PharmaSUG 2022 - Paper AD-131

SAS Macro to Colorize and Track Changes Between Data Transfers in Subject-Level Safety Listings

Inka M Leprince, Elizabeth Li, and Carl C Chesbrough, PharmaStat, LLC

Abstract

Most clinical trials have an independent data monitoring committee (DMC). In order to provide quality data to the DMC, sponsors conduct internal reviews of available data prior to each DMC meeting. Subject-level listings are used to aid the clinical data review of the ongoing studies. To ease the burden on the data reviewers, providing color coding for old/deleted records from the previous data transfer and highlighting new records from the current data transfer is increasingly requested by sponsors. We have developed a SAS macro, TrackCHG, to meet this request. This paper describes the SAS macro code of TrackCHG in detail. It also provides sample datasets and code for the reader to create an adverse event listing using the TrackCHG macro. After reading this paper, SAS users should be able understand and implement the TrackCHG macro into their existing listing programs to color code their output listings.

Introduction

Most clinical trials have an independent data monitoring committee (DMC) who reviews accumulating data from clinical trials on an ongoing basis. In order to provide quality data to the DMC, sponsors conduct an internal review of available data prior to each DMC meeting. Ongoing clinical data reviews rely on subject-level listings in conjunction with data management activities. This review task becomes onerous as the amount of data accumulates and the number of data transfers increases.

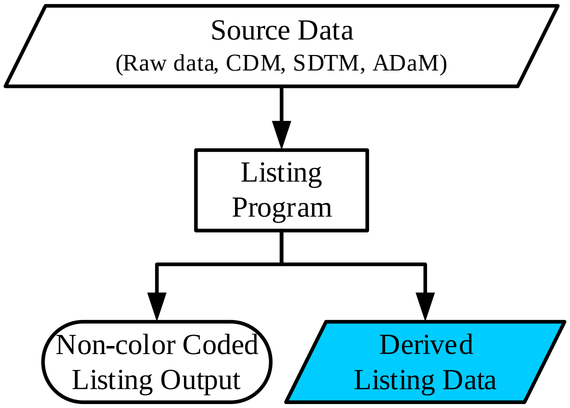
Color-coded subject-level listings are one of the tools to reduce the workload and increase the efficiency for ongoing clinical data review. Color-coded records highlight the differences between two versions of the data and allow reviewers to focus on the changes between the versions. We have developed a SAS macro, TrackCHG, that can be easily integrated with existing listing generation programs to programmatically apply customized text formats to records.

First, we present how the TrackCHG macro fits into the workflow of creating subject-level listings. Then, we walk through the SAS macro code as it performs the following functions: 1) record the date stamp of the current data, 2) use a PROC SORT step and DATA MERGE step to compare the previously derived data records against the current derived data records, 3) apply a text format to color code the new and old records in a customizable fashion. Finally, we provide sample datasets and example SAS code to create an adverse event listing using the TrackCHG macro.

Workflow of Creating a Color-Coded Subject-Level Listing

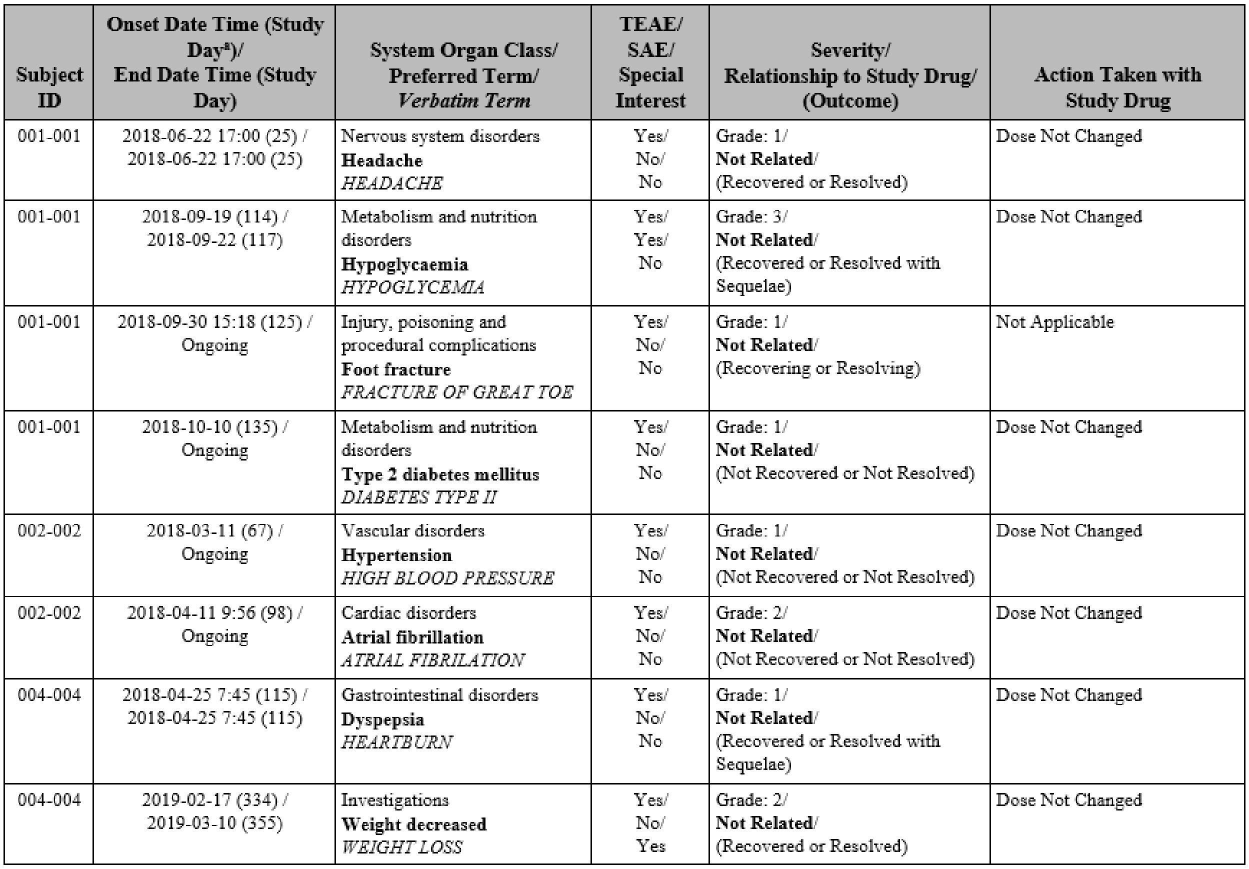
Figure 1 Flow Chart of Generation of Listings shows our workflow to generate a non-color-coded subject-level listing output at PharmaStat.

Figure 1 Flow Chart of Generation of Listings



The listing program reads the source data, processes the data, and saves a derived listing dataset (also referred to as AFILE) before calling a PROC REPORT procedure to generate an output (Figure 2). The source data can be one or more clinical data management (CDM) datasets, study data tabulation model (SDTM) datasets, or analysis data model (ADaM) datasets.

