PharmaSUG2010 - Paper CC13

Non Printable & Special Characters: Problems and how to overcome them

Sridhar R Dodlapati, i3 Statprobe, Basking Ridge, NJ
Praveen Lakkaraju, Naresh Tulluru and Zemin Zeng
Forest Laboratories Inc. Jersey City, NJ

ABSTRACT

Non printable & special characters in clinical trial data create potential problems in producing quality deliverables. There could be major issues such as incorrect statistics / counts in the deliverables, or minor ones such as incorrect line breaks, page brakes or appearance of strange symbols in the reports. Identifying and deleting these issues could pose challenges. When faced with this issue in Pharmaceutical & Biotech industry, it is imperative to clean them up. We need to understand the underlying cause and use various techniques to identify and handle them.

KEY WORDS

Non Printable, Invisible, Special, ASCII table, TRANTAB, Compress, K & W modifiers, Indexc, Byte and Rank.

INTRODUCTION

When SAS programmers encounter any non printable & special character related issues in clinical trial data for the first time, it might be time consuming to figure out the reason that is causing the problem. In this paper we are trying to provide an awareness of non printable & special characters, discuss issues that might be caused by them and provide corresponding solutions.

In this paper we have used "NPSC" as a short form for non printable & special characters for convenience. The macros and the examples used in this paper are implemented on UNIX operating system with SAS version 9.1.3.

BACKGROUND INFORMATION

Some of the most common non printable characters are carriage return, form feed, line feed, backspace, escape, horizontal tab and vertical tab. These might not have a visible shape but will have effects on the output. To further understand them, we have to look into ASCII table.

ASCII TABLE

ASCII stands for American Standard Code for Information Interchange. ASCII was originally designed for use with teletypes. Computers can only understand numbers; hence an ASCII code is the numerical representation of a character such as 'a' or 'A' or an action such as 'ESC' or 'DEL'. There are total of 256 ASCII characters (including extended ASCII characters) (decimal values range from 0 to 255). Tables 1, 2 & 3 in Appendix show details of the ASCII characters.

For the purpose of our topic, we can broadly classify the characters into 3 groups:

1. 33 non printable special characters. The first 32 characters (decimal value from 0 to 31) and the DEL char (decimal value 127).
2. 94 standard printable characters (decimal value range from 33 to 126) which represent letters, digits, punctuation marks, and a few miscellaneous symbols.
3. 128 special characters (Extended ASCII or ISO-8859-1. Decimal values range from 128 to 255). Decimal values from 128 to 159 in the Extended ASCII set are non printing control characters.

The "Space" character (decimal value 32) denotes the space between words, as produced by the space bar of a keyboard and it is considered as an invisible graphic rather than a control character.

All the characters that correspond to decimal values between 0 and 127 represent the standard ASCII character set (Standard across the operating environments i.e. operating system / application / font). Other ASCII characters that correspond to decimal values between 128 and 255 are available on certain ASCII operating environments, but the information those characters represent varies with the operating environment. As the need for understanding

additional characters and non printing characters by computers has risen, the standard character set of ASCII became restrictive and a few varying 'extended' sets have been put in place.

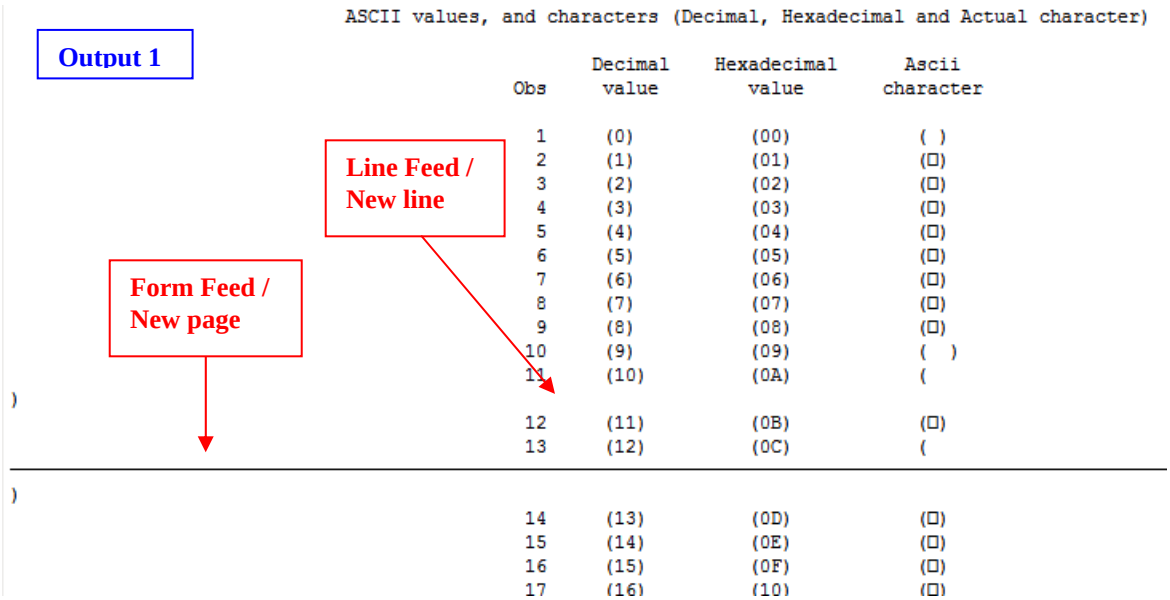
PROBLEMS CAUSED BY NON PRINTABLE & SPECIAL CHARACTERS

In Clinical trials data, we do not expect to have any characters outside the decimal values range from 32 to 127 because of the problems mentioned below. There are some exceptions though which are later presented in this paper.

