

EP90SAS: A Clinical Trials Data Management Tool

Phil Busby, President, Live Data Systems, Inc., and
Keith M. Bowden, Medical Data Systems Analyst,
Glaxo Wellcome, Inc., Research Triangle Park, North Carolina

EP90SAS is a conversion utility for clinical trials data. It was written for Glaxo Wellcome with SAS® and SAS/AF® software version 6.08 under VMS, and designed to address three special needs:

- The input is not generated all at once, but must be collected and processed in small batches;
- An easy to use user interface is needed so that non-programmers may process batches;
- The data sets names, variables, and directories will change from one study to the next, and it must be possible to add support for a new study without modifying the user interface.

These data are typically gathered from source documents in batches which are keyed into files using the Entrypoint 90® data entry software product (EP90). From there the batches are exported into sequential files, which are read with SAS INPUT statements and appended to master SAS data sets for subsequent analysis. EP90SAS helps the medical data analyst (MDA) manage the data conversion process via a SAS/AF control screen.

The feature which makes EP90SAS different from most data management applications is that the actual data fields, file layouts, file extensions, and data set names are dynamically determined by simple control files and %INCLUDEd SAS code. This makes the addition of new studies, data types, or records a straightforward task that does not require the SAS/AF application to change, allowing the MDA to focus on the data being analyzed, and not to have to worry about the SAS/AF screen control programming.

For each study the MDA composes the INPUT statement to be submitted from the SAS/AF control program. In the simple case of one record—one observation, the input statement is all that is needed. When multiple records are used to produce one observation, however, the analyst may need to use the RETAIN and OUTPUT statements. The third case, producing multiple output data sets, typically requires looking at control fields on the input record with the trailing @ INPUT statement option in order to route the data to the correct SAS data set. This is the model of the example which follows: all of the data for a Case Report Form (CRF) were exported from EP90 as one file, and the master SAS data sets for the study were loaded using EP90SAS.

The example shown was the sixth study processed by EP90SAS at GlaxoWellcome. After the first one was operational, subsequent studies were added without programming assistance or modification to the SAS/AF screens. The MDAs benefitted from having an easy to use on-line data management tool, but more importantly, they controlled the data, took responsibility for the correctness of the conversion, and found it easy to hook in enhancements to the conversion code.

The EP90SAS SAS/AF control screen is characterized by pushbuttons for pop-up pick lists where you can ask to see the available choices for parameter entry fields. When a pick list is displayed, you need only cursor down to the desired selection and press ENTER or Return to make a selection. When you select a study, all of the default values for that study are read from a control file and displayed, minimizing required keystrokes. When you press Return, all modified fields are validated (see Screen 1).

```

Glaxo Wellcome, Inc.
EP90 to SAS Conversion System

Enter study: ZOCA3001          <LIST>

Data type:  HSQ                <LIST>

Enter keyed batch (exported filename) and extension:

      _____ DAT          <LIST>

from file in ZQ8D04::EP90_XPT$BOOT:[EP90]
backed up to DRG$KP_ZOC:[ZOCA3001.DATA.EP90SAS]
and used to append to the master SAS data set
in library  DRG$KP_ZOC:[ZOCA3001.SASDATA]
with code in DRG$KP_ZOC:[ZOCA3001.SASPGMS.EP90SAS]

      <OK>          <CANCEL>          <HELP>

```

Screen 1: the EP90SAS Control Screen

On the EP90SAS control screen shown above, the uppercase data are fictitious default values that can be changed by typing over the data.

A helpful feature is the backup file check: each batch of input data is copied to a backup directory, and the system warns you when a batch has already been processed; however, the system allows you to process a batch even if it already exists on the backup directory. In this case you take responsibility for insuring that the new batch really should replace the previously-copied batch with the same name. The situation may arise from a batch which has been keyed in incorrectly and it is easier to delete all of the observations in the batch from the master SAS data sets and reenter them. Finally, the message field is used to give you feedback about the progress of each batch when it is appended to the master SAS data set(s).

The only other screen you see is the pick list: a SAS/AF program screen with an extended table that is used as a pop-up selection list (see Screen 2). The same screen is used for study selection, data type selection, and batch selection, with the appropriate prompt message displayed. After the display is the calling program code.

<RETURN>

Select one protocol:

BH700

CAVA922

NQ340

ZOC8841

ZOC8898

Screen 2: the EP90SAS Pick List

```

if btn = 'S' then do;      /* Study List wanted */
    btn = '';
    exten = 'STU';
    subject = 'protocol';
    study = '';
    char8 = study;
    call display('upnslst.program', char8,
        appath, exten, subject);
    study = char8;
    if study > '' then do;
        link defstudy;
        call symput('study', study);
    end;
return;
end;

```

The conversion of an exported batch of records from EP90 to SAS may take several forms; for example, one EP90 record yields one SAS data set observation (the simple case); multiple EP90 records yield a single SAS data set observation; or a single EP90 file yields multiple SAS data sets. Since a study typically has multiple data files (patient, summary, adverse event, drug, lab, etc.) each input record has to have a record ID field. The records exported from EP90 have a record identifier called a Batch Format Identifier (BFI) identifying the record layout.

