Paper 148-31

Subject Profiles Generated with Macro Profile System (MPS)

John Iwaniszek, Stat-Tech Services, LLC, Chapel Hill, North Carolina

Abstract:

Subject profiles are commonly requested as part of the general deliverables in a Food and Drug Administration (FDA) submission. Profiles usually take the form of a by-subject fact sheet detailing key data from different pages of the Case Report Form (CRF). The Macro Profile System (MPS), written in Base SAS®, provides a convenient way to generate profiles in Rich Text Format (RTF), which can be easily converted to Portable Document Format (PDF), and bookmarked to provide easy access to reviewers. This paper describes the MPS macros, provides an outline for programming subject profiles, and gives tips for book-marking individual profiles for easy access from an index.

Introduction:

Subject profiles are often an important component of the set of deliverables for drug approval submission to the US Food and Drug Administration (FDA). Profiles provide detailed information about individual subjects using key data from different parts of the Case Report Form (CRF). Creating subject profiles requires blending data that is stored in different data structures and presenting it in an organized and readable format. The Macro Profile System (MPS), written in Base SAS® and designed by George DeMuth, President of Stat-Tech Services, LLC, and the author, provides a simple way to generate subject profiles in Rich Text Format (RTF) that are accessible using bookmarks and default features of MSWord® and Adobe Acrobat®.

The Macro Profile System:

The MPS is a family of Base SAS macros that formats output using the RTF standard. RTF files are word process files made using a markup language like HTML that open in standard word processing packages like MS Word and Open Office. MPS initializes an output RTF file, creates RTF tables and notes, populates RTF tables, and closes the output RTF file. MPS is distributed freely by its owners, Stat-Tech Services, LLC and is available through its website (Stattechservices.com) or by contacting the author by email. A description of each macro and lists of their parameters follows the conclusion of this paper along with author contact information.

The Structure of a Subject Profile Program:

A subject profile program has a definite and orderly structure composed of two main parts. The first part of the program generates data sets required for the profiles. This involves counting and identifying the subjects to be included in the profiles, organizing the individual sets of data to be displayed, and building variables that append whatever RTF commands that are needed to augment the headers, tables, and sections that will make up the final profile.

The second part is comprised of the macro calls that set up the RTF document, and format the individual subject profiles. The following is a sample program that illustrates the various parts of the typical subject profile program.

The effort to organize the data sets for display and the structure of the macro calls that ultimately format the profiles should be guided by an approved profile specification. The specification should have been first reviewed by the project team and the end-users to make certain that it meets the needs of the people for whom the profiles are ultimately targeted.

Preparing CRF data for MPS:

The first step involves accounting for every subject who is to be included in the profiles. This is done by creating a data set with one record per subject. It is sometimes convenient to also include subject level variables like treatment group, study population (safety, efficacy), completer, gender, race, age, etc. in this data set.

```
data
        profile
        ;
        set
                mpl /* master patient list */
                ;
        by
                site subject
                ;

        length
                subjid $12 subjlink $100 dline 8
                ;

        /* combines site and subject number to produce a unique subject identifier */
        subjid = trim(left(site))||'-'||trim(left(subject));


        /* creates a field to be used later to bookmark individual subject profiles */
        subj_link = '{\field {\*\fldinst  HYPERLINK \\l "_'||
                    trim( left( translate(subjid, '_', '-')))||
                    '"}{\fldrslt {\cf2 '||trim( left( subjid))||' }}}'
                    ;

         dline=_n_; /* non-printing sorting variable used by macros */

         /* outputs various variables as subscripted macro variables
         */
         call symput( “subject”||trim( left( _n_)), trim( left ( subjid)) );
         call symput( “pat”||trim( left( _n_)), trim( left ( subjid)) );
         call symput( “init”|| trim( left( _n_)), trim( left ( initials)) );
         call symput( “screen”|| trim( left( _n_)), put( screen_date, date9.) );
         call symput( “fdose”|| trim( left( _n_)), put( first_dose_date, date9.) );
         call symput( “term”|| trim( left( _n_)), put( termination_date, date9.) );
         call symput( “pat_spec”|| trim( left( _n_)), trim( left (subj_link)) );


run;
```

