PharmaSUG 2017 - Paper BB04

Reporting Non-Printable and Special Characters for Review in Excel

Abhinav Srivastva, Gilead Sciences

ABSTRACT

Data in clinical trials can be transmitted in various formats such as Excel, CSV, tab-delimited, ASCII files or via any Electronic Data Capture (EDC) tool. A potential problem arises when data has embedded special characters or even non-printable characters which affects all downstream analysis and reporting; in addition to being non-compliant with CDISC standards. The paper will briefly present a discussion on these characters and how to identify them but the emphasis will be on creating an excel report which summarizes these in a way that can be easily reviewed and appropriate action can be planned. Creating a summary report as this can be used by the programmers as initial steps in data cleaning activities with each data transfer. Some of the features of the excel report include traffic-lighting effects, hyperlinks and tool-tips for providing additional information.

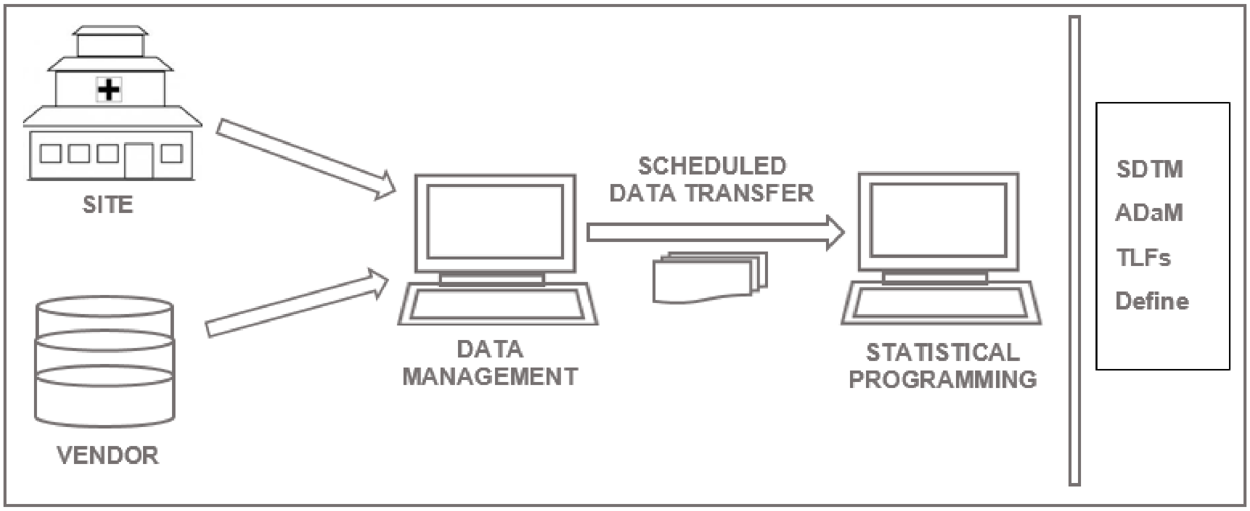
INTRODUCTION

Identifying unusual characters such as special characters and non-printable characters is a critical step in achieving quality analysis and reporting. The raw data collected as per Protocol/Case Report Form (CRF) should be assessed for these potential issues and rectified.

The paper presents a brief discussion on these characters paving the way for reporting them in a convenient excel destination using ODS EXCEL in SAS®.

DATA FLOW

Display 1 is a simplified version of data flow occurring between different entities. The data from sites and vendors are transmitted to the Data Management group who is responsible for data integrity checks and other edit checks in the form of conducting reviews and issuing queries either manually or in-built into the EDC system used. It is then transferred to the Statistical Programming group at a pre-determined frequency for analysis and reporting.



Display 1: Clinical Trials Data Flow

It is critical for the Statistical programmers to have a mechanism to validate the routine transfer in terms of keeping the data free from non-printable and special characters; even before evaluating them for compliance (CDISC) and other logical considerations.

