**ABSTRACT**
This paper describes two macros used in converting ASCII output files to SAS data sets. The first macro (READASCI) reads the ASCII output into SAS and creates a data set. The second macro (UNWRAPTX) takes information in one column that is displayed on multiple rows and 'unwraps' that information so that it is displayed on one row.

**MOTIVATION**
The application where these macros are used most often is to validate data listings. In this context, validate means to verify that the output correctly represents the values in the corresponding data sets. The data listing output is typically laid out as a series of columns that tend not to overlap each other. These data listings can get quite lengthy and become extremely tedious to validate by hand. By using the macros described in this paper, an electronic version of the output can be created which in turn can be used to validate the data listings electronically (Guttadauro, 2001).

**SPECIFIC EXAMPLE**
The following is an example of a data listing that will be used throughout this paper.

<table>
<thead>
<tr>
<th>PT ID</th>
<th>TRT</th>
<th>SEX</th>
<th>(a) RACE</th>
<th>OUTCOME</th>
</tr>
</thead>
<tbody>
<tr>
<td>101</td>
<td>A</td>
<td>F</td>
<td>AFRICAN AMERICAN</td>
<td>LOST TO FOLLOW-UP</td>
</tr>
<tr>
<td>102</td>
<td>A</td>
<td>M</td>
<td>HISPANIC</td>
<td>WITHDRAWN</td>
</tr>
<tr>
<td>103</td>
<td>B</td>
<td>M</td>
<td>CAUCASIAN</td>
<td>LOST TO FOLLOW-UP</td>
</tr>
<tr>
<td>104</td>
<td>A</td>
<td>F</td>
<td>CAUCASIAN</td>
<td>COMPLETED STUDY</td>
</tr>
<tr>
<td>105</td>
<td>B</td>
<td>F</td>
<td>HISPANIC</td>
<td>COMPLETED STUDY</td>
</tr>
<tr>
<td>106</td>
<td>A</td>
<td>F</td>
<td>CAUCASIAN</td>
<td>WITHDRAWN</td>
</tr>
<tr>
<td>107</td>
<td>B</td>
<td>M</td>
<td>AFRICAN AMERICAN</td>
<td>WITHDRAWN</td>
</tr>
</tbody>
</table>

(a) Age in years at time of screening.

**THE READASCI MACRO**
The READASCI macro is designed to read in ASCII files that use a column format. The user must specify the input ASCII file, the number of columns, the position of each column and the line size. The macro uses this information to create a SAS data set with the columns as variables. The macro contains additional code that will send messages to the log notifying the user if any of this information is missing. The macro also creates some additional variables that can be helpful in manipulating the newly created data set. The code for the READASCI macro can be found in Appendix 1.

**APPLY READASCI MACRO TO THE EXAMPLE**
The call for the READASCI macro for the example data listing specifies that the number of columns is 8. This is two more than the number of columns that appear in the output. The two additional columns will be needed to identify rows with wrapped text when the UNWRAPTX macro is used.

```
** Read in the original output **;
%READASCI (INFILE=LOCATION\L_DEMO.LIS,
    NUMCOLS=8,
    C_WIDTHS=%str(1-5^6-10^11-15^16-20^21-22^23-32^33-34^35-43),
    LINESIZE=43,
    OUTDATA=ORIGINAL);
```

Once the ASCII output has been read into SAS, the ORIGINAL data set can be modified to remove any unnecessary information. For the data listing example, we are only interested in keeping the body of the data listing and therefore the header and footer information can be removed. Since the columns for race and study outcome wrap text, we need to retain the patient id so that each row of the output will be identified with a specific patient.

```
** Modify the ORIGINAL data set **;
data ORIGINAL (drop=PAGE LINECNT LINE);
set ORIGINAL;
** Remove unneeded information **;
if LINECNT<12 or LINE=''
or
index(COL1,'____')>0 or index(COL1,'(a)')>0
then delete;
** Retain pt id and count the lines per pt **;
retain PT;
if COL1 ne '' then PT=COL1;
run;
```