The rest of this first part of the program is devoted to formatting data from various pages of the CRF to match the Subject Profile specification. For example, if the specification calls for a table outlining demographic data as laid out in the sample profile, then the data should be transposed so that the variable labels are assigned to one text variable and the

corresponding data is assigned to another text variable. The following data step illustrates how the demographic data is prepared for a display of this type.

```
data
        demographic(keep=subjid name result)
        ;
        merge
                demog
                mpl(keep=subjid in=in_p)
                ;
        by
                subjid
                ;

        if in_p;

        length name result $300.;

        name='Date of Birth:' ;                                 result= put(BirthDt, date9.);   output;
        name='Age:' ;                                           result= put(dmage, 3.);         output;
        name='Sex:';                                            result= put(sex, $sex.);        output;
        name='Race:';                                           result= put(race, race.);       output;
        name='Height (in.):';                                   result= put(height, 8.2);       output;
        name='Weight (lb.):';                                   result= put(weight, 8.2);       output;
        name='Is the subject currently a cigarette smoker?';    result= put(ynsmoker, $yesno.); output;
        name='Female fertility status:';                        result= put(females, child.);   output;
        name='Pregnancy test results (females only):';          result= put(pregc, $preg.);     output;
        name='Contraceptive use (females only):';               result= put(contracep, $250.);  output;
run;
```

Notice that one record per variable is created, that the LABEL is hard-coded, and that the data to be displayed is put to the RESULT field using whatever format is appropriate for that particular variable.

The following example shows a way to deal with multi record data, adverse events being a prime example. To produce the Adverse Event table shown in the Sample Profile, the data will not have to be transposed, as in the previous example, but it will have to be reorganized so that some of the variables in the original data set are formatted and consolidated into variables that will later be displayed as table cells in the resulting profiles. The raw AE text is consolidated with the preferred term, the start, stop and number of days on study are consolidated into a single field, and the severity/relatedness/etc. ratings are combined into a single text variable. The consolidation also involves adding RTF code to provide text formatting in the final display (the RTF reserved word “\par” was inserted between data fields so that a hard return would appear in the final RTF table).

```
data
        adverse_events
        ;
        merge
                ae
                profile (in=in_m)
                ;
        by
                subjid
                ;
        if in_m;

        attrib aeterm length=$250 label='AE Term';
        attrib start_stop length=$150 label='AE Start Stop Datetime';
        attrib status length=$250 label='Status';

        if aetext=" " then do;
                  aeterm="NO ADVERSE EVENTS WERE REPORTED";
                  start_stop=" ";
                  time_fram=" ";
```

```
                status=" ";
                treated=" ";
        end;

        else do;

                if p_term^=" " then aeterm = trim(left(aetext))||' ('||trim(left(p_term ))||")";
                else aeterm= trim(left(aetext));

                start_stop =    "Start \par "||put( aestdt , date9.)||
                                "/ \par Stop \par "||put(aeenddt, date9.)||
                                "\par Study Day:"||put( ((aestdt - screening_date) +1) , 3.)
                                ;

                status =        "Serious: "||put(aeser, $yesno.)||
                                "\par Freq: "||put(aefreq2, aefreq.)||
                                "\par Sev: "||put(aesev3, aesev.)||
                                "\par Out: "||put(aeout3, $outcome.)||
                                "\par Rel: "||put(aerel6, $aerel.)||
                                "\par Action: "||put(aeact4, $aeact.)
                                ;
        end;

        keep subjid aeterm start_stop status;
run;
```

Note that for many data sets, depending on the specification, this kind of reorganization of the data into special display data sets may not be necessary. For example, adverse events might be in nearly the exact shape needed to produce the specified portion of the profile devoted to adverse events. However it is likely that every data set will require some reorganizing before it is ready to be entered into the profile structure.

Creating the Profile:

Once the data have been prepared, the next step is to create the profiles. This is done using the MPS macros. A brief description of each macro follows:

Macro Name	Description
Rtfprofi.sas	Initializes the RTF file containing the profiles
Rtfproft.sas	Creates a text note in a profil
Rtfprofq.sas	Initializes a table
Rtfcol.sas	Creates a colum in a table
Rtfspan.sas	Creates a spanning header across one or more columns in a table
Rtfspend.sas	Terminates a spanning header
Rtfprofo.sas	Terminates a table
Rtfprofp.sas	Inserts a hard page break in a profile
Rtfprofe.sas	Closes the RTF file

(A complete list of each macro's parameters follows at the end of this paper.)