WHAT ARE NON-PRINTABLE AND SPECIAL CHARACTERS?

ASCII code associates an integer value (0-255) for each symbol in the character set such as letters, digits, punctuation marks and control characters. The ASCII codes can be broadly classified into 3 categories: Non-printable characters (Code 0-31), Printable characters (Code 32-127) and Special characters (Code 128-255). Please see Reference [1] and [2] for more details. The decimal and hexadecimal code corresponding to these characters can be generated in SAS as below:

```
data ascii_table (drop=i);
  length decimalv $5 hexadecv $6 ascichar $3 ;
     do i=0 to 255;
       ascichar="("||byte(i)||")";
       decimalv="("||strip(put(rank(byte(i)), best.))||")";
       hexadecv="("||strip(put(byte(i), $hex4.))||")";
       output;
     end;
    label ascichar="Ascii character"
          decimalv="Decimal value"
          hexadecv="Hexadecimal value";
run;
```

While the printable characters are duly welcomed, the non-printable (Code 0-31) and special characters (Code 128-255) need to be treated. In addition, ‘Delete’ character (Code 127) is also added to the exclusion list. The simplest and the most effective way of removing them is using a COMPRESS function as shown below.

```
data out;
set in;
length chars2excl $200;
 retain chars2excl;
 do i=0 to 31, 127 to 255;
        if i=0 then chars2excl=byte(i);
   else chars2excl=trim(chars2excl)||byte(i);
 end;
 Variable=compress(Variable,chars2excl);
run;
```

Other two approaches below will delete non-printable characters but not special characters:

```
/* COMPRESS with 'K'=Keep and 'W'=Writable modifier */
data out;
 set in;
  Variable = COMPRESS(Variable,'kw');
run;

/* VERIFY function to keep Printable characters */

data out;
 set in (keep=string);
  do until(test=0);
   test=notprint(string);
    test=verify(upcase(string),' ABCDEFGHIJKLMNOPQRSTUVWXYZ,.1234567890');
      * If a non-printable is found...replace it with a space ;
         if test>0 then do;
            substr(string,test,1)=' ';
         end;
  end;
run;
```

CREATING A REPORT

With each data transfer between Data Management and Statistical Programming group as depicted in Display 1, a summary report as presented here can be created to identify unusual characters and take steps to eliminate them.

TEST DATA

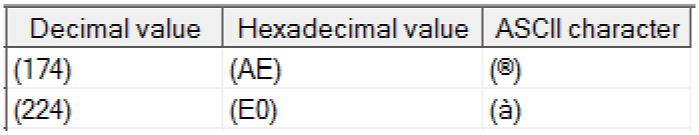
To demonstrate the process, let's create 2 datasets from SASHELP library and forcefully add non-printable and special characters in some text fields:

```
*-- Dataset # 1--*;
data class;
  set sashelp.class;
  length Sex1 $2;
  Sex1=Sex;
    if mod(_n_,5)=0 then Name=cats(byte(224)||Name);
    if mod(_n_,6)=0 then Name=cats(byte(174)||Name);
    if mod(_n_,4)=0 then Sex1=cats(byte(27)||Sex);
   drop Sex;
run;

*-- Dataset # 2 --*;
data shoes;
  set sashelp.shoes;
    where region ^= 'Central America/Caribbean';
    if mod(_n_,60) =0 then Region     =cats(strip(region)||byte(235));
    if mod(_n_,100)=0 then Subsidiary=cats(byte(9)||subsidiary);
run;
```

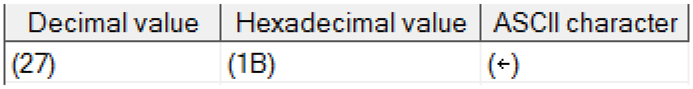
- Dataset CLASS –

  Variable '*Name*' contain 2 types of special characters as below:



Decimal value	Hexadecimal value	ASCII character
(174)	(AE)	(®)
(224)	(E0)	(à)

