keep = center subject pid letcode period draw_dt blddraw ba_ctni
                  status date time event );
set form07 (where=(reeddate = . and count=4));

format date mmddyy10.;
```

```
date = draw_dt;
time = . ;              /* Time not collected on this form */

if period = 1 then event = 2;
if period = 2 then event = 5;
if period = 3 then event = 7;
if period = 4 then event = 9;
run;
```

Form 09 captured discharge data. Since some subjects remained in hospital after the study endpoint (30 days) it was necessary to calculate a temporary discharge date by adding 30 to the randomization date.

```
**** Form 09 ****;
data fm09_rand (keep = center subject pid letcode dis_dt statdls ror rorrmk
                       ran_date ran_time rand30 reeddate event);
merge in.form09 (in=a) in.random (in=b);
by center subject;

format rand30 mmddyy10.;
rand30 = ran_date + 30;
if a and b then output;
run;

data fm09 (keep = center subject pid letcode dis_dt statdls ror rorrmk
                  date time rand30 event);
set fm09_rand (where=(reeddate = . and (dis_dt > 0 or statdls = 2)));

format date mmddyy10.;
time = . ;               /* Time not collected on this form */

if dis_dt > 0 then do;
   date = dis_dt ;
   event = 18;
end;

else if dis_dt = . and statdls = 2 then do;
   date = rand30 ;       /* Use calculated date for discharge */
   event = 19;
end;
run;

**** Form 12 ****;
data fm12 (keep = center subject pid letcode dead_dt deaddtuk autopsy
                  date time event);
set in.form12 (where=(reeddate = . ));

format date mmddyy10.;

date = dead_dt;
time = . ;              /* Time not collected on this form */

if autopsy = 0 then event = 16;
if autopsy = 1 then event = 17;
run;
```

Five forms contained records with more than one event per subject. They had to be manipulated to create one observation for each event. Arrays and do loops were used to complete the task.

Three of these forms (06, 08 and 81) contained data for ECGs performed and cardiac serum marker results. The basic code below was used to gather the data points needed for the report. Minor modifications were applied for the individual forms to identify visits and capture other form specific items.

```
**** Form 08 ****;
data temp08 (keep = center subject pid letcode ckuln ckunit tropuln tropunit
                    troptype tropdxl ck trop ecg date time event);
set in.form08 (where=(reeddate = .));
```

```
by center subject;

format ck 9.1;
format trop 8.2;
format date mmddyy10.;
format time time5.;

time_3=hms(sm3_hr,sm3mn,0);
time_4=hms(sm4_hr,sm4mn,0);
time_5=hms(sm5_hr,sm5mn,0);
time_6=hms(sm6_hr,sm6mn,0);
time_7=hms(sm7_hr,sm7mn,0);
time_8=hms(sm8_hr,sm8mn,0);

array date_{6} sm3_dt sm4_dt sm5_dt sm6_dt sm7_dt sm8_dt;
array time_{6} time_3 time_4 time_5 time_6 time_7 time_8;
array ck_{6} ck3 ck4 ck5 ck6 ck7 ck8;
array ckna_{6} ck3na ck4na ck5na ck6na ck7na ck8na;
array trop_{6} trop3 trop4 trop5 trop6 trop7 trop8;
array tropna_{6} trop3na trop4na trop5na trop6na trop7na trop8na;
array ecg_{6} ecg3 ecg4 ecg5 ecg6 ecg7 ecg8;

do i = 1 to 6;
   if (ck_{i} > . and ckna_{i} ne '99') or trop_{i} > . and tropna_{i} ne '99')
   or ecg_{i}= 1 then do;
      date = date_{i};
      time = time_{i};
      ck = ck_{i};
      trop = trop_{i};
      ecg = ecg_{i};
      row + 1;
      if date  = . then delete;         /* Remove obs with no date */
      output;
   end;
end;

run;

data fm08;
length ck_uln trop_uln ck_val trop_val $15;
set temp08;

event = 10;
ck_uln = ' ';
trop_uln = ' ';
ck_val = ' ';
trop_val = ' ';

