

Tired of Typing? Got a layout definition with no column headers? Have WE got a fix for YOU!

SESUG Author, Michele Laughlin, Catalyst Technologies, Snellville, Georgia
 SESUG Co-Author, Thom Clay, Catalyst Technologies, Snellville, Georgia
 SESUG Co-Author, Jamila Ervin, Catalyst Technologies, Snellville, Georgia

ABSTRACT

In SAS processing, programmers need a way to transform raw data into a SAS dataset. This is generally achieved via proc import or by an input data step. However, when column headings are missing from the input file, the former is not a good option and the latter requires redundant coding for the various pieces of a data step.

INTRODUCTION 'Create_Input_Template'

This utility is a simple program that generates code for the inputting of either positional or delimited file types. Even non-programmers can generate filename, length, input, label, informat, format, infile, comments and output datasets with simple data entry. Time can then be better spent enhancing the program by eliminating redundant coding that is required for basic input logic. The resulting template will conform to company coding standards every time. This paper will demonstrate a simple process of creating basic code for various file types.

First complete the entries on the input spreadsheet and save it as a comma delimited file, as shown in Figure 1. The input spreadsheet should contain columns for an input filename, file type, delimiter (if applicable), logical record length, field beginning position (if applicable), field label, SAS variable name, variable type (character, numeric, integer), the length of the field and any formatting that needs to be done to any of the variables (i.e. date fields).

A	B	C	D	E	F	G	H	I	J	K
Input File Name	File Type (d=delimited p=positional)	Delimiter (use "tab" if file is tab delimited)	Logical Record Length	Field Beginning Position	Field Label	SAS Variable Name	Variable Type (c=character n=numeric)	Field length (this will be used in length statement)	SAS Informat	SAS format
e:\catalyst\ABC_healt h_plan\infile.txt	d	tab	1000	1	Sequence Number	seqno	c	2		
				3	Observation Value Code	valtype	c	2		
				5	Filler Order Number	accnum	c	12	mmddyy10.	mmddyy10.
				17	Observation ID from OBR	obrid	c	6		
				23	Observation ID	obxid	c	6		
				29	Observation Text	obxtxt	c	2		
				31	LOINC Coding System	loinc	c	15		
				46	Alternative ID	altid	c	6		
				52	Alternative Text Name	alttext	c	20		
				72	Alternative Coding	altcode	c	3		
				75	Test Result	labval	c	5		
				80	Low Reference Range	low	c	5		
				85	High Reference Range	high	c	5		
				90	Normalcy in Test	abnormfl	c	2		
				92	Status	status	c	2		
				94	LabCorp ID	labid	c	5		
				99	LabCorp Coding	labcode	c	5		
				104	NTE Flag	nteflag	c	5		
				109	Update Time	xupdate	c	15		

Figure 1. Sample Excel spreadsheet

```

data layout;
  length hindsn      $100
         hfiletyp    $1
         hdel         $10
         hlrecl       $5
         indsn        $100
         filetype     $1
         delimitr     $10
         lrecl        $5
         pos          $4
         label        $40
         sasname      $8
         vartype      $1
         length       $4
         informat     $15
         format       $15
  ;

infile "&dir.\&layout" delimiter=', ' dsd lrecl=500 missover pad;

input  hindsn      $      /* physical location and name of input file      */
      hfiletyp    $      /* file type - d=delimited p=positional */
      hdel        $      /* delimiter - use "tab" for tab delimiter */
      hlrecl      $      /* logical record length */
      pos         $      /* beginning position of field */
      label       $      /* field label - to use in label statement */
      sasname     $      /* sas variable name */
      vartype     $      /* variable type c=character n=numeric */
      length      $      /* length of field - to use in length/input stmts */
      informat    $      /* informat - to use in input/informat stmts */
      format      $      /* format - to use in format statement */
  ;

hindsn  = lowercase(hindsn);
hfiletyp = lowercase(hfiletyp);
hdel    = lowercase(hdel);
sasname = lowercase(sasname);
vartype = lowercase(vartype);
format  = lowercase(format);
informat = lowercase(informat);

if _n_ = 1 then delete;
else if _n_ = 2 then do;
  indsn      = hindsn;
  filetype   = hfiletyp;
  delimitr   = hdel;
  lrecl      = hlrecl;
  retain indsn filetype delimitr lrecl;
end;
drop hindsn hfiletyp hdel hlrecl;
run;

```

Using the above code, the Create Input Template utility will then read the data from the input file field and will create a filename statement, similar to the one below.

```
filename _in_ "e:\catalyst\ABC_health_plan\infile.txt";
```

A comment section will be added to the utility as shown below by using the following code. Once the program is created this section can be modified as desired.

```
data _null_;
  length date $7;
  set layout nobobs=nobs end=eof;
  file "&dir.\&convname";
  if _n_ = 1 then do;
    put @1 'filename _in_ "' indsn '"';
    ///@1 '*****'
    /@1 "member:      &convname"
    /@1 'directory: c:\programs\conversions'
    /@1 '*****'
    /@1 'This programs converts '
    /@1 '*****'
    /@1 '  Date      Analyst Walkthru Description'
    /@1 '-----'
    /@1 "&sysdate              New "
    /@1 '*****';
    /@1 'data _out_';
    /@1 '      length' @;
  end;

*****
member:      create_conversion_testpgm.sas
directory: c:\programs\conversions
*****
This programs converts
*****
Date      Analyst Walkthru Description
-----
30JUN06              New
*****;
```