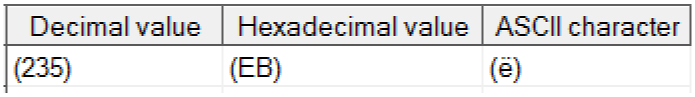
  Variable '*Sex1*' contain a non-printable character ('escape' character) as below:



Decimal value	Hexadecimal value	ASCII character
(27)	(1B)	(←)

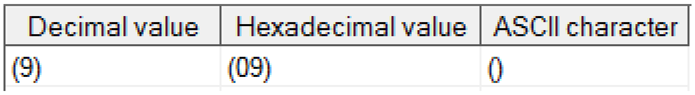
- Dataset SHOES –

  Variable '*Region*' contain a special character as below:



Decimal value	Hexadecimal value	ASCII character
(235)	(EB)	(ë)

  Variable '*Subsidiary*' contain a non-printable character ('*horizontal tab*' character) which doesn't get displayed as below:



Decimal value	Hexadecimal value	ASCII character
(9)	(09)	()

Additionally, let’s create a perfectly clean dataset and a dataset with zero observations as below:

```
*-- Dataset # 3: Clean dataset --*;
data prdsale;
   set sashelp.prdsale;
run;
```

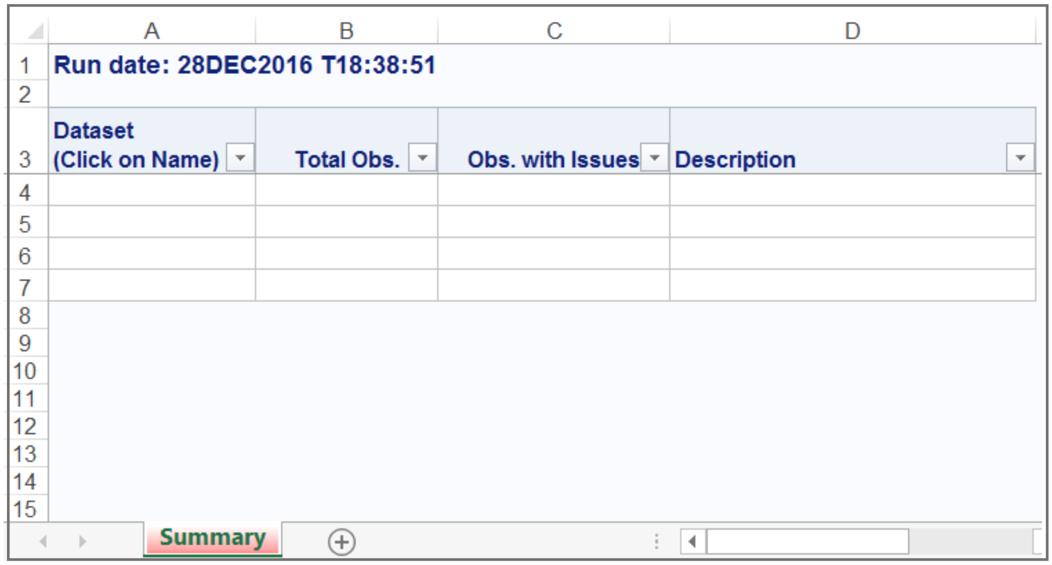
```
*- Dataset # 4: Zero Obs dataset
   data nodata;
     stop;
     set prdsale;
   run;
```

OVERVIEW OF THE REPORT

Based on the 4 datasets as created in Test Data section; non-printable and special characters in each of the datasets (as applicable) are identified using the integer codes as explained in - What are Non-Printable and Special characters? section and those records are flagged as ‘Y’ as seen in the DATA step below.

```
data out;
  <... more lines of code...>
    array char_vars{*}   _character_;
      do i=1 to dim(char_vars);
            do j=0 to 31,127; *-- Identify Non-printable characters ;
              if index(char_vars{i},byte(j))>0 then flag='Y';
            end;
               do k=128 to 255; *-- Identify Special Characters ;
                   if index(char_vars{i},byte(k))>0 then flag='Y';
               end;
      end;
run;
```

To render the result, an excel report template (Display 2) is designed which contain a minimum of Summary tab and additional tabs as needed based on the datasets containing unusual characters of interest.



