FILE LOCKING FACILITY FOR SAS/SHARE® SOFTWARE

BY Michael R. Gibson
The Gates Rubber Company

• ABSTRACT

Are you using SAS/SHARE and the "CONFLICTING LOCK" error message keeps bombing your batch jobs? Have you ever wanted exclusive control over datasets in your batch job? This paper will provide a method of locking datasets to gain that exclusive control when needed. The method uses a macro variable that is new to the SAS® System version 5.18, but can be modified to run on any version.

• INTRODUCTION

At Gates in the Advertising Dept. we have a system that requires the user to submit a batch job for processing. The user, however, is still able to enter SAS/FSEDIT® on dataset(s) used in the batch job. If the user happens to enter any dataset before the batch job can process it, the batch job gets the "CONFLICTING LOCK" message and bombs.

The macro described below is an attempt to lock all datasets used in the batch job so that NO user in interactive mode can access them. The secret behind this macro is the dataset READ password protection. Using the MODIFY statement of PROC DATASETS®, the macro changes the read password to LOCKED for datasets listed in the DSN parameter.

• LOCKOUT USAGE

There are five steps needed to use this method of locking files.

1. Enter the macro code into your AUTOCALL® library or compile it before using.

2. Search your batch job for a list of ALL datasets you would like to lock and the libref of those datasets. You will needed to call the macro for each libref.

3. Call the macro at the top of your batch code to lock desired datasets.
4. Add the dataset option READ=LOCKED to each reference to the locked datasets in the job. References will be those that read or write the datasets. Places to look are: DATAstep dataset names. SET, MERGE, and UPDATE statements. Any PROC statements that read or write.

5. Call the macro again at the bottom of the batch job using the UNLOCK=YES parameter to remove the READ password. This is required to allow normal interactive processing.

• LOCKOUT MACRO CODE

```
%MACRO LOCKOUT(LIB=,DSN=,UNLOCK=NO,PW=LOCKED);
  %* *---------------------------------------------------------------*;
  %* This macro will assign or remove a READ only password to SAS dataset(s). *
  %* The purpose of this password is to prevent access to datasets in a shared environment *
  %* via SAS/Share server, thereby allowing a batch JOB to run to completion. *
  %* WARNING: This macro was designed for BATCH mode and will ENDSAS if it cannot modify the password or find the dataset(s). *
  %* INPUT VARIABLES: *
  %* LIB = Library reference of datasets. *
  %* DSN = Dataset(s) to be locked i.e. multi DSN=DSN1 TEST DSN3 ... etc. *
  %* UNLOCK = Switch to tell macro to add READ only password or remove it. *
  %* PW = This is the READ only PASSWORD used for the modify statement. *
  %* SERVER FILE LOCKING MACRO FOR USE WITH VERSION 5.18 */

%let unlock = %upcase(&unlock);
%local dname i;
```
%let dname = %scan(&dsn,1); /* Initial priming value for do while;
%let i = 1;

PROC DATASETS LIBRARY=&lib MT=DATA NOLIST;

/* Create a MODIFY statement for every dataset listed in DSN parm;
%* scan function will return a blank when it reads past end of parm;
%do %while(&dname ne );

MODIFY &dname
  %if &unlock EQ YES %then (READ=&pw/) %str();
  %else (READ=&pw) %str();
  %let i = %eval(&i + 1);
  %let dname = %scan(&dsn,&i);
%end; /* end of dataset do loop;
RUN;

