

# You Want the Program to Run on the PC in New York, Unix in London, and That Nutty System in Manchester, without Modification??

David Franklin, Independent SAS® Consultant, Litchfield, NH

## INTRODUCTION

"That program you wrote for New York last week, can you make it run on the system in London? And by the way, Manchester were impressed with it and wondered if you could adapt it to go on their system -- remember they are still on 8.2."

It would have been easy to take the original version made for New York and adapt it for each site, ending up with three different versions. But what if we could just adapt the one version for use on all three sites?

## OUR INITIAL PROGRAM

```
** Bring in the data. The record for Colorado is
duplicated for instructional purposes.
Source: USGS 87-242, May 1990;
data River_US
  (label='Rivers in US over 1000 miles in Length');
  attrib name length=$40 label='River Name'
  distance length=8 label='Distance (miles)';
  input Name $ & distance @@;
cards;
Mississippi 2340 Columbia 1240 Colorado 1450
St. Lawrence 1900 Colorado 1450 Missouri 2540
Ohio 1310 Rio Grande 1900 Brazos 1280
Snake 1040 Atchafalaya 1420 Yukon 1980
Red 1290 Arkansas 1460
;
run;

libname derived "E:\TAUI\DERIVED";

** Remove Duplicates & put in dataset _DUPRECS;
proc sort data=River_US out=derived.River_US
  noduprec dupout=_duprecs;
  by descending distance;
run;

** Output Listings;
title "Duplicate Records Found and Removed in "
  "Load of RIVER_US Data";
proc print data=_DUPRECS label;
  var name distance;
  format distance comma8.;
run;

title "Longest Rivers in US Over 1,000 miles, "
  "By Distance";
proc print data=derived.River_US label;
  var name distance;
  format distance comma8.;
run;
```

*This program must run in AIX and Windows, and on 9.1.3 and 8.2, and in different locations!*



## KNOWING WHAT SAS KNOWS ABOUT YOUR OPERATING ENVIRONMENT

SAS already knows quite a bit about your operating environment during SAS initialization. Some of this information can be located in the SAS View VMACRO that can be found in the SAS Help library. These macro variables are automatically set up during SAS initialization and cannot be changed by the users. For the purposes of this paper the four macro variables will be looked at are:

```
SYSSCP
SYSSCPL
SYSSITE
SYSVER
```

*SAS knows where, what version and OS you are running*

The SYSSCP and SYSSCPL are linked and contain an identifier for your operating environment. It is useful to use these variables to execute applicable system commands. For our example, we are going to run this under Windows and Unix so the relevant system macro variables are:

Platform	SYSSCP Value	SYSSCPL Value
R64 or AIX64 or AIX on POWER	AIX 64	AIX
Windows XP Pro	WIN	XP_PRO

Refer to the documentation listed in the Reference section below for a more detailed list of SYSSCP and SYSSCPL values based on SAS version.

SYSSITE is the value that SAS has assigned to your site license. This number is displayed in the SAS Log. Every site license number is unique.

Finally, SYSVER. This is the SAS version of the SAS software that you are running on your site. With this information it is possible to use features that are available to a site that is running a later version that is not available in a previous version.

## ADAPTING OUR PREVIOUS CODE TO RUN ON DIFFERENT ENVIRONMENTS

Examining our request, and refining it after asking some questions, we know that New York (where our program was originally run on) is running SAS version 9.1.3 using Windows XP Pro with license number 9999US, London is also running SAS version 9.1.3 using Unix on an AIX box with license number 9999EU, and Manchester is running SAS version 8.2 using Windows XP Pro with the New York license.

The first task to do is identify what code is operating system or SAS version dependent. In the above example there are two sections.

First is the LIBNAME statement. This statement is dependent on operating system and on SAS site requirements -- directory structure is dependent on SAS site. In our example New York has the directory "E:\TAUI\DERIVED" while London has the structure "//CharingCross/Production/Projects/Tau/Derived". Manchester uses the same directory as New York.

With this in mind the code for the LIBNAME statement could be adapted to:

```
%if &syssite=9999US %then %do;
  libname derived "E:\TAUI\DERIVED";
%end;
%else %if &syssite=9999UK %then %do;
  libname derived
  "//CharingCross/Production/Projects/Tau/Derived";
%end;
%else %do;
  %put %str(ERR) OR: SYSSITE &syssite is not recognized.;
  %put Consult your site administrator for help.;
  %put Program will now terminate.;
endsas;
run;
%end;
```

*Dealing with different locations*

In the code above, if the SYSSITE value is not recognized then the program will put a message to the SAS LOG and terminate. This makes the code very site specific.

