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 St-241, May 1990.

```
data River_US;
  label='Rivers in US over 1000 miles in Length';
  attrib name length distance short;
  input Name $ &length @@ &distance @@;
  format length comma8. distance comma8. ;
  format distance comma8. ;
run;
```

** Output Listings;

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

```
title "Load of River_US Data";
proc sort data=River_US; run;
```

```
libname derived "E:\TAUI\DERIVED";
```

```
%if (&sysver >= 9.1) %then %do;
  %if %str(ERR) OR: SYSSITE &syssite is not recognized, &program will now terminate.;
%end;
endnas;
run;
```

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:

```
%if (&sysver >= 9.1) %then %do;
  %if %str(ERR) OR: SYSSITE &syssite is not recognized, &program will now terminate.;
%end;
```

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 %str(ERR) OR: SYSSITE &syssite is not recognized, &program will now terminate.;
%end;
```

```
%put %str(ERR) OR: SYSSITE &syssite is not recognized, &program will now terminate.;
```

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

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: 603-275-8809
E-mail: dfranklin@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® Consultant, Litchfield, NH

CODE AFTER ADAPTATION FOR USE ON MULTIPLE ENVIRONMENTS

** Bring in the data. The record for Colorado is duplicated for instructional purposes. Source: USGS 87-242, May 1990;
 attrib name length=40 label='River Name';
 length name $40;
 length 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 1230 Arkansas 1460
 run;

%macro runlist;
 ** SET LIBNAME based on site – abort if not recognized;
 %if &syssite=999
 libname derived "E:\TAUI\DERIVED";
 %end;
 %else %if &syssite=9999UK %then %do;
 libname derived "\CharingCross\Production\Projects\Tauli\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.;
 %endas;
 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
 noduproc 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;