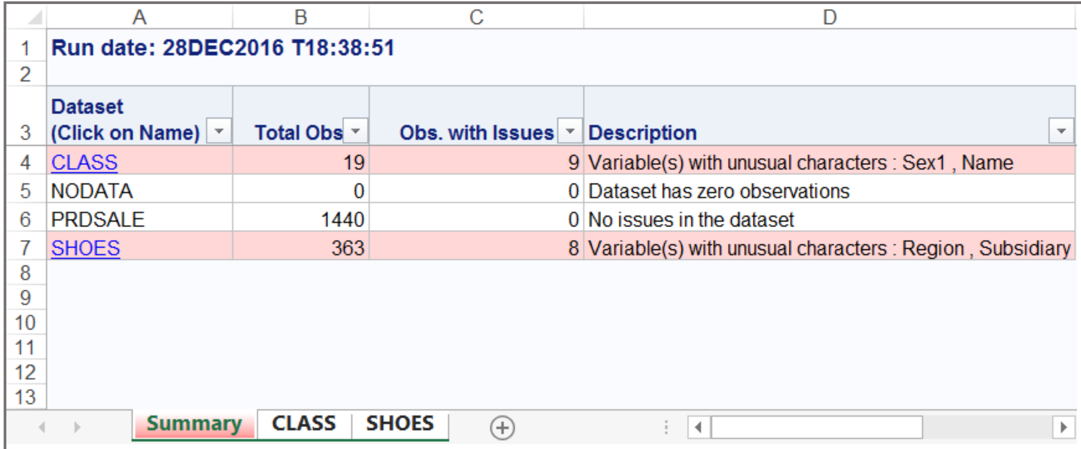
	A	B	C	D
1	Run date: 28DEC2016 T18:38:51			
2				
3	Dataset (Click on Name)	Total Obs.	Obs. with Issues	Description
4				
5				
6				
7				
8				
9				
10				
11				
12				
13				
14				
15				

Display 2: Summarized view

All datasets are processed and the “Summary” tab in the excel workbook lists them. The datasets with issues are hyperlinked which points to the respective tab by clicking on the dataset name. “Total Obs.” column displays the observation count in each dataset; while the “Obs. with Issues” column displays the observation count for records with issues. Additional descriptions are provided in the last column.

EXECUTION OF THE REPORT

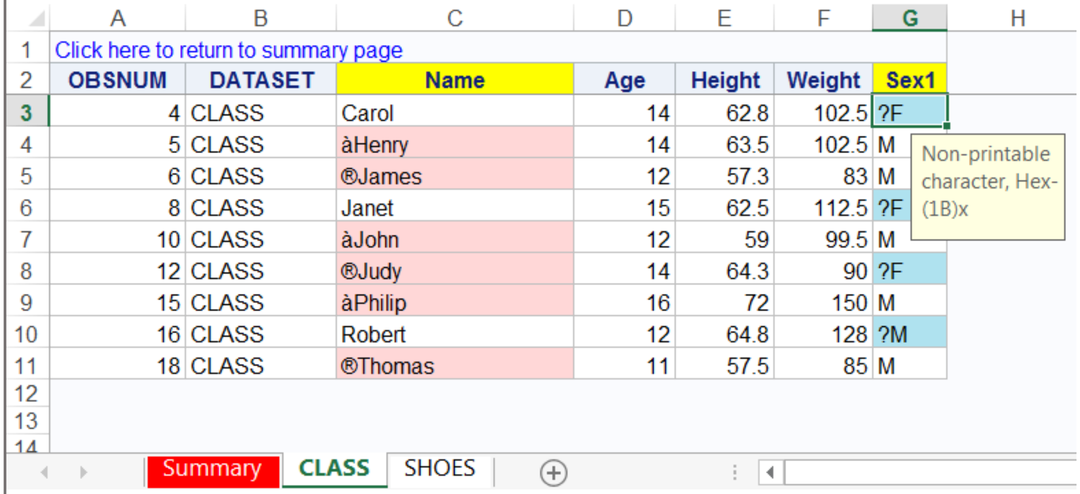
The 4 datasets created in Test Data section is put to test and Display 3, 4 and 5 exhibits the result.