The utility then creates the length statement using the variables that were entered on the Excel spreadsheet in the field length column. The statement also determines whether the variable is character or numeric based on the variable type field column from the Excel spreadsheet.

```
if vartype = 'c' then put @13 sasname $char8. ' $' length;
else                  put @13 sasname $char8. ' ' length;

if eof then do;
  put @13 ';;'
  /@6 'infile _in_ lrecl=' lrecl @;

data _out_;
  length seqno      $2
         valtype    $2
         accnum     $12
         obrid      $6
         obxid      $6
         obxtext    $2
         loinc      $15
         altid      $6
         alttext    $20
         altcode    $3
         labval     $5
         low        $5
         high       $5
         abnormfl   $2
         status     $2
         labid      $5
         labcode    $5
```

```

nteflag $5
xupdate $15
;

```

Next created are the infile and input statements. These use data captured from various columns on the Excel spreadsheet.

```

    if filetype = 'd' then do;
        if delimitr = 'tab' then put " delimiter='09'X dsd pad missover;";
        else put " delimiter='" delimitr +(-1) "' dsd pad missover;";
    end;
    else
        put ' ';
    call symput('noobs', left(put(nobs,4.)));
end;
run;
data _null_;
set layout end=eof;
length fmt $15;
array fvars (&noobs) $8 _temporary_;
array ivars (&noobs) $8 _temporary_;
array fmts (&noobs) $15 _temporary_;
array infmts (&noobs) $15 _temporary_;

file "&dir.\&convname" mod;
if _n_ = 1 then put @6 'input ' @;

if filetype = 'd' then do;
    if vartype = 'c' then put @12 sasname $char8. ' $ /* ' label $char40. '*/';
    else
        put @12 sasname $char8. ' /* ' label $char40. '*/';
end;
else do;
    if informat ~= ' ' then fmt = informat;
    else if vartype = 'c' then fmt = '$char'!!trim(length)!!'.';
    else
        fmt = trim(length)!!'.';
    put @12 '@' pos $char5. ' ' sasname $char8. ' ' fmt $char15. ' /* ' label
    $char40. '*/';
end;

if format ~= ' ' then do;
    fndx + 1;
    fvars{fndx} = sasname;
    fmts{fndx} = format;
end;

if informat ~= ' ' then do;
    indx + 1;
    ivars{indx} = sasname;
    infmts{indx} = informat;
end;

if eof then do;
    put @12 ' ';
    if fndx > 0 then do;
        put @6 'format ' @;
        do i = 1 to fndx;
            put @15 fvars{i} $char8. ' ' fmts{i};
        end;
        put @15 ' ';
    end;
    if indx > 0 then do;
        put @6 'informat ' @;
        do i = 1 to indx;
            put @15 ivars{i} $char8. ' ' infmts{i};
        end;
    end;
end;

```

```

        end;
        put @15 ' ';
    end;
end;
retain fndx indx 0;
run;

```

The logical record length is already assigned to the infile statement based on the length was specified in the Excel spreadsheet. The program also adds a 'dsd' 'pad' and 'missover' options to the infile statement. The input statement will have pre-filled comments with each.

```

infile _in_ lrecl=1000 delimiter=',' dsd pad missover;
input seqno $ /* Sequence Number */
      valtype $ /* Observation Value Code */
      accnum $ /* Filler Order Number */
      obrid $ /* Observation ID from OBR */
      obxid $ /* Observation ID */
      obxtext $ /* Observation Text */
      loinc $ /* LOINC Coding System */
      altid $ /* Alternative ID */
      alttext $ /* Alternative Text Name */
      altcode $ /* Alternative Coding */
      labval $ /* Test Result */
      low $ /* Low Reference Renge */
      high $ /* High Reference Range */
      abnormfl $ /* Normalcy in Test */
      status $ /* Status */
      labid $ /* LabCorp ID */
      labcode $ /* LabCorp Coding */
      nteflag $ /* NTE Flag */
      xupdate $ /* Update Time */
;

```

Finally the utility creates label statements using the field label column from the Excel spreadsheet.

```

data _null_;
set layout nobs=nobs end=eof;
file "&dir.\&convname" mod;
if _n_ = 1 then put @6 'label ' @;
put @12 sasname $char8. ' = "' label $char40. '"';

if eof then put @12 ' ';
              /@1 'run;';
run;

```

CONCLUSION

The output from the utility is a fully functional SAS program that can be used to input any delimited or positional file into a SAS dataset format. This program can then be modified by any SAS user to perform additional data processing as needed.

CONTACT INFORMATION

Thom Clay
2386 Clower Street
Suite C201
Snellville, GA 30078
770-982-8022 ext. 164
tclay@catalysttech.com
www.catalysttech.com

Jamila Ervin
2386 Clower Street
Suite C201
Snellville, GA 30078
770-982-8022 ext. 126
jervin@catalysttech.com
www.catalysttech.com

Michele Laughlin
2386 Clower Street
Suite C201
Snellville, GA 30078
770-982-8022 ext. 133
mclaughlin@catalysttech.com
www.catalysttech.com

SAS and all other SAS Institute Inc. product or service names are registered trademarks or trademarks of SAS Institute Inc., in the USA and other countries. ® Indicates USA registration.