The second piece of code that will not work right is the DUPOUT option in the SORT procedure -- this was not available until SAS version 9.1 but Manchester is running SAS version 8.2. There SYSVER option is useful for this situation, as the following code will show:

```
** Remove Duplicates and put in dataset _DUPRECS;
%if (&sysver >= 9.1) %then %do;
  ** If SAS version 9.1 or higher;
  proc sort data=River_US out=derived.River_US
    noduprec dupout=_duprecs;
  by descending distance;
run;
%else %do;
  ** If SAS version before 9.1;
  proc sort data=River_US;
  by descending distance name;
run;
data derived.River_US _duprecs;
  by descending distance name;
  if first.name then output derived.River_US;
  else output _duprecs;
run;
%end;
```

*Dealing with different SAS versions*

To finish off the code we have to cater for the case where if the operating system was Unix based then any sorting of data was to use the host sort when the size of the dataset was greater than 50M. This can be done using the SYSSCPL global macro variable as the following code demonstrates:

```
%if &sysscp=AIX %then %do;
  options sortpgm=host sortcut=50M;
%end;
```

*Dealing with different OS's*

In the example we used the SYSSCPL value but we could have used the SYSSCP as well -- the SYSSCP value is a superset of the SYSSCPL value (refer to SAS documentation relating to your OS for respective values).

After all the adaptations to the code noted above, it is possible to now copy the program (code is in Appendix) onto the systems in New York, London and Manchester without adaptation.

## CONCLUSION

A brief introduction to four SAS automatic variables then resulted in the original program created for New York being able to be adapted for use on all three sites without having three separate programs being written. It is also possible to use these same techniques shown here when using two or more operating environments and/or SAS versions within the one site, or when upgrading from one SAS version to another.

## CONTACT INFORMATION

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

Name: David Franklin  
Enterprise: Independent SAS Consultant  
Address: 16 Roberts Road  
City, State ZIP: Litchfield, NH 03052  
Work Phone: 603-275-6809  
E-mail: [dfranklinuk@compuserve.com](mailto:dfranklinuk@compuserve.com)  
Web: <http://www.TheProgrammersCabin.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

# You Want the Program to Run on the PC in New York, Unix in London, and That Nutty System in Manchester, without Modification??

David Franklin, Independent SAS<sup>®</sup> Consultant, Litchfield, NH

## CODE AFTER ADAPTION FOR USE ON MULTIPLE ENVIRONMENTS

```
** Bring in the data. The record for Colorado is duplicated for
instructional purposes. Source: USGS 87-242, May 1990;
data River_US (label='Rivers in US over 1000 miles in Length');
  attrib name length=$40 label='River Name'
         distance length=8 label='Distance (miles)';
  length name $40 distance 8;
  input Name $ & distance @@;
cards;
Mississippi 2340 Columbia 1240 Colorado 1450
St. Lawrence 1900 Colorado 1450 Missouri 2540
Ohio 1310 Rio Grande 1900 Brazos 1280
Snake 1040 Atchafalaya 1420 Yukon 1980
Red 1290 Arkansas 1460
;
run;

%macro runlist;

  ** SET LIBNAME based on site - abort if not recognized;
  %if &syssite=9999US %then %do;
    libname derived "E:\TAUI\DERIVED";
  %end;
  %else %if &syssite=9999UK %then %do;
    libname derived "//CharingCross/Production/Projects/Tau/Derived";
  %end;
  %else %do;
    %put %str(ERR) OR: SYSSITE &syssite is not recognized.;
    %put Consult your site administrator for assistance.;
    %put Program will now terminate.;
  endsas;
  run;
%end;

  ** IF AIX set some additional options for SORT;
  %if &sysscpl=AIX %then %do;
    options sortpgm=host sortcut=50M;
  %end;

  ** Remove Duplicates and put in dataset _DUPRECS;
  %if (&sysver >= 9.1) %then %do;
    ** If SAS version 9.1 or higher;
    proc sort data=River_US out=derived.River_US
      noduprec dupout=_duprecs;
      by descending distance;
    run;
  %end;
  %else %do;
    ** If SAS version before 9.1;
    proc sort data=River_US;
      by descending distance name;
    run;
    data derived.River_US
      _duprecs;
      by descending distance name;
      if first.name then output derived.River_US;
      else output _duprecs;
    run;
  %end;

  ** Output Listings;
  title1 "Duplicate Records Found and Removed in Load of RIVER_US Data";
  proc print data=_DUPRECS label;
    var name distance;
    format distance comma8.;
  run;

  title1 "Longest Rivers in US Over 1,000 miles, By Distance";
  proc print data=derived.River_US label;
    var name distance;
    format distance comma8.;
  run;

%mend runlist;
%runlist;
run;
```