The following section shows a portion of the SAS/AF program behind the control screen which drives the conversion. You can see that when multiple SAS data sets are being loaded from one input record, the %INCLUDED code communicates the appropriate names for the SAS data sets through SAS macro variables, and the number of output SAS data sets is implicitly communicated through the numeric suffix used in the macro variable naming convention.

```
if btn ne 'K' then return;
/* Proceed if OK selected */
btn = '';
if keybat > '' then do;
  call symput('keybat', keybat);
  msg1 = keybat;
  /* display keyed batch number */
  refresh; /* update displayed messages */
  call symput('namedir', ep90dir);
  /* UPSTU program just submits this:
  %inc "&codepath&datatype.sas"; */
  call display('upstu.program');
  /* update study data, create batch ds. */
  refresh; /* update macro vars */
  recs = symget('recs');
  if recs < '0' then recs = '0';
  /* included code sets outds1 if multiple */
  /* output data sets are created. */
  workds = symget('outds1');
  if recs > '0' or workds > '' then do;
    refresh; /* update displayed message */
    if workds > '' then do;
      do j=1 to 99 while(workds > '');
        workds = symget('outds'||
          left(trim(put(j,2)))));
        if workds > '' then do;
          recs = symget('outcnt' ||
            left(put(j,2)));
```

```
if recs > '0' then
  /* append to master data set */
  call display('mergestu.program',
    workds, workds);
end;
end;
end;
else
  /* default = datatype.saseb$data */
  call display('mergestu.program',
    datatype, keybat);
refresh; /* updates macro vars. */
/* copy batch from ep90dir to ep90in */
iname = trim(ep90dir) || trim(keybat)
  || '.' || trim(extensio);
syscmd = 'copy ' || trim(iname) || ' ' ||
  fname;
rc = system(syscmd);
end;
msg1 = 'Done.';
if recs > '0' then substr(msg1,6,5) = recs;
end;
```

The following section pertains to the %INCLUDED code for a particular study.

The data step is required for EP90SAS, but other files or macros can be created to make processing more efficient. To better organize the various files and output, the following directory structure is recommended (where '-' represents protocol id):

[-.saspgms.ep90sas] contains the input file(s), data step, and any miscellaneous or optional macros called by the program, such as an error detection program for a keyed, dedicated symbol (in this case, a '?').

- See APPENDIX 1 for code typical of the input file. A suggested file name is *.INP where * = SAS data set name.
- See APPENDIX 2 for code typical of the data step. A suggested file name is CRF.SAS.
- See APPENDIX 3 for code typical of the error detection program. A suggested file name is ERRLLIS.SAS.

[-.data.ep90sas] contains a copy of the uploaded batch file which prevents that file from being uploaded more than once.

[-.code] contains the output from the error-detection program and an edit /compare file corresponding to each data set.

[-.sasdata] contains the master SAS data sets produced by EP90SAS.

APPENDIX 1 : ADVERSE.INP

INPUT

```
@ 9 PTID $ 6.
@ 15 INVID $ 4.
@ 19 TMTNO $ 3.
@ 22 PERIOD $ 2.
@ 24 ADVTX $ 66.
@ 90 ADVFR $ 1.
@ 91 ADVSEV $ 1.
@ 92 ADVDTON $ 6.
@ 98 ADVTMON $ 6.
@104 ADVCRS $ 1.
@105 ADVDTRS $ 6.
@111 ADVDUR $ 6.
@117 ADACK1-ADACK7 $ 1.
@124 ADEXSR $ 1.
@125 ADLFTH $ 1.
@126 ADVDGRL $ 1.
@127 PTCOND $ 1.
;
```

APPENDIX 2: CRF.SAS

* Set up EP90SAS parameters and define /initialize macro counters;

```
filename drive "&namedir.&keybat..pat";
%let recs=0; %let outcnt1=0;
%let outcnt2=0; %let outcnt3=0;
%let outcnt4=0;
%let outds1=screen;
%let outds2=antimicr;
%let outds3=summary;
%let outds4=adverse;
```

* Set up data sets, identify or define variables, and initialize data set counters;

```
data screen(keep=ptid invid tmtno period dob
sex wtkg ethorig female pretest ptstneg)
antimicr(keep=ptid invid tmtno period dgds1
dgfr1 dgsttm1 dgstdt1 dgspdt1
```

```
dgstpm1 dgds2 dgfr2 dgsttm2 dgstdt2
dgspdt2 dgspdm2)
summary(keep=ptid invid tmtno period stycomp
wdricat wdrltx1-wdrltx4 wdrldt)
adverse(keep=ptid invid tmtno period advtx
advfr advsev advdton advtmon
advcrs advdtrs advdur adactk1-adactk7
adexsr adlfth advdgrl ptcond aecd) ;
infile drive missover end=eof;
input @1 bfi $8. @;
retain counta countb countc countd 0;
length aecd $ 8.;
```