Following are some of the issues that might be caused by NPSC.

1. The line / page alignment in the output generated will be disrupted when some of these characters are present in the output. Most common problem is, even though there is plenty of space available in a line / page, with out using all of it, the data will spill over to the next line / page.
2. Depending on their presence in the critical variables, one might get wrong statistics or counts in the outputs.
3. Unexpected conditional statement results and/or incorrect number of records get selected during subset.
4. Some characters (Extended ASCII characters) are not same across operating systems / applications/ fonts. When such characters are present in a SAS dataset, it is possible that the character might have had a different form or meaning in the source application compared to the final destination which is SAS dataset.

We make an attempt to print all ASCII characters to examine their effects. The SAS code used to generate the below output (Output 1) is presented in the APPENDIX as output1.sas. Below is the partial output:



ASCII values, and characters (Decimal, Hexadecimal and Actual character)

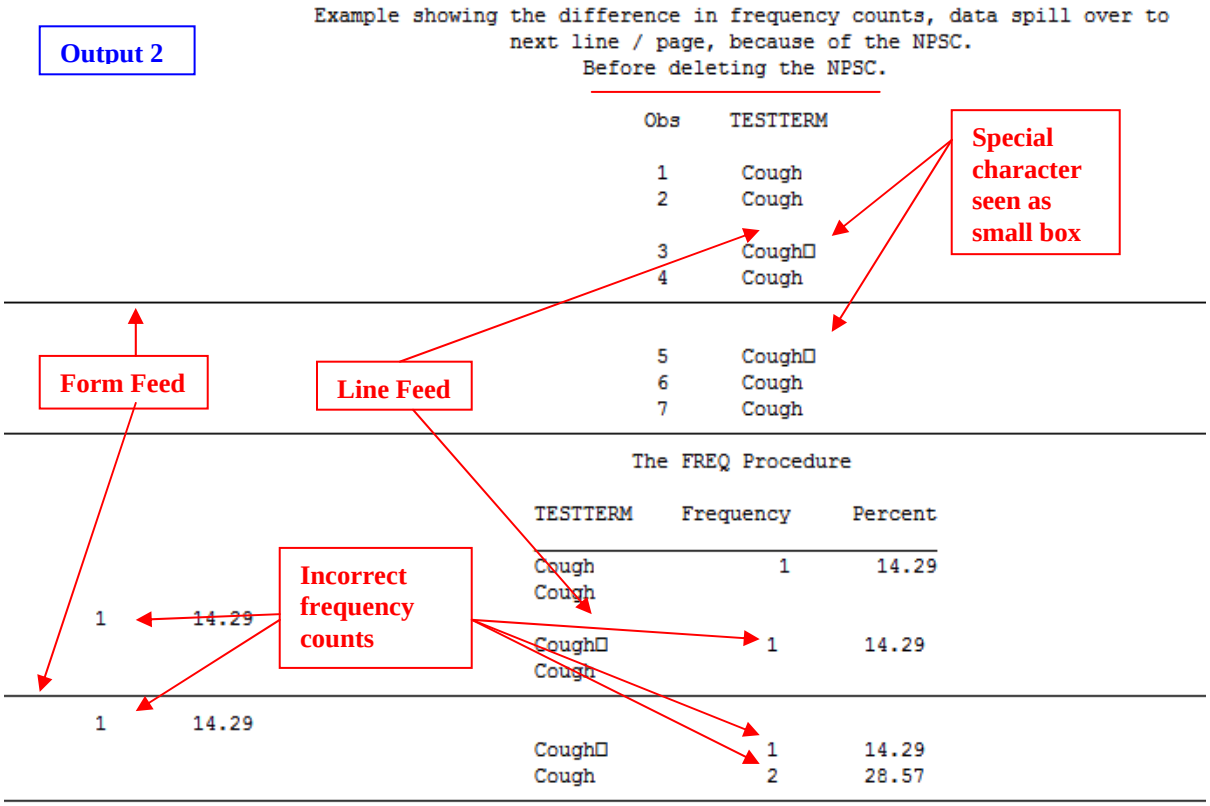
Obs	Decimal value	Hexadecimal value	Ascii character
1	(0)	(00)	( )
2	(1)	(01)	(□)
3	(2)	(02)	(□)
4	(3)	(03)	(□)
5	(4)	(04)	(□)
6	(5)	(05)	(□)
7	(6)	(06)	(□)
8	(7)	(07)	(□)
9	(8)	(08)	(□)
10	(9)	(09)	( )
11	(10)	(0A)	(
12	(11)	(0B)	(□)
13	(12)	(0C)	(
14	(13)	(0D)	(□)
15	(14)	(0E)	(□)
16	(15)	(0F)	(□)
17	(16)	(10)	(□)

In the above output there are 3 variables. The first one has decimal value, the second has hexadecimal value and the third one has the character. All of them are enclosed in parenthesis. Observation 11 has non printable character that corresponds to new line (NL line feed, DECIMAL value = 10, HEXADECIMAL value = '0A') and Observation 13 has non printable character that corresponds to new page (NP form feed, DECIMAL value = 12, HEXADECIMAL value = '0C'). In the 11th observation when the character (new line) was printed, it has been forced to the next line. The same way, in the 13th observation when the character (new page) was printed, it has been forced to the next page. Also observe that some of the characters were printed as small boxes.