The data set produced as a result of the READASCI macro and the modification code is displayed below. Notice that the variables COL5 and COL7 are only two characters wide. These variables will be used in the UNWRAPTX macro to identify which rows contain wrapped text.
PREPARE DATA SET FOR UNWRAPTX MACRO

Once the ORIGINAL data set has been created, there is some additional manipulation that must be done before the UNWRAPTX macro can be used. Each column that contains text to be unwrapped must be isolated in its own data set. In addition, all the columns containing text that does not need to be unwrapped, also need to be isolated. When isolating these columns, be sure to include any columns needed for unique identification. For the data listing example, the PT variable is the only variable needed for unique identification. The following code was used to create three data sets, one for the race column, one for the study outcome column and one for the rest of the columns.

** Create data sets for columns to unwrap **;
** and one for the rest of data. **;
data RACE(keep=PT COL5 COL6)
  OUTCOME(keep=PT COL7 COL8)
  REST(keep=PT COL2 COL3 COL4);
  set ORIGINAL;
run;

** Keep only one line per patient for **;
** data not needing to be unwrapped. **;
proc sort data=REST nodupkey;
  by PT;
run;

THE UNWRAPTX MACRO

The UNWRAPTX macro is designed to unwrap text that has been displayed as one column in multiple rows. In order to use the UNWRAPTX macro, the user must prepare the data sets by isolating the column variables that require unwrapping. The macro will be used to unwrap one column of information for each call of the macro. Once the data sets are ready, the user needs to specify the input data set, the flag variable, the variable containing the rest of the text and the unique identification variable names. The macro uses this information to create a SAS data set with one variable containing all the text for the column. The flag variable is used to identify rows with additional information that must be included in the unwrapping process. In the data listing example, COL5 is the flag variable for race and COL6 contains the rest of the text for race. If COL5 is missing, this indicates a row containing wrapped text which will need to be unwrapped. The code for the UNWRAPTX macro can be found in Appendix 2.

APPLY UNWRAPTX MACRO TO THE EXAMPLE

For this example, the columns for race and study outcome need to be unwrapped. The PT variable is the only one needed for unique identification.

** Unwrap the race and study outcome columns **;
%UNWRAPTX(INDATA=RACE, FLAGVAR=COL5, VAR=COL6, UNIQ_ID=PT);
%UNWRAPTX(INDATA=OUTCOME, FLAGVAR=COL7, VAR=COL8, UNIQ_ID=PT);

** Merge the data sets back together **;
data NEW_ORIG;
  merge RACE OUTCOME REST;
  by PT;
runcn

The data set produced as a result of merging the data sets created by the UNWRAPTX macro and the REST data set is displayed below.

CONCLUSION

The concepts used in the macros presented in this paper are not very complicated. However, if used in the appropriate situation they can result in enormous resource savings.

REFERENCES


ACKNOWLEDGMENTS

Thanks to Mike White for his assistance in fine-tuning the READASCI macro.

CONTACT INFORMATION

Lara E.H. Guttadauro
Kendle International Inc.
1200 Carew Tower 
441 Vine Street 
Cincinnati, OH  45202 
Work Phone: (513) 763-1363  
Email: guttadauro.lara@kendle.com
APPENDIX 1: READASCI MACRO

/*===============================================================
| PROGRAM NAME: READASCI.SAS |
| PROGRAMMER: Lara Guttadauro |
| PURPOSE: Macro used to read in ASCII output files with any |
| number of columns and create a data set. |
--------------------------------------------------------------------*/

%macro READASCI (INFILE=, /* the ASCII file to be read in */
  NUMCOLS=, /* the number of columns in INFILE */
  C_WIDTHS=, /* the position of the columns separated by ^ */
  LINESIZE=, /* the linesize of INFILE */
  OUTDATA=); /* the name of the data set to be created */

