PhUSE 2016

Paper AD09

# Programmed Patient Narratives Using SAS®

Prafulla Girase, Biogen, Cambridge, USA
Daniel Boisvert, Biogen, Cambridge, USA
Vineeth Kurnala, Biogen, Cambridge, USA

## ABSTRACT

The process of writing patient narratives for a Clinical Study Report (CSR) can be very time consuming. It generally involves manual review of data listings, patient profiles etc. followed by manual writing of narratives as plain English sentences. The off-the-shelf tools were explored for automation but customization required in the narrative contents prevented its use. Therefore, a SAS-based programmed approach was developed in a large Phase III clinical trial to automate creation of a major portion of the narratives. This approach saved hundreds of hours of manual work writing narratives for this study. It also became a foundation for developing a more robust and re-usable SAS-based macro application for future use. This paper describes the programmed approach along with some important snippets of SAS code. The paper will provide helpful insights to others who would like to automate creation of patient narratives.

## INTRODUCTION

The ICH E3 Clinical Study Report guidelines require that there be a brief narrative describing each death, each other serious adverse event, and those of the other significant events that are judged to be of special interest because of clinical importance [1]. As per these guidelines, each narrative should describe the nature and intensity of event, the clinical course leading up to event, relevant laboratory measurements, whether the drug was stopped, and when [1]. In addition, it should include subject identifier, age and sex of subject, disease being treated, relevant concomitant illnesses, relevant concomitant medications, and test drug/investigational product administered [1]. The purpose of this paper is to describe the SAS-based programmed approach to author narratives that meet these guidelines as we have implemented at Biogen.

## TRADITIONAL PROCESS OF NARRATIVE GENERATION

The process of generating narratives can vary in each organization based on the organizational structure, operating model, and resourcing. The following figure describes the traditional process of narrative generation within our organization.

