Paper CC154

A macro to copy, rename, update and move your SAS® programs- all in one click

Julio Ruiz, Rho Inc., Chapel Hill, NC, USA

ABSTRACT

As SAS programmers we often have to copy, rename, update, and/or move SAS programs from one directory to another. Performing these tasks can be time-consuming, particularly when two or more of them need to be performed manually. This paper presents a macro developed in SAS that gives the end-user the ability to programmatically accomplish any of these tasks with one simple click. The macro's main goal is to offer the end-user the ability to save time, as it can perform any or all of these tasks in a matter of seconds.

INTRODUCTION

Most of us would agree that time is a high value commodity that we wish we had more of nowadays. As SAS programmers we are constantly trying to find new ways to be more efficient at what we do to save time. The *ExprtProgs* macro was written with this goal in mind. The macro provides the end-user the ability to save time by allowing him/her to programmatically copy, rename, update, and/or move SAS programs from one directory to another with one simple click. The macro's flexibility allows the end-user to execute any or all of these tasks simultaneously. Depending on the number of programs and tasks needed to be performed, the macro can significantly cut down the amount of time and effort needed to get the job done.

MACRO PARAMETERS

Parameter	Description
InPath	Fully qualified file path where programs will be copied from (REQUIRED).
OutPath	Fully qualified file path where output will be copied to (REQUIRED).
IncList	List of programs to be included (OPTIONAL). This parameter keeps only the programs specified. Each program name should be separated by at least one blank space. Using this parameter can be convenient when only a selected few programs are desired.
ExcList	List of programs to be excluded (OPTIONAL). It does not exclude anything if not specified. Each program name should be separated by at least one blank space. Using this parameter can be convenient when only a selected few programs are to be excluded.
Addnmext	Name extension (suffix) to be added to all program names, if specified (OPTIONAL).
CPnamelist	List of current program names (OPTIONAL). This parameter lists the program names from the input directory that will be renamed when exported to the output directory. Each program name should be separated by at least one blank space.
NPnamelist	List of new program names (REQUIRED- If CPnamelist is specified). This parameter lists the new program names the programs listed in CPnamelist will take when exported to the output directory. Each program name should be separated by at least one blank space.
Updtline1 – Updtline7	Each Updtline corresponds to a particular string that is to be updated in the program(s) (OPTIONAL). If invoked, each text string MUST be double-quoted AND separated by at least one blank space. For example: Updtline1 = "Sponsor X" "Sponsor Y";

Table 1. Macro Parameters

PARAMETER ERROR CHECKING

As the first step of its execution process, the macro performs parameter error checking and will end its execution and provide an informative log message if any of the following conditions are found:

- The macro is running on a SAS version earlier than 9.1.
- The input file path (InPath) parameter is missing and/or the specified input directory does not exist.
- The output file path (OutPath) parameter is missing and/or the specified output directory does not exist.
- If current program name list (CPnamelist) parameter is specified and new program name list (NPnamelist) is not or if the number program names from both lists are not the same.
- The text strings in any of the update lines (Updtline*) parameters are not specified correctly.

Additionally, the macro will continue its execution, but will provide an informative log message if one or more of the following conditions are found:

- Both inclusion (IncList) and exclusion (ExcList) list parameters are specified concurrently. Under this condition, the macro defaults to the list provided in the inclusion macro parameter.
- If any of the programs specified in the inclusion list is not found in the input directory.
- A suffix is added to the existing program name using the Addnmext macro parameter.

PROGRAM LIST CREATION

If the macro parameter error checking process does not end the execution of the macro, the macro proceeds to its second step, the creation of a program list. In this step the macro creates a data set with the names of all SAS programs available in the specified input directory, applies any subsets specified in IncList or ExcList, renames programs as specified in the CPnamelist and NPnamelist lists, adds any suffix specified in Addnmext, and creates macro variables with all the information needed for the following step.

