

Automation using SAS Makes it Easy to Monitor Dynamic Data in Clinical Trial

Wenjun He

The Emmes Company, LLC Rockville MD

ABSTRACT

A successful clinical trial is the teamwork composed of study startup, conduct, data collection, monitoring and data cleaning, among which it is critical that the study is implemented as specified in the protocol and the data is clean and of good quality. As most electronic data capture (EDC) system is setup to export data as SAS® datasets, the clinical SAS® programming team can leverage these datasets and provide metrics and tools to facilitate dynamic data monitoring, improving the quality of research study data and enhancing human subject protection especially in risk-based monitoring. This paper presents example SAS® code using SAS® macros and other SAS® facilities to automate tracking enrollment progression, real time site metrics reporting, and safety-based medical monitoring, which can benefit EDC clinical reporting and trial management processes.

INTRODUCTION

The integrity of clinical trial outcomes is dependent upon the accuracy and completeness of the study data. Data collection processes start from direct data entry into the EDC system, or data entry transcribed from paper source documents into the EDC system. Web-based EDC system is used to replace manual paper methods and assist in clinical trial electronic data capture and management. Periodic monitoring of site activities such as participant recruitment, visits, protocol deviation, and captured research data is required to insure the information complete and of highest quality for analysis in a trial. In addition to clinical data, EDC system can provide backend tables to facilitate trial management and following risk-based monitoring during the course of a clinical trial, such as the tracking of case report form (CRF) completion at the site level and the resolution of queries with the site regarding the discrepancies in collected data.

SAS® software is one application to be used for evaluating and analyzing the operational data and generate custom reports that benefit trial monitoring and management process. The Macro facility in SAS® can be utilized in automatically reporting to inform trial personnel of ongoing study activities. The reports based on EDC system data can be enhanced by SAS® Output Delivery System (ODS) in generating reports in multiple formats such as HTML, RTF, PDF, and Excel files. Efficiency of SAS® program can be improved when using PROC SQL and ODBC connection to SQL based EDC system.

This paper and presentation will use real case examples to demonstrate the basic SAS® tools that can efficiently assist with the dynamic clinical trial data monitoring.

ACCESS TO SQL RELATIONAL DATABASE THROUGH SAS®

Multi-site trials are always necessary for trials with the large sample sizes in the design. In the multi-site trials with various visits and numerous CRFs, frequent monitoring forms and data quality are required to ensure adequate and timely capture of clinical data throughout a study. Following is the flowchart of data collection and monitoring in clinical trial (Figure 1). It is important to have clinical trial data entered into e-CRFs to be available in an expeditious manner for analyses in a SAS® environment.

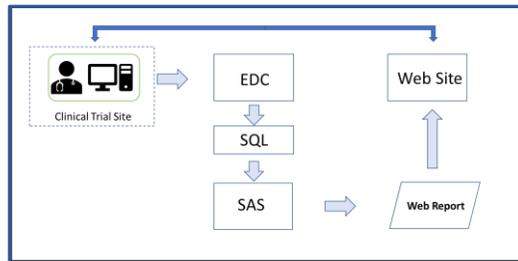


Figure1. Flowchart of data collection and monitoring in clinical trial

Electronic data capture systems are built on a relational database framework. One access method is through an Open Database Connectivity (ODBC) application provides an interface and wizard administrative tool to assign a Database System Name (DSN) that can be used to link one's SAS environment and a SQL database platform. The DSNs can be established with windows authentication using network login or through SQL authentication if the datasets are password protected.

```

LIBNAME protlib ODBC NOPROMPT="driver=SQL SERVER;
                                server=P3SK1;
                                uid=dataroPHU_SAS;
                                pwd=XXXXXXXXXX;
                                database=PHUprod"
                                SCHEMA=sas;
  
```

Figure 2. ODBS and LIBNAME declaration in SAS for connection to the study data

Once the ODBC connection is setup, one approach for connection to the study database is a data connection through a LIBNAME declaration (Figure 2). The ODBC DSN was specified in this example by DATABASE=PHUprod. In defining the ODBC connection through the wizard, the SQL server location and database name must be declared. Since this is a password protected SQL database, the LIBNAME statement incorporates USERID (specified in this example by "uid = ") and PASSWORD (specified in this example by "pwd="). ODBC connection option is specified by "NOPROMPT=" with separate multiple options separated by a semicolon for the data source. If the programmer does not specify enough correct connection options, an error returns and no dialog box is to be displayed to help complete the connection string.