** Send messages to the log for problems **;
%if &INFILE= %then %put WARNING! NO INPUT FILE WAS SPECIFIED!;
%if &NUMCOLS= %then %put WARNING! NUMBER OF COLUMNS WAS NOT SPECIFIED!;
%if &LINESIZE= %then %put WARNING! LINE SIZE WAS NOT SPECIFIED!;
%if &OUTDATA= %then %put WARNING! NO OUTPUT DATA SET WAS SPECIFIED!;

** Count the number of column widths specified and send messages to the log **;
%local ONESP LEN STD;
%let STD =%str(STR0123456789-);
%if &C_WIDTHS= %then %do;
  %let CHK_NUM=0 ;
  %put WARNING: NO COLUMN WIDTHS WERE SPECIFIED!;
%end;
%else %do;
  %let ONESP = % trim(%left(%qsysfunc(compress(&C_WIDTHS))));
  %let LEN=%qsysfunc(compress(&ONESP,&STD));
  %if &LEN = &ONESP %then %let CHK_NUM=0 ;
  %else %let CHK_NUM=%eval(%length(&LEN)+1) ;
%end;
%if &NUMCOLS ne &CHK_NUM %then %do;
  %put WARNING! THE # OF COLUMNS AND # OF COLUMN WIDTHS SPECIFIED ARE NOT THE SAME!;
%end;

** Read in the columns of data in the listing and create a data set. **;
data &OUTDATA;
  infile "&INFILE" missover lrecl=&LINESIZE pad;
  input
      %do i=1 %to &NUMCOLS;
        %let WIDTH=%scan(&C_WIDTHS,&i,^);
        COL&i $ &WIDTH
      %end;
      $1 LINE $CHAR&LINESIZE..;
  @1 run;

** Determine the page and line counts **;
data &OUTDATA (drop=PAGEBRK);
  set &OUTDATA;
  PAGEBRK=substr(LINE,1,1);
  retain PAGE 0 LINECNT 0;
  if PAGEBRK='OC'x then do;
    PAGE=PAGE+1;
    LINECNT=0;
  end;
  LINECNT=LINECNT+1;
  run;
%mend READASCI;
APPENDIX 2: UNWRAPTX MACRO

/*---------------------------------------------
| PROGRAM NAME: UNWRAPTX.SAS |
| PROGRAMMER: Pingping Zhang |
| PURPOSE: Macro used to unwrap text. |
| ---------------------------------------------*/

** Before using this macro you need to **;
** create a sub-data set from the ORIGINAL **;
** data set containing only flagvar, var, **;
** and unique id variables. **;

%macro UNWRAPTX
(INDATA=, /* the sub-data set containing the 
variable to be unwrapped */
  FLAGVAR=, /* variable that can be used to 
identify if text has been 
wrapped */
  VAR=, /* variable that needs to be 
unwrapped */
  UNIQ_ID= /* unique id variables that will 
be used for merging 
e.g. UNIQ_ID=PT */
);

** Delete the rows with no information, **;
** count the number of rows used for each **;
** observation containing wrapped text. **;

data ONE;
  set &INDATA;
  retain COUNT;
  if &VAR='' & &FLAGVAR='' then delete;
  if &FLAGVAR='' then COUNT=0;
  COUNT=COUNT+1;
run;

proc sort data=ONE out=TEMP;
  by COUNT;
run;

** Create macro variable TOT, which is **;
** the maximum number of rows used for **;
** each observation containing wrapped **;
** text. This will determine how many **;
** do-loops will be needed. **;

data _null_; 
  set TEMP; 
  call SYMPUT ('TOT', COUNT); 
run;

** Create a series of data sets. Each one **;
** contains only the ith row of wrapped **;
** text. Later, these data sets will be **;
** merged and the variables will be **;
** concatenated into one variable. **;
%do i=1 %to &TOT;
  ** Get the ith row of wrapped text **;
  data TWO&i;
    set ONE;
    if COUNT=&i;
    rename &VAR=&VAR._&i;
  run;

  proc sort data=TWO&i;
    by &UNIQ_ID;
  run;
%end;
%end;

** Rename data set to INDATA value. **;
data &INDATA;
  set THREE;
run;
%mend UNWRAPTX;