ABSTRACT:
With the availability of SAS/SHARE software in February 1987, new opportunities for developing information systems became a reality. Although the product initially addresses only the concurrent access and update of "data" type members, it significantly expanded the versatility and interactive potential of SAS-based information systems. This paper will first discuss what is SAS/SHARE, what will SAS/SHARE do and not do, and how to install/implement SAS/SHARE. Secondly it will discuss writing macros to control SAS/SHARE server assignment, programming considerations and conversion hints.

WHAT IS SAS/SHARE?
SAS/SHARE software provides concurrent access and update to data-type members within a SAS data library by means of server. The server is a program which synchronizes the input/output requests for access and update to the members of the SAS data library. There can be multiple servers to improve performance.

WHAT WILL SAS/SHARE DO?
SAS/SHARE allows multiple users to update "same" data-type member with PROC FSEEDIT "locking out" only on the "record" or observation level. PROC APPEND allows multiple users to add data on to the end of a member. All other procedures and the DATA step require exclusive control of a member by a single user for update. However "different" members in the same library can be updated at the same time as long a user has exclusive control. SAS/SHARE is natural complement to the the SAS/AF product in the area of data entry when you are developing dialogs for multiple users.

WHAT WILL SAS/SHARE NOT DO?
SAS/SHARE does not function with "catalog" type members. This means you can not modify or perform maintenance to FSEEDIT screens, SAS/AF menus or programs while other users are allocated to the library. This can be very limiting to the SAS/AF programmer who needs to make one little programming change. A programmer can not update the member until all users have released the library. SAS/SHARE also does not provide a "status-type" command to interrogate a "member" to determine if any user has exclusive control before attempting to gain exclusive control for update.

INSTALLATION CONSIDERATIONS (MVS):
SAS/SHARE installation is not complicated. However, SAS/SHARE does require systems programming assistance. Follow the "OS SAS/SHARE Installation Instructions" carefully. There are no short cuts. You or your systems programmer must install SVCO, install SASIUCV in a link list library, define an inactive MVS subsystem to provide an anchor point of SAS/SHARE control blocks, and provide a started task for each server in 'SYSL.PROCLIB'. If SAS/SHARE is to be supported from multiple CPUs, then VTAM definitions must be defined. A word of advice is test out SAS/SHARE on one CPU, without VTAM, before attempting the VTAM use.

IMPLEMENTATION AT SDDPC:
(San Diego Data Processing Corp)
The heart of SAS/SHARE are servers which synchronizes the input/output requests. The servers are started tasks which run the PROC SERVER procedure continuously. The servers are automatically started at IPL time and remain up at all times. If you are using SAS/SHARE with VTAM, VTAM must be completely up before executing the server started task. Likewise, if you must bring VTAM down, the operator must manually first cancel the SAS/SHARE started tasks or VTAM won't come down. This is to be corrected in the maintenance release or Version 6.

TYPES OF LIBRARIES: USER-DEFINED
User-defined libraries are easier to use and understand. You should first gain experience with them. They require no administration with the exception of knowing the name of the "server" to use. They can be allocated to and cleared from the server by the programmer or user. However, the only sure method to gain exclusive control of a member for "update" (other than FSEEDIT or APPEND) is to manually stop the libraries or physically log users off TSO.
Administrator-defined libraries are harder to use because they require administrator coordination. An administrator is a person who is designated to control the implementation and use of SAS/SHARE. The administrator controls the allocation of administrator libraries to a designated "server". Libraries are allocated to a server with PROC OPERATE. At SDDPC we automatically allocate all our administrator-defined libraries immediately after we bring up the "server" started task. Administrator libraries require a unique DDname/LIBREF within the "server". This is because the libraries are allocated to the same started task region. User-defined libraries don't require unique DDname/LIBREFs, the server generates one, in the form SYSnnnnn, when it dynamically allocates the library. The big advantage of administrator-defined libraries is that you can programatically, with PROC OPERATE, STOP or QUIESCE libraries when you need exclusive control and then START the libraries after the update. Both of these commands should be used carefully.

CONNECTING USERS TO SAS/SHARE LIBRARIES:

User-defined libraries:
LIBNAME libref 'dsn' SERVER=serverid;

Administrator-defined libraries:
LIBNAME libref SERVER=serverid;

"serverid" is the name of the started task. Note that "serverid" is hard-coded in the statement. The LIBNAME statement can be buried in SAS code. What if you want to change servers for performance reasons? It is like finding a needle in a haystack. The answer to this problem is my next subject: macros.

USE OF MACROS FOR ALLOCATION:

Macros are an ideal way to control server names in SAS/SHARE. The installation tape for SAS/SHARE contains examples of macros that you can use. "SAS Communications", First Quarter 1988, also contains an article called "Using SAS/SHARE Software Autocall Macros" on this subject. Data library names and associated DDname/LIBREF, server names can be maintained in a macro variable table(s). If name is isn't on the table then a default server name can be assigned. In this way server names are transparent to the programmer or the user and programmers never have to change code for server assignment.