Another example is presented below to demonstrate the non printable & special characters effects in conjunction with data. The SAS code used to generate the below outputs (Output 2 and 3) is presented in the APPENDIX as output_2_3.sas. Upon closely examining the output 2, we can see that, after the second 'Cough', there is an extra line skip, and after the fourth 'Cough', there is a page break (here it is seen as the solid line). Even though the value 'Cough' in the TESTTERM looks alike, they have different frequency counts. This is because of the last invisible non printable character in them.

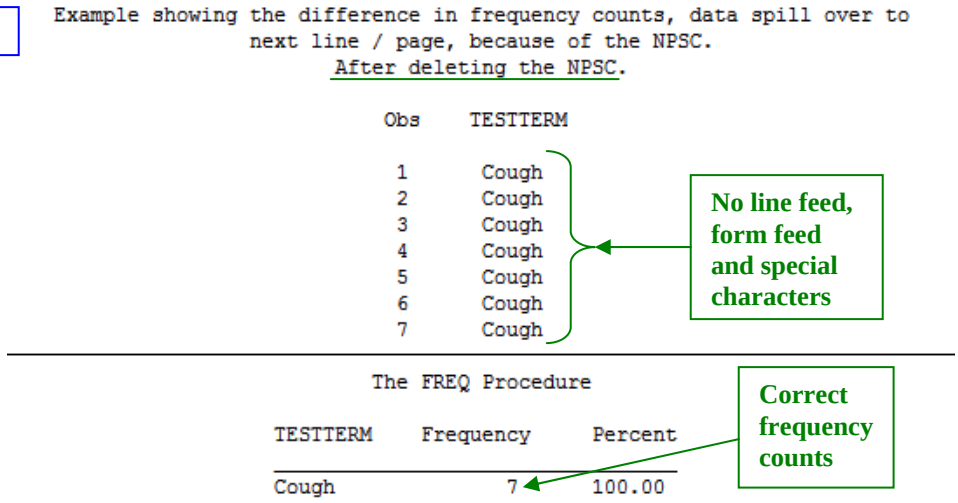
When using conditional statements, inaccurate results are possible, and incorrect number of records can get selected during subset for the same reasons mentioned above. Ex: Value **'YES'** is not same as **'YES?'**; Value **'COUGH'** is not same as **'COUGH?'** where **'?'** is a NPSC. For this reason the statement ***upcase(varx) = 'YES';*** doesn't work, but ***index(upcase(varx)) = 'YES';*** works.

Output 2



Below is the output 3 which is created after deleting the NPSC from the same dataset that is used to generate the output 2. Output 3 is appropriate without any line skip, page break, and correct frequency count as expected.

Output 3



As some special characters are not same across all the environments, they might not mean / look like what they were meant / looked like in the source. In such instance, the special character does not make sense in the context.

SOURCE

We do need, and use some of these non printable & special characters in various applications such as Word, Excel and other editors. However in clinical trial data, these characters can cause problems as explained above. Hence they are not acceptable in the data. If they are not allowed, then how they were entered in to the clinical data in the first place?

NPSC might be introduced into database when the data is imported from applications such as Excel sheets, Word document or other editors. It is not possible to enter some of these special characters / symbols into our data just by using the key board, unless those were entered programmatically by using some special techniques e.g. BYTE

function in SAS. When viewed in the dataset, some of these characters might appear as a small box in the data, but it might not be the case always.

Most of the times, SAS programmers will be given data from other departments (usually data management), and do not have any control over it. Providing quality data is data management's responsibility, and providing quality / accurate reports is statistical programmer's responsibility. **Hence, it is both data management and statistical programming department's responsibility to identify the NPSC and take necessary action**.

IDENTIFYING AND GENERATING A REPORT ABOUT THE NPSC IN A GIVEN DATABASE

What ever might be the source for these NPSC in our database, problems caused by them are often difficult to identify or go unrecognized / overlooked. Hence a robust approach is needed to find them in a given database. For this purpose we have developed a macro called **RPTNPSC** which is presented in the APPENDIX. This macro identifies and generates a detailed report of all occurrences of NPSC in a given database. Once all the datasets and the variables containing NPSC are identified, we have to analyze them to decide what necessary action can be taken depending on ones requirements. RPTNPSC macro can be used as a powerful **edit check tool** to ensure cleaner data without NPSC.

Below is the first part of the report (output 4) generated by RPTNPSC macro that gives summary information of the datasets and variables with number of observations having NPSC.

Datasets, variables that contain NPSC and their counts in /testdrug/rawdata.

Output 4

Dataset name	Variable name	Num of Obs with NPSC
AE	AETERM	6
ECG	ECGCOM	2
CM	CMTRT	1
LB	UNIT	1

Below is the partial second part of the report (output 5) generated by RPTNPSC macro. For all the datasets in the specified database, a detailed report of NPSC containing dataset, observation number, variable name, number of NPSC instances (NPSC count) in that observation for that variable, complete information of each NPSC (i.e. position within that variable, Decimal & Hexadecimal values), and finally the variable value. We can see the disrupted alignment in the below sample output, which is due to line feed.

Detailed report of NPSC in the directory /testdrug/rawdata.