CREATING A FILE LIST

The macro creates a data set with information about file path and program names for all files having the 'SAS' extension like this:

```
data filelist;
   length prog_fldr $200 prog_name itemname $50;
   prog_fldr = "&inpath";
   rc = filename('files',prog_fldr);
   dirid = dopen('files');

   do i=1 to dnum(dirid);
      prog_name = dread(dirid,i);
      itemname= upcase(scan(prog_name,1,'.'));
      if upcase(scan(prog_name,-1,'.')) = 'SAS' then output;
   end;
   rc=dclose(dirid);
   drop rc dirid i;
run;
```

INCLUSION/EXCLUSION FILTER

Once the data set with all available SAS programs from the input directory is created, the inclusion/exclusion subset specified in the IncList/ExcList parameter is applied, if specified. This procedure contains code that takes the unquoted inclusion/exclusion program name list and adds single quotes to every program name in the list so that it can be used in a WHERE statement.

```
proc sort data= filelist;
   by itemname;
%if %nrbquote(&IncList.)^= and %nrbquote(&ExcList.)= or %nrbquote(&ExcList.)= NONE
   %then %do;
      where itemname in(%unquote(%str(%')%qsysfunc(tranwrd(%sysfunc(compbl(
                        %upcase(&IncList.))),%str( ),%str(%' %')))%str(%')));
%end;
%else %if %nrbquote(&IncList.)= and %nrbquote(&ExcList.)^= %then %do;
      where itemname ^in(%unquote(%str(%')%qsysfunc(tranwrd(%sysfunc(compbl(
                         %upcase(&ExcList.))),%str( ),%str(%' %')))%str(%')));
%end;
run;
```

If the specified subset yields no results, the macro terminates the execution and provides an informative log message. If no subset is specified (i.e., both IncList and ExcList are blank), all eligible programs are used.

RENAMING PROGRAMS

If the end-user decides to rename the existing programs, the macro takes the specified program names from both CPnamelist and NPnamelist macro parameters and renames each program name from the CPnamelist list to the corresponding name in the NPnamelist list, in the specified order. If any of the program names specified in CPnamelist is not found in the input directory, the macro provides a log message indicating that the program does not exist.

ADDING NAME EXTENSION (SUFFIX)

The Addnmext macro parameter allows the end-user the ability to attach any suffix to the program names. While this option can be used to simply add a suffix to the existing programs, it can also be used when renaming programs concurrently (see Example 3).

PUTTING INFORMATION INTO MACRO VARIABLES

Once the program name list has been finalized, the macro loads the program name and file path information and number of programs into macro variables, which drive the do loop in the exporting/updating section.

```
proc sql noprint;
   select count(itemname), progdir, outprgdir, itemname, newnm
      into: Nprogs,
          : iprogdir separated by ',',
          : oprogdir separated by ',',
          : oprognms separated by ',',
          : nprognms separated by ','
   from proglist;
quit;
```

EXPORTING/UPDATING PROGRAMS

Most of the macro's functionalities is carried out by the code in this section. It is in this step that the macro is able to copy, rename, update, and/or move SAS programs from the directory specified in Inpath macro parameter to the directory specified in Outpath macro parameter. This section of the macro loops through the number of programs found in the previous step and performs the tasks at hand one program at a time.

To accomplish its objective, the macro uses the INFILE statement in combination with the FILE statement to copy, rename, and/or move the programs as needed. The automatic _INFILE_ variable, which references the contents of the current input buffer from the INFILE statement, is utilized to make the specified text string updates to the programs.

The text string updates from the update lines (i.e., Updtline1 - Updtline7) macro parameters are accomplished by replacing the specified text strings using the TRANWRD function.

```
%do j= 1 %to &N_Updtlns;
   %if &&Updtline&j ^= %then _infile_ = tranwrd(_infile_,&&Updtline&j);
%end;
```

In addition to the text strings specified in the Updtline macro parameters, the macro contains code that may be used to update the program name in the program header block.

```
/*-------------------------------------------------------------------------*
   ******************************************************
   *** Copyright <Company> 2015, all rights reserved   ***
   ******************************************************
   PROJECT:    Sponsor Project X
   PROGRAM:    AEXP_TAA
   PURPOSE:    Create table AEXP_TAA
   INPUT:      <Project Dir.>\<Sponsor>\<Project>\<Data>\ADAE.sas7bdat
               <Project Dir.>\<Sponsor>\<Project>\<Data>\ADSL.sas7bdat

   OUTPUT:     <Project Dir.>\<Sponsor>\<Project>\<Output>\AEXP_TAA.(SAS LOG LST)
   CALLS:      None

   PROGRAM HISTORY:
   DATE        PROGRAMMER        DESCRIPTION
   ---------   ---------------   ------------------------------------------------
   27SEP2015   Julio Ruiz        Create Table
*-------------------------------------------------------------------------*/

%let TBL = AEXP_TAA;
```