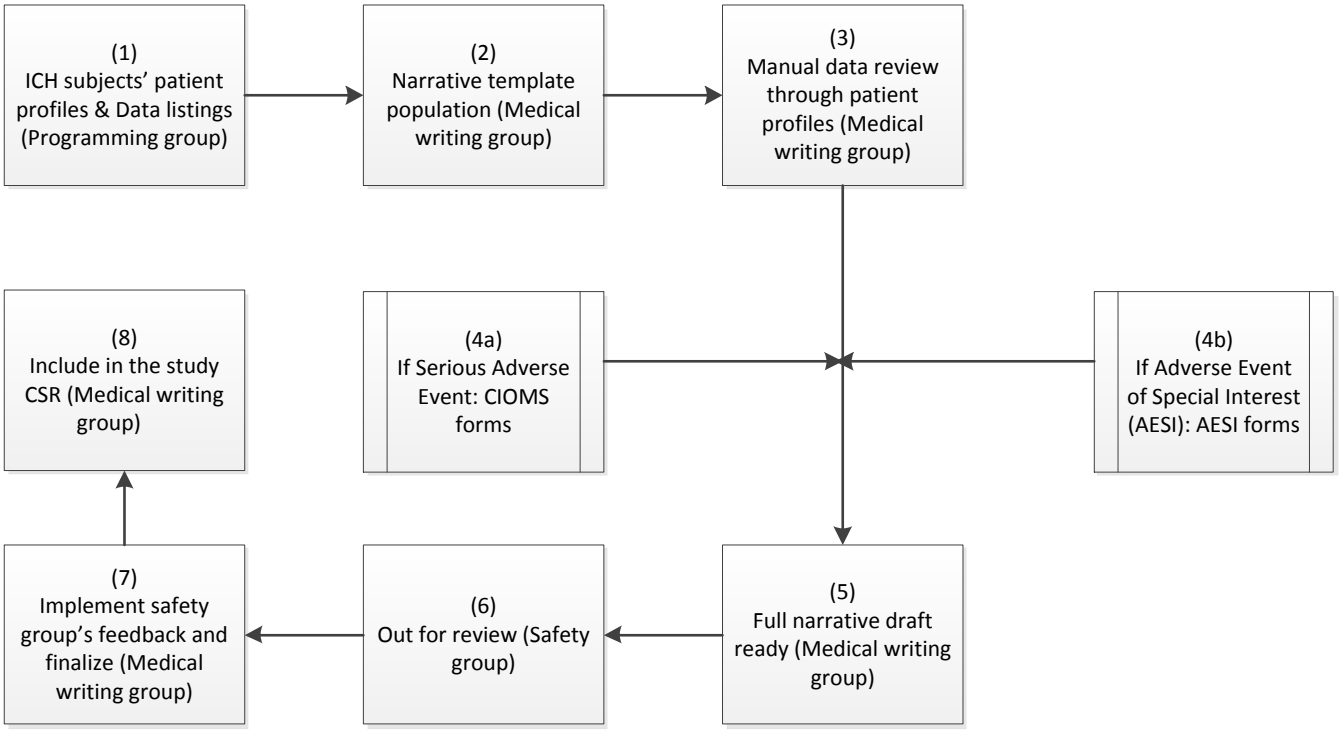



**Figure 1. Overview of Traditional Narrative Generation Process**

As noted in box (3) above, the medical writing (MW) group has to do manual review of patient profiles and data listings, extract the data and then manually write narrative sentences in plain English. For 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 this example, the information required for underlined text was manually read from patient profiles and/or data listings and then written by the writer. As one can imagine, this is a very resource and time intensive task and one that occurs in the critical time after database lock and prior to CSR finalization. The concept described in this paper is about automating the step of reading various data points and writing narrative text. The steps (4a) and (4b) could not be automated as it involved clinical judgment and collecting information from free text populated on the forms.

## NEW PROCESS WITH PROGRAMMED APPROACH

The following figure describes the new process of narrative generation with programmed approach.

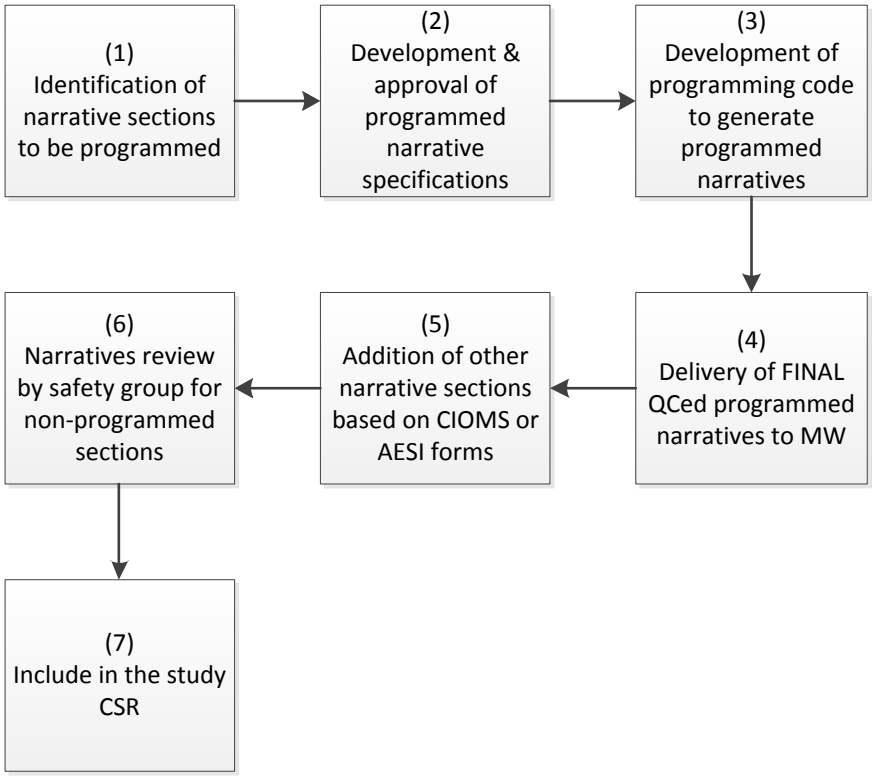



**Figure 2. Overview of Narrative Generation Process with Programmed Approach**

The traditional process of reading patient profiles and data listings and then manually writing narratives has a benefit of clinical judgment which cannot be automated. Secondly, some of the sections of narratives may be based on sources that are not part of study's clinical database (e.g. CIOMS forms for serious adverse events) which also could not be automated. The project team wanted to avoid mixing of programmed sentences with sentences written manually. The goal was to keep programmed narrative section intact without any manual intervention. As a result, the traditional presentation format of narratives used in the past could no longer be used. Reaching a cross-functional agreement on the programmed narrative specifications and its presentation format was very critical for the success of this project. The step of developing specifications for the programmed narratives is described separately in this paper.