Output 5

Dataset name	Obs number	Variable name	NPSC count	NPSC information (position, decimal value, hexadecimal value)	Value
AE	203	AETERM	2	(20, 13, '0D'), (21, 10, '0A')	cellulitis left leg
AE	274	AETERM	2	(29, 13, '0D'), (30, 10, '0A')	Possible tendonitis left arm
AE	305	AETERM	2	(32, 13, '0D'), (33, 10, '0A')	Pneumonia - outpatient ER Visit
AE	546	AETERM	4	(52, 13, '0D'), (53, 10, '0A'), (54, 13, '0D'), (55, 10, '0A')	Tin
AE	682	AETERM	2	(71, 13, '0D'), (72, 10, '0A')	Subject reported swelling in her right leg from knee to ankle. Rash upper back. right wrist tendinitis and weight gain.
AE	737	AETERM	2	(16, 13, '0D'),	dermatitis face

Line feed at the end of the value causing line break.

Carriage return (OD) and Line feed (OA) information.

Line feed in the middle of the value causing line break and data spill to next line.

SOLUTIONS

Once the NPSC are identified in the database, it is essential to analyze all NPSC occurrences as there are many different solutions available depending on various situations and their specific requirements.

Special characters like “µ” (which stands for “micro”) in the lab units or any special characters that were entered in the database intentionally are required; hence these special characters need to be kept as is in the database. Once we identify the characters that are to be deleted, then the following are the options / solutions:

1. Report them to data management and get clean data in the next transfer.
2. Replace NPSC with other characters.
3. Delete NPSC.

REPORT THEM TO DATA MANAGEMENT AND GET CLEAN DATA IN THE NEXT TRANSFER

If the identified NPSC are not supposed to be present in the database, and if data management can address these occurrences, then this is the most preferred option.

REPLACE NPSC WITH OTHER CHARACTERS

Replacing NPSC with other characters such as space is usually rare, but is some times required to handle some special situations. In such cases, first the position of the NPSC is identified and then by using the SUBSTR (left of =) or TRANSLATE functions, it is replaced by another character.

DELETE NPSC

After identifying the NPSC, if none of the above two solutions is an option, then we have to delete them from the database. There are many ways to do this, and in this paper we discuss some methods along with their pros and cons.

Method I:

The simplest approach is to use the compress function as below:

```
unit = compress(unit, '09'x); *remove the horizontal tab;
```

In this method, specific NPSC occurrences can be targeted and removed within a single variable. This method could be cumbersome, if there are many variables with NPSC and all of them need to be deleted.

Method II:

Compress function with modifiers ‘K’ and ‘W’ is a new feature in SAS version 9. Following is the syntax:

```
varname = compress(varname, , 'kw');
```

The modifier “k” stands for ‘KEEP’ and the modifier “w” stands for ‘WRITABLE’. Note that there is no second parameter in the above code. When compress function is used in combination of K & W modifiers, it keeps all the writable characters which means it deletes all the non writable characters.

Here are the disadvantages of this approach:

1. Only available from SAS version 9 onwards.
2. This method will delete all the non printable characters, but does not delete any special characters.
3. Even while handling only non printable characters, this method has certain restrictions. Characters to be considered as non-printable by SAS, depends on TRANTAB system option settings. The results depend directly on the translation table that is in effect and indirectly on the ENCODING and LOCALE system options. Translation tables are used internally by the SAS supervisor to implement NLS (National Language Support). Hence changing the TRANTAB options is strongly not recommended without proper purpose and knowledge. Especially the non printing control characters (special characters with decimal values from 128 to 159 in the Extended ASCII set) are considered as printable characters in some settings and non printable characters in other settings. So, the compress function in combination with K & W modifiers does not guarantee consistent result on which we can rely on.

Method III:

In this method the NPSC range is explicitly specified in the compress function. This can be done by creating a variable with all the characters that need to be deleted and passing it as a second parameter in the compress function. Below is the code to create a variable “npschars” that has all the NPSC:

```
do i=0 to 31, 127 to 255;
   if i=0 then npschars=byte(i);
   else npschars=trim(npschars)||byte(i);
end;
```

This is the best approach as it gives us the control over the characters that we want to delete and doesn’t involve any alteration of the system options settings.

Keeping the pros & cons of each method in mind, we have developed the **DELNPSC** macro to delete or replace the NPSC in the given dataset which is presented in the APPENDIX.

DIFFERENCE BETWEEN USING THE “KW” MODIFIER AND EXPLICITELY SPECIFYING THE RANGE OF CHARACTERS THAT NEEDS TO BE DELETED

The importance of explicitly mentioning the NPSC in compress function is shown below by comparing this method with compress function with “KW” modifiers. The SAS code used to generate the below output (Output 6) is presented in the APPENDIX as output6.sas. Only partial output is shown below:

Output 6

Difference between using the 'KW' modifier and Explicitly specifying the characters that need to be deleted in 'COMPRESS function'.