As stated previously, the profile has two major sections. The first section involves formatting the data while the second section is devoted to stepping through the list of subjects and appending individual profiles to the output file. The output portion of the program can be broken down into individual steps.

Initialize the Profile File:

The following macro call initializes the profile output file, sets the margins, the page layout, and sets the document properties title to "`Protocol XYZ Subject Profiles`".

```
%rtfprofi(filename=..\..\output\listing\profile.rtf, topm=1.0, leftm=0.5, land=N,
title=%bquote(Protocol XYZ Subject Profiles));
```

The initialization section is run once for the entire set of profiles. The balance of the profile macro calls are placed in a macro (%PATPROF) that contains a macro loop that processes the subjects individually.

Set Up Headers:

The MPS can make use of the Header section in MSWord. Users might find it convenient to place subject identification and baseline information in the header. The following code assumes that the header variables are stored in macro variable arrays. These macro variable arrays can be set up in the first section of the program where the data sets are organized for display. They can also be populated inside a macro loop using a DATA _NULL_ data step that uses a master patient list data set containing all the relevant subject data.

Note that the only difference between the two %RTFPROFT calls is that the first one has parameter ADDSECT=N and the second one has ADDSECT=Y. The first version is necessary to initialize the output file, while the second version is executed for every other subsequent subject after the first.

```
%macro patprof ;

        %do _i=1 %to 5 ;                      %* assuming 5 subjects as an example ;

           %if _i=1 %then %do;
                   %rtfproft(
                        text1=%bquote(Subject Profiles\par\par Site: &&site&_i
                        Subject: &&subject&_i  Initials: &&init&_i  Population:
                         Safety\par Screen Date: &&screen&_i  First Dose Date:
                         &&fdose&_i  Term Date: &&term&_i\par),
                        font1=3,
                        fonts1=10,
                        jst1=L,
                        header=Y,
                        footer=N,
                        addsect=n
                        );
                 %end;

                 %else %do;
                   %rtfproft(
                        text1=%bquote(Subject Profiles\par\par Site: &&site&_i
                        Subject: &&subject&_i  Initials: &&init&_i  Population:
                         Safety\par Screen Date: &&screen&_i  First Dose Date:
                         &&fdose&_i  Term Date: &&term&_i\par),
                        font1=3,
                        fonts1=10,
                        jst1=L,
                        header=Y,
                        footer=N,
                        addsect=Y
                        );
                 %end;
```

The following call will place a bookmark at the beginning of a subject's profile that MSWord and Adobe Acrobat will recognize and use to populate an index to the subject profiles. The bookmark is created using the variable SUBJ_LINK created in the PROFILE data set displayed above. The variable is stored in a macro variable array in

the same manner as one would store subject IDs for sub-setting the display data sets below.

```
%rtfproft(
          text1=%bquote(
\s1 {\*\bkmkstart _&&pat_spec&_i}{&&pat&_i}{\*\bkmkend _&&pat_spec&_i} \par),
          font1=3,
          fonts1=10,
          jst1=L,
          bold1=Y
          );
```

Profile Main Body Data Tables:

Once the subject profile output file is initialized and the headers are established, the next step is to build the data display portion of the profile. The routine for this part of the profile is to select the records for display by subject ID in a data step, and pass the temporary data set to a series of macro calls. The example below uses the subject ID stored in a macro variable array that was constructed in a previous data step.

The temporary data set contains only those records belonging to the particular subject contained in the display data set.

```
data temp1 ;                    %* Subset subject &_i ;
    set demographic;
    if subjid = "&&pat&_i" ;
    run ;
```

The following series of macro calls places a title “Demographics:” in open text, initializes a table, and formats two columns in the tables that will display the demographic data reorganized in the data step described above. The final macro call (%RTFPROFO) closes the table.

```
%rtfproft(text1=%bquote(Demographics: \par), font1=3,
          fonts1=10, jst1=L, bold1=Y);
%rtfprofq(data=temp1, bodysize=10, labsize=10,
        line=dline, style=box);
        %rtfcol(var=name, label=%bquote(Item),
              fmt=$80., just=L,  width=40);
        %rtfcol(var=result, label=%bquote(Value),
               fmt=$80., just=L, width=30);
 %rtfprofo;
```