OUTPUT DELIVERY SYSTEM FOR GENERATING CUSTOM REPORTS

The SAS® Output Delivery System (ODS) is a remarkable tool for generating professional style reports in variety of output platform, such as HTML, PDF, and RTF, among which, HTML reports are an ideal format to submit to site staff due to its easy accessibility through any web browser. Figure 3 is a web report example generated by SAS® and updated on a daily basis to monitor the enrollment of a multi-site clinical trial.

Accrual Report
All Sites
As of Tuesday, April 5, 2022
(Click column header to sort.)

Subject ID	Sex	Age at Enrollment	Enrollment Date
(all)	(all)	(all)	(all)
0001	Female	31 (weeks)	11FEB2021
0002	Female	31 (weeks)	11FEB2021
0003	Male	22 (weeks)	11FEB2021
0004	Male	30 (weeks)	16FEB2021
0005	Female	33 (weeks)	16FEB2021
0006	Female	45 (weeks)	16FEB2021

Figure 3. Example HTML output generated using TABLEDITOR TAGSET

The TABLEDITOR TAGSET can be used to create HTML output. The output can be easily exported to EXCEL or other filetypes. Some features can be added to the output using OPTIONS. For example, the column header and row headers can be frozen like in Excel when the FROZEN_ROWHEADERS= and FROZEN_HEADERS= options are specified. These options can be added in conjunction with the PAGEHEIGHT= and PAGEWIDTH= to generate an explicit page width and page height. A mouseover color can be specified using the HIGHLIGHT_COLOR= option. The SORT= option enables you to click on the column headers to sort the columns. Autofilter can be added using the AUTOFILTER="YES" option. To filter specific columns, the FILTER_COLS= option can be specified with the list of columns separated by commas. The below code adds frozen headers to the page and apply filters to the table.

```
ods tagsets.tableeditor path="xxxxx"
  body=body (title="XXXXXXXX") style=custom
  options(sort="yes"
    frozen_headers="yes"
    pageheight="650"
    highlight_color='pink'
    scrollbar_color="silver"
    top_margin="0.8"
    autofilter='yes' |
    sort_arrow_color="blue");
```

Example code using ODS TABLEDITOR TAGSET to create HTML output

The ODS TEXT is used in the example program to insert the text “No enrollment have been reported” to output destination that is open, in case there is no enrollment occurs at some sites.

```
%if &Ncnt=0 %then %do;
  ods text="<div align='center'>^S={font_size=10.5pt font_weight=bold color=green} %str(No enrollments have been reported.)</div>";
%end;
```