if ckuln > . and ckunit > . then do;
   ck_uln = put(ckuln,9.1) || ' ' || put(ckunit,ck.);
   ck_uln = trim(left(ck_uln));
end;
if tropuln > . and tropunit > . then do;
   trop_uln = put(tropuln,8.2) || ' ' || put(tropunit,trop.);
   trop_uln = trim(left(trop_uln));
end;
if ck > . then do;
   ck_val = put(ck,9.1) || ' ' || put(ckunit,ck.);
   ck_val = trim(left(ck_val));
end;
if trop > . then do;
   trop_val = put(trop,8.2) || ' ' || put(tropunit,trop.);
   trop_val = trim(left(trop_val));
end;
run;
```

Form 13 was used to capture serious adverse event (SAE) data. Since events that occurred after discharge could not be included in the report it was necessary to check discharge status and the event dates against the discharge date recorded on Form 09. Form 15 (not shown) was used to record acute coronary event (ACE) data and is similar to Form 13, so the code below was used twice. Only a few modifications were needed to identify visits and collect relevant data points.

```
**** Form 13 ****;
data disch (keep = center subject pid letcode dis_dt statdls ran_date
                   rand30 reeddate );
length ae_remrk $30;
set fm09_rand (where=(reeddate = . and (dis_dt > 0 or statdls = 2)));
run;

data fm13 (keep = center subject pid letcode seqno ae_dt aedtuk ae_remrk
                  aeded_dt aedeaduk date time event statdls dis_dt rand30);
merge disch in.form13
            (in=a where=(reeddate = . and ae_dt > 0 and aedtuk NE 88 ))  ;
by center subject ;

format date mmddyy10.;
time = .;                  /* Time not collected on this form */
ae_remrk = ' ';
ae_remrk = trim(left(aermk1)) || ' ' || trim(left(aermk2));

if a and dis_dt > 0 and ae_dt LE dis_dt then do;
   date1 = ae_dt;
   if aedeaduk > . then date2 = . ;
   else date2 = aeded_dt;

   array date_{2} date1 date2;
   do i = 1 to 2;
      date = date_{i};
      if i = 1 then event = 14;
      if i = 2 and date > 0 then event = 15;
      if date = . then delete;
      output;
   end;
end;

else do;
   if a and dis_dt = . and statdls = 2 and ae_dt LE rand30 then do;
      date1 = ae_dt;
      if aedeaduk > . then date2 = . ;
      else date2 = aeded_dt;

      array date_2{2} date1 date2;
      do i = 1 to 2;
         date = date_2{i};
         if i = 1 then event = 14;
         if i = 2 and date > 0 then event = 15;
         if date = . then delete;
         output;
      end;
   end;
end;
run;
```

STEP 3

Once the new form data sets were ready, several additional DATA steps were executed to further filter the data and combine it into new data sets. Finally, after all the needed elements were collected a mass merge was performed to combine the information into a master data set to use for the report.

The first step combined header data (age, gender and various dates) into one file that would be used later to build the master.

```
data one;
```

```
merge random (in=a drop=date time event) fm00 (in=b) fm09 (in=c drop=date time
event);
by center subject;
if a and b and c then output;
run;
```

The next step was executed on four of the required forms and created data sets with one observation for each subject ID. Note: XX = form number.

```
data checkXX (keep = center subject);
set fmXX;
by center subject;
if first.center or first.subject then output;
run;
```

The check data sets created above were merged with data set one to create a file containing all subject numbers (allpts). Data set two becomes the first version of the master data set.

```
data allpts (keep=center subject ) two;
merge  one (in=a) check2 (in=b) check6 (in=c) check7 (in=d) check9(in=e);
by center subject;
if a and b and c and d and e then do;
   if a then output two;
   output allpts;
end;
run;
```

Data set one is then merged with each form file to store subject identifiers for display in the report. The new files were then merged with allpts to gather available data for all subjects. The code displayed below reflects execution of these steps against the randomization file. Form 07 required additional modifications in order to create an exclusion data set containing records where Troponin specimens were not obtained or assay values were not available (noresult).