Obs	Decimal value	Hexadecimal value	Ascii character	Char after 'COMPRESS func with KW mod'	Char after 'NPSC specified in COMPRESS'
58	(152)	(98)	(˜)	()	()
59	(153)	(99)	(™)	()	()
60	(154)	(9A)	(š)	()	()
61	(155)	(9B)	(›)	()	()
62	(156)	(9C)	(œ)	()	()
63	(157)	(9D)	(□)	()	()
64	(158)	(9E)	(ž)	()	()
65	(159)	(9F)	(Ÿ)	()	()
66	(160)	(A0)	( )	( )	()
67	(161)	(A1)	(¡)	(¡)	()
68	(162)	(A2)	(¢)	(¢)	()
69	(163)	(A3)	(£)	(£)	()
70	(164)	(A4)	(¤)	(¤)	()
71	(165)	(A5)	(¥)	(¥)	()
72	(166)	(A6)	(¦)	(¦)	()
73	(167)	(A7)	(§)	(§)	()
74	(168)	(A8)	(¨)	(¨)	()
75	(169)	(A9)	(©)	(©)	()
76	(170)	(AA)	(ª)	(ª)	()
77	(171)	(AB)	(«)	(«)	()

Difference

In this output, for each observation, the third variable contains the character before it is deleted, the fourth variable contains the character after using the compress function with modifiers ‘K’ and ‘W’, and finally the fifth variable contains the character after using the algorithm that explicitly specifies the NPSC. One can observe that, in the fourth variable, some observations have the special character still present in them, but in the fifth variable, there is no character present for all the observations.

CONCLUSION

Non printable and special characters in the clinical trial data are hard to identify and can cause problems that could take lot of time to identify and fix. We have discussed identifying and generating reports of NPSC occurrences and provided various solutions with examples. We hope this paper provides all needed information to deal with Non Printable Special Characters in clinical trail data.

REFERENCES

SAS online documentation

http://www.asciitable.com/

http://www.danshort.com/ASCIImap/

http://msdn.microsoft.com/en-us/goglobal/cc305145.aspx

http://www.lammertbies.nl/comm/info/ascii-characters.html

ACKNOWLEDGMENTS

The Authors would like to thank the management at i3 Statprobe and Forest Laboratories Inc.

CONTACT INFORMATION

We appreciate your valuable comments and suggestions. Please contact the authors at:

Sridhar R Dodlapati
i3 Statprobe
131 Morristown Road
Basking Ridge, NJ 07920
Sridhar.Dodlapati@i3statprobe.com

Praveen Lakkaraju
Forest Laboratories, Inc.
Harborside Financial Center, Plaza V
Jersey City, NJ 07311
PraveenLakkaraju@gmail.com

Naresh Tulluru
Forest Laboratories, Inc.
Harborside Financial Center, Plaza V
Jersey City, NJ 07311
Naresh.Tulluru@yahoo.com

Zemin Zeng
Forest Laboratories, Inc.
Harborside Financial Center, Plaza V
Jersey City, NJ 07311
Zemin.Zeng@frx.com

APPENDIX

OUTPUT1.SAS

```
/*************************************************************************************************
Program:  Output1.sas

Purpose:  Generate and print all the ASCII characters including the NPSC.
*************************************************************************************************/

data test (drop=i);
   length decimalv $5 hexadecv $6 ascichar $3 ;

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

options nodate nonumber;
title "ASCII values, and characters (Decimal, Hexadecimal and Actual character)";
proc print data=test label;
run;

/**********************/
/*** End of program ***/
/**********************/
```

OUTPUT_2_3.SAS

```
/*************************************************************************************************
Program:  Output_2_3.sas

Purpose:  Example showing the difference in frequency counts, because of the NPSC.
*************************************************************************************************/

data test (keep=testterm);
   length testterm $200;

   do i=9 to 13;
      testterm="Cough"||byte(i); *Concatenating the NPSC with data;
      output;
   end;
   do i=1 to 2;
      testterm="Cough";
      output;
   end;
run;

options nodate nonumber;
title "Example showing the difference in frequency counts, data spill over to";
title2 "next line / page, because of the NPSC.";
title3 "Before deleting the NPSC.";
proc print data=test;
run;

title;
proc freq data=test;
   tables testterm / nocum;
run;

/* Call DELNPSC macro to delete the NPSC in the test dataset. */
/* DELNPSC macro is explained in the APPENDIX. */
%delnpsc (ds=test, cvarlist=, replchar=%str());

title "Example showing the difference in frequency counts, data spill over to";
title2 "next line / page, because of the NPSC.";
title3 "After deleting the NPSC.";
proc print data=test;
run;
```

```
title;
proc freq data=test;
   tables testterm / nocum;
run;

/*********************/
/*** End of program ***/
/*********************/
```

IDENTIFYING AND GENERATING A REPORT ABOUT THE NPSC IN A GIVEN DATABASE

RPTNPSC.SAS (Generates the output 4 and output 5)

```
/*************************************************************************************************
Program:  rptnpsc.sas

Programmer: Sridhar R Dodlapati

Date:     2009-12-15

Purpose:  General purpose macro to report the non printable/special characters in the specified
          Dataset folder.

Input:    Dataset directory path in which NPSC need to be identified in the datasets.

Output:   Report "rptnpsc.lst" will be generated in the same folder where the program is run.

Macro parameter description:
datapath= Path of the dataset folder in which the presence of non printable/special characters
          need to be identified and reported. datapath is not enclosed in quotation marks.

Usage:    rptnpsc macro is called outside the data step as follows:
          %rptnpsc (datapath=/sasdata/neb/NEBMD04);
*************************************************************************************************/

%macro rptnpsc(datapath=);
.....
.....
      /*** Get all the datasets info in the specified data folder ***/
      proc sql noprint;
         *Get all the dataset names that are present in the given directory;
         create table dsindir as
            select distinct upcase(memname) as dsname
            from dictionary.tables
            where libname="DIRPATH" and memtype="DATA";

         *Get the number of datasets for looping;
         %let numds=&sqlobs;

         *Create a macro variable with all the dataset names separated by space;
         select dsname into: dsnames separated by ' '
         from dsindir;
      quit;
.....
.....
         /*** Process all the datasets in the specified data folder ***/
         %do i=1 %to &numds;
            %let ds&i=%lowcase(%sysfunc(scan(&dsnames, &i)));

            data &&ds&i (keep= dsname varname obsnum varvalue npsccnt npscinf);
               set dirpath.&&ds&i;

               .....
               .....
               /*** Process all the character variables in each dataset ***/
               .....
               .....
                     indexc(charvars{k}, npschars)
               .....
               .....

         %end;
```

```
.....
.....

/*** Report generation ***/
.....
.....
%mend;
/**********************/
/*** End of program ***/
/**********************/
```