Using the Adverse Event data set to show a slight variation on the format of a profile section, the following series of macro calls places the title “Adverse Events” in a heading row of the table using spanning headers, and follows by appending the rows of data formatted in the adverse event data step outlined above.

```
%rtfprofq(data=temp15, bodysize=10, labsize=10,
          line=dline, style=box, runhead=y);
         %rtfspan(text=Adverse Events:, bold=y, just=l);
                 %rtfcol(var=aeterm,     label=%bquote(Description (preferred term)),
                         fmt=$250.,   just=L,  width=22);
                 %rtfcol(var=start_stop, label=%bquote(Start/Stop),
                         fmt=$250.,   just=L,  width=16);
                 %rtfcol(var=status,     label=%bquote(Status),
                         fmt=$250.,   just=L,  width=10);
         %rtfspend;
 %rtfprofo;
```

Once all the components of the profile have been arranged within the profile portion of the driving macro, the last step is to terminate the loop that iterates through the master list of subjects and close the rtf document by issuing a call to the macro %Rtfprofe .

```
%end;

%rtfprofe;
```

Conclusion:

The MPS provides a convenient means to produce well organized and flexible subject profiles. The software is user-friendly and provides a rich mix of capabilities to maximize output styles necessitated by input data set structures and end-user specifications. Native features of RTF allow the embedding of bookmarks and hot links into the body of the profile and these features are automatically translated into PDF when the RTF version is converted to PDF using widely available conversion software. Readers are also referred to the Microsoft reference for the RTF language.

Sample Profile

Subject: XXXXX **Treatment Arm:** Placebo **Population:** Safety

Demographics:

Item	Value
Date of Birth:	01JUN1976
Age:	52
Sex:	FEMALE
Race:	White
Height (in.):	78.00
Weight (lb.):	235.00
Is the subject currently a cigarette smoker?	No
Female fertility status:	Sterile
Pregnancy test results (females only):	N.A.
Contraceptive use (females only):	N.A.

Adverse Events:

Description (preferred term)	Start/Stop	Status
DIARRHEA (Diarrhea)	Start 26JUL2004/ Stop 26JUL2004 Study Day: 11	Serious: No Freq: Single episode Sev: Mild Out: Resolved Rel: Probable Action: None
PAINFUL SENSATION IN STOMACH (Dyspepsia)	Start 04AUG2004/ Stop 01AUG2004 Study Day: 8	Serious: No Freq: Single episode Sev: Moderate Out: Resolved Rel: Possible Action: Hosp
STOMACH PAIN (Abdominal pain upper)	Start 28JUL2004/ Stop 28JUL2004 Study Day: 35	Serious: No Freq: Single episode Sev: Mild Out: Resolved Rel: Probable Action: None

Macro Parameters:

Rtfprofi.sas: This macro initializes the RTF file containing the profiles.

Rtfprofi.sas	
Parameters	**Description**
filename=tmp.rtf	The fully qualified name of the output file containing the profiles
land=Y	Specifies whether profiles will be Landscape or Portrait
paper=STAND	Paper size, use LEGAL for legal size
title=	Document information title
author=	Document information author
company=	Document information company
comment=	Document information comment
topm=1.5	Top margin
bottom=1	Bottom margin
leftm=1	Left margin
rightm=1	Right margin
font1=SAS Monospace	Font name for fontstyle 1
family1=fmodern	Font name for fontstyle 1
font2=New Times Roman	Font name for fontstyle 2
family2=froman	Font name for fontstyle 2
font3=Arial	Font name for fontstyle 3
family3=fswiss	Font name for fontstyle 3
font4=Symbol	Font name for fontstyle 4
family4=ftech	Font name for fontstyle 4

%Rtfproft – This macro puts text into to an RTF paragraph via a variety of approaches

- As a normal text paragraph OR as a table with up to 5 cells
- Paragraph formatting/text options are available on individual paragraphs and within table cells.
- If both text and a variable are entered, then issue text followed by the variable.
- Programmer controls the width of table cells.
- Each cell can be specified as text OR as input from a file.
- For input from a file, records can be separated by a carriage return or with blanks.