Output 1. Program Header Block Sample

As you can see from the header block sample, the program name is mentioned in four different places. The following

code searches for those lines where the program name would be expected within the automatic variable _INFILE_ and replaces the program name, if found. This section of code can be updated by the end-user to fit his/her specific needs:

```
%*Update program names in program header block, if needed;
%if "&oprognm" ^= "&nprognm" %then %do;
   if index(upcase(_infile_),'PROGRAM') or index(upcase(_infile_),'PURPOSE') or
      index(upcase(_infile_),'%LET PROG') or index(upcase(_infile_),'%LET TBL') or
      index(upcase(_infile_),'OUTPUT') then do;
           %*If program name text in upper case;
           _infile_ = tranwrd(_infile_,"&oprognm","&nprognm");
    %*If program name text in lower case;
           _infile_ = tranwrd(_infile_,%lowcase("&oprognm"),%lowcase("&nprognm"));
   end;
%end;
```

JOB CLEANUP

As its last step, the macro deletes any temporary data sets that get created during the macro execution by using the DELETE statement in PROC DATASETS.

MACRO CALL EXAMPLES

For purpose of the following examples, let's assume that you are working with an existing project (Project X) and need to create a new project (Project Y) for the same sponsor (Sponsor XYZ) and are dealing with ten programs (programs TAA-TAJ).

Example 1. You want to create/move a copy of all programs from Project X to the Project Y directory, while updating the name of the project, date of creation, and programmer name. You also want to rename the last five programs from Project X from TAF-TAJ to TBF-TBJ in Project Y, while keeping the original name on the first five. This is how the macro call would look like:

```
%ExprtProgs( Inpath     =  Project Dir.\SponsorXYZ\ProjectX,
             Outpath    =  Project Dir.\SponsorXYZ\ProjectY,
             CPnamelist =  TAF TAG TAH TAI TAJ,
             NPnamelist =  TBF TBG TBH TBI TBJ,
             Updtline1  =  "Project X" "Project Y",
             Updtline2  =  "04JUL2014" "04JUL2015",
             Updtline3  =  "John Doe" "Bob Smith"
            );
```

Example 2. You want to create/move a copy of the last five programs from Project X (TAF-TAJ) to the Project Y directory, while renaming the programs and updating the project name, date of creation, and programmer name as in Example 1. This is how the macro call would look like:

```
%ExprtProgs( Inpath     =  Project Dir.\SponsorXYZ\ProjectX,
             Outpath    =  Project Dir.\SponsorXYZ\ProjectY,
             IncList    =  TAF TAG TAH TAI TAJ,
             CPnamelist =  TAF TAG TAH TAI TAJ,
             NPnamelist =  TBF TBG TBH TBI TBJ,
             Updtline1  =  "Project X" "Project Y",
             Updtline2  =  "04JUL2014" "04JUL2015",
             Updtline3  =  "John Doe" "Bob Smith"
            );
```

Example 3. You want to create/move a copy of everything, except the last five programs from Project X to the Project Y directory, while adding "_NEW" as a suffix and updating the project name and date of creation. This is how the macro call would look like:

```
%ExprtProgs( Inpath     =  Project Dir.\SponsorXYZ\ProjectX,
             Outpath    =  Project Dir.\SponsorXYZ\ProjectY,
             ExcList    =  TAF TAG TAH TAI TAJ,
             Addnmext   =  _NEW,
             Updtline1  =  "Project X" "Project Y",
             Updtline2  =  "04JUL2014" "04JUL2015"
            );
```

Example 4. You want to update text strings without creating/moving programs (this example assumes that the programs already exist in the Project Y directory). This is how the macro call would look like:

```
%ExprtProgs( Inpath     =  Project Dir.\SponsorXYZ\ProjectY,
             Outpath    =  Project Dir.\SponsorXYZ\ProjectY,
             Updtline1  =  "Old Text 1" "New Text 1",
             Updtline2  =  "Old Text 2" "New Text 2",
             Updtline3  =  "Old Text 3" "New Text 3",
             Updtline4  =  "Old Text 4" "New Text 4",
             Updtline5  =  "Old Text 5" "New Text 5",
             Updtline6  =  "Old Text 6" "New Text 6",
             Updtline7  =  "Old Text 7" "New Text 7"
            );
```

CONCLUSION

The macro presented in this paper provides an easy and time-efficient way to copy, rename, update, and/or move SAS programs from one directory to another- tasks that SAS programmers often have to do as part of their everyday work. The macro was written under the assumption that the end-user would not want to update more than seven lines of text strings simultaneously. However, if needing to update more than seven lines of text strings, the end-user can either run the macro a second time (as shown in Example 4) using the different text strings or update the macro by adding new “Updtline” macro parameters and updating number of iterations in the last section of the parameter error checking process.