Examples of SDDPC macros:

Normal SAS/SHARE allocation:
User - %SHARE(ALLOC,libref,'dsn')
SAS code ....; RUN;
%SHARE(FREE,libref)

Admin - %SHARE(ALLOC,libref)
SAS code ....; RUN;
%SHARE(FREE,libref)

Exclusive Control: Admin only
*Stop the library from server; %SHARE(STOP,libref);
*Gain exclusive control; LIBNAME libref 'dsn';
*Execute SAS code; SAS code ....; RUN;
*Free exclusive control; LIBNAME libref CLEAR;
*Start the library to the server; %SHARE(START,libref);

Examples of SDDPC SAS/SHARE macro code:

%SHRBUILD - Build DSN Table:
%MACRO SHRBUILD;
%SHRLIB(A;server;'dsn',libref);
%SHRLIB(U;server,'dsn ' ;libref);
...;
%SHRLIB(U,server,'DEFAULT',@USERDEF);
%MEND SHRBUILD;

Each entry defines:
A/U Admin or user library server Server name 'dsn' Data library name libref Admin - unique libref User - @USERDEF (constant)

%SHRLIB - Build macro variable entry:
%MACRO SHRLIB(SHRTYP,SHRSER,SHRDSN, SHRDDN);
%GLOBAL SHRNUM
SHRTBL;
%LET SHRNUM = %VAL(&SHRNUM + 1);
%GLOBAL SHRTYP&SHRNUM
SHRDSN&SHRNUM
SHRDDN&SHRNUM
SHRSER&SHRNUM;
%LET SHRTYP&SHRNUM = &SHRTYP;
%LET SHRDSN&SHRNUM = &SHRDSN;
%LET SHRDDN&SHRNUM = &SHRDDN;
%LET SHRSER&SHRNUM = &SHRSER;
%LET SHRTBL = YES;
%MEND SHRLIB;
%SHARE - Allocate SAS/SHARE libraries:

(selected portions)

%MACRO SHARE(ACTION, DDN, DSN, SERVER);

%GLOBAL SHARERC;
%LET SHARERC=0; %LET ERRTYP = NOTE;
%* ---- DETERMINE IF SHARE DSN TABLE DSN HAS BEEN BUILT -------------------;
%PROCESS:
%GLOBAL SHRTBL;
%IF "&SHRTBL" = "YES" %THEN %GOTO SEARCH; %ELSE %SHRBUILD;
%* ---- DETERMINE IF LIBRARY IS ON SHARE TABLE ----------------------------;
%SEARCH:
%LOCAL i;
%DO i = 1 %TO &SHRNUM %BY 1;
%IF &DSN = &SHRDSN1 %THEN %GOTO FOUND; %END;
%ELSE DO;
%LET SHARERC=16;
%MEXIT;
%END;
%NOTFOUND:
%IF &DSN = AND "&ACTION" = "ALLOC" %THEN %DO;
%PUT &ERRYP: SDDPC *****************************************;
%PUT &ERRYP, SDDPC * MACRO, SHARE(&ACTION, &DDN, &DSN, &SERVER);
%PUT &ERRYP, SDDPC * NO DSN NAME SPECIFIED, CAN NOT ALLOCATE;
%LET SHARERC=16;
%MEXIT;
%END;
%* ---- LAST ENTRY CONTAINS ON DEFAULT SERVER NAME -----------------------;
%LET i = &SHRNUM;
%FOUND:
%IF &SERVER = %THEN %LET OPERSER = &SERVER;
%ELSE %LET OPERSER = &SHRSER1;
%IF "&&SHRTYP1" = "A" %THEN %GOTO ADMIN;
%* ---- USER-DEFINED SERVER ---------------------------------------------;
%IF "&ACTION"="ALLOC" %THEN %DO
LIBNAME &DDN &DSN SERVER=&OPERSER;
%LET SHARERC=SYSLIBRC;
%IF &SYSLIBRC = 0 %THEN %LET ERRTYP = WARNING;
%PUT &ERRYP: SDDPC *****************************************;
%PUT &ERRYP, SDDPC * MACRO, SHARE(&ACTION, &DDN, &DSN, &SERVER);
%MEND SHARE;
%END;
%ELSE %IF "&ACTION"="FREE" %THEN %DO
LIBNAME &DDN CLEAR;
%MEND SHARE;
%END;
%ADMIN:
%* ---- FINAL EXIT ------------------------------------------------------;
%MEXIT:
%PUT &ERRYP: SDDPC * SHARE RETURN CODE=&SHARERC;
%MEND SHARE;

Note: The macro variable &ERRYP contains either NOTE or WARNING.

