

PhUSE 2017

Paper AD09

A Laborsaving SAS[®] Macro for Programmed Patient Narratives

Vineeth Kurnala, Biogen, Cambridge MA, USA

Prafulla Girase, Biogen, Cambridge MA, USA

ABSTRACT

Writing patient narratives is a tedious and time consuming task. There are some off-the-shelf tools to generate patient narratives but they either tend to be very expensive or non-customizable. In 2016 a programmer at Biogen developed a programmatic narrative writing system using SAS[®] and published a paper about it [3]. Although this was very effective and saved hundreds of manual writing hours to the Medical Writing team, the code needed to be adjusted to fit each study. This led to an effort to automate the SAS[®] program by creating a macro such that it can be used on any study with simple modifications to macro parameters. A macro based approach further enhanced efficiency and created a more robust solution. The purpose of this paper is to describe how the patient narratives were automated using a SAS[®] macro. This macro has the flexibility to use SDTM/ADaM/analysis datasets which makes it easier to use across studies following CDISC standards. This macro led to an effort to standardize imputation rules for partial dates in analysis datasets for adverse events and concomitant medications across Biometrics.

INTRODUCTION

ICH Clinical Study Report guidelines (E3), section 12.3.2 has guidance on the contents of a patient narrative.

There should be brief narratives describing each death, each other serious adverse event, and those of the other significant adverse events that are judged to be of special interest because of clinical importance.

In general, the narrative should describe the nature and intensity of event, the clinical course leading up to event, with an indication of timing relevant to test drug/investigational product administration; relevant laboratory measurements, whether the drug was stopped, and when; countermeasures; post mortem findings; investigator's opinion on causality, and sponsor's opinion on causality, if appropriate. In addition, the following information should be included:

- Patient identifier
- Age and sex of patient; general clinical condition of patient, if appropriate
- Disease being treated (if the same for all patients this is not required) with duration (of current episode) of illness
- Relevant concomitant/previous illnesses with details of occurrence/duration
- Relevant concomitant/previous medication with details of dosage
- Test drug/investigational product administered, drug dose, if this varied among patients, and length of time administered.

A SAS[®]-based programmatic system to compose the narratives based on the ICH guidelines was developed at Biogen and published in a paper [3]. The SAS[®]-based programmatic approach was further

enhanced by developing a SAS[®]-based macro application to enable a more robust solution across different studies at Biogen. This paper describes the process of developing the SAS[®]-based macro application.

LONG-ESTABLISHED NARRATIVE PROCESS

Traditionally Medical Writers in various organizations review different sources of information to compose the narratives such as the Council for International Organizations of Medical Sciences (CIOMS) forms, Case Report Forms (CRFs), MedWatch forms, Data Clarification Forms, and Clinical Database Listings. Figure 1 represents a long-established process of writing narratives at Biogen.