How this macro works:

In the macro, first the directory path specified in the macro definition is checked and a user defined warning notice is issued if it is incorrect or not specified at all. If the path provided is correct, it creates a macro variable containing the number of datasets in the folder, another macro variable is created that has all the datasets names separated by space. Then a loop is designed to go over each observation for all character variables in each dataset. Inside the loop, while processing the first observation, it creates a string containing all the non printable & special characters that need to be identified. The importance of using a string to specify all NPSC characters is further explained in detail in the Solutions Section in this paper. The identification of NPSC is done by using various functions such as indexc, vname, vvalue, rank & substr and a dataset is created from which the final report is generated.

OUTPUT6.SAS

```
/*************************************************************************************************
Program:  Output6.sas

Purpose:  Example showing the difference between the 2 approaches in compress function.
          1) using 'KW' modifier and
          2) Explicitly specifying the characters that need to be deleted
*************************************************************************************************/

data test (drop=i npschars);
   length decimalv $5 hexadecv $6 ascichar compress specnpsc $3 npschars $161;

   do i=0 to 31, 127 to 255;
      if i=0 then npschars=byte(i);
      else npschars=trim(npschars)||byte(i);
   end;

   do i=0 to 31, 127 to 255;
      decimalv="("||strip(put(rank(byte(i)), best.))||")";
      hexadecv="("||strip(put(byte(i), $hex4.))||")";
      ascichar="("||byte(i)||")";
      compress="("||compress(byte(i), ,'KW')||")";
      specnpsc="("||compress(byte(i),npschars)||")";
      output;
   end;
   label
      decimalv="Decimal value"
      hexadecv="Hexadecimal value"
      ascichar="Ascii character"
      compress="Char after 'COMPRESS func with KW mod'"
      specnpsc="Char after 'NPSC specified in COMPRESS'";
run;

options nodate nonumber;
title "Difference between using the 'KW' modifier and 'Explicitly specifying the";
title2 "characters that need to be deleted' in 'COMPRESS function'.";
proc print data=test  label;
run;

/**********************/
/*** End of program ***/
/**********************/
```

DELETING OR REPLACING THE NPSC IN THE GIVEN DATASET

DELNPSC.SAS

```
/*************************************************************************************************
Program:  delnpsc.sas
```

```
Programmer: Sridhar R Dodlapati

Date:     2009-12-15

Purpose:  General purpose macro to delete or replace the non printable/special characters in
          the specified dataset.

Input:    Dataset that need NPSC clean up.

Output:   The same input dataset will be output in which the NPSC will be deleted or replaced.
          If no NPSC are were present in the input dataset, then the input and output datasets
          will be exactly same.

Macro parameter description:
ds=       Dataset in which the non printable/special characters need to be deleted or replaced.
          This is a required parameter. If not provided, then a warning note will be issued in
          the log.

cvarlist= List of character variables separated by space, that may contains the non
          printable/special characters and need to be deleted or replaced. If nothing is
          specified for the cvarlist, then all the character variables in the specified dataset
          will be used as the character variable list.

replchar= Character used to replace the non printable/special characters. If none specified,
          then the non printable/special characters will be completely deleted. replchar is not
          enclosed in quotations marks, but it should be specified / enclosed in the
          parenthesis of the %str().

Usage:    delnpsc macro is called outside the data step as follows:
          %delnpsc (ds=ae, cvarlist=, replchar=%str());  *Deleting the NPSC;
          %delnpsc (ds=cm, cvarlist=aeterm, replchar=%str( )); *Replacing NPSC with 'SPACE';
          %delnpsc (ds=ex, cvarlist=var1 var2, replchar=%str(?)); *Replacing NPSC with '?';
*************************************************************************************************/

%macro delnpsc (ds=, cvarlist=, replchar=%str());

   %local nullstr warn1 warn2;
   %let warn1=WAR;
   %let warn2=NING;

   %if "&ds"="&nullstr" %then %do;
      %put &warn1&warn2 (User defined): No dataset has been specified in the delnpsc macro call.;
   %end;
   %else %if not %sysfunc(exist(&ds)) %then %do;
      %put &warn1&warn2 (User defined): The specified dataset in the macro call does not exist;
      %put Please specify a valid dataset name.;
   %end;
   %else %do;
      %if "&cvarlist"="&nullstr" %then %let cvarlist=_character_;

      data &ds (drop=i npschars %if "&replchar"^="&nullstr" %then position;);
         set &ds;

         array charvars &cvarlist;
         length npschars $161; *161 is the total number of characters from 0 to 31, 127 to 255;
         retain npschars;

         if _N_=1 then do;
            do i=0 to 31, 127 to 255;
               if i=0 then npschars=byte(i);
               else npschars=trim(npschars)||byte(i);
            end;
         end;

         do over charvars;
            %if "&replchar"="&nullstr" %then %do;
               charvars=compress(charvars, npschars);
            %end;
            %else %do;
               do until (position=0);
                  position=indexc(charvars, npschars);
                  if position>0 then substr(charvars, position, 1)="&replchar";
               end;
            %end;
         end;
      run;
   %end;
%mend;
```

```
/**********************/
/*** End of program ***/
/**********************/
```

How this macro works:

While processing the first observation, first it creates a character variable (NPSCHARS) containing all the non printable & special characters, that need to be deleted or replaced. Then by going over all the character variables one by one for each observation, it either uses the compress function to delete the NPSC or combination of indexc & substr function to replace the NPSC with replace character specified in the macro definition.