Example code using ODS TEXT to insert text in open output destination

The complete example SAS program using MACRO RPT to generate both site specific accrual report and the main report is added as below.

```

%macro rpt;
  %do i=1 %to 4;
    %if %i EQ 1 %then %do;
      data indnsi; set rptout; run;
    %end;
    %if %i EQ 2 %then %do;
      data indnsi; set rptout; where strip(enrlsite) in ("27BRI"); run;
    %end;
    %if %i EQ 3 %then %do;
      data indnsi; set rptout; where strip(enrlsite) in ("28FKH"); run;
    %end;
    %if %i EQ 4 %then %do;
      data indnsi; set rptout; where strip(enrlsite) in ("29BAN"); run;
    %end;

  %let Ncnt=0;
  PROC SQL noprint;
    select count(*) into :Ncnt from indnsi;
  quit;
  %put %Ncnt;

  filename body "&rpt_dir2.\&prot._accruals&sitecodesi...html";

  ods tagsets.tableeditor path="&rpt_dir2." body=body (title="Accrual Report") style=custom
  options(sort="yes" frozen_headers="yes" pageheight="650" highlight_color='pink' scrollbar_color="silver"
    top_margin="0.8" autofilter='yes' sort_arrow_color="blue");
  title1 h=18pt color=blue bold font=arial "&protname.";
  title2 h=16pt color=blue bold font=arial "Accrual Report";
  title3 h=14pt color=blue bold font=arial "&ssitesi";
  title4 h=12pt color=blue bold font=arial "As of &today";
  %if %Ncnt=0 %then %do;
    ods text="<div align='center'>^S=[font_size=10.5pt font_weight=bold color=green] %str(No enrollments have been reported.)</div>";
  %end;
  title6 h=10pt color=green font=arial "(Click column header to sort.)";
  footnote2 j=c h=3 italic "Data cleaning and monitoring remain ongoing. This data should not be considered clean.";

  %if %i EQ 1 %then %do;
  proc report data=indnsi nowd split=''
    style(report)=[font_face=arial cellspacing=.5 outputwidth=6in font_size=&fsize.]
    style(header)=[background=lightgrey font_weight=bold font_size=&fsize.]
    style(column)=[font_face=arial just=center font_size=&fsize.];

    column enrlsite patid sex age enrldate ;
    define enrlsite / display "Site Code" style(column)={cellwidth=1in};
    define patid / display "Subject ID" style(column)={cellwidth=1in};
    define sex / display "Sex" style(column)={cellwidth=1in};
    define age / display "Age at Enrollment" style(column)={cellwidth=1in};
    define enrldate / display "Enrollment Date" style(column)={cellwidth=1in};

  %end;

  %else %do;
  proc report data=indnsi nowd split=''
    style(report)=[font_face=arial cellspacing=.5 outputwidth=6in font_size=&fsize.]
    style(header)=[background=lightgrey font_weight=bold font_size=&fsize.]
    style(column)=[font_face=arial just=center font_size=&fsize.];

    column patid sex age enrldate ;
    define patid / display "Subject ID" style(column)={cellwidth=1in};
    define sex / display "Sex" style(column)={cellwidth=1in};
    define age / display "Age at Enrollment" style(column)={cellwidth=1in};
    define enrldate / display "Enrollment Date" style(column)={cellwidth=1in};

  %end;
  run;
  ods tagsets.tableeditor close;
  filename body clear;

%end;
%mend rpt;

```

USING SAS MACROS TO FACILITATE REAL TIME MONITORING ON EDC DATABASE

Example 1: Tracking Enrollment Progression and Date Projection for Enrollment Target

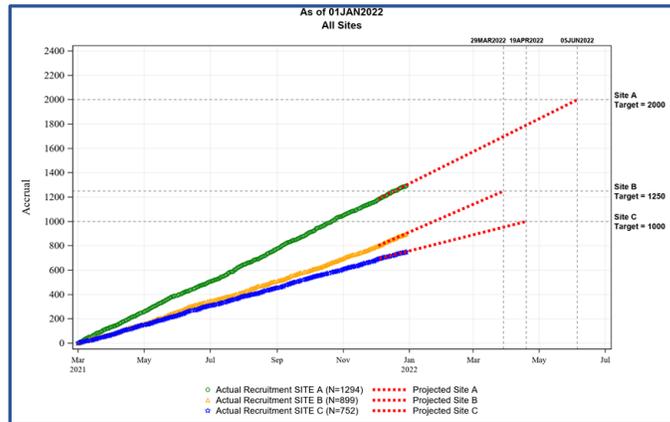


Figure 4. Example output for daily accrual progression with the projected date of enrollment completion based on the target for enrollment and the observed enrollment rate in the last 28 days prior to the report date

A successful completion of a clinical trial requires pre-specified enrollment targets to be met. This example is to use SAS Macro to program for the projected date of enrollment completion, based on the pre-designed enrollment target and the observed enrollment in the last 28 days prior to the report date.