## SPECIFICATIONS FOR THE NARRATIVES

Below are examples of the specification format that was developed for this project. There are two critical components in the specification. The first component is the template that generically describes the presentation format of programmed narrative. The second component is the programming specification for the narrative text. The Display 1 below is a sample of portion of narrative specification that describes the template for presentation format of narrative. The orange font indicates alternative narrative sentence. The green font indicates a hyperlink that navigated to programming specification for the respective narrative text. The sample of programming specification portion is presented in Display 2. This specification format could be used for cross functional agreement on the narrative presentation format and it also served as specifications to programmers who developed production and QC code.

**Subject ID:** <dummy ID> (xxx-xxx) {1.1}
**Study Treatment in 111XXX000:** Drug A 100 times a week {1.2}
**Treatment Period in 111XXX000:** <date first dose as dd month yyyy> to <date last dose as dd month yyyy> {1.6}
**Total Number of Drug A Doses Received in 111XXX000:** x {1.7}
**Total Number of Placebo Doses Received in 111XXX000:** x

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

{ 1.3 } **Adverse Event:** <Reported term>
**MedDRA Term:** <Preferred term>
**Total Number of Drug A Doses Received at Event Onset** {1.4}**:** x {1.5}
**Total Number of Placebo Doses Received at Event Onset:** x

---

Subject <dummy ID> (xxx-xxx) {2.1}, a 48 {2.2}-year-old white (race not recorded) {2.3} female {2.4}, was enrolled into Study 111XXX000 {2.5} and began a treatment regimen of placebo 100 times a weeks {2.6} on 11 May 2010 {2.7} (Day 1 {2.8}).
No medical history was reported for this subject at the time of study enrollment {2.9}. The subject's reported medical history at the time of study enrollment included (verbatim terms) allergy to tv, movies. Allergies included (verbatim terms) allergy to soap operas, action movies. **No allergies were reported**.

**Display 1. Sample of Template Portion of Narrative Specifications.**

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}
   - 2.7. DM.RFSTDTC

**Display 2. Sample of Programming Specification**.

## PICTORIAL OVERVIEW OF FLOW WITHIN THE SAS PROGRAM

Figure 3 below provides an overview of flow within the SAS program. The green font is used for CDISC SDTM/ADaM datasets and orange font is used for temporary datasets created within the program. Some detailed programming steps have been excluded from the flow diagram. The initial project plan was to only use CDISC SDTM datasets as a source to make this project more re-usable for other studies. However, to ensure consistency with certain derivations done in the CDISC ADaM datasets for CSR outputs, two ADaM data sources (ADSL and ADAE) were added.