REFERENCES

SAS Institute Inc., SAS® 9.2 Language Reference, Version 9.2, Cary, NC: SAS Institute Inc., 2011

CONTACT INFORMATION

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

Name: Julio Ruiz
Enterprise: Rho, Inc.
Address: 6330 Quadrangle Drive Suite 500
City, State ZIP: Chapel Hill, NC 27517
Work Phone: 919-599-6389
Fax: 919-408-0999
E-mail: Julio_Ruiz@RhoWorld.com
Web: www.rhoworld.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.

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

APPENDIX- EXPRTPROGS MACRO CODE

```
%MACRO ExprtProgs(
           Inpath     =  ,
           Outpath    =  ,
           IncList    =  ,
           ExcList    =  ,
           Addnmext   =  ,
           CPnamelist =  ,
           NPnamelist =  ,
           Updtline1  =  ,
           Updtline2  =  ,
           Updtline3  =  ,
           Updtline4  =  ,
           Updtline5  =  ,
           Updtline6  =  ,
           Updtline7  =
                 );

   %put;
   %put ' => *==================================';
   %put ' => *   ExprtProgs MACRO IS EXECUTING   ';
   %put ' => *==================================';
   %put;

   %*****************************************;
   %* PARAMETER ERROR & SAS VERSION CHECKING  *;
   %*****************************************;

   %put;
   %put ' => *================================================';
   %put ' => *   PARAMETER ERROR CHECKING & GENERAL MESSAGES     ';
   %put ' => *================================================';
   %put;

    %* CHECK IF SAS VERSION IS 9.1 OR HIGHER;
    %if &sysver < 9.1 %then %do;
       %put ExprtProgs: -> MACRO MUST BE RUN USING SAS VERSION 9.1 OR HIGHER,
Execution terminating.;
      %goto exit;
   %end;

   %* CHECK IF INPUT PATH EXIST;
   %if %nrbquote(&InPath.) = %then %do;
      %put ExprtProgs: -> INPUT FILE PATH PARAMETER IS MISSING, Execution
terminating.;
      %goto exit;
   %end;
   %else %if %nrbquote(&InPath.) ^= %then %do;
      %let rc = %sysfunc(filename(fileref,&inpath)) ;
      %if %sysfunc(fexist(&fileref))= 0 %then %do;
         %put ExprtProgs: -> INPUT FILE PATH DOES NOT EXIST, Execution
terminating.;
         %goto exit;
      %end;
   %end;

   %* CHECK IF OUTPUT PATH EXIST ;
   %if %nrbquote(&Outpath.) = %then %do;
      %put ExprtProgs: -> OUTPUT FILE PATH PARAMETER IS MISSING, Execution
terminating.;
      %goto exit;
   %end;
   %else %if %nrbquote(&Outpath.) ^= %then %do;
      %let rc = %sysfunc(filename(fileref,&Outpath)) ;
```

```
      %if %sysfunc(fexist(&fileref))=0 %then %do;
         %put ExprtProgs: -> OUTPUT FILE PATH DOES NOT EXIST, Execution
terminating.;
         %goto exit;
      %end;
   %end;

   %* CHECK IF &IncList AND/OR &ExcList ARE SPECIFIED & CHECK FOR PROGRAM
EXISTANCE. DEFAULT TO &IncList WHEN BOTH
      PARAMETERS ARE SPECIFIED;
   %if %nrbquote(&IncList.)^= and %nrbquote(&ExcList.)^= %then %do;
      %put ExprtProgs: -> BOTH [IncList] AND [ExcList] WERE SPECIFIED. ONLY ONE CAN
BE USED AT A TIME.;
      %put %str(              ) [IncList] WILL BE USED INSTEAD.;
      %put;
      %let ExcList= NONE;
   %end;
   %if %nrbquote(&IncList.)^= %then %do;
      %* count number of program names in inclusion list;
      %let lcount=0;
      %do %while(%qscan(&IncList,&lcount+1,%str( )) ne %str());
         %let lcount = %eval(&lcount+1);
      %end;

      %do i= 1 %to &lcount;
         %let prognmck  = %upcase(%scan(%bquote(&IncList),&i,%str( )));
         %** Check for existance of program in input directory;
         %let rc = %sysfunc(filename(fileref,&inpath\&prognmck..sas)) ;
         %if %sysfunc(fexist(&fileref))= 0 %then %do;
            %put ExprtProgs: -> PROGRAM [&prognmck..sas] DOES NOT EXIST IN INPUT
DIRECTORY.;
         %end;
      %end;
   %end;

   %* CHECK IF &CPnamelist AND &NPnamelist ARE SPECIFIED & HAVE THE SAME NUMBER OF