The complete SAS code of the Macro generate_proj_data() for projected date of enrollment completion is added below. The basic logic for this macro is to use GLM procedure with the cumulative counts of enrolled participant at each observed date as the predictor. The projected enrollment completion date will be the fitted value for the regression model prediction at the expected enrollment target.

```
%macro generate_proj_data(igrp=,itarget=, today=);
  %Global projected1 projected2 projected3;

  data gsigrp.curv;
    set B2(where=(site="Site &igrp")) ;
  run;
  proc sort data=gsigrp.curv; by SITE ENRLDATE; run;

  data gsigrp.curvp1;
    format _grp $10.;
    set gsigrp.curv;
    by site enrldate;
    _grp="observed";
    output;
    if last.site then do;
      enrldate=.; count=.; cum=%eval(&itarget.); _grp="projected"; output;
    end;
  run;

  ods output ParameterEstimates=pars;
  Proc GLM data=gsigrp.curvp1(where=(enrldate>="&today"d-28 or missing(enrldate)));
  Model enrldate=cum;
  output out=gsigrp.curvp2 predicted=fit;
  Run; quit;
  ods output close;

  data gsigrp.curvp3;
    format _grp $10. fit date9.;
    set gsigrp.curvp2;
    _grp="projected";
    enrldate=fit; output;
    keep _grp site enrldate CUM;
  run;

  data gsigrp.curvp4;
    set gsigrp.curvp1(where=(_grp="observed")) gsigrp.curvp3;
    if "&igrp"="A" then call symput("projected1",put(enrldate,date9.));
    else if "&igrp"="B" then call symput("projected2",put(enrldate,date9.));
    else if "&igrp"="C" then call symput("projected3",put(enrldate,date9.));
  run;

%mend generate_proj_data;
```

Another Macro generate_graph is added below, which is using SGPLOT procedure to generate the graph (Figure 4) with all of the observed accrual as well as the projected enrollment completion date.

```

%macro generate_graph;

proc sgplot data=rptout;
  styleattrs datacontrastcolors=(green orange blue purple) datasymbols=(circle triangle star diamond);
  format enrldate date7.
  group1 group1f.
  group2 group2f. |
  group3 group3f.
  group21 group21f.
  group22 group22f.
  group23 group23f.;

  scatter x=enrldate y=Cum_1 /group=group1 markerattrs=(size=6 color=green symbol=circle) name = 'D1' legendlabel='Actual Count';
  scatter x=enrldate y=Cum_2 /group=group2 markerattrs=(size=6 color=orange symbol=triangle) name = 'D2' legendlabel='Actual Count';
  scatter x=enrldate y=Cum_3 /group=group3 markerattrs=(size=6 color=blue symbol=star) name = 'D3' legendlabel='Actual Count';

  series x=enrldate y=Cum_1p /group=group21 lineattrs=(thickness=4 color=red pattern=shortDash) name = 'P1' legendlabel='Projection Line';
  series x=enrldate y=Cum_2p /group=group22 lineattrs=(thickness=4 color=red pattern=shortDash) name = 'P2' legendlabel='Projection Line';
  series x=enrldate y=Cum_3p /group=group23 lineattrs=(thickness=4 color=red pattern=shortDash) name = 'P3' legendlabel='Projection Line';

  xaxis display=(nolabel) grid min=&min_dt max=&projected1"D valueattrs=(size=8);
  yaxis label = 'Accrual' values=(0 to 2500 by 200) grid;

  refline 2000/axis=y LINEATTRS=(pattern=shortDash color=grey) label=('Site A *Target = 2000') LABELLOC = OUTSIDE splitchar="*" labelattrs=(size=8 weight=bold family=arial);
  refline 1250/axis=y LINEATTRS=(pattern=shortDash color=grey) label=('Site B *Target = 1250') LABELLOC = OUTSIDE splitchar="*" labelattrs=(size=8 weight=bold family=arial);
  refline 1000/axis=y LINEATTRS=(pattern=shortDash color=grey) label=('Site C *Target = 1000') LABELLOC = OUTSIDE splitchar="*" labelattrs=(size=8 weight=bold family=arial);

  refline &projdt_A2000/axis=x LINEATTRS=(pattern=shortDash color=gray) label("&projdt_A2000c") labelattrs=(size=6 weight=bold family=arial) LABELLOC = OUTSIDE LABELPOS=max;
  refline &projdt_B1250/axis=x LINEATTRS=(pattern=shortDash color=gray) label("&projdt_B1250c") labelattrs=(size=6 weight=bold family=arial) LABELLOC = OUTSIDE LABELPOS=max;
  refline &projdt_C1000/axis=x LINEATTRS=(pattern=shortDash color=gray) label("&projdt_C1000c") labelattrs=(size=6 weight=bold family=arial) LABELLOC = OUTSIDE LABELPOS=max;

  keylegend 'D1' 'P1' 'D2' 'P2' 'D3' 'P3' / across=2 down=3 title="" valueattrs=(Family=Arial size=9pt) location=outside position=bottom NOBORDER;
run;
%mend generate_graph;