```
data rand_one (drop=ran_date ran_time dis_dt rorrmk ror statdls );
merge one (in=a) random (in=b);
by center subject;
if a and b then output;
run;

data chkrand (drop=age sex);
merge allpts (in=a) rand_one (in=b);
by center subject;
if a and b then output;
run;

data check7b chkrslt;
merge allpts (in=a) fm07(in=b);
by center subject;
   if blddraw = 0 or (blddraw = 1 and ba_ctni = . and status ne 7) then do;
      output chkrslt;
      delete;
   end;
   else if a and b then output check7b;
run;


data noresult (keep=center subject);
set chkrslt;
by center subject;
if first.center or first.subject then output;
run;
```

The various chk data sets created above were set together to get all form data into one file.

```
data tempa ;
set chkrand check2b check6b check7b check8b check81b check9b
    check12b check13b check15b ;
```

```
by center subject date time;
run;
```

The tempa data set is merged with data set two (master) to combine form data with header data. Note that the new data set (tempb) included subjects that may have had “as needed” forms with unresolved queries (they will be removed below). Data set tempb becomes a new version of the master.

```
data tempb;
merge tempa two  ;
by center subject;
run;
```

The next two steps create new exclusion data sets. Data set allbad is a combination of records flagged with unresolved queries (errors), and subjects where Troponin specimens were not obtained or assay values were not available (noresult). Data set exclude combines the allbad records with subjects that appeared in a previous report (good_pts).

```
data allbad;
merge errors noresult ;
by center subject;
run;

data exclude;
merge allbad in.good_pts (drop=pid letcode);
by center subject;
run;
```

The last step merges the master with the exclusion file and a final version of each is stored.

```
data out.final out.exlude;
merge tempb (in=a) exclude (in=b);
by center subject;

if b then output out.exclude;
else output out.final;
run;
```

STEP 4

As part of the tracking process, internal flag variables were created and used as a mechanism to eliminate the possibility of sending a subject’s chronology more than once. The code below was used to perform this task. The auditing process also included tracking delivery and receipt of classification results. The results file was received from the core ECG lab throughout the course of the study and was maintained in a separate database which was used by statistical programmers during final analysis. The description of those tasks is beyond the scope of this paper and will not be included.

.

```
data new_good_pts (keep=center subject pid letcode date_sent);
set in.final;
by center subject;
date_sent = today();
if first.center or first.subject then output;
run;

data out.good_pts (keep=center subject pid letcode date_sent);
merge new_good_pts in.good_pts;
by center subject;
run;
```

Finally, PROC FORMAT, ODS RTF and PROC REPORT were employed to generate the user-friendly chronology report in Microsoft Word format. A separate listing of the subject IDs along with a total count of subjects was also generated during the final reporting process.

The code below was used to create the required formats.

```
proc format;
   value gender  1 = 'Male'
                 2 = 'Female' ;
   value stat    1 = 'Discharged from hospital alive'
                 2 = 'Still in hospital at day 30 after surgery'
                 3 = 'Death in hospital' ;
   value ror     1 = 'Yes'
                 0 = 'No' ;
   value yesno   1 = 'Yes'
                 2 = 'No' ;
   value period  1 = 'Baseline prior to surgery'
                 2 = 'Post-op prior to randomization'
                 3 = 'Day 1 post-randomization'
                 4 = 'Day 4 post-randomization or discharge' ;
   value ck      1 = 'ng/ml'
                 2 = 'U/L'
                 3 = 'ug/L'
                 4 = '%'
                 5 = 'Index' ;
   value trop    1 = 'ng/ml' ;
   value trtype  1 = 'I'
                 2 = 'T' ;
   value mytime  0-high = [time5.]
                 .U     = 'Unknown' ;
   value result  999999 = '   T'
                 .      = ' '
                 other  = [8.2] ;
   value event   1 = 'Pre-surgery ECG'
                 2 = 'Baseline Troponin'
                 3 = 'Surgery'
                 4 = 'Post-op ECG'
                 5 = 'Post-op Troponin'
                 6 = ‘Subject Randomized'
                 7 = 'Day 1 post-rand Troponin'
                 8 = 'Day 4 post-rand ECG'
                 9 = 'Day 4 post-rand Troponin'
                10 = 'Serum Marker Results/ECG'
                11 = 'Acute Coronary Event'
                12 = 'CABG'
                13 = 'PCI/PTCA'
                14 = 'Adverse Event'
                15 = 'AE - Death'
                16 = 'Death in Hospital - No Autopsy'
                17 = 'Death in Hospital - with Autopsy'
                18 = ‘Subject Discharged'
                19 = 'Subject still in hospital at 30 days'
                20 = 'Subject Withdrew' ;
run;
```

The code below was used to generate the list of subjects appearing in the report.