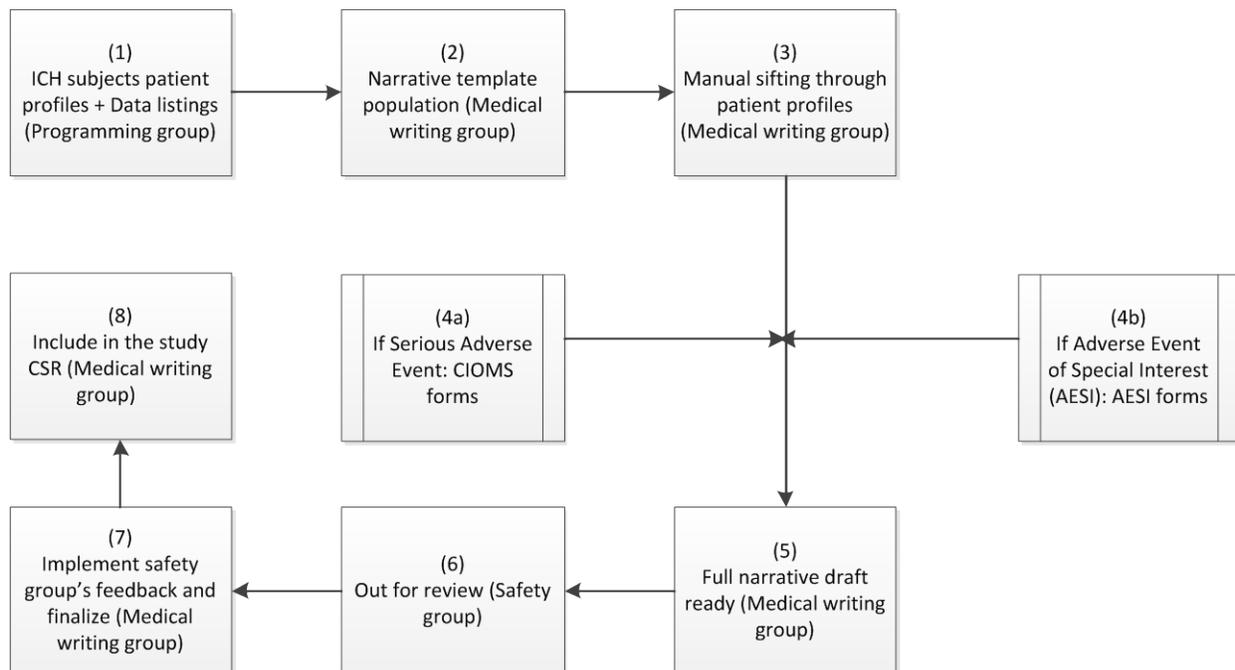


Figure 1: Overview of Traditional Narrative Generation Process [3]

Medical Writers review the data through the patient profiles and patient listings generated by the statistical programming group to compose the narratives as denoted in step (3). They manually go through steps (4a) and (4b) to construct a meaningful patient safety summary. One can imagine the strenuous effort while writing narratives for studies with hundreds or thousands of patients. This task was often could not begin until after the database lock when the data was complete in order to avoid rework. A programmatic system was developed which reduced hours of manual work by automating the steps (1), (2) and (3). The steps (4a) and (4b) could not be automated as it involved clinical judgment and collecting information from free text populated on the forms [3]. However, the programmatic system was designed and tested for a single study having a specific design. Since this approach was proven to be successful, we furthered our thought of developing centralized code that can work for studies having similar design.

A MACRO BASED PROGRAMMATIC SYSTEM

CDISC SDTM and ADaM datasets (as required) were taken into consideration while developing the macro. The identification of patients who needed the narratives was made an integral part of the macro development process. Figure 2 describes the various stages of narrative development with the macro based approach. With this approach the narrative development process was started ahead of the data base lock since the program can be re-run any number of times during the course of study development. The statistical programming group collaborated with Medical Writers early in this effort to identify if there were any Adverse Events of Special Interest (AESI) related to the study. This helped the programming group to derive flags for those patients at the ADaM dataset level in ADAE which in turn saved time for Step (1). Using the derived flags a patient level dataset is created and used as a subset for the macro. This patient subset is fed into Step (2). Step (2) is further detailed in the next section of this paper. Step (3) and (4) remained unchanged due to the need for subject matter expertise of Medical Writers and Drug Safety personnel.

