

Paper 287-25**SAS Administration—Making Your Ship Go**

Paul D. McDonald, SPIKEware, Inc., Schaumburg, IL

ABSTRACT

This paper shows how to effectively manage your PC SAS Software System so that upgrades, add-ins, free software, and other such goodies are easily and rapidly distributed to everyone who uses SAS. PC SAS Users will get the most out of this paper, but everyone can learn something new.

INTRODUCTION

Managing multiple users of the SAS Software System on a PC Network can be a headache. Some sites have as many as 500 (and some have even many more!) installations of PC SAS. This paper shows you how to begin effectively managing an extensive system of PC SAS.

This system assumes the following:

- ◆ You are operating in a PC SAS environment
- ◆ Each SAS programmer has read-write access to one unique network drive
- ◆ SAS is installed locally on each user's PC

THE SECRET: AUTOEXEC.SAS

The AUTOEXEC.SAS file is a SAS program that runs when each new SAS session is started. Using this utility, you can then have the SAS supervisor check for and install any required updates and maintenance automatically.

THE LOCAL USER'S OPTIONS

A "DO NOT MAKE ANY CHANGES BELOW THIS LINE" comment should be included, and all administrative statements should be entered after that comment. This way, the user can use the AUTOEXEC.SAS without modifying the administrative capabilities.

CREATING THE SASUSER TAG

Each SAS User needs to have a unique identifier of some type. The easiest way to do this is to create a macro variable &SASUSER using a %LET statement in the AUTOEXEC.SAS.

```
%let sasuser = PMCDONALD ;
```

This username must be unique among all SAS users on the system.

SETTING UP A MACRO LIBRARY

I have found it useful to have all standard macros stored and compiled on the local drive. I have also found it extraordinarily helpful to create a MACRO library, where all site-standard macros write their datasets. This prevents any overwriting of work datasets of the same dataset name.

```
libname macro 'c:\sas\macros' ;
```

Write your site-standard macros to only create datasets in the MACRO library (unless the dataset is expected output of the macro).

RUNNING A NETEXEC.SAS

The next step is to have every users' AUTOEXEC.SAS program call a unique SAS program using the %INCLUDE statement (referred to as NETEXEC.SAS).

```
%include 'k:\sasadmin\netexec.sas' ;
```

In this example, we assume that the drive 'k' is a network drive.

The NETEXEC file is then functioning as a "common" AUTOEXEC.SAS program. The NETEXEC.SAS program can then be used to do the following:

- ◆ Set up the NETEXEC libraries for the NETEXEC process
- ◆ Verify the active status of the SAS User
- ◆ Check for the need for any system update
- ◆ Perform any needed system update
- ◆ Set up site-standard libraries and catalogs
- ◆ Release the NETEXEC libraries upon completion

SET UP THE NETEXEC LIBRARIES FOR THE NETEXEC PROCESS

The first step of the NETEXEC.SAS program is to set up any of the NETEXEC libraries required. A simple LIBNAME statement should suffice:

```
libname netexec 'k:\sasadmin\data' ;
```

The NETEXEC library should hold a SAS dataset named NETEXEC.SASUSER. This dataset should contain a list and description of every site-registered SASUSER. The following variables are recommended:

- ◆ SASUSER (an equivalent of the &SASUSER macro variable)
- ◆ STATUS (ACTIVE/INACTIVE flag for the SASUSER)
- ◆ UPDATE1-UPDATEX (one Y/N variable [or date if you prefer] for each of the required system update)
- ◆ LAST_DT (last access date)

VERIFY THE ACTIVE STATUS OF THE SAS USER, CHECK FOR THE NEED FOR A SYSTEM UPDATE, AND PERFORM THOSE SYSTEM UPDATES.

This process is simple: simply create a DATA _NULL_ that scans the NETEXEC.SASUSER table for the status of the user. The source code might look something like this:

```
data _null_ ;
  set netexec.sasuser nobs=nobs ;
  flag = '0' ;

  if sasuser = "&sasuser" then do ;
    flag = '1' ;

    if update1 = 'N' then do ;
      call execute ("%include
'k:\sasadmin\updates\update1.sas' ;");
      update1 = 'Y' ;
    end ;

    if update2 = 'N' then do ;
      call execute ("%include
'k:\sasadmin\updates\update2.sas' ;");
      update2 = 'Y' ;
    end ;

    stop ;
  end ;
  if _n_ = nobs then do ;
    put 'ERROR: You are not a registered
SAS User at your site. Please see the
administrator.' ;
  end ;
```

```
run ;
```

Here are the critical parts of this data step:

1. The variable FLAG is set to zero, so that if the entire dataset NETEXEC.SASUSER is processed without finding a user, a non-registered user routine can be activated. This can be anything from a NOTE or WARNING message in the log to shutting down SAS using ABORT ABEND.
2. Each time a new UPDATE is added to the system, a corresponding IF statement must be added to the administrative data step. The CALL EXECUTE call routines are used to INCLUDE appropriate files that will then process the necessary updates.

EXAMPLE: Here is an example of an INCLUDE file to process the update for the update for SAS/ACCESS for MICROSOFT EXEL97. This particular update came as a nice executable file, called SASEXEC97.EXE. A simple X command is used to run the update. The %INCLUDE file would contain the following source code:

```
options noxwait xsync ;
x 'k:\sasadmin\updates\saseexec97.exe' ;
```

This particular setup program from the SAS Institute runs nicely—it's a nice little “follow-your-nose” setup routine. However, not all setup routines are that nice. For more complex routines, try writing some Windows HELP files to guide the user.

Other ideas include modifying the user’s local AUTOEXEC.SAS for any administrative changes, copying macro catalogs to the local drive, and updating a local drive SASNEWS file.

SET UP SITE-STANDARD LIBRARIES AND CATALOGS

If there are any standard LIBRARIES and CATALOGS for the site, this is the place to set them up with a LIBNAME statement.

RELEASE THE NETEXEC LIBRARIES UPON COMPLETION

Use a LIBNAME CLEAR statement to clear the NETEXEC library for the remainder of the SAS Session. This statement looks like this:

```
libname netexec clear ;
```

Including this statement ensures that SAS USERS will not inadvertently modify any administrative files and datasets.

MAINTENENCE REPORTING

Simple maintenance reporting can be completed daily, weekly, or as desired by the SAS System Administrator. These reports can be “who is on the system” and “who needs an update” for examples.

CONCLUSION

I have found PC SAS Site Administration (and mainframe administration, for that matter) to be widely ignored. Yet, it is a vital part of providing quality results in the data analysis field.

These ideas can make your site run more smoothly and make supporting a large or small number of SAS users much easier and therefore much more likely to be fulfilled. In turn, the end results and output created by the SAS site will be more robust, more reliable, and less restrained.

And that makes using SAS better for everyone!

CONTACT INFORMATION

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

Paul D. McDonald
SPIKEware, Inc.
1450 East American Lane Suite 1400
Schaumburg, IL 60173
Work Phone: (847) 330-4460
Fax: (847) 798-0732
Email: PDM@SPIKEware.com
Web: <http://www.SPIKEware.com/>