Introduction

We in the aquatic ecology team of the environmental division use a large SAS/AF catalog as our data management system. Each team member has a copy on their personal computer. We keep the data and a master catalog copy on a local area network (LAN) server. A big challenge for me as the application developer has been to motivate my coworkers to copy a new version of the master catalog off of the server whenever I make fixes or improvements. I sent out e-mail announcements about the changes with instructions on which source and target directories to use for copying. Unfortunately, it is all too seductive for users to procrastinate. Hence, they may encounter the same bugs as before, not take advantage of a new quality assurance, measure or see whatever new feature I have added.

This can be very frustrating, but why struggle with human nature when you can let the SAS® System do the work? To brief, I found a way to make the SAS/AF application check whether the master copy on the LAN is newer than the one on the user's hard disk. If so, the application quits, downloads the new version and restarts itself. For a speed boost, the application can detect that it does not need to check the modification status again. The objective of this paper is to describe how you can accomplish this with PROC SQL and some macros within Screen Control Language.

Step One-A macro to check whether this is the first invocation of the application

A master copy of the SAS/AF catalog resides on a LAN server drive, along with the environmental database. Each team member has a copy of the application catalog on their hard disk in the working directory for the SAS System. Upon starting SAS, the Autoexec.Sas file runs the SAS Display Manager command to invoke the AF application with Main.Program. The application is designed to check whether the modification date of the master catalog is newer than the date on the user's copy and, if so, then quit AF and copy the newer version. Then the application will restart. Since the users would be paying a time penalty by waiting for the check of modification dates, I wanted to make sure they did not pay it again on restart of the application.

The solution to this is to check for the existence of a global macro variable upon invocation. If it does not exist, then this must be the first invocation and the program should go ahead and check the modification dates. If the variable exists, then this must be the second (or greater) invocation, and the check should be skipped.

The macro Symchk (Fig. 1) was taken from page 254 of the SAS Guide to Macro Processing (SAS Institute, Inc. 1990). A clear explanation of the macro is given in the manual, so I will not repeat it here. Basically, the macro evaluates the question of whether the macro variable Moddate exists as NO when the AF application starts for the first time in a SAS session (Fig 1, line 5). Then step 2, described below, will execute. The variable Moddate will only exist if PROC SQL creates it. If the AF application is restarted, the macro expression will evaluate to YES...
Step Two- Using PROC SQL to obtain modification dates

The next objective was to find a way to detect the modification dates of the master and local copies of the application. I knew that it was possible to use SCL functions to obtain a lot of information from SAS data sets, but functions dealing with SAS catalogs are less comprehensive. The clue for using PROC SQL to obtain modification dates of data sets came from Stanley (1994, page 47). Help with the necessary syntax for dealing with catalogs came from the technical support people at the SAS Institute.

Firstly, a macro named Sqlchk is defined that enables us to conditionally execute the SQL step (Fig. 2). The macro evaluates, using macro Symchk (Fig. 2, line 2), whether the value of &chktest is NO. If so, then two macro variables are defined as global in line 3, so that they will be available to Screen Control Language (remember that they are being created in a submit block). Finally, PROC SQL is started.

In lines 5-7 of Figure 2, we see how PROC SQL obtains the maximum modification date from the master copy of the catalog Dat.Zoodb. The modification date is placed into one of the two global macro variables. The date from the master catalog is assigned to &Moddate. The same action occurs for the local catalog (U.Zoodb) found on the hard disk (Fig. 2, lines 9-11). Note that the MAX function is needed to obtain the latest modification date from all the entries in the catalog because there is no single modification date, as is the case for data sets. The final action for this section is to execute the macro Sqlchk (line 15) and end the submit block. Note that the parameter supplied to macro Sqlchk is Moddate, so we are testing in line 2 whether SQL has already run.

Step 3- Take action based upon the results of a date comparison

Now we are ready to use Screen Control Language to test whether the date of the master catalog is newer than the date of local catalog. First, in lines 1-3 of Figure 3, we get the value of the macro variable Yesno and ask if it equals NO. If it does, then this must be the first invocation of the application, and it is necessary to go ahead and compare the dates. If Yesno= YES, then this step is skipped and MAIN executes.

The modification dates that were assigned to macro variables are gotten with the SYMGETC function and assigned to SCL variables (Fig 3, lines 4-5). Now we test to see if the date of the master catalog is newer than the local copy. If so, another submit block is started. The master catalog is copied onto the local hard disk using PROC DATASETS. Also, a Display Manager command is submitted that will restart the application.

However, this submit block will not execute until the AF application is ended, so that no problems can arise by copying onto the active catalog, i.e. the one on the local hard disk. Notice that the submit block did not contain any options (fig. 3, line 8), so it runs after AF termination. That makes it possible to embed the restart command (Fig. 3, line 12). The last task is to force termination of the AF application by setting the status variable to halt (line 13). Now the submitted commands execute, the master catalog is copied, and the application restarts. Mission accomplished!

Todd C. Folsom
Environmental Division, (MGQ3A3)
Duke Power Company
13339 Hagers Ferry Rd.
Huntersville, NC 28078
704-875-5426
tcfolsom@dpcmail.dukepower.com

References


SAS and SAS/AF are registered trademarks or trademarks of SAS Institute, Inc. in the USA and other countries. © indicates USA registration.