```
data good_pts (keep=center subject pid letcode);
set in.final;
by center subject;
if first.center or first.subject then output;
run;

%let dsid=%sysfunc(open(good_pts));
%let num=%sysfunc(attrn(&dsid,nobs));
%let rc=%sysfunc(close(&dsid));

ods listing close;
ods rtf file = " C:\My Documents\Reports\ECG_LIST_&SYSDATE..doc"
   style=sasweb columns=3;

data _null_;
  file print ods;
```

```
  set good_ecg_pts (keep= pid letcode);
  label pid = 'Subject ID'
        letcode = 'Subject Letcode';
  title 'List of Subjects to be included in ECG Report for Core Lab';
  footnote "Total number of subjects: "&num          "Report generated &sysdate";
  put _ods_;
run;

ods _all_ close;
ods listing;
```

The code below was used to generate the chronology report. Note: line wrapping is used for display purposes only.

```
ods listing close;
ods rtf file="C:\My Documents\Reports\CHRONOLOGY_&SYSDATE..doc" ;

proc report data = in.final headline headskip split='/' nowd
            style(header)=[font_face="sas monospace" font_size=8 pt
                           background=#c0c0c0]
            style(column)=[font_face="sas monospace" font_size=8 pt]
            style(report)=[frame=void rules=rows];
by center subject ;
title 'Subject Event Chronology';
format rorrmk $30.;
format center z3. ;
format subject z4.;

column
   center subject  pid letcode age gender dis_dt statdls ror rorrmk ran_date
   ran_time date time event end_time copy ba_ctni ck_uln trop_uln troptype tropdxl
   ck_val trop_val ecg ae_remrk  ;

** Header Data **;
define center   /  noprint ;
define subject  /  noprint ;
define pid      /  display noprint ;
define letcode  /  display noprint ;
define age      /  display noprint ;
define gender   /  display noprint ;
define dis_dt   /  display noprint ;
define ran_date /  display noprint ;
define ran_time /  display noprint ;
define statdls  /  display noprint ;
define ror      /  display noprint ;
define rorrmk   /  display noprint ;

** CRF Data **;
define date     /  'Date' width = 10 f=mmddyy10. style(column)=[just=left];
define time     /  'Time' width = 12 f=mytime. style(column)=[just=left];
define event    /  'Event' width = 25 f=event. style(column)=[just=left] ;
define end_time /  'Surgery/End Time' width = 12 f=mytime.
                    style(column)=[just=left];
define copy     /  'Req. ECG/Copy/Obtained' width = 10 f=yesno.
                    style(column)=[just=center];
define ba_ctni  /  'Troponin/Result*' width = 8 f=result.
                    style(column)=[just=center];
define ck_uln   /  'CK/ULN' width = 12 style(column)=[just=center];
define trop_uln /  'Trop/ULN' width = 12 style(column)=[just=center];
define ck_val   /  'SM CK/Result' width = 10 style(column)=[just=center];
define trop_val /  'SM Trop/Result' width=10 style(column)=[just=center];
define troptype /  'Trop/Type' width=4 f=trtype. style(column)=[just=center];
define tropdxl  /  'Diag/Level' width=6 f=5.2 style(column)=[just=center];
define ecg      /  'ECG/Obtained' width = 8 f=ror. style(column)=[just=center];
define ae_remrk /  'AE/Remark' width = 30 style(column)=[just=left];