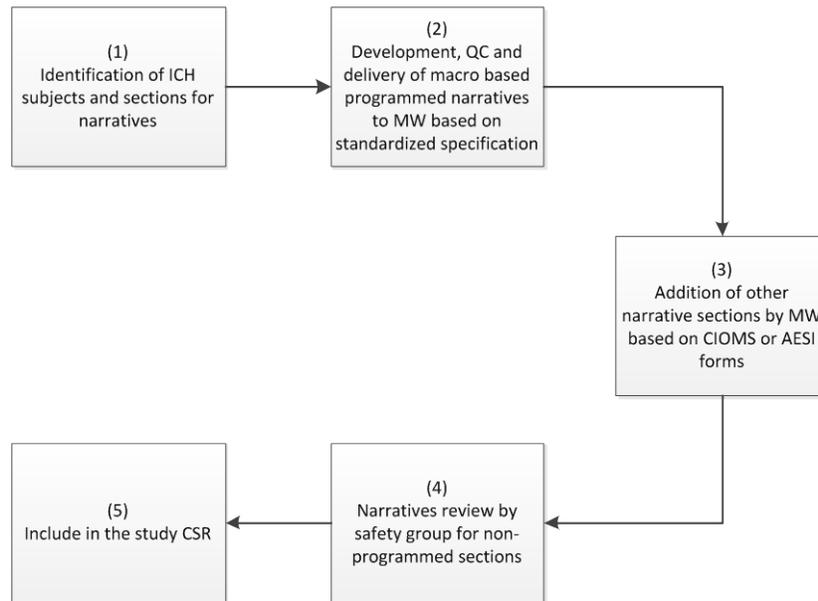


Figure 2: Overview of Macro-based Narrative Generation Process

A detailed programming specification developed previously in the programmatic system was still applicable for the macro based approach. Therefore, it was simply inherited for this project. Studies with a slight variation in the design can utilize the macro with easy updates. For example, consider a study having 2 parts and the CSRs are written separately for each part but the same subjects who completed from Part 1 are enrolled into Part 2. The macro has the capability to add Part 1 study information when composing Part 2 narratives. Display 3 is a specification for such study.

Subject ID: xxx-xxx (<dummy ID>){1.1}
Study Treatment in <Study ID>: <Study treatment label> {1.2}
Treatment Period in <Study ID>: <date of first dose as dd Month yyyy> to <date of last dose as dd Month yyyy> {1.6}

Total Number of <Study treatment label 1> Doses Received in <Study ID>: x {1.7}
Total Number of <Study treatment label 2> Doses Received in <Study ID>: x {1.7}

Death {1.8}
Date of Death: dd Month yyyy {1.9}

{1.3} **Adverse Event:** <Reported term>
MedDRA Term: <Preferred term>
Total Number of <Study treatment label 1> Doses Received at Event Onset: x
Total Number of <Study treatment label 2> Doses Received at Event Onset: x

Display 1: Specification for the structure of narratives for a double-blind or open label study

2. Part 2 – Patient Introduction

2.1. DM.SUBJID

2.2. DM.AGE

2.3. DM.RACE. Note the capitalization here: white, African American, Asian. If race is not reported in the data due to confidentiality reason, then insert "(race not recorded)". If race is other, then insert "(race listed as 'other')".

2.4. DM.SEX values mapped to "male" or "female".

2.4.1. Note that this is used to refer to the patient as He/She for the entire narrative.

2.5. DM.STUDYID

2.6. The text "and began a treatment regimen of" concatenated with treatment description as described in section {1.2}

Display 2: Specification of algorithms for a double-blind or open label study

Subject ID: xxx-xxx (<dummy ID>){1.1}
Study Treatment in <Study ID> Part 2: <Study treatment label> {1.2}
Treatment Period in <Study ID> Part 2: <date of first dose as dd Month yyyy> to <date of last dose as dd Month yyyy> {1.6}

Total Number of Days on <Study Treatment 1> in <Study ID> Part 1: x {1.7}
Total Number of Days on <Study Treatment 1> in <Study ID> Part 2: x {1.7}

Death {1.8}
Date of Death: dd Month yyyy {1.9}

{1.3} **Adverse Event:** <Reported term>
MedDRA Term: <Preferred term>
Total Number of Days on <Study treatment label 1> at Event Onset: x {1.4}
Total Number of <Study treatment label 2> Doses Received at Event Onset: x

Display 3: Specification for structure of narratives for studies having different parts.