Run date: 28DEC2016 T18:38:51

Dataset (Click on Name)	Total Obs	Obs. with Issues	Description
CLASS	19	9	Variable(s) with unusual characters : Sex1 , Name
NODATA	0	0	Dataset has zero observations
PRDSALE	1440	0	No issues in the dataset
SHOES	363	8	Variable(s) with unusual characters : Region , Subsidiary

Display 3: Summarized View of Test Datasets

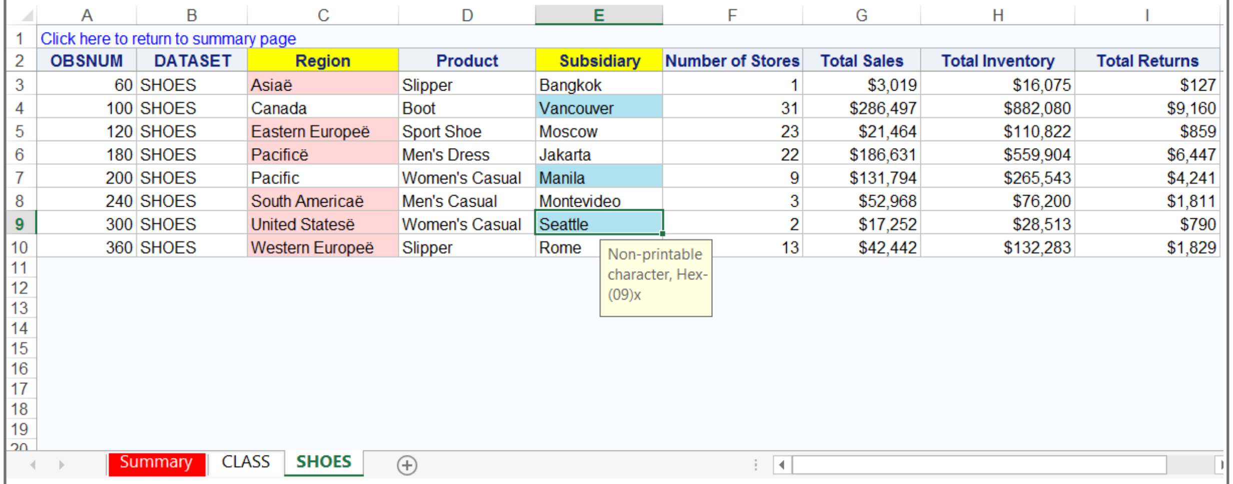


Click here to return to summary page

OBSNUM	DATASET	Name	Age	Height	Weight	Sex1
4	CLASS	Carol	14	62.8	102.5	?F
5	CLASS	àHenry	14	63.5	102.5	M
6	CLASS	®James	12	57.3	83	M
8	CLASS	Janet	15	62.5	112.5	?F
10	CLASS	àJohn	12	59	99.5	M
12	CLASS	®Judy	14	64.3	90	?F
15	CLASS	àPhilip	16	72	150	M
16	CLASS	Robert	12	64.8	128	?M
18	CLASS	®Thomas	11	57.5	85	M

Non-printable character, Hex-(1B)x

Display 4: Detailed View of Test Datasets (Part I)