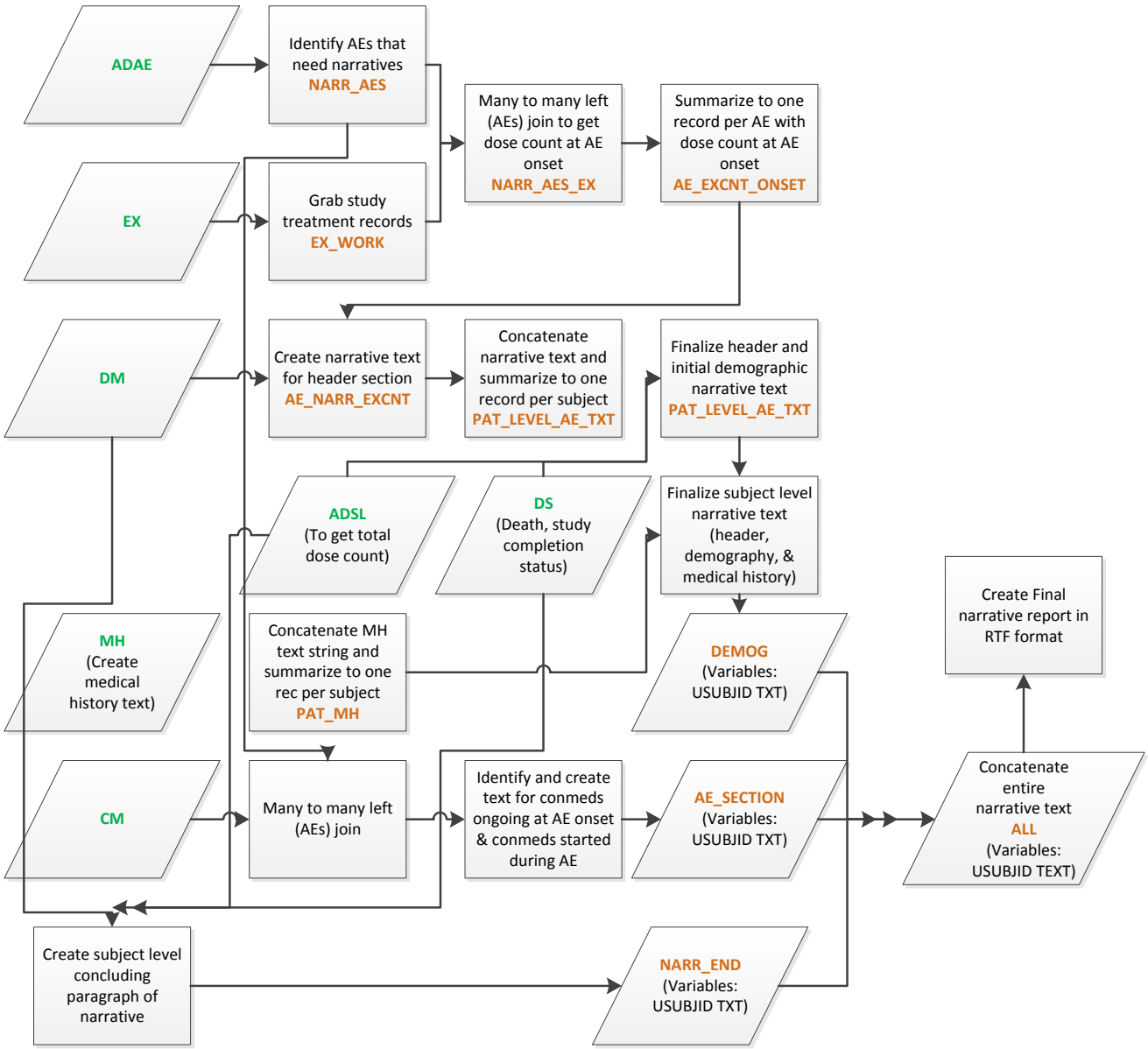