In this case, the primary information is from Part 2 data but the Medical Writer wanted to address some key aspects from Part 1 of the study. Two new rows were added as show in {1.7}. The variable values that are obtained from a study dataset were accessed through the information saved in the macro variables.

DESIGN OF SAS® MACRO FOR PATIENT NARRATIVES

The SDTM or ADaM variables were directly called in the SAS® code in the previous programmatic system. In the macro based approach we defined the study variables as macro variables. Consider the following example: “*Subject 0000, a 50-year-old white female, was enrolled into Study 111XX000 and began a treatment regimen of Drug A once every 4 weeks and placebo once weekly on 10 January 2012 (Day 1)*”. In the programmatic system the underlined text was obtained by calling the SDTM/ADaM variable names in the text assignment variable as follows.

```
text = "Subject "||strip(subjid)||", a "||strip(agec)||"-year-old "||strip(lowercase(race))||" "||strip(lowercase(sexc))||
      ", was enrolled into Study "||strip(studyid)||" and began a treatment regimen of "||strip(arm)||"on "||
      strip(rfststc)||" (Day "||strip(rfstdyc)||").";
```

In the macro based approach most of these study variables are assigned to macro variables and a list of all macro variables required for the code are maintained in a driver file (3) “narratives-<studyid>.sas” as shown in Figure 3. The variables age, race and sex are called directly in the macro program without creating additional macro variables as their variable names are standard across SDTM or ADaM datasets in our organization. The patient level dataset (1) was created with patients requiring narratives based on ICH/AESI criteria is fed into the driver file as a macro variable and the main macro (2) “m-narratives.sas” is called at the end of the driver file. Using these two inputs, the main macro program will generate the output (4) in the RTF file format.