	00	01	02	03	04	05	06	07	08	09	0A	0B	0C	0D	0E	0F
00	NUL 0000	STX 0001	SOT 0002	ETX 0003	EOT 0004	ENQ 0005	ACK 0006	BEL 0007	BS 0008	HT 0009	LF 000A	VT 000B	FF 000C	CR 000D	SO 000E	SI 000F
10	DLE 0010	DC1 0011	DC2 0012	DC3 0013	DC4 0014	NAK 0015	SYN 0016	ETB 0017	CAN 0018	EM 0019	SUB 001A	ESC 001B	FS 001C	GS 001D	RS 001E	US 001F
20	SP 0020	! 0021	" 0022	# 0023	$ 0024	% 0025	& 0026	' 0027	(0028	) 0029	* 002A	+ 002B	, 002C	- 002D	. 002E	/ 002F
30	0 0030	1 0031	2 0032	3 0033	4 0034	5 0035	6 0036	7 0037	8 0038	9 0039	: 003A	; 003B	< 003C	= 003D	> 003E	? 003F
40	@ 0040	A 0041	B 0042	C 0043	D 0044	E 0045	F 0046	G 0047	H 0048	I 0049	J 004A	K 004B	L 004C	M 004D	N 004E	O 004F
50	P 0050	Q 0051	R 0052	S 0053	T 0054	U 0055	V 0056	W 0057	X 0058	Y 0059	Z 005A	[005B	\ 005C	] 005D	^ 005E	_ 005F
60	` 0060	a 0061	b 0062	c 0063	d 0064	e 0065	f 0066	g 0067	h 0068	i 0069	j 006A	k 006B	l 006C	m 006D	n 006E	o 006F
70	p 0070	q 0071	r 0072	s 0073	t 0074	u 0075	v 0076	w 0077	x 0078	y 0079	z 007A	{ 007B	| 007C	} 007D	~ 007E	DEL 007F
80	€ 20AC		‚ 201A	ƒ 0192	„ 201E	… 2026	† 2020	‡ 2021	ˆ 02C6	‰ 2030	Š 0160	‹ 2039	Œ 0152		Ž 017D	
90		‘ 2018	’ 2019	“ 201C	” 201D	• 2022	– 2013	— 2014	˜ 02DC	™ 2122	š 0161	› 203A	œ 0153		ž 017E	Ÿ 0178
A0	NBSP 00A0	¡ 00A1	¢ 00A2	£ 00A3	¤ 00A4	¥ 00A5	¦ 00A6	§ 00A7	¨ 00A8	© 00A9	ª 00AA	« 00AB	¬ 00AC	- 00AD	® 00AE	¯ 00AF
B0	° 00B0	± 00B1	² 00B2	³ 00B3	´ 00B4	µ 00B5	¶ 00B6	· 00B7	¸ 00B8	¹ 00B9	º 00BA	» 00BB	¼ 00BC	½ 00BD	¾ 00BE	¿ 00BF
C0	À 00C0	Á 00C1	Â 00C2	Ã 00C3	Ä 00C4	Å 00C5	Æ 00C6	Ç 00C7	È 00C8	É 00C9	Ê 00CA	Ë 00CB	Ì 00CC	Í 00CD	Î 00CE	Ï 00CF
D0	Ð 00D0	Ñ 00D1	Ò 00D2	Ó 00D3	Ô 00D4	Õ 00D5	Ö 00D6	× 00D7	Ø 00D8	Ù 00D9	Ú 00DA	Û 00DB	Ü 00DC	Ý 00DD	Þ 00DE	ß 00DF
E0	à 00E0	á 00E1	â 00E2	ã 00E3	ä 00E4	å 00E5	æ 00E6	ç 00E7	è 00E8	é 00E9	ê 00EA	ë 00EB	ì 00EC	í 00ED	î 00EE	ï 00EF
F0	ð 00F0	ñ 00F1	ò 00F2	ó 00F3	ô 00F4	õ 00F5	ö 00F6	÷ 00F7	ø 00F8	ù 00F9	ú 00FA	û 00FB	ü 00FC	ý 00FD	þ 00FE	ÿ 00FF

Table 1: All ASCII characters with Hexadecimal values.