Figure 2 Sample Listing of Adverse Events



Subject ID	Onset Date Time (Study Day[a])/ End Date Time (Study Day)	System Organ Class/ Preferred Term/ *Verbatim Term*	TEAE/ SAE/ Special Interest	Severity/ Relationship to Study Drug/ (Outcome)	Action Taken with Study Drug
001-001	2018-06-22 17:00 (25) / 2018-06-22 17:00 (25)	Nervous system disorders **Headache** *HEADACHE*	Yes/ No/ No	Grade: 1/ **Not Related**/ (Recovered or Resolved)	Dose Not Changed
001-001	2018-09-19 (114) / 2018-09-22 (117)	Metabolism and nutrition disorders **Hypoglycaemia** *HYPOGLYCEMIA*	Yes/ Yes/ No	Grade: 3/ **Not Related**/ (Recovered or Resolved with Sequelae)	Dose Not Changed
001-001	2018-09-30 15:18 (125) / Ongoing	Injury, poisoning and procedural complications **Foot fracture** *FRACTURE OF GREAT TOE*	Yes/ No/ No	Grade: 1/ **Not Related**/ (Recovering or Resolving)	Not Applicable
001-001	2018-10-10 (135) / Ongoing	Metabolism and nutrition disorders **Type 2 diabetes mellitus** *DIABETES TYPE II*	Yes/ No/ No	Grade: 1/ **Not Related**/ (Not Recovered or Not Resolved)	Dose Not Changed
002-002	2018-03-11 (67) / Ongoing	Vascular disorders **Hypertension** *HIGH BLOOD PRESSURE*	Yes/ No/ No	Grade: 1/ **Not Related**/ (Not Recovered or Not Resolved)	Dose Not Changed
002-002	2018-04-11 9:56 (98) / Ongoing	Cardiac disorders **Atrial fibrillation** *ATRIAL FIBRILATION*	Yes/ No/ No	Grade: 2/ **Not Related**/ (Not Recovered or Not Resolved)	Dose Not Changed
004-004	2018-04-25 7:45 (115) / 2018-04-25 7:45 (115)	Gastrointestinal disorders **Dyspepsia** *HEARTBURN*	Yes/ No/ No	Grade: 1/ **Not Related**/ (Recovered or Resolved with Sequelae)	Dose Not Changed
004-004	2019-02-17 (334) / 2019-03-10 (355)	Investigations **Weight decreased** *WEIGHT LOSS*	Yes/ No/ Yes	Grade: 2/ **Not Related**/ (Recovered or Resolved)	Dose Not Changed

In order to generate color-coded subject-level listings, extra steps need to be taken. This process has been consolidated into the TrackCHG macro. The TrackCHG macro is designed to be incorporated inside a listing program, right before the PROC REPORT procedure. Figure 3 shows the process and Figure 4 shows the colorized output.

Figure 3 Flow Chart of Generation of Color-Coded Listing

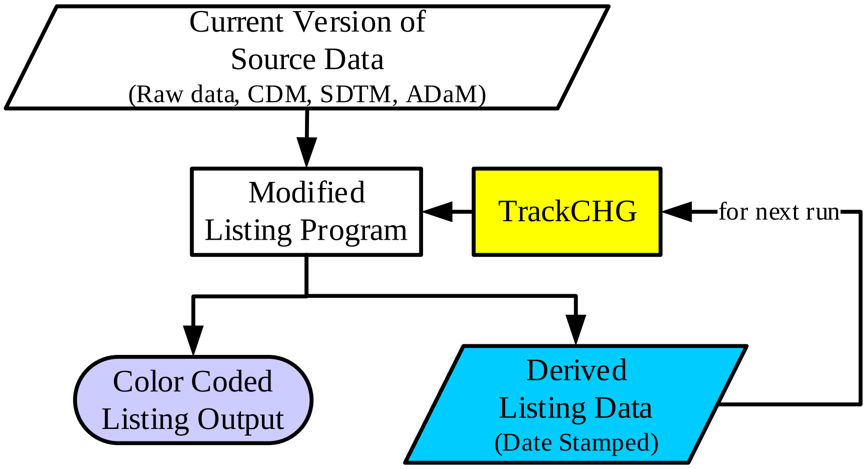
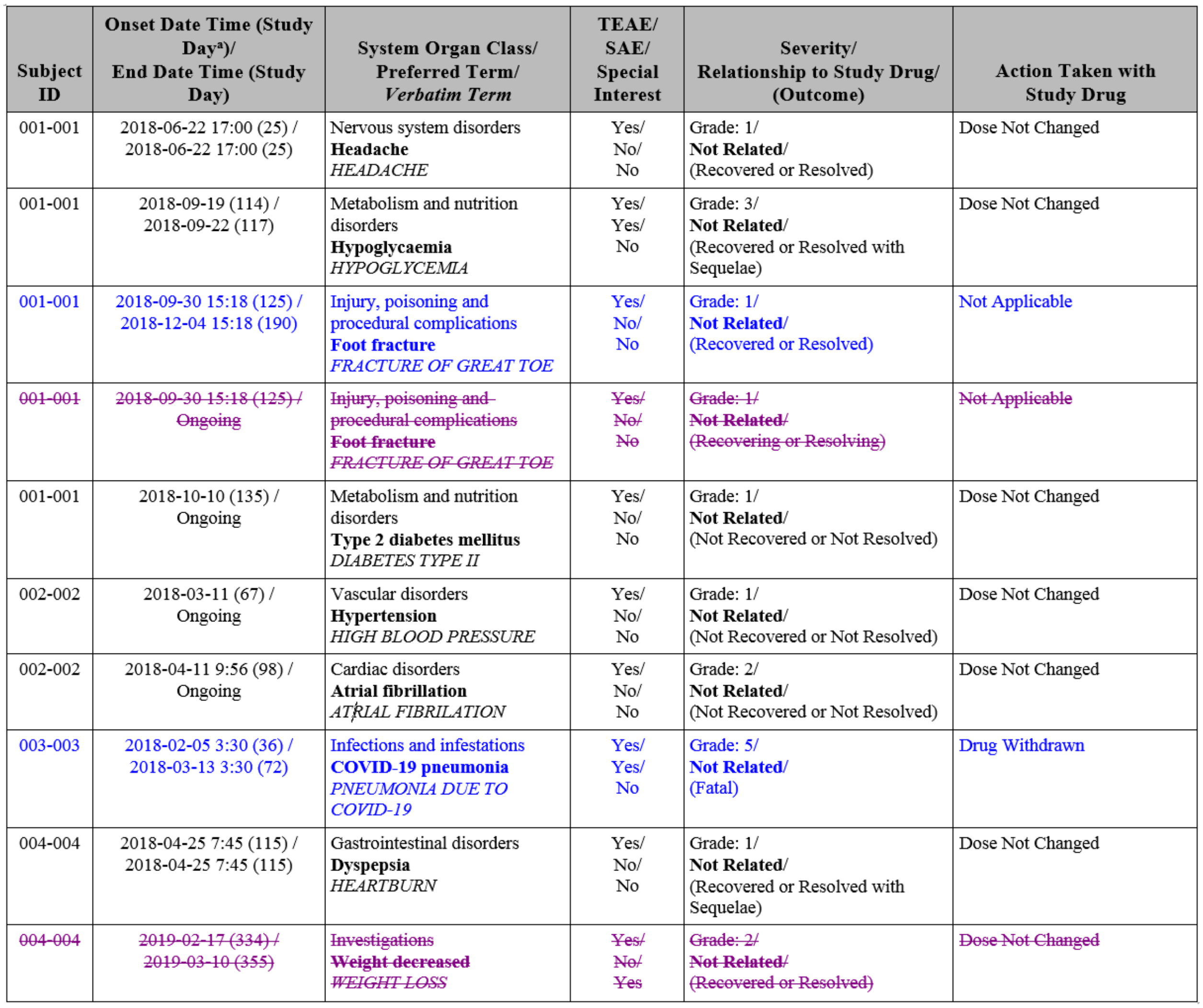