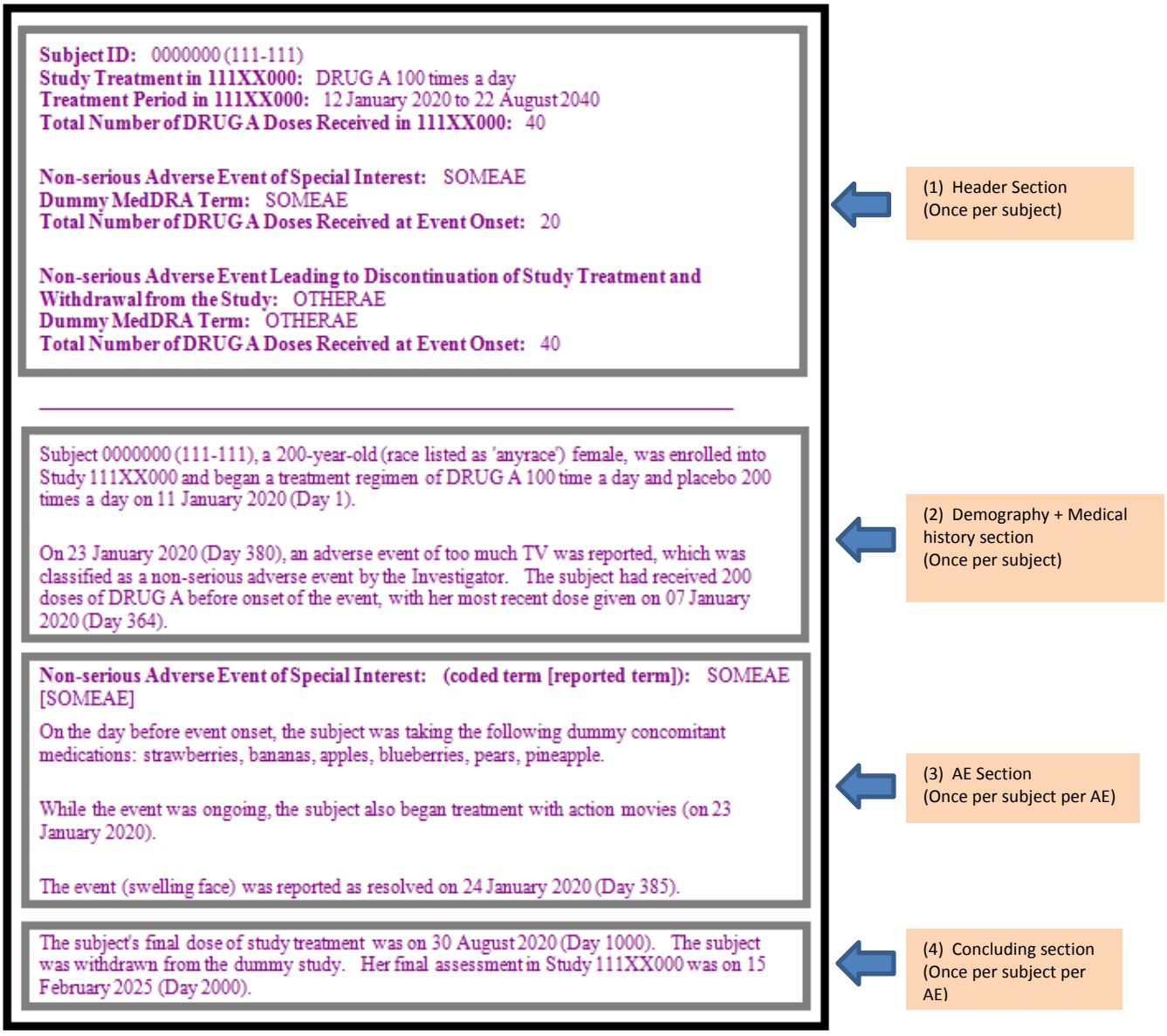