PROGRAM NAMES;
   %if %nrbquote(&CPnamelist.)= and %nrbquote(&NPnamelist.)^= or
%nrbquote(&CPnamelist.)^= and
    %nrbquote(&NPnamelist.)= %then %do;
          %put ExprtProgs: -> BOTH [CPnamelist] AND [NPnamelist] NEED TO BE
SPECIFIED, Execution terminating.;
          %goto exit;
   %end;
   %else %if %nrbquote(&CPnamelist.)^= and %nrbquote(&NPnamelist.)^= %then %do;
      %if %sysfunc(countw(&CPnamelist.)) ^= %sysfunc(countw(&NPnamelist.)) %then
%do;
         %put ExprtProgs: -> PARAMETERS [CPnamelist] AND [NPnamelist] DO NOT HAVE
SAME NUMBER OF PROGRAMS.;
         %put %str(              ) EACH PROGRAM NAME IN [CPnamelist] SHOULD HAVE A
CORRESPONDING PROGRAM;
         %put %str(              ) NAME IN [NPnamelist], Execution terminating.;
         %goto exit;
      %end;
   %end;

   %* CHECK IF NAME EXTENSION WAS SPECIFIED & DISPLAY LOG MESSAGE;
   %if %nrbquote(&Addnmext.)^= %then %do;
      %put ExprtProgs: -> EXTENSION [ &Addnmext. ] WILL BE ADDED TO PROGRAM NAMES.;
      %put;
   %end;

   %* CHECK IF TEXT STRINGS FOR LINES TO BE UPDATES WERE CORRECTLY SPECIFIED &
PREPARE TEXT STRINGS FOR _INFILE_ USE;
   %let N_Updtlns= 0;
```

```
   %do i= 1 %to 7; ***Update if need to increase number of update lines in the
macro***;
      %if &&updtline&i^= %then %do;
         %let N_Updtlns= %eval(&N_Updtlns+1); %*count number of line updates;
         %let k= &N_Updtlns; %*Re-numerate update lines, if not used in sequence;

         %* Check if parameters has two text strings enclosed by double quotes.
End execution when not correctly specified;
         %if %sysfunc(countc(&&Updtline&i,'"'))^=4 %then %do;
            %put ExprtProgs: -> TEXT STRINGS FOR [Updtline&i] WERE NOT CORRECTLY
SPECIFIED. MAKE SURE LINE;
            %put %str(              ) CONTAINS TWO PIECES OF TEXT STRING (TEXT TO
BE UPDATED) AND EACH TEXT;
            %put %str(              ) STRING IS ENCLOSED BY DOUBLE QUOTES.
Execution terminating.;
            %goto exit;
         %end;
         %* Prepare text strings for _infile_ use;
         %let Updtline&k=%unquote(%qsysfunc(tranwrd(%sysfunc(compbl(&&Updtline&i)),
                        %str(" "),%str(%" , %"))));;
      %end;
   %end;

   %put;
   %put;
   %put ' => *========================================';
   %put ' => *   CREATING  LIST FOR PROGRAMS          ';
   %put ' => *========================================';
   %put;

   data filelist;
      length prog_fldr $200 prog_name itemname $50 ;
      prog_fldr = "&inpath";
      rc = filename('files',prog_fldr);
      dirid = dopen('files');

      do i=1 to dnum(dirid);
         prog_name = dread(dirid,i);
         itemname= upcase(scan(prog_name,1,'.'));
         if upcase(scan(prog_name,-1,'.')) = 'SAS' then output ;
      end;
      rc=dclose(dirid);
      drop rc dirid i;
   run;

   proc sort data= filelist;
      by itemname;
      %if %nrbquote(&IncList.)^= and %nrbquote(&ExcList.)= or %nrbquote(&ExcList.)=
NONE %then %do;
         where itemname
in(%unquote(%str(%')%qsysfunc(tranwrd(%sysfunc(compbl(%upcase(&IncList.))),
                              %str( ),%str(%' %')))%str(%')));
      %end;
      %else %if %nrbquote(&IncList.)= and %nrbquote(&ExcList.)^= %then %do;
         where itemname
^in(%unquote(%str(%')%qsysfunc(tranwrd(%sysfunc(compbl(%upcase(&ExcList.))),
                              %str( ),%str(%' %')))%str(%')));
      %end;
   run;

   proc sql noprint;
      select count(0) into :Nobs from filelist;
   quit;
```

```
    %if (&Nobs > 0) %then %do;
       options source nonotes;

       data renamed(where=(newnm ne ''));
          length oldnm newnm $50;
          oldlist= "&CPnamelist";
          newlist= "&NPnamelist";

          do until (oldnm= ' ');
             n_+1;
             oldnm= upcase(scan(oldlist,n_, ' '));
             newnm= upcase(scan(newlist,n_, ' '));