Figure 4 Sample Color-Coded Listing of Adverse Events



Subject ID	Onset Date Time (Study Day[a])/ End Date Time (Study Day)	System Organ Class/ Preferred Term/ *Verbatim Term*	TEAE/ SAE/ Special Interest	Severity/ Relationship to Study Drug/ (Outcome)	Action Taken with Study Drug
001-001	2018-06-22 17:00 (25) / 2018-06-22 17:00 (25)	Nervous system disorders **Headache** *HEADACHE*	Yes/ No/ No	Grade: 1/ **Not Related**/ (Recovered or Resolved)	Dose Not Changed
001-001	2018-09-19 (114) / 2018-09-22 (117)	Metabolism and nutrition disorders **Hypoglycaemia** *HYPOGLYCEMIA*	Yes/ Yes/ No	Grade: 3/ **Not Related**/ (Recovered or Resolved with Sequelae)	Dose Not Changed
001-001	2018-09-30 15:18 (125) / 2018-12-04 15:18 (190)	Injury, poisoning and procedural complications **Foot fracture** *FRACTURE OF GREAT TOE*	Yes/ No/ No	Grade: 1/ **Not Related**/ (Recovered or Resolved)	Not Applicable
~~001-001~~	~~2018-09-30 15:18 (125) /~~ ~~Ongoing~~	~~Injury, poisoning and procedural complications~~ ~~**Foot fracture**~~ ~~*FRACTURE OF GREAT TOE*~~	~~Yes/~~ ~~No/~~ ~~No~~	~~Grade: 1/~~ ~~**Not Related**/~~ ~~(Recovering or Resolving)~~	~~Not Applicable~~
001-001	2018-10-10 (135) / Ongoing	Metabolism and nutrition disorders **Type 2 diabetes mellitus** *DIABETES TYPE II*	Yes/ No/ No	Grade: 1/ **Not Related**/ (Not Recovered or Not Resolved)	Dose Not Changed
002-002	2018-03-11 (67) / Ongoing	Vascular disorders **Hypertension** *HIGH BLOOD PRESSURE*	Yes/ No/ No	Grade: 1/ **Not Related**/ (Not Recovered or Not Resolved)	Dose Not Changed
002-002	2018-04-11 9:56 (98) / Ongoing	Cardiac disorders **Atrial fibrillation** *ATRIAL FIBRILATION*	Yes/ No/ No	Grade: 2/ **Not Related**/ (Not Recovered or Not Resolved)	Dose Not Changed
003-003	2018-02-05 3:30 (36) / 2018-03-13 3:30 (72)	Infections and infestations **COVID-19 pneumonia** *PNEUMONIA DUE TO COVID-19*	Yes/ Yes/ No	Grade: 5/ **Not Related**/ (Fatal)	Drug Withdrawn
004-004	2018-04-25 7:45 (115) / 2018-04-25 7:45 (115)	Gastrointestinal disorders **Dyspepsia** *HEARTBURN*	Yes/ No/ No	Grade: 1/ **Not Related**/ (Recovered or Resolved with Sequelae)	Dose Not Changed
~~004-004~~	~~2019-02-17 (334) /~~ ~~2019-03-10 (355)~~	~~Investigations~~ ~~**Weight decreased**~~ ~~*WEIGHT LOSS*~~	~~Yes/~~ ~~No/~~ ~~Yes~~	~~Grade: 2/~~ ~~**Not Related**/~~ ~~(Recovered or Resolved)~~	~~Dose Not Changed~~

TrackCHG Components

In this section, we describe the set of three input datasets, three output datasets, and seven macro parameters that are used in the SAS macro, TrackCHG.

TrackCHG Inputs

1. TLGDATA.DOTsaved: a dataset that stores the date stamps of data transfers. It contains 4 variables:

Table 1 DOTSaved Variables

Variable Name	Variable Label	Type	Format	Length
DataDate	Date of Current Data Transfer	Integer	YYMMDD10.	8
PrevDataDate	Date of Previous Data Transfer	Integer	YYMMDD10.	8
TimeStamp	Date Stamp of Current Data Transfer	Char	$9	9
PrevStamp	Date Stamp of Previous Data Transfer	Char	$9	9

This dataset can also be configured to store other variables, for example, the run date of the listing, or the date when the source SDTM or ADaM dataset was updated.

- Date of Current Data Transfer (integer, date), *This is the date last modified of the dataset specified within the TrackCHG macro.* In the sample code (Appendix A: TrackCHG.SAS), the SAS macro grabs the date last modified from the Adverse Events Analysis dataset (Appendix B3: ADAE (Adverse Event Analysis Dataset)), but I recommend modifying the macro code to grab the date that the raw data or SDTM demographics dataset was last updated to better capture the date when the files were transferred.

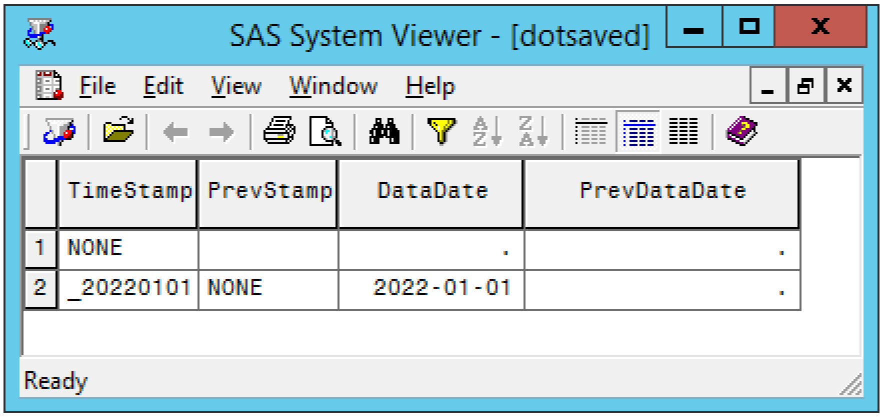
- Date of Previous Data Transfer (integer, date), *This is the date when the previous version of data files was transferred, derived using the lag function in a DATA step.*

- Date Stamp of Current Data Transfer (character, $9) *This is the character value of the Date of Current Data Transfer, preceded by an underscore. These values follow the format "_YYYYMMDD"*

- Date Stamp of Previous Data Transfer (character, $9) *This the character value of the Date of Previous Data Transfer, preceded by an underscore. These values follow the format "_YYYYMMDD"*

Using the sample SAS code in Appendix B1: DOTsaved (Date of Transfer Saved), the TLGDATA.DOTsaved dataset can be created with initial values (see image below for example).

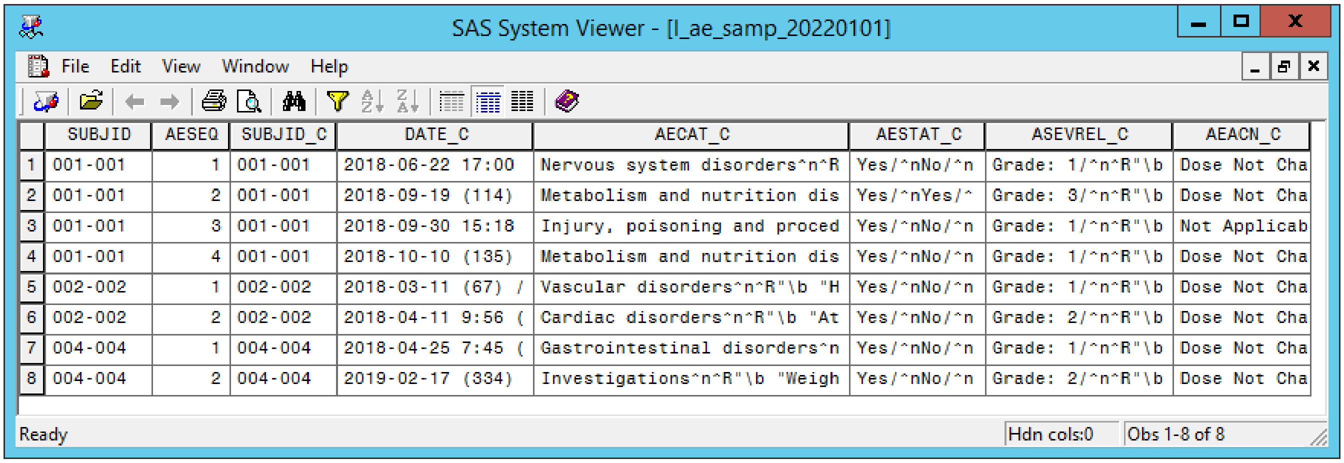
Figure 5 DOTsaved Date of Transfer Dataset



	TimeStamp	PrevStamp	DataDate	PrevDataDate
1	NONE		.	.
2	_20220101	NONE	2022-01-01	.