After testing use the OPTIONS NONOTES to bypass NOTE messages.
PROGRAMMING/CONVERSION CONSIDERATIONS

First consideration is: Does this need to library need to be on SAS/SHARE library? Unless your users have a need to update the same member or the same library, it shouldn’t be a SAS/SHARE library because of the additional overhead and administration.

Once you have made the decision for SAS/SHARE, assign a unique DDname/libref. It is required for an admin-defined library and makes sense for a user-defined library in case you may later want to make it an admin-defined library.

If you are converting an existing application, you must remove the //DDname cards in batch and the TSO ALLOCATE/FREE statements in TSO (ex: SAS/AF). These are replaced by LIBNAME or macro statements which control the allocation. This is because SAS/SHARE libraries are allocated to the server started task region, not to the batch or TSO region as are non-SAS/SHARE libraries.

SAS DATA steps and PROC statements require coding additions. To gain concurrent access to the member data, many PROCs or the DATA step require a new option (CNTLLEV=BLK) be added to the DATA=SASdataset or SET statements.

PROC SORT DATA=libref.member (CNTLLEV=BLK) OUT=libref.member;
DATA libref.member;
SET libref.member (CNTLLEV=BLK);
PROCs dependent on BY variable processing, such as PROC PRINT, PROC GCHART, require that the data be copied before processing.

PROC COPY IN=libref OUT=libref;
SELECT member;
To update you must get exclusive control of the member with the the exception of PROC FSEDIT and APPEND which allow concurrent update.

WHAT DOES AN SAS/SHARE ADMINISTRATOR DO?

A administrator can do some or all of these the following tasks depending on the division of responsibility in your organization.
- Install SAS/SHARE
- Determine if libraries should be SAS/SHARE
- Write SAS/SHARE macros
- Maintain table of administrator library names
- Assign servers
- Allocate and monitor disk space
- Monitor performance
- Write SAS/SHARE installation dependent documentation
- Maintain security

NAMING CONVENTIONS ARE IMPORTANT!

Establishing naming conventions has many benefits. First of all, you can identify by sight if they are SAS/SHARE libraries. Second, and very important to our installation, we can write general purpose SAS/AF or SPF dialogs for PROC FSEDIT, PROC PRINT, PROC DATASETS, PROC SORT. In coding behind the dialogs we can them generate the correct allocation code depending on the SAS data library name depending on whether the library is SAS/SHARE or not (%SHARE(ALLOC,...) vs. TSO ALLOCATE ...). Third, it is easier to monitor performance.

HOW DO YOU MONITOR SAS/SHARE PERFORMANCE

The tools are very limited and it is mainly trial and error. The best yardstick are users complaining about response time. My usual action, providing we are not having an overall system response time problem, is to assign that data library to another server. At our installation that is easy because of the macros and the data library macro table. You can also review the log (FTIIFOO1) from the server execution in question. The log tracks SAS/SHARE activity by date, time, action, libref, member and user. It also contains error messages, if any. The only crucial piece of information missing from the log is SAS data library name which hopefully will be added in future maintenance.

I understand some installation have analyzed SMF records. An interesting SMF chargeback problem arises with SAS/SHARE because all files are allocated to "server" or the started task. Therefore, the I/O associated with the SAS/SHARE libraries are being charged against the started task region and not the user region. In most installations, SAS/SHARE users are getting a free ride because started tasks are non-billable or spread over all the users.
OTHER ISSUES:

Security is always a concern. Currently we use our installation security package to provide security for SAS/SHARE library like any other library. There are also user exits available which are documented in Technical Report: Y-103R "User Exit Facilities for the SAS System, Version 5". I understand these have been used for SAS/SHARE security.

Libraries in our installation are backed up regularly by full pack or incremental backups. These backups on SAS/SHARE libraries will fail if they are allocated to the server. Therefore, as a part of the backup job the administrator libraries are automatically stopped before backup and restarted after the backup.

Restoring a library or increasing space in a library can be a problem when users are allocated to it. In that case of administrator libraries, it is easy to stop the libraries and then restart after the restore or increase of space.

SAS/SHARE WRAPUP:

SAS/SHARE has provided our corporation with a useful tool particularly in the area of PROC FSEDIT. We have had very few problems and would recommend it with the following guidelines. Your company should be willing to expend system programming and communication programming (if VTAM is used) support in the installation. It's very minimal.

SAS/SHARE macros should be coded so allocation to servers can be controlled dynamically. Naming conventions, procedures, and installation dependent documentation should be written before releasing SAS/SHARE to users. With effective preplanning SAS/SHARE can be a successful product in your installation and require very little time to administer after the initial installation.

SAS, SAS/AF, SAS/FSP, SAS Communications are registered trademarks of SAS Institute, Inc., Cary, NC, USA.

SAS/SHARE is the trademark of SAS Institute, Inc.

San Diego Data Processing Corporation
5975 Santa Fe Street
San Diego, CA 92109
(619) 236-5977

469