Click here to return to summary page

OBSNUM	DATASET	Region	Product	Subsidiary	Number of Stores	Total Sales	Total Inventory	Total Returns
60	SHOES	Asiaë	Slipper	Bangkok	1	$3,019	$16,075	$127
100	SHOES	Canada	Boot	Vancouver	31	$286,497	$882,080	$9,160
120	SHOES	Eastern Europeë	Sport Shoe	Moscow	23	$21,464	$110,822	$859
180	SHOES	Pacificë	Men's Dress	Jakarta	22	$186,631	$559,904	$6,447
200	SHOES	Pacific	Women's Casual	Manila	9	$131,794	$265,543	$4,241
240	SHOES	South Americaë	Men's Casual	Montevideo	3	$52,968	$76,200	$1,811
300	SHOES	United Statesë	Women's Casual	Seattle	2	$17,252	$28,513	$790
360	SHOES	Western Europeë	Slipper	Rome	13	$42,442	$132,283	$1,829

Non-printable character, Hex-(09)x

Display 5: Detailed View of Test Datasets (Part II)

Key Notes:

- Display 3: Dataset names CLASS and SHOES are hyperlinked to indicate the presence of non-printable and special characters in atleast one of the records. Also, the corresponding rows are highlighted in ‘light red’ to indicate the same.
- Display 4 and 5: Clicking on the dataset name in Display 3 takes to the corresponding tab in Display 4 and 5. Some of the key features in these detailed tabs are – variable names containing unusual characters are highlighted in ‘yellow’; cells containing special characters are highlighted in ‘light red’; cells containing non-printable characters are highlighted in ‘cyan’ along with a tool-tip indicating the hexadecimal code associated with that character; each detailed tab has a link at the first row “*Click here to return to summary page*” for easy navigation to the summary tab.

Please refer to Appendix for the complete SAS program that was used to create the report.

CONCLUSION

It is strongly encouraged to check for non-printable and special characters as the initial steps in data cleaning activities. To facilitate this, a report should be generated which can be shared with Data Management or other cross-functional groups as applicable. The report template presented here is one of the sample template which can an expanded to meet individual requirements.

REFERENCES

[1] ASCII Code, “ASCII Code – The extended ASCII table”, http://www.ascii-code.com/

[2] Dodlapati, Sridhar R, Lakkaraju Praveen, 2010. “Non Printable & Special Characters: Problems and how to overcome them”. PharmaSUG 2010 – Paper CC13.

[3] SAS Support, “Sample 24716: Deleting unprintable characters from character variables”, http://support.sas.com/kb/24/716.html

CONTACT INFORMATION

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

Name: Abhinav Srivastva
Enterprise: Gilead Sciences, Inc.
E-mail: srivastvaabhinav@gmail.com