Dec	Hx	Oct	Char	Dec	Hx	Oct	Html	Chr	Dec	Hx	Oct	Html	Chr	Dec	Hx	Oct	Html	Chr
0	0	000	NUL (null)	32	20	040		Space	64	40	100	@	@	96	60	140	`	`
1	1	001	SOH (start of heading)	33	21	041	!	!	65	41	101	A	A	97	61	141	a	a
2	2	002	STX (start of text)	34	22	042	"	"	66	42	102	B	B	98	62	142	b	b
3	3	003	ETX (end of text)	35	23	043	#	#	67	43	103	C	C	99	63	143	c	c
4	4	004	EOT (end of transmission)	36	24	044	$	$	68	44	104	D	D	100	64	144	d	d
5	5	005	ENQ (enquiry)	37	25	045	%	%	69	45	105	E	E	101	65	145	e	e
6	6	006	ACK (acknowledge)	38	26	046	&	&	70	46	106	F	F	102	66	146	f	f
7	7	007	BEL (bell)	39	27	047	'	'	71	47	107	G	G	103	67	147	g	g
8	8	010	BS (backspace)	40	28	050	(	(	72	48	110	H	H	104	68	150	h	h
9	9	011	TAB (horizontal tab)	41	29	051	)	)	73	49	111	I	I	105	69	151	i	i
10	A	012	LF (NL line feed, new line)	42	2A	052	*	*	74	4A	112	J	J	106	6A	152	j	j
11	B	013	VT (vertical tab)	43	2B	053	+	+	75	4B	113	K	K	107	6B	153	k	k
12	C	014	FF (NP form feed, new page)	44	2C	054	,	,	76	4C	114	L	L	108	6C	154	l	l
13	D	015	CR (carriage return)	45	2D	055	-	-	77	4D	115	M	M	109	6D	155	m	m
14	E	016	SO (shift out)	46	2E	056	.	.	78	4E	116	N	N	110	6E	156	n	n
15	F	017	SI (shift in)	47	2F	057	/	/	79	4F	117	O	O	111	6F	157	o	o
16	10	020	DLE (data link escape)	48	30	060	0	0	80	50	120	P	P	112	70	160	p	p
17	11	021	DC1 (device control 1)	49	31	061	1	1	81	51	121	Q	Q	113	71	161	q	q
18	12	022	DC2 (device control 2)	50	32	062	2	2	82	52	122	R	R	114	72	162	r	r
19	13	023	DC3 (device control 3)	51	33	063	3	3	83	53	123	S	S	115	73	163	s	s
20	14	024	DC4 (device control 4)	52	34	064	4	4	84	54	124	T	T	116	74	164	t	t
21	15	025	NAK (negative acknowledge)	53	35	065	5	5	85	55	125	U	U	117	75	165	u	u
22	16	026	SYN (synchronous idle)	54	36	066	6	6	86	56	126	V	V	118	76	166	v	v
23	17	027	ETB (end of trans. block)	55	37	067	7	7	87	57	127	W	W	119	77	167	w	w
24	18	030	CAN (cancel)	56	38	070	8	8	88	58	130	X	X	120	78	170	x	x
25	19	031	EM (end of medium)	57	39	071	9	9	89	59	131	Y	Y	121	79	171	y	y
26	1A	032	SUB (substitute)	58	3A	072	:	:	90	5A	132	Z	Z	122	7A	172	z	z
27	1B	033	ESC (escape)	59	3B	073	;	;	91	5B	133	[	[	123	7B	173	{	{
28	1C	034	FS (file separator)	60	3C	074	<	<	92	5C	134	\	\	124	7C	174			|
29	1D	035	GS (group separator)	61	3D	075	=	=	93	5D	135	]	]	125	7D	175	}	}
30	1E	036	RS (record separator)	62	3E	076	>	>	94	5E	136	^	^	126	7E	176	~	~
31	1F	037	US (unit separator)	63	3F	077	?	?	95	5F	137	_	_	127	7F	177	&#127;	DEL

Table 2: Standard ASCII character set (Standard across the operating environments). Characters corresponding to decimal values range from 0 and 127.

128.	€	136.	ˆ	144.		152.	˜	160.		168.	¨	176.	°	184.	¸			
129.		137.	‰	145.	‘	153.	™	161.	¡	169.	©	177.	±	185.	¹			
130.	‚	138.	Š	146.	’	154.	š	162.	¢	170.	ª	178.	²	186.	º			
131.	*f*	139.	‹	147.	“	155.	›	163.	£	171.	«	179.	³	187.	»			
132.	„	140.	Œ	148.	”	156.	œ	164.	¤	172.	¬	180.	´	188.	¼			
133.	…	141.		149.	•	157.		165.	¥	173.	-	181.	µ	189.	½			
134.	†	142.	Ž	150.	–	158.	ž	166.	¦	174.	®	182.	¶	190.	¾			
135.	‡	143.		151.	—	159.	Ÿ	167.	§	175.	¯	183.	·	191.	¿			
192.	À	200.	È	204.	Ì	208.	Ð	210.	Ò	215.	×	217.	Ù	221.	Ý			
193.	Á	201.	É	205.	Í	209.	Ñ	211.	Ó	216.	Ø	218.	Ú	222.	Þ			
194.	Â	202.	Ê	206.	Î			212.	Ô			219.	Û	223.	ß			
195.	Ã	203.	Ë	207.	Ï			213.	Õ			220.	Ü					
196.	Ä							214.	Ö									
197.	Å																	
198.	Æ																	
199.	Ç																	
224.	à	232.	è	236.	ì	240.	ð	242.	ò	247.	÷	249.	ù	253.	ý			
225.	á	233.	é	237.	í	241.	ñ	243.	ó	248.	ø	250.	ú	254.	þ			
226.	â	234.	ê	238.	î			244.	ô			251.	û	255.	ÿ			
227.	ã	235.	ë	239.	ï			245.	õ			252.	ü					
228.	ä							246.	ö									
229.	å																	
230.	æ																	
231.	ç																	

Table 3: Extended ASCII character set. Characters corresponding to decimal values range from 128 and 255.