2. AFILE: the derived dataset that supports the listing, which is produced from the current data transfer. This is the dataset that is usually read by the PROC REPORT procedure, but instead will be input and processed by the TrackCHG macro. In the sample code, AFILE is derived from the adverse event analysis dataset (Appendix B3: ADAE (Adverse Event Analysis Dataset)) in the listing code, Appendix C: Sample Adverse Event Listing Code.
3. TIMEDATA.&BASEPROG.&ORDER.&PREVSTAMP: a derived listing dataset that supports the listing, which was produced from the previous data transfer. The sample SAS code in Appendix B2: L_AE_SAMP_20220101 (Sample Adverse Event Listing Derived Dataset), provides L_AE_SAMP_20220101, a derived listing dataset for adverse events with an artificially assigned date stamp of January 1, 2022 can be created with initial values (see image below for example).

Figure 6 Derived Adverse Event Listing Data Set with Date Stamp 2022-01-01



	SUBJID	AESEQ	SUBJID_C	DATE_C	AECAT_C	AESTAT_C	ASEVREL_C	AEACN_C
1	001-001	1	001-001	2018-06-22 17:00	Nervous system disorders^n^R	Yes/^nNo/^n	Grade: 1/^n^R"\b	Dose Not Cha
2	001-001	2	001-001	2018-09-19 (114)	Metabolism and nutrition dis	Yes/^nYes/^	Grade: 3/^n^R"\b	Dose Not Cha
3	001-001	3	001-001	2018-09-30 15:18	Injury, poisoning and proced	Yes/^nNo/^n	Grade: 1/^n^R"\b	Not Applicab
4	001-001	4	001-001	2018-10-10 (135)	Metabolism and nutrition dis	Yes/^nNo/^n	Grade: 1/^n^R"\b	Dose Not Cha
5	002-002	1	002-002	2018-03-11 (67) /	Vascular disorders^n^R"\b "H	Yes/^nNo/^n	Grade: 1/^n^R"\b	Dose Not Cha
6	002-002	2	002-002	2018-04-11 9:56 (	Cardiac disorders^n^R"\b "At	Yes/^nNo/^n	Grade: 2/^n^R"\b	Dose Not Cha
7	004-004	1	004-004	2018-04-25 7:45 (	Gastrointestinal disorders^n	Yes/^nNo/^n	Grade: 1/^n^R"\b	Dose Not Cha
8	004-004	2	004-004	2019-02-17 (334)	Investigations^n^R"\b "Weigh	Yes/^nNo/^n	Grade: 2/^n^R"\b	Dose Not Cha

Ready   Hdn cols:0   Obs 1-8 of 8

TrackCHG Macro Parameters

Table 2 TrackCHG Macro Parameters

Parameter	Valid Value	Sample Value	Description
INDATA	Dataset name	`INDATA = AFILE`	Input dataset name, the derived dataset that supports the listing to be generated
BASEPROG	text	`BASEPROG = L_AE`	The name of the base listing program
ORDER	text	`ORDER = _SAMP`	The order of the listing program, may have values to indicate the subsets or variations of the base programs
NEWCOLOR	text	`NEWCOLOR = ^S={color=blue}`	The text format to be applied to new/changed data from the current data transfer. The default is set to blue font color
PREVCOLOR	text	`PREVCOLOR = ^S={textdecoration= line_through color=purple}`	The text format to be applied original/deleted data from the previous data transfer. The default is set to a purple font color with strikethrough
SORTVAR	Variable names	`SORTVAR = subjid aeseq subjid_c date_c aecat_c aestat_c aesevrel_c aeacn_c`	The list of variable names, delimited by a space, for the sort and merging order of AFILE with its timestamped predecessor. This list typically contains all the variables that are used in the COLUMN statement in PROC REPORT
ARRAYVAR	Character variable names	`ARRAYVAR = subjid_c date_c aecat_c aestat_c aesevrel_c aeacn_c`	The list of variable names, delimited by a space, for the **character** variables whose values will be formatted with either the new or previous text formatting

TrackCHG Outputs

1. TLGDATA.DOTsaved: a dataset that stores the date stamps of data transfers, updated to include the current date stamp.
2. work._&BASEPROG.&ORDER: a derived dataset that supports the listing output, which has colored text formatting applied to new and old records. This is the dataset that will be read into the PROC REPORT procedure in place of its predecessor, AFILE. The TrackCHG macro applies the following changes to work._&BASEPROG.&ORDER:

Table 3 work._&BASEPROG.&ORDER Created Variables

Variable Name	**Variable Label**	**Type**	**Format**	**Length**
NewDataFL	New Data Record Flag	Char	$1	1
ColorData	Apply Color/Text Formatting to this Record?	Char	$3	3
Data_Color	Color/Text Formatting	Char	$50	50

- New Data Record Flag (character, $1), *This is set to 'Y' if a record in AFILE has new data, null otherwise*. This flag can be used in PROC REPORT or additional data steps for extra processing of new data.
- Apply Color/Text Formatting to this Record? (character, $3), *This is set to 'YES' to identify that the character variables identified in &ARRAYVAR should be assigned color/text formatting*.
- Color/Text Formatting (character, $50) *This variable is assigned values from &NEWCOLOR or &PREVCOLOR, as applicable.* For records where ColorData = 'YES', all of the variables identified in &ARRAYVAR will have the text formatting specified by Data_Color appended to the beginning of their values.

3. TIMEDATA.&BASEPROG.&ORDER.&TIMESTAMP: AFILE saved in the TimeData folder with a date stamp of the current data transfer. This dataset will be used for future comparisons.

TrackCHG Macro Function

This macro is designed to be called by a listing program, right before the PROC REPORT procedure. The derived listing dataset, AFILE, that is normally read in by the PROC REPORT will instead be processed by the TrackCHG macro, which will compare the records from AFILE to the records generated from a previous iteration of the listing program. TrackCHG will apply text formats to character variables in AFILE to color-code changes in the data.

This macro consists two parts: time stamping and data comparison.

Creating Time Stamped Variables

In order for the TrackCHG macro to run with the sample data provided in Appendix C: Sample Adverse Event Listing Code, the macro is currently designed to go into the ADaM derived data directory to retrieve a date (e.g., the date of the ADAE.sas7bdat file was last updated). This date is saved in the macro variable, RAWDATE.

```
%let rc      = %sysfunc(filename(onefile, &adamdata\adae.sas7bdat)) ;
%let fid     = %sysfunc(fopen(&onefile)) ;
%let rawdate = %substr(%qsysfunc(finfo(&fid,Last Modified)),1,9) ;
%let fidc    = %sysfunc(fclose(&fid)) ;
```

The first DATA step creates a temporary dataset, NewSaved, with one observation and two variables DATADATE, which is a numeric variable containing the SAS date of RAWDATE, and TIMESTAMP which is the character variable storing digits of RAWDATE preceded by an underscore. Additionally, the macro variable TIMESTAMP is created in this step and will be used to save date stamped versions of AFILE, the derived listing dataset.

```
data NewSaved ;
   length TimeStamp $9. ;
   DataDate = input("&rawdate", date9.) ;
   TimeStamp = '_' || compress(put(DataDate,yymmdd10.), '-') ;
   call symput('timestamp', timestamp);
   format datadate yymmdd10.;
run ;
%put &=timestamp ;
```