APPENDIX

```
/*-------------------------------------------------------------------------*/
/* Macro Input : Path/Directory where all datasets are stored              */
/* Macro Output: .XLS file                                                 */
/*-------------------------------------------------------------------------*/

options noquotelenmax;
%macro check_datasets (path= );

/* Read all datasets and store into macro variables */
libname DIR "&path.";

proc sql noprint;
  create table datasets as
    select memname from dictionary.members
    where libname='DIR'
    ;
  select count(*) into:N_DS
    from datasets
    ;
  select memname into:dtname1 - :dtname%sysfunc(strip(&N_DS))
     from datasets
    ;
quit;

/* Loop through all Datasets to identify NPSC=Non Printable & Special Chars*/
%do ds=1 %to &N_DS;

data _null_;
  if 0 then set DIR.&&dtname&ds. nobs=n;
    call symputx('rows',put(n,best.));
    stop;
run;

%if &rows. ^= 0 %then %do;

data _null_;
   set DIR.&&dtname&ds.;
    array char_vars{*}   _character_;
    call symputx("char_num",put(dim(char_vars),best.));
run;

data &&dtname&ds.._1;
   retain OBSNUM;
   retain DATASET;
   set DIR.&&dtname&ds.;
    array char_vars{*}   _character_;
      array new_vars {*} $500 ___c1-___c&char_num.;
      ___max_chars = &char_num.;
      length ___varname $500 DATASET $50;
      retain ___varname ' ';
          do i=1 to dim(char_vars);
                 do j=0 to 31,127; *-- Non-printable character ;
                if index(char_vars{i},byte(j))>0 then do;

new_vars{i}=vname(char_vars{i})||"#NP#("||strip(put(byte(j),$hex4.))||")x";
```

```
                           if index(___varname,vname(char_vars{i}))=0 then
___varname=catx(' , ',strip(___varname),vname(char_vars{i})) ;
                              ___flag='Y';
                        end;
                   end;
                 do k=128 to 255; *-- Special Characters ;
                if index(char_vars{i},byte(k))>0 then do;
                  new_vars{i}=vname(char_vars{i})||'#SP';
                           if index(___varname,vname(char_vars{i}))=0 then
___varname=catx(' , ',strip(___varname),vname(char_vars{i})) ;
                            ___flag='Y';
                      end;
                   end;
              end;

      ___DUMMY = ' ';
      OBSNUM   = _n_;
      DATASET  = "&&dtname&ds.";

      drop i j k;
run;

/* Get Overall Summary from above DATA step into macro variables */
proc sql noprint;
  select count(*) into:N_obs
     from &&dtname&ds.._1
  ;
  select count(*) into:I_obs
     from &&dtname&ds.._1
       where ___flag='Y';
  ;
  select strip(___varname) into:Var_list
     from &&dtname&ds.._1 (firstobs=&N_obs)
  ;
quit;

%end;

/* For zero observation datasets */
%else %do;

   data &&dtname&ds.._1;
     length DATASET $50;
       DATASET="&&dtname&ds.";
         call symputx('N_obs',0);
         call symputx('I_obs',0);
         call symputx('Var_list',' ');
   run;

%end;

/* Summary tab view */
data part1;
   length DATASET $50 DESC $1000;
        DATASET = "&&dtname&ds.";
        TOT_OBS = &N_obs.;
        I_OBS   = &I_obs.;
        DUMMY   = I_OBS;
              if &N_obs. = 0 then DESC = 'Dataset has zero observations';
         else if &I_Obs. = 0 then DESC = 'No issues in the dataset';
         else                     DESC = "Variable(s) with unusual characters
: &Var_list.";
run;

proc append base = summary data = part1;
run;

%end;

/* Create a Format for hyperlinks on Summary tab */
data fmtset;
   set summary (where=(I_OBS ne 0)) end=last;
   length label $50;
    retain fmtname 'ds_names' type 'C';
    start=dataset;
    end  =dataset;
    label="#"||strip(dataset)||'!A1';
    output;
      if last then do;
            hlo  ='O';
            label=' ';
            output;
      end;
run;

proc format cntlin=fmtset;
run;

/* Calculate # of datasets for detailed view */
proc sql noprint;
  create table detailed_ds as
    select * from summary
        where I_OBS ^= 0
  ;
  select count(*) into:cnt_d
   from detailed_ds
  ;
   %if &cnt_d ^= 0 %then %do;
     select dataset into:ddtname1-:ddtname%sysfunc(strip(&cnt_d))
          from detailed_ds
     ;
    %do s=1 %to &cnt_d;
     select tranwrd(strip(___varname),',',' '), ___max_chars
            into:arr_list&s.,
                :arr_len&s.
        from &&ddtname&s.._1
        having obsnum=max(obsnum)