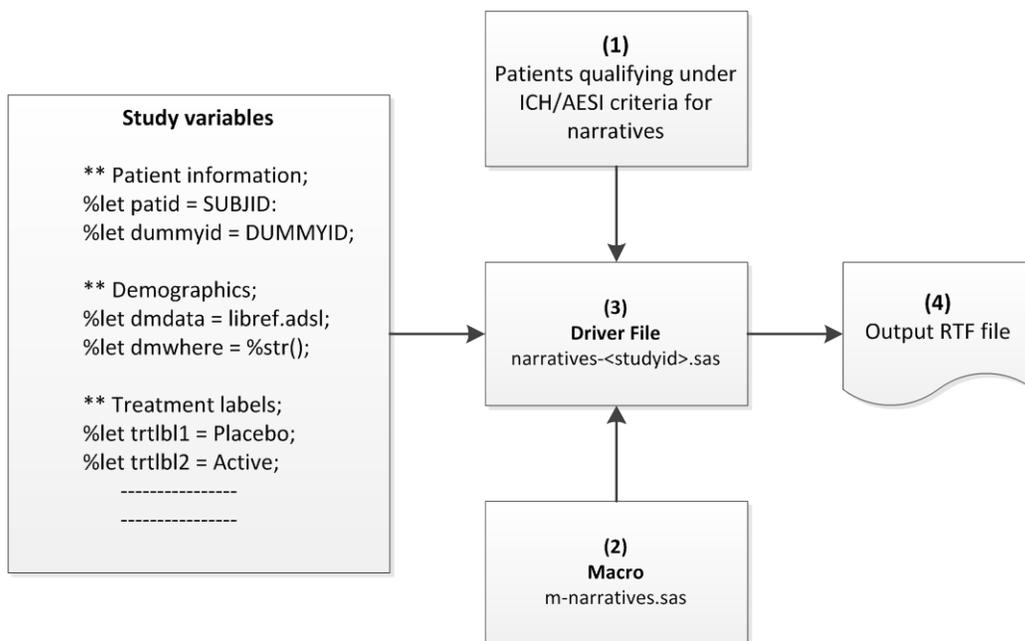


Figure 3: Design of SAS® macro for programmed patient narratives

The entire structure of the output is divided into 2 parts as shown in Display 1 and 2. Part 1 is called the “header” and Part 2 is called the “body” of the output, both separated by a line break. The header was designed to be more flexible so that any key study level information can be added to the output. Display 3 is the specification for a default header output of the macro but additional information can also be added. Display 4 shows the customized specification on the left side and the associated driver file “narratives- <studyid>.sas” on the right side. The default header and custom header specifications are marked in Display 4 for clarity.

```

Subject ID: xxx-xxx [<dummy ID>]{1.1}
Study Treatment in <Study ID>: <Study treatment label> {1.2}
Treatment Period in <Study ID>: <date of first dose as dd Month yyyy> to <date of last dose as dd
Month yyyy> {1.6}

Total Number of <Study treatment label 1> Doses Received in <Study ID>: x {1.7}
Total Number of <Study treatment label 2> Doses Received in <Study ID>: x {1.7}

Death {1.8}
Date of Death: dd Month yyyy {1.9}

{1.3} Adverse Event: <Reported term>
MedDRA Term: <Preferred term>
Total Number of <Study treatment label 1> Doses Received at Event Onset: x
Total Number of <Study treatment label 2> Doses Received at Event Onset: x

XXXX Use Prior to Study Drug Administration: x {1.10}

XXXXX XXXX Score: {1.11}
Visit: Score
Screening: xx
Baseline: xx

XXXXXX Assessment: {1.12}
Visit: Evaluator, Result
Baseline: R1, xx,xxx.xx
24 Hours: R1, xx,xxx.xx
24 Hours: R2, xx,xxx.xx

Time from XXXX to Treatment: xxx {1.13}
Baseline XXXXX Category: xxx {1.14}

```

```

*** Any additional information (from source 1);
%let add1data = libref.adsl;
%let add1where = ;

%let add1var1 = <variable name>;
%let add1var2 = ;
%let add1var3 = ;
%let add1var4 = ;
%let add1var5 = ;
%let add1var6 = ;

%let add1lbl1 = XXXX use prior to study drug administration;
%let add1lbl2 = ;
%let add1lbl3 = ;
%let add1lbl4 = ;
%let add1lbl5 = ;
%let add1lbl6 = ;

*** Any additional information (from source 2);
%let add2data = libref.adxxxx;
%let add2where = %str(paramcd = 'XXXXXX' and aval ne .);
%let add2day = avisit;
%let add2srt = avisitn;
%let add2var1 = aval;
%let add2var2 = ;
%let add2lbl1 = XXXXX XXXX Score;
%let add2s1bl = %str(Visit: Score);

*** Any additional information (from source 3);
%let add3data = libref.xx;
%let add3where = %str(xtstod = 'XXXXX' and xkorres ne ' ');
%let add3day = visit;
%let add3srt = xxseq;
%let add3var1 = xxeval;
%let add3var2 = xkorres;
%let add3lbl1 = XXXXXX Assessment;
%let add3s1bl = %str(Visit: Evaluator, Result);

*** Any additional info that needs to be appended (from source 4);
%let add4data = libref.adsl;
%let add4where = %str();

%let add4var1 = <variable name 1>;
%let add4var2 = <variable name 2>;
%let add4var3 = ;
%let add4var4 = ;
%let add4var5 = ;
%let add4var6 = ;

%let add4lbl1 = Time from XXXX to Treatment;
%let add4lbl2 = Baseline XXXXX Category;
%let add4lbl3 = ;
%let add4lbl4 = ;
%let add4lbl5 = ;
%let add4lbl6 = ;

```

Display 4: Customized specification for header part of patient narratives

We have added 4 different sections of macro variables for this type of additional information from different sources. For example, (1) %let add1data= requires the programmer to specify the name of the source dataset. %let add1where= is optional for any conditions. %let add1var1= to %let add1var6= are macro variables for variable names from the source specified previously in %let add1data=. %let add1lbl1= to %let add1lbl6= are corresponding text labels that are to be displayed as per specification. As many as 6 rows with 6 different variable values from the same source dataset can be added in this section. This can be extended to any number of rows within the same source dataset with easy adjustments.