The second DATA step is designed to record the unique values of DATADATE in DOTsaved. Datasets WORK.NewSaved and TLGDATA.DOTsaved are merged by DATADATE in a full outer join. If DATADATE is a new date that is not contained in the TLGDATA.DOTsaved, it creates a new record in TLGDATA.DOTsaved. If the DATADATE already exists in TLGDATA.DOTsaved, then no changes are made. In this data step, we also define the two new variables PREVDATADATE and PREVSTAMP, which holds the YYMMDD10. and character values, respectively, of the previously stored value of DATADATE. These two macro variables will aid in the comparison and saving of date stamped derived datasets. The value of PREVSTAMP is also stored in the macro variable, PREVSTAMP, in order to be able to call the derived listing made from the previous date.

```
/* Record date of current data transfer in tlgdata.DOTsaved */
data tlgdata.DOTsaved ;
   merge tlgdata.DOTsaved (in=SavedDates)
         newSaved ;
   by dataDate ;

   lagPrevDataDate = lag(DataDate) ;
   lagPrevStamp = lag(TimeStamp) ;

   if not SavedDates then do ;
      PrevDataDate = lagprevDataDate ;
      PrevStamp = lagprevStamp ;
   end ;

   call symput('prevstamp',prevstamp);
   drop lagprevStamp lagprevDataDate ;
   format DataDate PrevDataDate YYMMDD10. ;
run ;
```

Comparing Two Versions of Data

The second part of the macro indicates new or changed data within the derived dataset that will be used to create the listing. A data saves AFILE with the timestamp of TIMESTAMP and stores the dataset in the TimeData folder for future comparisons.

```
data timedata.&baseprog.&order.&timestamp ;
   set afile ;
run ;
```

If there is no previous record (&PREVSTAMP= NONE) then the macro passes the dataset through without adding any color text formats to the data because the previous version of the dataset does not exist and would cause the MERGE step to fail.

Otherwise, after saving AFILE, the previous (TIMEDATA.&BASEPROG.&ORDER.&PREVSTAMP) version of the derived listing dataset and the current version (AFILE) of the derived listing dataset are merged by the variables listed in &SORTVAR in an full outer join. Three new variables, NewDataFL, ColorData, and Data_Color, are defined (Table 3) in this data step as well. Any records that do not fall within the inner join of the two datasets will be marked for text formatting (ColorData = ‘YES’). If the records are not in the previously derived dataset, then it must be “new” data, and is flagged NewDataFL = ‘Y’ and marked for text formatting ColorData = ‘YES’, and the Data_Color is set to &NEWCOLOR. If the records are not in the new data then they must have been changed or removed. In this case, the previous data records are also marked for text formatting ColorData = ‘YES’ and the Data_Color is set to the &PREVCOLOR. Then, a DO Loop is called and goes through the array of character variables defined in

&ARRAYVAR and it appends the text formatting at the beginning of the character value of each of the array variables, when ColorData = ‘YES’.

```
data _&baseprog.&order ;
   length &arrayvar $250 data_color $50;
%if &prevstamp = NONE %then %do ;
   set &indata ;
   array cols{*} &arrayvar ;
%end ;

%else %do ;
   merge timedata.&baseprog.&order.&prevstamp (in=inprev)
                  &indata (in=innew) ;
   by &sortvar ;

   /* if not in previous data, then must be new data */
   if not inprev then do ;
      newdatafl = 'Y' ;
      colordata = 'YES' ;
      data_color = "&newcolor" ;
   end ;

   /* if not in new data, then be removed/changed records in previous data */
   if not innew then do ;
      colordata = 'YES' ;
      data_color = "&prevcolor" ;
   end ;

   if colordata='YES' then do ;
      do iy = 1 to dim(cols) ;
         cols{iy}  = strip(data_color) || strip(cols{iy}) ;
      end ;
   end ;
%end ;
   drop iy;
run ;
```

This dataset with the text formatting is saved in the temporary dataset _&BASEPROG.&ORDER and used by a PROC REPORT procedure to produce the colorized adverse event listing shown in Figure 4. This figure demonstrates how the colored output can be used to identify changed, new, and deleted records. The third adverse event for Subject 001-001 can be identified as a changed record because there is a new record in blue text immediately followed by the previous record in purple strikethrough text that describe the same adverse event. The previous record shows that the foot fracture did not have an end datetime and had an outcome of “Recovering or Resolving” because the event was ongoing at the time of the previous data transfer, 2022-01-01. As of the current data transfer, the foot fracture is “Recovered or Resolved” and has a populated end date, end time, and study day. Subject 003-003 has a new adverse event of COVID-19 pneumonia, displayed in blue text. Subject 004-004 had a second AE of weight decreased that was removed because this record is not in the current data transfer but was included in the previous data transfer.

Customizations

The text formatting can be customized for old/changed data and new/current data. These text formats include strikethrough, highlights, bold, italics, colors, and more. Please refer to SAS documentation, SAS® 9.4 Output Delivery System: Advanced Topic and Eric Gerbhart’s SAS Global Forum Paper 222-2009, on SAS text formats. These text formats can also be applied to outputs in Microsoft Excel workbooks.

If the previous old/changed records are not the review focus, the previous records can easily be suppressed by implementing the following SAS code:

```
   /* if not in previous data, then must be new data */
   if not inprev then do ;
      newdatafl = 'Y' ;
      colordata = 'YES' ;
      data_color = "&newcolor" ;
   end ;

   /* if not in new data, then must be removed/changed records in previous data */
   if not innew then delete ;
```

If there is a set of listings that are always produced and delivered to the Sponsor or DMC following each data cut, this set of listings can all use the same DOTsaved dataset, instead of creating a unique TLGDATA.DOTsaved dataset to be used by each listing program. It is important to note that all the listing programs associated with a TLGDATA.DOTsaved dataset should be run at least once per data transfer in order to log and store the timestamped derived listing datasets of all the listings from each data transfer.

Cautionary Tales

One weakness of this TrackCHG macro is that it is not flexible to changes in the derived dataset that supports the listing.

If, for example, a sponsor wants to add a new column like a flag for COVID-related events, the macro will crash. Workaround: Update the PROC REPORT to include the new variable, but do not include the new variable name in the &SORTVAR and &ARRAYVAR values until the next data transfer, when the new variable becomes available in the previous and current datasets.

If you modify one or more of the values in the listings (e.g., displaying AETERM in proper case instead of uppercase), all affected observations will be flagged as new records. Workaround: implement these updates *prior* to the data transfer.

Conclusion

With only minor updates to the environment and listing macros, the add-on TrackCHG macro can be easily implemented to generate colorized listings that highlight the differences between two sets of source data. Modifications can be made to the macro to customize the text formats to add strikethrough, bold, italicize, highlighting, etc. These text formats also can be carried over to Microsoft Excel listings. A SAS user can also preferentially display only new records, only old records, or both in the color-coded subject listings.

References

Gebhart, E. (2009). “Paper 222-2009 Inline Formatting with ODS Markup”. *SAS Global Forum*. https://support.sas.com/resources/papers/proceedings09/222-2009.pdf

SAS Institute Inc. (2016). *SAS® 9.4 Output Delivery System: Advanced Topics, Third Edition*. Cary, NC: SAS Institute Inc. (pp. 216-220, 253–259).