* Set up to process records by batch format identifier (bfi) and input file;

```
if (bfi = 'SCREEN') then do;
%inc
```

```
'logicalname:[prot.saspgms.ep90sas]screen.inp';
output screen; counta + 1; end;
```

```
else if (bfi = 'ANTIMICR') then do;
%inc
```

```
'logicalname:[p.saspgms.ep90sas]antimicr.inp';
output antimicr; countb + 1; end;
```

```
else if (bfi = 'SUMMARY') then do;
%inc
```

```
'logicalname:[p.saspgms.ep90sas]summary.inp';
output summary; countc + 1; end;
```

```
else if (bfi = 'ADVERSE') then do;
%inc
```

```
'logical_name:[p.saspgms.ep90sas]adverse.inp';
if advtx = '' then delete;
```

```
output adverse; countd + 1; end;
```

```
if bfi ^in ('SCREEN' 'ANTIMICR'
'SUMMARY' 'ADVERSE') then do;
```

```
put 'WARNING: Unknown BFI: ' bfi;
delete;
```

```
end;
```

* Set up to display results from counters; drop bfi counta countb countc countd;

```
if eof=1 then do;
```

```
put / 'Output count screen = ' counta 7.;
```

```
call symput('outcnt1', left(put(counta,8.)));
```

```
put / 'Output count antimicr = ' countb 7.;
```

```
call symput('outcnt2',left(put(countb,8.)));
```

```
put / 'Output count summary = ' countc 7.;
```

```
call symput('outcnt3',left(put(countc,8.)));
```

```
put / 'Output count adverse = ' countd 7.;
```

```
call symput('outcnt4',left(put(countd,8.)));
```

```
end;
```

```
%inc
```

```
'logicalname:[p.saspgms.ep90sas]errlis.sas';
run;
```

APPENDIX 3: ERRLLIS.SAS

```

proc sort data=work.screen;
  by invid tmtno period; run;
proc sort data=work.antimicr;
  by invid tmtno period; run;
proc sort data=work.summary;
  by invid tmtno period; run;
proc sort data=work.adverse;
  by invid tmtno period; run;

data work.errlis(keep=invid tmtno period item
  result);
merge work.screen work.antimicr
  work.summary work.adverse;
by invid tmtno period;

array data [*] $ ptid invid tmtno dob sex wtkg
  ethorig female pregtest ptstneg
  dgds1 dgfr1 dgsttm1 dgstdt1 dgspdt1
  dgsptm1 dgds2 dgfr2 dgsttm2 dgstdt2
  dgspdt2 dgsptm2 stycomp wdrlcat wdrltx1
  wdrltx2 wdrltx3 wdrltx4 wdrltd advtx advfr
  advsev advdton advtmon advcrs advdtrs
  advdur adactk1 adactk2 adactk3 adactk4
  adactk5 adactk6 adactk7 adexsr adlftth
  advdgrl ptcond;

array names [*] $ names1-names48
  ('PTID' 'INVID' 'TMTNO' 'DOB' 'SEX'
  'WTKG' 'ETHORIG' 'FEMALE'
  'PREGTEST' 'PTSTNEG' 'DGDS1'
  'DGFR1' 'DGSTTM1' 'DGSTD1'
  'DGSPDT1' 'DGSPTM1' 'DGDS2' 'DGFR2'
  'DGSTTM2' 'DGSTD2' 'DGSPDT2'
  'DGSPTM2' 'STYCOMP' 'WDRLCAT'
  'WDRLTX1' 'WDRLTX2' 'WDRLTX3'
  'WDRLTX4' 'WDRLDT' 'ADVTX'
  'ADVFR' 'ADVSEV' 'ADVDTON'
  'ADVTMON' 'ADVCRS' 'ADVDTRS'
  'ADV DUR' 'ADACTK1' 'ADACTK2'
  'ADACTK3' 'ADACTK4' 'ADACTK5'
  'ADACTK6' 'ADACTK7' 'ADEXSR'
  'ADLFTH' 'ADV DGR' 'PTCOND');

do i=1 to dim(data);
  if (index (data[i], '?') > 0) then do;
    item = names[i];
    result = data[i];
    output;
  end;
end;
run;

```

```

proc printto
  file='logical_name:[protocol.code]errlis.lis';
run;
proc print data=work.errlis;
  by invid tmtno;
  title2 'Data Entry Flag';
run;

```

Conclusion

The EP90SAS application demonstrates how effective teamwork can be to the successful processing of clinical trials data. The front-end person writes the screen control code, focusing on the ease of use of the user interface. This framework is then used by the MDA to specify the data sets and variables for the particular study, focusing on the accurate representation of the study data. The resulting system can be used to convert data rapidly and accurately as it comes in for the timely analysis of the clinical trial results to show the new drug's safety and efficacy.

SAS and SAS/AF are registered trademarks of SAS Institute Inc. in the USA and other countries.

® indicates USA registration.

Entrypoint 90 is a registered trademark of DataLex, Inc.

Other product names are trademarks of their respective companies.

Keith M. Bowden
Glaxo Wellcome, Inc.
5 Moore Drive
Research Triangle Park, NC 27709
919/483-8493

Phil Busby
Live Data Systems, Inc.
P.O. Box 13144
Research Triangle Park, NC 27709
800/252-3102
<http://www.interbuz.com>