(2) and (3) from Display 4 are constructed differently than (1), (4) and (5). Using these sections, we can print the patient data up to 3 levels of variables. For example, in (3), the patient data is displayed by Visit, Evaluator initials, and Result of an assessment. Section (4) is an example for displaying 2 different rows with values from 2 different variables from the same source dataset. After the statistical programmer fills in all the necessary information in the macro variables the main macro is called at the end of the driver file. A sample output of the customized header information is shown in Display 5.

<p>Subject ID: 111-111 Study Treatment in XXXXX: Placebo Treatment Period in XXXXX: 10 June 2020 to 11 June 2020 Total Number of Placebo Doses Received in XXXXXX: 40 Serious Adverse Event: Some AE MedDRA Term: SOME AE Total Number of Placebo Doses Received at Event Onset: 20</p> <p>XXXX use prior to study drug administration: Yes XXXXX XXXX Score: Visit: Score Screening: 1 Baseline: 2 xx Hours: 3 Day xx: 4 Day yy: 5 Day zz/Final Vst: 5</p>	<p>XXXXXX Assessment: Visit: Evaluator, Result Baseline: X1, 21,000.11 Baseline: X2, 21,001.11 xx Hours: X1, NA yy Hours: X2, NA Day xx: X1, NA Day xx: X2, NA Day yy: X1, 50,000.22 Day zz: X2, 50,023.22</p> <p>Time from XXXX to Treatment: >x to <= x.x hours Baseline XXXXX Category: <=2</p>
---	---

Display 5: Custom output from the narratives macro program

The specification for “body” of the output was kept the same as done in the original programmatic system [3]. Display 6 shows the specification.

<p>Subject xxx-xxx (<dummy ID>) {2.1}, a xx {2.2}-year-old white (race not recorded){2.3} female {2.4}, was enrolled into Study <Study ID> {2.5} and began a treatment regimen of <study treatment label 1 and study treatment label 2> {2.6} on dd Month yyyy {2.7} (Day 1 {2.8}).</p> <p>{3.1}Adverse Event: (coded term [reported term]): SOME AE {3.3}[SOME AE ON SOME SIDE OF THE BODY {3.3}] SAE Number: XXXXXXXXXX {3.4}</p> <p>On dd Month yyyy {4.1} (Day xx {4.2}), an adverse event of some ae {4.4} was reported, which was classified as a serious or non-serious adverse event {4.6} by the Investigator. The subject had received <number> doses of <Study treatment label 1> or <Study treatment label 2> before onset of the event, with his or her most recent dose given on <dd Month yyyy (Day xx)> {4.7}.</p>	<pre> %let study = %upcase(%Protocol); ** {2.5}; %let patid = subjid; ** {2.1}; %let dummyid = dummyid; ** {2.1}; %let trtval1 = %str(0); %let trtval2 = %str(1); %let trtlbl1 = %str(Placebo); ** {2.6}; %let trtlbl2 = %str(XXXXXXX); ** {2.6}; *** Treatment start and stop dates; %let trtstart = trtsdt; ** {2.7}; %let trtstop = trtedt; ***** ** Adverse Events ***** %let aedata = libref.adae; ** {3.1}; %let aestart = astdt; ** {4.1}; %let aeend = aendt; %let aestday = astdy; ** {4.2}; %let aeenday = aendy; </pre>
--	--

Display 6: Specification (left) and macro variables (right) for “body” of the output

As shown in Display 6 the references beside the text in the specification have corresponding macro variables on the right-hand side that a programmer should assign. We didn’t create separate macro variables for {2.2}, {2.3}, {2.4} as they are standard across all studies in SDTM/ADaM datasets at Biogen. Similarly, {4.4} is standard AEDECOD variable in the SDTM dataset and {4.6} is directly derived from AESER from the SDTM or ADaM AE dataset as specified in {3.1}.