Contact Information

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

Inka Leprince
PharmaStat, LLC
(510) 709-9107
ileprince@pharmastat.com

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

Appendix A: TrackCHG.SAS

```
** --------------------------------------------------------------------- ** ;
** Program:    TrackCHG.sas                                                 ;
** Author:     IMLeprince                                                   ;
** Created:    2021-11-15                                                   ;
** Inputs:     tlgdata.DOTsaved   Contains timestamps of data transfers     ;
**             timedata.l_baseprog&order.&prevstamp   AFILE created from    ;
**               previous (&prevstamp) data transfer                        ;
**             work.afile            Analysis file for listing              ;
**                                                                          ;
** Outputs:    tlgdata.DOTsaved   Contains timestamps of data transfers     ;
**             timedata.l_baseprog&order.&timestamp   AFILE saved with a    ;
**              timestamp                                                   ;
**             work._&baseprog.&order   Analysis file for listing with      ;
**               color-coded records                                        ;
**                                                                          ;
** Usage:      Track Changes between current vs previous data transfer      ;
**               for subject-level safety listings by color coding records  ;
** Revisions:                                                               ;
** --------------------------------------------------------------------- ** ;

%macro TrackCHG (
        indata   =  ,
        baseprog =  ,
        order    =  ,
        newcolor  = ^S={color=blue},
        prevcolor = ^S={textdecoration=line_through color=purple},
        sortvar   =  ,
        arrayvar  =
        ) ;

%*-------------------------------------------------------------------------;
%* Save most current data transfer date into DOTsaved                      ;
%* Create macro variables:                                                 ;
%* (1) Timestamp: _YYYYMMDD of current data transfer                       ;
%* (2) PrevStamp: _YYYYMMDD of previous data transfer                      ;
%*-------------------------------------------------------------------------;

/* Preference to get the date of data transfer from the raw demographics dataset: DM.sas7bdat */
/* For utilization with example AE listing, getting date of data transfer from ADaM ADAE.sas7bdat */
%let rc      = %sysfunc(filename(onefile, &adamdata\adae.sas7bdat)) ;
%let fid     = %sysfunc(fopen(&onefile)) ;
%let rawdate = %substr(%qsysfunc(finfo(&fid,Last Modified)),1,9) ;
%let fidc    = %sysfunc(fclose(&fid)) ;

data NewSaved ;
   length TimeStamp $9. ;
   DataDate = input("&rawdate", date9.) ;
   TimeStamp = '_' || compress(put(DataDate,yymmdd10.), '-') ;
   call symput('timestamp', timestamp);
run ;
%put &=timestamp ;


/* Record date of current data transfer in tlgdata.DOTsaved */
data tlgdata.DOTsaved ;
   merge tlgdata.DOTsaved (in=SavedDates)
         newSaved ;
   by dataDate ;

   lagPrevDataDate = lag(DataDate) ;
   lagPrevStamp = lag(TimeStamp) ;

   if not SavedDates then do ;
      PrevDataDate = lagprevDataDate ;
      PrevStamp = lagprevStamp ;
   end ;

   call symput('prevstamp',prevstamp);
   drop lagprevStamp lagprevDataDate ;
   format DataDate PrevDataDate YYMMDD10. ;
run ;

%*--------------------------------------------------------------------------;
%* Indicate New or Changed Data within Listing
%*--------------------------------------------------------------------------;
/* Sort input data by &SORTVAR */
proc sort data = &indata ;
    by &sortvar ;
run ;

/* Save a timestamped copy of the current dataset for future comparison */
data timedata.&baseprog.&order.&timestamp ;
   set &indata ;
run ;

/* Merge previous and current derived datasets                  */
/* Previous data transfer: timedata.&baseprog.&order.&prevstamp */
/* Current data transfer:  &indata                              */
data _&baseprog.&order ;
   length &arrayvar $250 data_color $50;
%if &prevstamp = NONE %then %do ;
   set &indata ;
   array cols{*} &arrayvar ;
%end ;

%else %do ;
   merge timedata.&baseprog.&order.&prevstamp (in=inprev)
                  &indata (in=innew) ;
   by &sortvar ;
   array cols{*} &arrayvar ;


   /* if not in previous data, then must be new data */
   if not inprev then do ;
      newdatafl = 'Y' ;
      colordata = 'YES' ;
      data_color = "&newcolor" ;
   end ;

   /* if not in new data, then must be removed/changed records in previous data */
   if not innew then do ;
      colordata = 'YES' ;
      data_color = "&prevcolor" ;
   end ;

   if colordata='YES' then do ;
      do iy = 1 to dim(cols) ;
         cols{iy}  = strip(data_color) || strip(cols{iy}) ;
      end ;
   end ;
%end ;
run ;

%mend ;
```

Appendix B: Sample Data

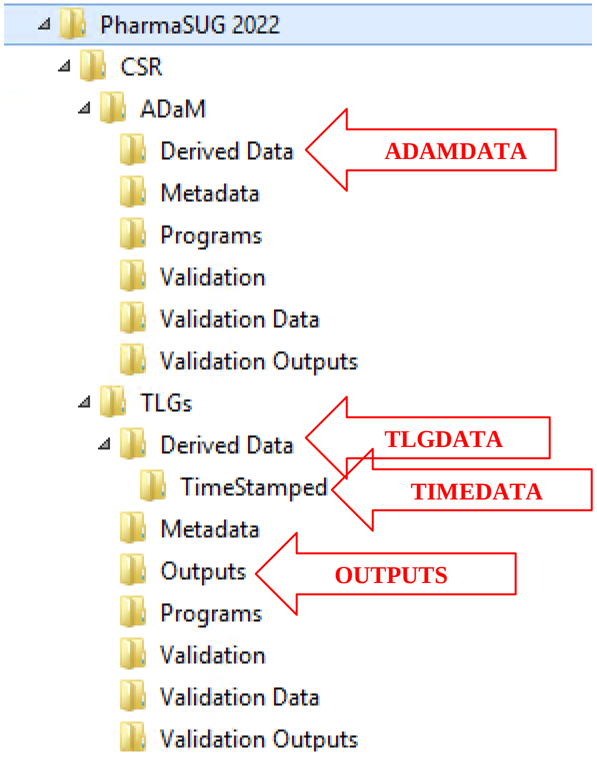
```
%*-------------------------------------------------------------------------;
%* Setting up environment: define the file paths
%*-------------------------------------------------------------------------;

/* ADAMDATA = where the ADaM datasets (ADAE) resides                  */
/* TLGDATA  = where the derived datasets that support outputs resides */
/* TIMEDATA = where the timestamped derived datasets that support     */
/*             outputs resides                                        */
/* OUTPUTS = where the AE listing.RTF will be saved                   */

%let ADAMDATA = ..\CSR\ADaM\Derived Data ;
%let TLGDATA =  ..\CSR\TLGs\Derived Data ;
%let TIMEDATA =  ..\CSR\TLGs\Derived Data\TimeStamped ;
%let OUTPUTS = ..\CSR\TLGs\Outputs ;

libname tlgdata "&tlgdata" ;
libname timedata "&timedata" ;
libname adamdata "&adamdata" ;
libname outputs "&outputs" ;
```

Figure 7 Sample Environment



```
PharmaSUG 2022
  CSR
    ADaM
      Derived Data
      Metadata
      Programs
      Validation
      Validation Data
      Validation Outputs
    TLGs
      Derived Data
        TimeStamped
      Metadata
      Outputs
      Programs
      Validation
      Validation Data
      Validation Outputs
```

Appendix B1: DOTsaved (Date of Transfer Saved)

```
%*-------------------------------------------------------------------------;
%* DOTsaved: Date of Transfer Saved
%* Stores the date stamps of data transfers
%*-------------------------------------------------------------------------;
data tlgdata.DOTsaved ;
        infile datalines missover ;
        informat TimeStamp PrevStamp $9. DataDate PrevDataDate YYMMDD10. ;
        input TimeStamp PrevStamp $ DataDate PrevDataDate ;
    label TimeStamp = "Date Stamp of Current Data Transfer"
          PrevStamp = "Date Stamp of Previous Data Transfer"
          DataDate  = "Date of Current Data Transfer"
          PrevDataDate = "Date of Previous Data Transfer";
        format DataDate PrevDataDate YYMMDD10. ;
    datalines ;
NONE
_20220101 NONE      2022-01-01
```

```
        ;
run ;
```