compute before ;
   line = ' ' ;
```

```
endcomp;

compute before _page_ / left style=[font_face="sas monospace" font_size=8 pt] ;
   line @15 'PID: ' @20 pid $8.  @40 'LETCODE: ' @50 letcode $3. @60 'AGE: ' age 3.
      @75 'GENDER: '               @100 gender gender. ;
   line @15 '_____________________________________________________________________________' ;
   line @15 'Date of Randomization: '           @40 ran_date mmddyy10.
      @75 'Time of Randomization: ' @100 ran_time time5. ;
   line @15 'Date of Discharge: '               @40 dis_dt mmddyy10.
      @75 'Status at Discharge: '   @100 statdls stat. ;
   line @15 'Return to OR: '        @40 ror ror.
      @75 'Reason: '                @100 rorrmk $30. ;
   line @15 ' ' ;
endcomp;

footnote font='arial' height=10pt italic "* T indicates the Troponin specimen was
   lost or the volume was not sufficient for testing." ;
footnote3 font='arial' height=8pt italic "Report date - &SYSDATE" ;
run;

ods rtf close;
ods listing;
```

A sample page of the final output from the PROC REPORT is displayed below.

Subject Event Chronology

PID: 999-1234 LETCODE: XYZ AGE: 65 GENDER: Male

Date of Randomization: 01/16/2010 Time of Randomization: 09:30
Date of Discharge: 01/25/2010 Status at Discharge: Discharged from hospital alive
Return to OR: No Reason:

Date	Time	Event	Surgery End Time	Req. ECG Copy Obtained	Troponin Result*	CK ULN	Trop ULN	Trop Type	Diag Level	SM CK Result	SM Trop Result	ECG Obtained	AE Remark
01/12/2010		Pre-surgery ECG		Yes									
01/12/2010		Baseline Troponin			0.01								
01/12/2010	19:30	Serum Marker Results/ECG					0.15 ng/ml		1.50		0.09 ng/ml	No	
01/13/2010	14:25	Acute Coronary Event											
01/14/2010	13:45	Surgery	16:20										
01/16/2010		Post-op ECG		Yes									
01/16/2010		Post-op Troponin			0.03								
01/16/2010	09:30	Subject Randomized											
01/17/2010		Day 1 post-rand Troponin			0.03								
01/19/2010	10:46	Serum Marker Results/ECG					0.15 ng/ml		1.50			Yes	
01/20/2010		Day 4 post-rand ECG		Yes									
01/20/2010		Day 4 post-rand Troponin			0.2								
01/22/2010	17:52	Serum Marker Results/ECG					0.15 ng/ml		1.50			Yes	
01/25/2010		Subject Discharged											

CONCLUSION

Utilizing SAS products, which is the standard data storage and retrieval software employed at the Perry Point coordinating center, staff have been successful in implementing a chronology reporting system that provided the central ECG core lab with a user friendly report for classifying subject events over the last five years.

ACKNOWLEDGEMENT

The author's would like to express their appreciation to the following people for their assistance on this project:

M. Terrin, M.D., University of Maryland School of Medicine, Baltimore, Maryland

R. Anne Horney, Department of Veterans Affairs, Perry Point Cooperative Studies Coordinating Center

CONTACT INFORMATION

Your comments and questions are valued and encouraged. Contact the authors at:

Christina Carty
CSPCC (151E)
VA Medical Center, P.O. Box 1010
Perry Point, MD 21902
Phone: 410-642-2411, ext. 5354
Fax: 410-642-1156
Email: Tina.Carty@va.gov

Elizabeth Spence
CSPCC (151E)
VA Medical Center, P.O. Box 1010
Perry Point, MD 21902
Phone: 410-642-2411, ext. 5308
Fax: 410-642-1156
Email: Elizabeth.Spence@va.gov