             output;
          end;

          drop oldlist newlist n_;
       run;

       proc sql noprint;
          create table renm_list as
             select *
          from work.filelist as a full join renamed as b
             on a.itemname = b.oldnm;
       quit;

       data _null_;
          set renm_list;
          if prog_name= ' ' then do;
             put 'ExprtProgs: -> PROGRAM NAME [ ' oldnm'] SPECIFIED IN CPnamelist
DOES NOT EXIST IN PROGRAM LIST.';
          end;
       run;

       data proglist (drop= oldnm prog_fldr);
          length progdir $150 ;
          set renm_list;
          if itemname ^= oldnm and oldnm= ' ' then newnm= itemname;

          progdir= prog_fldr;
       run;

       data proglist;
          set proglist;
          outprgdir= "&outpath"||subpad(progdir,(lengthn("&Inpath")+1),99);
          newnm= catt(of newnm, "&Addnmext.");
       run;

       proc sort data=proglist;
          by newnm;
       run;

       %**Put program info into macro vars ;
       options nosource nonotes;
       proc sql noprint;
          select count(itemname), progdir, outprgdir, itemname, newnm
             into: Nprogs,
                 : iprogdir separated by ',',
                 : oprogdir separated by ',',
                 : oprognms separated by ',',
                 : nprognms separated by ','
          from proglist;
       quit;

       options source notes;
       options noxwait noxsync xmin missing=' ';
```

```
      %do i=1 %to &Nprogs;
         %local Oprognm Nprognm indir outdir;
         %let Oprognm  = %upcase(%scan(%bquote(&oprognms),&i,%str(,)));
         %let Nprognm  = %upcase(%scan(%bquote(&nprognms),&i,%str(,)));
         %let indir    = %scan(%bquote(&iprogdir),&i,%str(,));
         %let outdir   = %scan(%bquote(&oprogdir),&i,%str(,));

            %put;
            %put *=> ================================================;
            %put *=>    EXPORTING/UPDATING: &nprognm                 ;
            %put *=> ================================================;
            %put;

            %** Copy and Modify Program Header Block, Setup info, etc.;
            data _null_;
               infile "&indir\&oprognm..sas";
               file   "&outdir\&nprognm..sas";
               input;

              %*Update program names in program header block, if needed;
               %if "&oprognm" ^= "&nprognm" %then %do;
                  if index(upcase(_infile_),'PROGRAM') or
index(upcase(_infile_),'PURPOSE') or
                     index(upcase(_infile_),'%LET PROG') or
index(upcase(_infile_),'%LET TBL') or
                     index(upcase(_infile_),'OUTPUT')  or
index(upcase(_infile_),'%SETUP_TLF') then do;
                       %*If program name text in upper case;
                       _infile_ = tranwrd(_infile_,"&oprognm","&nprognm");
                        %*If program name text in lower case;
                       _infile_ =
tranwrd(_infile_,%lowcase("&oprognm"),%lowcase("&nprognm"));
                  end;
               %end;

              %*Update specified text lines;
               %do j= 1 %to &N_Updtlns;
                  %if &&Updtline&j ^= %then _infile_ =
tranwrd(_infile_,&&Updtline&j);;
               %end;

               put _infile_;
            run;
      %end;
   %end;
   %else %do;
      %put ExprtProgs: -> NO OBSERVATIONS WERE FOUND FOR INPUT FILE TYPE.  THIS MAY
HAVE BEEN CAUSED BY;
      %put %str(             ) THE [IncList] SUBSET OR AN EMPTY DIRECTORY FOR THE
SPECIFIED FILE TYPE.;
      %goto exit;
   %end;

   %put;
   %put  *=======================================;
   %put  *   DELETING TEMPORARY DATASETS         ;
   %put  *=======================================;
   %put;

   proc datasets nodetails nolist;
    delete Ren: ;
   quit;

   %exit:
%MEND ExprtProgs;
```