Appendix B2: L_AE_SAMP_20220101 (Sample Adverse Event Listing Derived Dataset)

```
%*------------------------------------------------------------------------;
%* l_ae_samp_20220101: derived dataset of AE listing from the
%*                     2022-01-01 data transfer
%*------------------------------------------------------------------------;
data timedata.l_ae_samp_20220101 ;
    infile datalines dlm = '*' missover ;
    input SUBJID $7. AESEQ :8. SUBJID_C :$7. DATE_C :$50. AECAT_C :$125. AESTAT_C :$15.
          AESEVREL_C :$80. AEACN_C :$25. ;
    label SUBJID = 'Subject ID'
          AESEQ =  'Sequence Number'
          SUBJID_C = 'Subejct ID (C)'
          DATE_C   = 'Onset Date/End Date (C)'
          AECAT_C  = 'SOC/PT/Verbatim Term (C)'
          AESTAT_C = 'TEAE/SAE/Special Interest (C)'
          AESEVREL_C = 'Severity/Relationship to Study Drug/Outcome (C)'
          AEACN_C = 'Action Taken with Study Drug (C)'
    ;
datalines ;
001-001*1*001-001*2018-06-22 17:00 (25) /^n2018-06-22 17:00 (25)*Nervous system disorders^n^R"\b "Headache^R"\b0 "^n^R"\i
"HEADACHE^R"\i0"                                         *Yes/^nNo/^nNo  *Grade: 1/^n^R"\b "Not Related^R"\b0 "/^n(Recovered or
Resolved)              *Dose Not Changed
001-001*2*001-001*2018-09-19 (114) /^n2018-09-22 (117)           *Metabolism and nutrition disorders^n^R"\b "Hypoglycaemia^R"\b0
"^n^R"\i "HYPOGLYCEMIA^R"\i0"                      *Yes/^nYes/^nNo *Grade: 3/^n^R"\b "Not Related^R"\b0 "/^n(Recovered or Resolved with
Sequelae)*Dose Not Changed
001-001*3*001-001*2018-09-30 15:18 (125) /^nOngoing              *Injury, poisoning and procedural complications^n^R"\b "Foot
fracture^R"\b0 "^n^R"\i "FRACTURE OF GREAT TOE^R"\i0"*Yes/^nNo/^nNo  *Grade: 1/^n^R"\b "Not Related^R"\b0 "/^n(Recovering or
Resolving)            *Not Applicable
001-001*4*001-001*2018-10-10 (135) /^nOngoing                    *Metabolism and nutrition disorders^n^R"\b "Type 2 diabetes
mellitus^R"\b0 "^n^R"\i "DIABETES TYPE II^R"\i0"       *Yes/^nNo/^nNo  *Grade: 1/^n^R"\b "Not Related^R"\b0 "/^n(Not Recovered or Not
Resolved)      *Dose Not Changed
002-002*1*002-002*2018-03-11 (67) /^nOngoing                     *Vascular disorders^n^R"\b "Hypertension^R"\b0 "^n^R"\i "HIGH BLOOD
PRESSURE^R"\i0"                                *Yes/^nNo/^nNo  *Grade: 1/^n^R"\b "Not Related^R"\b0 "/^n(Not Recovered or Not Resolved)
*Dose Not Changed
002-002*2*002-002*2018-04-11 9:56 (98) /^nOngoing                *Cardiac disorders^n^R"\b "Atrial fibrillation^R"\b0 "^n^R"\i "ATRIAL
FIBRILATION^R"\i0"                           *Yes/^nNo/^nNo  *Grade: 2/^n^R"\b "Not Related^R"\b0 "/^n(Not Recovered or Not Resolved)
*Dose Not Changed
004-004*1*004-004*2018-04-25 7:45 (115) /^n2018-04-25 7:45 (115)*Gastrointestinal disorders^n^R"\b "Dyspepsia^R"\b0 "^n^R"\i
"HEARTBURN^R"\i0"                                      *Yes/^nNo/^nNo  *Grade: 1/^n^R"\b "Not Related^R"\b0 "/^n(Recovered or Resolved
with Sequelae)*Dose Not Changed
004-004*2*004-004*2019-02-17 (334) /^n2019-03-10 (355)           *Investigations^n^R"\b "Weight decreased^R"\b0 "^n^R"\i "WEIGHT
LOSS^R"\i0"                                        *Yes/^nNo/^nYes *Grade: 2/^n^R"\b "Not Related^R"\b0 "/^n(Recovered or Resolved)
*Dose Not Changed
;
run ;
```

Appendix B3: ADAE (Adverse Event Analysis Dataset)

```
%*-------------------------------------------------------------------------;
%* ADAE: analysis dataset of adverse events
%* Current data transfer
%*-------------------------------------------------------------------------;
data adamdata.adae ;
    infile datalines dlm = '*' missover ;
    input SUBJID $7. AESEQ :8. ASTDT MMDDYY10. ASTTM :time5. ASTDY :8. AENDT MMDDYY10. AENTM :time5. AENDY :8.
          AETERM :$25. AEDECOD :$24. AEBODSYS :$50. AEOUT :$35. ATOXGR :$8. AEREL :$11. AEACN :$16.
          TRTEMFL $1. AESER $1. AESIFL $1. ;
    label SUBJID   = 'Subject Identifier'
          AESEQ    = 'Sequence Number'
          ASTDT    = 'Analysis Start Date'
          ASTTM    = 'Analysis Start Time'
          ASTDY    = 'Analysis Start Relative Day'
          AENDT    = 'Analysis End Date'
          AENTM    = 'Analysis End Time'
          AENDY    = 'Analysis End Relative Day'
          AETERM   = 'Reported Term for the Adverse Event'
          AEDECOD  = 'Dictionary-Derived Term'
          AEBODSYS = 'Body System or Organ Class'
          AEOUT    = 'Outcome of Adverse Event'
          ATOXGR   = 'Analysis Toxicity Grade'
          AEREL    = 'Causality'
          AEACN    = 'Action Taken with Study Treatment'
          TRTEMFL  = 'Treatment Emergent Analysis Flag'
          AESER    = 'Serious Event'
          AESIFL   = 'AE of Special Interest Flag' ;
    format ASTDT AENDT YYMMDD10. ASTTM AENTM TIME5. ;
datalines ;
001-001*1*06/22/2018*17:00* 25*06/22/2018*23:00* 25*HEADACHE                 *Headache                *Nervous system disorders
*RECOVERED OR RESOLVED              *Grade: 1*NOT RELATED*DOSE NOT CHANGED*YNN
001-001*2*09/19/2018*     *114*09/22/2018*     *117*HYPOGLYCEMIA             *Hypoglycaemia           *Metabolism and nutrition
disorders            *RECOVERED OR RESOLVED WITH SEQUELAE*Grade: 3*NOT RELATED*DOSE NOT CHANGED*YYN
001-001*3*09/30/2018*15:18*125*12/04/2018*     *190*FRACTURE OF GREAT TOE    *Foot fracture           *Injury, poisoning and
procedural complications*RECOVERED OR RESOLVED              *Grade: 1*NOT RELATED*NOT APPLICABLE  *YNN
001-001*4*10/10/2018*     *135*          *     *   *DIABETES TYPE II         *Type 2 diabetes mellitus*Metabolism and nutrition
disorders            *NOT RECOVERED OR NOT RESOLVED      *Grade: 1*NOT RELATED*DOSE NOT CHANGED*YNN
002-002*1*03/11/2018*     * 67*          *     *   *HIGH BLOOD PRESSURE      *Hypertension            *Vascular disorders
*NOT RECOVERED OR NOT RESOLVED      *Grade: 1*NOT RELATED*DOSE NOT CHANGED*YNN
002-002*2*04/11/2018*09:56* 98*          *     *   *ATRIAL FIBRILATION       *Atrial fibrillation     *Cardiac disorders
*NOT RECOVERED OR NOT RESOLVED      *Grade: 2*NOT RELATED*DOSE NOT CHANGED*YNN
003-003*1*02/05/2018*03:30* 36*03/13/2018*     * 72*PNEUMONIA DUE TO COVID-19*COVID-19 pneumonia      *Infections and infestations
*FATAL                              *Grade: 5*NOT RELATED*DRUG WITHDRAWN  *YYN
004-004*1*04/25/2018*07:45*115*04/25/2018*12:30*115*HEARTBURN                *Dyspepsia               *Gastrointestinal disorders
*RECOVERED OR RESOLVED WITH SEQUELAE*Grade: 1*NOT RELATED*DOSE NOT CHANGED*YNN
;
run ;
```