The output of the specification from Display 6 is shown in Display 7. The number of doses that the patient received is calculated by the macro. The macro also has the capability to calculate the exposure duration instead of dose accountability.

Subject 111-111, a xx-year-old white male, was enrolled into Study XXXXXX and began a treatment regimen of Placebo on 10 June 2020 (Day 1).

Serious Adverse Event: (coded term [reported term]): SOME AE [SOME AE]
SAE Number: xxxxxxxxxxxx

On 08 September 2020 (Day xx), an adverse event of some AE was reported, which was classified as a serious adverse event by the Investigator. The subject had received x doses of Placebo before onset of the event.

Display 7: Sample output of specification from Display 6

The number of doses calculation in the underlined part is needed based on the onset of the event. The macro calculates the number of doses received before each onset of the event by interleaving the non-missing dose records from SDTM EX dataset with the Adverse Events dataset. Calculation is needed at each event because the events that are needed for the narrative is automatically selected based on the ICH/AESI criteria. Display 8 shows the code for this algorithm.

```

%if &excntyn = Y %then
%do;
  data exae;
    set libref.ex;
    xxstdt = exstdt;
    ord = 1;
    run;
%end;
  data ae;
    set libref.adae;
    ord = 2;
    xxstdt = aestdt;
    run;
  ** <sort both datasets by SUBJID, XXSTDT, ORD>
  data aex;
    set ae exae;
    by subjid xxstdt ord;
    retain doscnt 0;
    if first.subject then doscnt=0;
    if ord = 1 then doscnt = doscnt+1;
    run;

```

Display 8: Calculation of number of doses before the onset of an event

CONCLUSION

The macro based narrative was considered very successful and was utilized across many studies with similar design. The automation allowed us to output additional information in the header part which gave much flexibility to the Medical Writers to incorporate key end point information and saved hundreds of manual typing hours. While the automation opened up more opportunities for various studies it was only possible in studies with data organized by the CDISC standards. There were some aspects of the patient data such as analyzing “what adverse event has caused a trigger to the laboratory result abnormalities?” were not addressed by the macro and needed human interpretation. At the same time, the macro has helped us eliminate any inaccuracies and typographical errors caused by manual human intervention. The

macro development effort was possible because of strong collaboration between statistical programming, statistics, medical writing, and drug safety functions. In conclusion, automation tools similar to these can be developed using SAS[®] or other analytical applications by a mutual understanding of expectations between different functions within an organization.

REFERENCES

1. ICH Harmonized Tripartite Guideline: Structure and Content of Clinical Study Reports E3. Current Step 4 version dated 30 November 1995. Page 24.
2. SAS[®] 9.3 Language Reference:
<https://support.sas.com/documentation/cdl/en/lrcon/65287/HTML/default/viewer.htm#titlepage.htm>
3. Girase P, Boisvert D, Kurnala V, “Programmed Patient Narratives Using SAS[®],”PhUSE 2016, Paper AD09.

CONTACT INFORMATION

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

Vineeth Kurnala
Senior Analyst III, Statistical Programming
Biogen
300 Binney St.
Cambridge MA 02142
Email: vineeth.kurnala@biogen.com

Prafulla S Girase
Principal Analyst, Statistical Programming
Biogen
300 Binney St.
Cambridge MA 02142
Email: prafulla.girase@biogen.com

SAS[®] and all other SAS[®] Institute Inc. product or service names are registered trademarks or trademarks of SAS[®] Institute Inc. in the USA and other countries. ® indicates USA registration.

Other brand and product names are trademarks of their respective companies.