**Figure 3. Pictorial Overview of Flow Within SAS Program**

## NARRATIVE OUTPUT

Display 3 below shows a dummy sample of programmed narrative output for a dummy subject. As described in Figure 2, Medical Writing added additional narratives text that is not displayed here. Medical Writing did not want to make any modification to programmed output. Therefore, a different (purple) font was used in the programmed output to distinguish it from narrative text added later. The font color was changed back to black font once narratives were final. This simple changed increased the usability and integrity of the automated sections immensely. Narrative writers could easily see which text was not to be touched. The header section of programmed output describes subject's treatment information along with list of all adverse events for that subject which requires a narrative section. The second section describes subject's demographic and medical history. The third section is written for each adverse event along with information related to that adverse event. The last concluding section describes subject's last dose date, study completion status, and date of final assessment in the study. As discussed earlier in this paper, reaching an agreement to this presentation format was a critical component to success of this project.




**Display 3. Sample of Narrative Output**

## SNIPPETS OF PROGRAMMING CODE

The following global macro parameters were created. These were used in the narrative sentences to influence font boldness, carriage return in the narrative output.

```
%global bon boff cr ;
%let bon = ^S={font_weight=bold}  ;
%let boff = ^S={} ;
%let cr = ^-2n ;
```

Below is a sample of how sentences were built.

```
ae_header_txt = "&cr.&cr.&bon." || strip(ser_cat) || "  " || "(coded term [reported term]):
" || "&boff." || strip(aedecod) || " [" || strip(aeterm) || "]"  ;
```

The date presentation in the narrative needed to be in the format “dd Month yyyy”. A small macro was written as below to be used in the program for date format conversion.

```
%macro narrdt(indt=, outdt=);
  attrib &outdt. length=$17 ;
```

```
  _outdt = compress(strip(substr(&indt.,9,2)) || " " ||
strip(put(substr(&indt.,6,2),$monfmt.)) ||
" " ||
strip(substr(&indt.,1,4)), '-') ;
  &outdt. = _outdt ;
  drop _outdt ;
%mend narrdt ;
```

Initially, the entire narrative text for a subject was kept in a single variable on a single record. However, since the entire narrative for a subject was output in RTF file as one cell, it resulted in output truncation in instances where the cell was longer than the page. This issue was resolved by breaking the narrative output for a subject into multiple records. Below is a code that was used for this step. This step created a new record every time there was double carriage return in the narrative text.

```
data narr_eos_subj_m (keep=usubjid: text2 aedecod aestdtc aeendtc);
  attrib text1 length=$30000 text2 length=$30000;
  set here.narr_eos_subj;

  text1 = TRANWRD(text,"^-2n^-2n",'|');

  i=1;
  DO WHILE (SCAN(text1,i,"|") NE '');
    if i=1 and substr(text1,1,1) = "|" then text2= strip("|") || strip(SCAN(text1,i,"|")) ;
    else if scan(text1,i+1,"|") = '' and
      substr(reverse(text1),1) EQ "|" then text2 =   strip(SCAN(text1,i,"|")) || strip("|") ;
    else if scan(text1,i+1,"|") = '' and
      substr(reverse(text1),1) NE "|" then text2 = SCAN(text1,i,"|") ;
    else text2= strip(SCAN(text1,i,"|")) || strip("|") ;
    OUTPUT;
   i=i+1;
  END;

run ;
```

The MW team needed programmed narrative outputs in batches of subjects. The following macro was developed within the program to create RTF output in batches.

```
%macro narr_rpt ;
 %do i = 1 %to &batchcnt. ;
  ods noresults;
  ods listing close;
  ods escapechar='^';
  options nobyline nodate nonumber leftmargin=1in rightmargin=1in
          topmargin=1in bottommargin=1in ps=45;
  title;
  footnote;
  ods rtf file="narratives-111xx000-final-&&first_subj&i..to&&last_subj&i..-batch&&batchnum&i...rtf" ;
  proc report nowindows missing nofs style = {frame=void rules=none}
              STYLE(header column)=[foreground=purple] data = narr_eos_subj_m;
    where batchnum = &&batchnum&i ;
    columns usubjid text brk_var;
    define usubjid / order order=data noprint;
    define text / order=data width=5 left " " flow
           style=[backgroundcolor=white font_style=roman font_size=11pt];
    define brk_var / order order=internal noprint ;
    break after brk_var / page ;
  run;
  ods rtf close;
  ods listing;
  ods results;
 %end ;
%mend narr_rpt ;
```

## CONCLUSION

In the end this automated narrative process was very successful and resulted in saving hundreds of hours that would have required otherwise. The process was successfully repeated on multiple studies. Using this process as a foundation, we also developed a more robust and reusable SAS-based macro application. The discussion of the macro is beyond the scope of this

paper. It should be noted that the automation is only as good as the data. While a human would be able to interpret data and tell a story, the automated approach simply prints the data. Conversely, a human interpreting data can misinterpret it and make transcription errors. Overall, we have found that the time savings and accuracy improvements greatly outweigh the risks.

## 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


## CONTACT INFORMATION

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


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

Daniel Boisvert
Senior Principal Analyst, Data Analysis & Infrastructure
Biogen
300 Binney St.
Cambridge MA 02142
Email: Daniel.Boisvert@biogenidec.com

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