Appendix C: Sample Adverse Event Listing Code

```
** --------------------------------------------------- ** ;
** Program:    l_ae.sas                                ** ;
** Author:     IMLeprince                              ** ;
** Created:    2021-12-24                              ** ;
**                                                     ** ;
** Usage:      Generate a listings of Adverse Events   ** ;
**                                                     ** ;
** REQUIRES rawdata:                                   ** ;
** REQUIRES anadata: adae                              ** ;
** REQUIRES macros:  TrackCHG                          ** ;
**                                                     ** ;
** OUTPUT: l_ae&order.rtf                              ** ;
**                                                     ** ;
** Revisions:                                          ** ;
**                                                     ** ;
** --------------------------------------------------- ** ;

%*-------------------------------------------------------------------------;
%* Setting up environment: define the filepaths
%* Refer to Figure 7
%*-------------------------------------------------------------------------;
%let ORDER = _SAMP ;

/* ADAMDATA = where the ADaM datasets (ADAE) resides                  */
/* TLGDATA  = where the derived datasets that support outputs resides */
/* TIMEDATA = where the timestamped derived datasets that support     */
/*             outputs resides                                        */
/* OUTPUTS = where the AE listing.RTF will be saved                   */

%let ADAMDATA = ..\CSR\ADaM\Derived Data ;
%let TLGDATA =  ..\CSR\TLGs\Derived Data ;
%let TIMEDATA =  ..\CSR\TLGs\Derived Data\TimeStamped ;
%let OUTPUTS = ..\CSR\TLGs\Outputs ;

libname tlgdata "&tlgdata" ;
libname timedata "&timedata" ;
libname adamdata "&adamdata" ;
libname outputs "&outputs" ;

proc format ;
    value $ yn
        'Y' = 'Yes'
        'N' = 'No'
        ' ' = 'No'
    ;
    value $ grade
        'Grade 1' = 'Grade 1: Mild'
        'Grade 2' = 'Grade 2: Moderate'
        'Grade 3' = 'Grade 3: Severe'
        'Grade 4' = 'Grade 4: Life-threatening'
        'Grade 5' = 'Grade 5: Death'
    ;
    value $ aeout
        'NOT RECOVERED OR NOT RESOLVED' = 'Not Recovered or Not Resolved'
        'RECOVERED OR RESOLVED'         = 'Recovered or Resolved'
        'RECOVERING OR RESOLVING'       = 'Recovering or Resolving'
        'RECOVERED OR RESOLVED WITH SEQUELAE' = 'Recovered or Resolved with Sequelae'
        'UNKNOWN'                       = 'Unknown'
        'FATAL'                         = 'Fatal'
    ;
run ;

%*-----------------------------------------------------------------------------;
%* Bring in data ADAE
%*-----------------------------------------------------------------------------;
proc sort data = adamdata.adae out = file1 ;
   by subjid astdt aendt aebodsys ;
run;

data afile ;
    length subjid_c  $50
           stdate_c endate_c date_c aecat_c aestat_c aesevrel_c aeacn_c $250 ;
    set file1 ;
    by subjid astdt aendt aebodsys ;

    subjid_c = subjid ;

    /* process character dates */
    array daten {*} astdt aendt ;
    array times {*} asttm aettm ;
    array dates_c {*} $50 stdate_c endate_c ;
    array adys   {*} astdy aendy ;

    do ii = 1 to dim(daten) ;
        if adys{ii} > .z then do ;
            if asttm > .z then
                dates_c{ii} = strip(put(daten{ii},yymmdd10.))||' '||strip(put(asttm,time5.))||' ('||strip(put(adys{ii},best8.))||')' ;
            else dates_c{ii} = strip(put(daten{ii},yymmdd10.))||' ('||strip(put(adys{ii},best8.))||')' ;
        end ;
    end ;
    if missing(endate_c) then
        date_c = strip(stdate_c)||' /^nOngoing' ;
    else date_c = strip(stdate_c)||' /^n'||strip(endate_c) ;

       /* AE SOC, MeDDRA Term, Verbatim */
    if aedecod ^= '' then
        aecat_c = strip(aebodsys)|| '^n' ||'^R"\b "' ||trim(aedecod) || '^R"\b0 "' || '^n' ||
                  '^R"\i "' ||  trim(aeterm) || '^R"\i0"';
    else if aebodsys = '' and aedecod= ' ' and aeterm^='' then
        aecat_c = '^R"\b "NOT CODED^R"\b0 "' ||'^n' || '^R"\i "' || trim(aeterm) || '^R"\i0"';
       /* AE Status flags */
    aestat_c = put(trtemfl, $yn.)||'/^n'||put(aeser, $yn.)||'/^n'||put(aesifl, $yn.);
    aestat_c = compress(aestat_c, ' ') ;

       /* AE Severity, Relation to Study Drug Treatment, Outcome */
    aesevrel_c = strip(put(atoxgr,$grade.)) || '/^n^R"\b "'||strip(propcase(aerel))||'^R"\b0 "/^n(' ||strip(put(aeout,$aeout.)) || ')'
;
    if aeacn ^='' then aeacn_c = strip(propcase(aeacn))  ;
    keep subjid aeseq subjid_c date_c aecat_c aestat_c aesevrel_c aeacn_c;
run ;

%*-----------------------------------------------------------------------------;
%* Indicate New or Changed Data for Listing
%*-----------------------------------------------------------------------------;
%TrackCHG (
        indata  = afile ,
        baseprog = l_ae ,
        order    = _samp ,
        sortvar  = subjid aeseq subjid_c date_c aecat_c aestat_c aesevrel_c aeacn_c ,
        arrayvar = subjid_c date_c aecat_c aestat_c aesevrel_c aeacn_c
        ) ;

%*-----------------------------------------------------------------------------;
%* Generate Listing
%*-----------------------------------------------------------------------------;
ods _all_ close ;
options orientation=landscape ;
ods escapechar='^' ;

title1    j=c "SAMPLE AE LISTING USING TrackCHG MACRO" ;
title2    j=l "Company Co."  j=r "Study 001" ;
title3    j=c "Listing 1: Treatment-Emergent Adverse Events" ;
title4    j=c "Safety Population" ;
footnote1 j=l "SAE = serious adverse event;
 TEAE =  treatment-emergent event is defined as an event occurring during or after
 administration of the first dose of study drug until xx days after the final dose
 of study drug (follow-up visit)" ;
footnote2 j=l "Note: Adverse event terms were mapped according to Medical
 Dictionary for Regulatory Activities (MedDRA) version xx.x." ;
footnote3 j=l "^{super a} Study Day = onset date - first dose date + 1,
 if on or after the first study drug dosing; Study Day = onset date - first dose
 date, otherwise." ;

ods rtf file="&outputs\listing-01-00-00-l-ae_samp.rtf" style=RTF;
proc report data = _l_ae_samp /*afile*/ missing split = '~' spacing=1 headskip headline spanrows ;
    column (subjid aeseq subjid_c date_c aecat_c aestat_c aesevrel_c aeacn_c) ;

    define subjid       / order order = internal noprint ;
    define aeseq        / order order = internal noprint ;
    define subjid_c     / order order = internal 'Subject~ID'           center style=[width=0.65 in];
    define date_c       / display  "Onset Date Time (Study Day^{super a})/^nEnd Date Time (Study Day)"
          center style=[width=1.75in] ;
    define aecat_c      / display "System Organ Class/~Preferred Term/~^S={font_style=italic}Verbatim Term"
            left   style=[width=1.85 in] ;
    define aestat_c     / display "TEAE/~SAE/~Special Interest"
            center style=[width=0.85 in] ;
    define aesevrel_c   / display "Severity/~Relationship to Study Drug/~(Outcome)"
            left   style=[width=2.03 in] ;
    define aeacn_c      / display "Action Taken with~Study Drug"
            left   style=[width=1.85 in] ;
run ;

ods _all_ close ;
```