```

Example 2: Progression in Screening, Randomization, and Treatment by Site

The following SAS Macro is used to monitor the weekly update in screening, randomization, and study product administration. The side-by-side panels of bar charts for each site or all site are generated to display the counts of participants who are recruited for screening, screening-failed, randomized, or dosed, respectively. This plot can be setup for updating on a weekly basis, with the bar in red to show the newly reported counts occurred in the last 7 days, next to the corresponding bar in blue for the cumulative one in the same category.

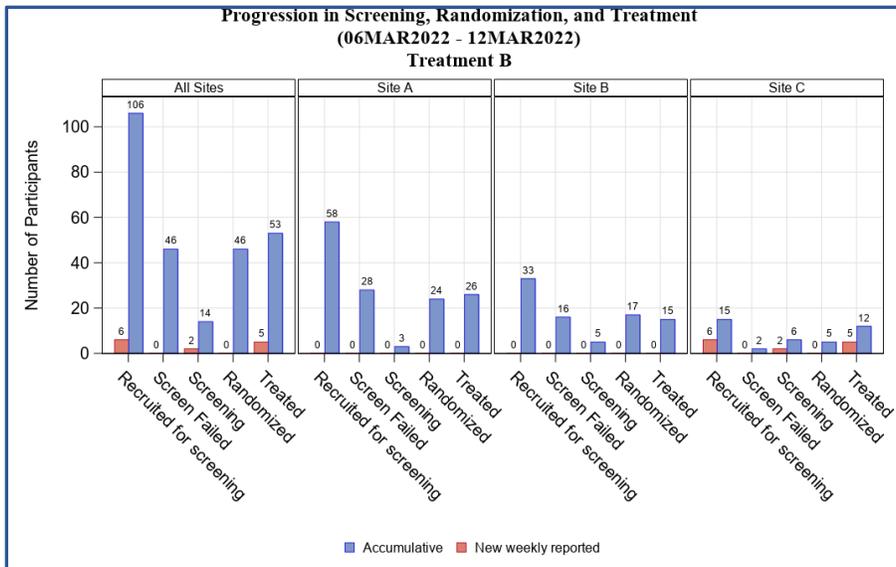


Figure 5. Example output to monitor progression in screening, randomization, and treatment by site

```

%macro graph2(gr=);
ods graphics on/reset=index border=no width=9in height=6in imagefmt=png imagename="PharmaSUG2022_Figure2";
ODS ESCAPECHAR="^";
Title1 font="Times New Roman" h=12pt bold "Figure 2";
Title2 font="Times New Roman" h=12pt bold "Progression in Screening, Randomization, and Treatment ";
Title3 font="Times New Roman" h=12pt bold "(&week_inv.)";
Title4 font="Times New Roman" h=12pt bold "Treatment &gr.";

proc spanel data=laskweek1;
  where cgrp_n="&gr." ;
  panelby site/rows=1 columns=4 novarname spacing=2 uniscale=all;
  vbar ngrp/missing
      response=nfreq
      group=grpl datalabel
      groupdisplay=cluster
      grouporder=descending
      CLUSTERWIDTH=.8
      DATALABELFITPOLICY=NONE ;
  rowaxis grid label="Number of Participants" discreteorder=data labelattrs=(size=11pt) VALUEATTRS=(SIZE=11 weight=normal);
  colaxis grid label=" " max=150 discreteorder=data labelattrs=(size=11pt) VALUEATTRS=(SIZE=11 weight=normal);
  keylegend / SORTORDER=ASCENDING
      position=bottom noborder |
      title=" "
      TITLEATTRS=(Family='Times New Roman' Size=12);

run;
ods graphics off;
ods rtf close;
%mend;
%graph2(gr=B)

```

Example 3: Safety-Based Medical Monitoring

The following graph is an example to monitor the frequency of unsolicited adverse events (AEs) by MedDRA System Organ Class (SOC) and severity by cohort. The SAS code for the Macro to generate the panels is added below. The summary of the AEs can be updated periodically for safety monitoring.