%Rtfproft	
Parameters	**Description**
width1=	Cell/paragraph width (cell/paragraph 1)
font1=3	Font (cell/paragraph 1)
fonts1=10	Font size (cell/paragraph 1)
text1=	Literal text string (cell/paragraph 1)
bold1=N	Bold (Y/N) (cell/paragraph 1)

%Rtfproft	
Parameters	**Description**
ital1=N	Italicize (Y/N) (cell/paragraph 1)
under1=N	Underline (Y/N) (cell/paragraph 1)
data1=	Data set to populate cell/paragraph 1
var1=	Variable to populate cell/paragraph 1
fmt1=	SAS Software® format for value of *var* (cell/paragraph 1)
break1=Y	Appends /par to end of text in cell (cell/paragraph 1)
find1=0	First indent parameter (cell/paragraph 1)
lmarg1=0	Left margin (cell/paragraph 1)
jst1=C	Cell justification (L/R/C) (cell/paragraph 1)
…	…
width5=	Cell/paragraph width (cell/paragraph 5)
font5=3	Font (cell/paragraph 5)
fonts5=10	Font size (cell/paragraph 5)
text5=	Literal text string (cell/paragraph 5)
bold5=N	Bold (Y/N) (cell/paragraph 5)
ital5=N	Italicize (Y/N) (cell/paragraph 5)
under5=N	Underline (Y/N) (cell/paragraph 5)
data5=	Data set to populate cell/paragraph 5
var5=	Variable to populate cell/paragraph 5
fmt5=	SAS Software format for value of *var* (cell/paragraph 5)
break5=Y	Appends /par to end of text in cell (cell/paragraph 5)
find5=0	First indent parameter (cell/paragraph 5)
lmarg5=0	Left margin (cell/paragraph 5)
jst5=C	Cell justification (L/R/C) (cell/paragraph 5)
btop=N	Controls border around cell/table – Top border (Y/N)
bleft=N	Controls border around cell/table – Left border (Y/N)
bright=N	Controls border around cell/table – Right border (Y/N)
bbot=N	Controls border around cell/table – Bottom border (Y/N)
grid=N	Controls whether grid is displayed (Y/N)
header=N	Controls whether table will occupy header (Y/N)
footer=N	Controls whether table will occupy footer (Y/N)
addsect=N	Controls whether table is inserted in new section (Y/N)

%Rtfprofq – This macro initializes an RTF quick table for the RTF profile system.

%Rtfprofq	
Parameters	**Description**
data=	Input data set name
headskip=N	Skip line after table header (Y/N)
bodyfont=3	Table body Font
bodysize=10	Table body Font size
labfont=3	Column label font
labsize=10	Column label font size

%Rtfprofq	
Parameters	**Description**
labbold=Y	Bold column label (Y/N)
labital=N	Italicize column label (Y/N)
labunder=N	Underline column label (Y/N)
land=Y	Table is landscape (Y/N)
useleft=Y	
runhead=N	Tells the output system to generate headers as running headers (Y/N)
noheadrow=n	Controls the inclusion of a header row on the table (Y/N)
style=lines	Controls the appearance of lines around table/celss (Lines/ Outline/Boxes)
line=	Line variable, new value of line variable triggers line
split=*	Split character for column headers

%Rtfcol – this macro is used to define columns in tables initialized with %Rtfprofq

%Rtfcol	
Parameters	**Description**
var=	Variable to supply data to column
fmt=	SAS Software format
label=	Column label
just=c	Justification (C/R/L)
width=10	Width (tenths of inches)
type=	BY creates a sorted variable
hang=0	Hanging indent (tenths of inches)
context=%bquote((Continued))	Text to be appended to by variable when by group records span two or more pages

%rtfprofo – closes table and is used in conjunction with %rtfprofq. <no parameters >
%Rtfprofp – inserts hard page break. <no parameters >
%Rtfspan – This macro sets up spanning header over columns identified with %Rtfcol.

% Rtfspan	
Parameters	**Description**
text=	Text to serve as spanning header
fontnum=3	Font of spanning header
fontsize=8	Font size of spanning header
bold=Y	Bold spanning header (Y/N)

%Rtfspend – terminates spanning header. < no parameters >
%Rtfprofe – closes RTF document. <No parameters >

Contact Information

John Iwaniszek
Stat-Tech Services, LLC
400 Silver Cedar Ct.
Chapel Hill, NC 27514

Work Phone: 919 929 5015
Fax: 919 928 9320

JIiwaniszek@StatTechServices.com
WWW.StatTechServices.com