     ;
      %end;
   %end;
quit;

/* Generate Report Using ODS Excel */
ods excel file="&path\data_check.xlsx";
ods excel options (AUTOFILTER           = 'ALL'
                   ABSOLUTE_COLUMN_WIDTH ='16,14,18,52'
                   FROZEN_HEADERS        = 'ON'
                   SHEET_INTERVAL        = 'NONE'
                   EMBEDDED_TITLES       = 'ON'
                   ROW_REPEAT            = '1-3'
                   ORIENTATION           = 'LANDSCAPE'
                   TAB_COLOR             = 'red'
                   SHEET_NAME            = 'Summary');

/* Summary tab PROC REPORT */
  title j=l "Run date: %sysfunc(date(),date9.) T%sysfunc(time(),tod8.)";

  proc report data = summary nowd split='*';
    column DUMMY DATASET TOT_OBS I_OBS DESC;
    define DUMMY  /display noprint;
    define DATASET/display 'Dataset*(Click on Name)' style(header)=[just=l];
    define TOT_OBS/display 'Total Obs.';
    define I_OBS  /display 'Obs. with Issues';
    define DESC   /display 'Description' style(header)=[just=l];
      compute DATASET;
       if DUMMY ^= 0 then call define(_col_,'style',"style=[url=$ds_names.
textdecoration=underline color=blue]");
      endcomp;
      compute I_OBS;
       if I_OBS ^= 0 then call
define(_row_,'style',"style=[background=cxFFD7D7]");
      endcomp;
  run;

/* Detailed tab PROC REPORT */
%if &cnt_d ^= 0 %then %do;
    %do t=1 %to &cnt_d;

    ods excel options (AUTOFILTER            = 'NONE'
                       ABSOLUTE_COLUMN_WIDTH = 'NONE'
                       FROZEN_HEADERS        = 'ON'
                       ROW_REPEAT            = '1-2'
                       ORIENTATION           = 'LANDSCAPE'
                       SHEET_INTERVAL        = 'PROC'
                       TAB_COLOR             = 'white'
                       SHEET_NAME            = "&&ddtname&t.");

     title;
     proc report data=&&ddtname&t.._1 (where=(___flag='Y')) nowd
style(lines)=[url="#Summary!A1" foreground=blue just=l];
         *-- Highlight column headers with NPSC --*;
       %let v=1;
        %do %while(%scan(&&arr_list&t.,&v.,%str( )) ne );
             %let tmp=%scan(&&arr_list&t.,&v.,%str( ));
               define &tmp / style(header)=[background=cxFFFF00];
                  %let v=%eval(&v.+1);
        %end;
         *-- Suppress printing temporary variables --*;
           %if %sysfunc(strip(&&arr_len&t.)) = 1 %then %do;
               define ___c1        / display noprint;
             %end;
```

```
          %else %do;
               define ___c1-___c%sysfunc(strip(&&arr_len&t.)) / display
noprint;
              %end;
               define ___max_chars / display noprint;
               define ___varname   / display noprint;
               define ___flag      / display noprint;
               define ___dummy     / display noprint;
         *-- Add background color with cells containing NPSC --*;
                 compute ___dummy;
               array sel_char{*} &&arr_list&t.;
                         %if %sysfunc(strip(&&arr_len&t.)) = 1 %then %do;
                   array sel_tmp {*} ___c1 ;
                         %end;
                  %else %do;
                           array sel_tmp {*} ___c1-
___c%sysfunc(strip(&&arr_len&t.)) ;
                          %end;
                           do i=1 to dim(sel_char);
                              do j=1 to dim(sel_tmp);
                     if "'"||strip(vname(sel_char{i}))||"'" =
"'"||strip(scan(sel_tmp{j},1,'#'))||"'" then
                                 do;
                                     if strip(scan(sel_tmp{j},2,'#'))='SP'
then do;
                          call
define(vname(sel_char{i}),'style','style=[background=cxFFD7D7]');
                                end;
                       else if strip(scan(sel_tmp{j},2,'#'))='NP' then do;
                          call define(vname(sel_char{i}),'style',
                          "style=[background=cxAFE2EF flyover='Non-printable
character, Hex-"||strip(scan(sel_tmp{j},3,'#'))||"']");
                       end;
                      end;
                               end;
                              end;
                 endcomp;
              compute before _page_ ;
                         line "Click here to return to summary page";
              endcomp;
     run;
      %end;
%end;

ods excel close;

%mend;

%check_datasets (path = C:\temp);
```