/* CHECK to see if MODIFY worked;
%* using NEW system MACRO var SYSERR;
%* available only in version 5.19;
%if &syserr NE 0 %then %do;
  %put %str( );
  %put %511("* ...
  %put %str( );
  %put %str(COULD NOT LOCK or UNLOCK specified datasets );
  %put %str( CHECK for CONFLICTING user on SERVER );
  %put %str( or INVALID LIB= or DSN= options. );
  %put %str(DATASET%(s%) Requested are: );
  %put &dsn;
  %put %str( );
  %put %str( ------> SAS ENDED DUE TO ERROR <------ );
  %put %str( );
  %put %str(* * * LOCKOUT ERROR * * * LOCKOUT ERROR * * * );
ENDSAS; /* END the SAS section */
%end;
%else %do;
  %put %str( );
  %put %str(DATASET%(s%)
  %put &dsn;
  %put %str( );
  %put %str( );
  %put %str(* * * LOCKOUT ERROR * * * LOCKOUT ERROR * * * );
ENDSAS; /* END the SAS section */
%end;

1582
%if &unlock EQ YES %then %do;
  %put %str(UNLOCK was completed successfully
  %put %str(  
  %put %str(*** UNLOCK SUCCESSFUL ***
  %end;
%else %do;
  %put %str(LOCK was completed successfully
  %put %str(  
  %put %str(*** LOCK SUCCESSFUL ***
  %end;
%MEND LOCKOUT;

* LOCKOUT APPLICATION

//SAS EXEC SAS

LIBNAME DEPT SERVER=MCR 'CIC.ADVERT.SASLIB';
/* *-----------------------------------------------+ */
/* * AUTOREL -- AUTOMOTIVE STOCK ORDERS          */
/* * RELEASE ORDERS TO PERMANENT FILE            */
/* * INSTALLED 2/15/88 BY PETE DILLINGHAM       */
/* * MOD: 9/26/88 BY MICHAEL GIBSON             */
/* * INSTALLED SAS/SHARE LOCKOUT MACRO          */
/* *-----------------------------------------------+ */

/* LOCK FILES IN DEPT LIBRARY */

%LOCKOUT(LIB=DEPT,DSN=AUTOHOLD AUTOSTCK AUTOSHIP AUTOALLO);

DATA WORK.AUTOREL
  DEPT.AUTOHOLD(READ=LOCKED);
SET DEPT.AUTOHOLD(READ=LOCKED);
IF KEY EQ ' ' THEN DELETE;
IF RELEASE EQ 'R'
  THEN OUTPUT WORK.AUTOREL;
ELSE OUTPUT DEPT.AUTOHOLD;

***

PROC SORT DATA=DEPT.AUTOALLO(READ=LOCKED);
   BY REGALLOC;
PROC SORT DATA=WORK.ALLOTRAN;
   BY REGALLOC;

1583
DATA DEPT.AUTOSTCK(DROP=SHIP MTH ENTRUTE READ=LOCKED);
   UPDATE DEPT.AUTOSTCK(IN=B READ=LOCKED)
       WORK.AUTOTRAN(IN=A);
   BY NUMBER;

PROC APPEND BASE=DEPT.AUTOSHIP(READ=LOCKED)
   DATA=WORK.AUTOREL(DROP=DESCRPI-DESCRP40);
RUN;
   /* UNLOCK FILES IN DEPT LIBRARY */
   LOCKOUT(LIB=DEPT,DSN=AUTOHOLD AUTOSTCK AUTOSHIP AUTOALLO, UNLOCK=YES);   /* Same as above except for UNLOCK=YES */
   /*
   //
   */

• FUTURE

I did not attempt to adapt the lockout macro to the interactive environment, because where would you go if PROC DATASETS failed to lock the dataset(s)? If using SAS/AF, would this be a MENU, CBT, PROGRAM or drop back to the Display Manager? The macro, however, can be modified to work in an inactive mode, it just requires more work.

This method of locking datasets while under SAS/SHARE should work on any operating system. I hope it can be of some use in your environment.

Michael R. Gibson
Computer Information Center
The Gates Rubber Co.
P.O. Box 5887
Denver, CO 80217
(303)-744-5684

SAS, SAS/FSP and SAS/SHARE are registered trademarks of the SAS Institute Inc. Cary, NC, USA.