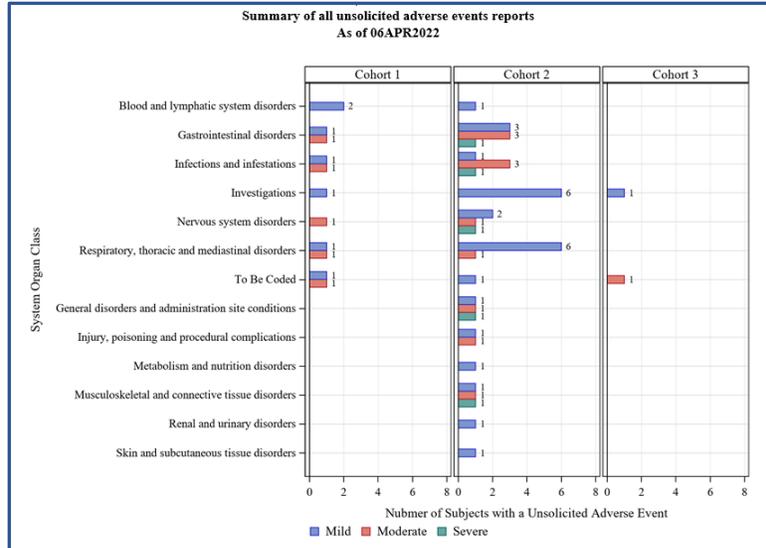


Figure 6. Example output to monitor unsolicited adverse events (summary) by MedDRA System Organ Class and severity by cohort

```

goptions reset=all htext=8pt device=png;
options orientation=landscape nodate nonumber;
ods listing image_dpi=120 gpath="C:\Users\whe\Desktop\submission\2022\output" style=myjournal3;

ods graphics on/reset=index border=no width=9in height=6in imagefmt=png imagename="PharmaSUG2022_Figure3";
ODS ESCAPECHAR="^";

%macro graph3(date=, title=);
ods rtf file="C:\Users\whe\Desktop\submission\2022\output\PharmaSUG2022_Figure3_sdate.rtf" nogtitle nogfootnote bodytitle;
Title1 h=12pt bold font='Times New Roman' "FIGURE 3";
Title2 h=12pt bold font='Times New Roman' "Summary of &title. unsolicited adverse events reports";
Title3 h=12pt bold font='Times New Roman' "As of &sdate.";

proc sgpanel data=work1 dattmap=attrmap;
panelby_grp /rows=1 columns=3 novarname spacing=2 uniscale=all;
hbar SOC/missing
response=n_subj
group=max_sev datalabel
groupdisplay=cluster
grouporder=descending
CLUSTERWIDTH=0.8
DATALABELFITPOLICY=NONE ;
rowaxis grid label='System Organ Class' discreteorder=data labelattrs=(size=10pt) VALUEATTRS=(SIZE=9 weight=normal) grid gridattrs=(color=greyE0) ;
colaxis grid label='Number of Subjects with a Unsolicited Adverse Event' max=8 discreteorder=data labelattrs=(size=10pt) VALUEATTRS=(SIZE=9 weight=normal);
keylegend / SORTORDER=ASCENDING
position=bottom noborder |
title=' '
TITLEATTRS=(Family='Times New Roman' Size=12);
run;

ods rtf close;
%mend graph3;
%graph3(date=stoday, title=all)

```

CONCLUSION

This paper presents example SAS code using SAS macros and other SAS facilities to automate tracking enrollment progression, real time site metrics reporting, and safety-based medical monitoring, which can benefit EDC clinical reporting and trial management processes.

CONTACT INFORMATION

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

Wenjun He

Senior SAS Programmer

The EMMES Company

whe@emmes.com