Changes and Enhancements

SAS/SHARE Services Extend to the Internet

A SAS/SHARE*NET server is a SAS/SHARE server for which SAS/SHARE*NET software is licensed. SAS/SHARE*NET software represents the Data Services component of SAS/IntrNet software. SAS/SHARE*NET is the licensable right to send requests to a SAS/SHARE server from a client that is not a SAS application. Examples of clients that are not SAS applications are:

htmSQL

runs a Web server and offers a gateway to your SAS data from a Web browser. It enables you to incorporate data into a Web page by using SQL queries.

Java applet or application

uses SAS/SHARE*NET driver for JDBC, which enables you to write Java applets or applications that can view and update data through a direct connection to a SAS/SHARE*NET server.

C program

uses the SAS SQL Library for C. The SAS SQL Library is an applications programming interface (API) that enables you to create applications that use SQL queries and statements to access data in SAS data sets and in other database management systems.

An application that uses the ODBC driver, such as Microsoft Excel

uses the ODBC driver, which provides ODBC-compliant Windows applications that have “read” and “write” access to local and remote SAS data sets.

It is also possible for a SAS/SHARE server to be a client to another server such as an ORACLE DBMS server. The preceding client interfaces or applications are documented by their own respective vendors or developers.

PROC SERVER Options for Performance and Security

The following new options in the PROC SERVER statement improve the performance and security of the SAS/SHARE server.
ACCTLVL=

is a performance-tuning parameter. It enables you to examine server resource consumption during different parts of your application's execution.

LOG=values

is a performance-tuning parameter. Three new values (ACTIVETIME, BYTECOUNT, and ELAPSED TIME) provide more efficient resource usage statistics than the now obsolete values IO and CPU.

ACTIVETIME

logs a client's cumulative elapsed time in a single session.

BYTECOUNT

logs a client's cumulative bytes in a single session.

ELAPSED TIME

logs a client's total time in a single session (including idle time).

AUTHENTICATE=

is a security feature that specifies whether a client must supply a valid userid and password in order to access the server.

PT2DBPW=

is a security feature that specifies the password that is required for clients that run an SQL CONNECT TO REMOTE statement that includes the DBMS= option.

LIBNAME Statement Option for the Remote Engine

The LIBNAME statement supports the ROPTIONS option.

ROPTIONS=

specifies options to the remote engine that are used to process the SAS data library in the server session.

Version 6 and Version 7 Compatibility

SAS/SHARE provides comprehensive Version 6 to Version 7 compatibility for client/server applications. This enables you to convert your applications incrementally rather than at one time. For example, you can convert your client from Version 6 to Version 7 and still communicate with a Version 6 server. After successfully converting the client, the server can then be converted. Another approach would be to convert the server first and then the client. Otherwise, both client and server would need to be converted simultaneously, which generally increases the cost and risk of the conversion.

SAS System Options for Data Encryption

Encryption services protect data that is sent between hosts across a network. Encryption services use a reversible algorithm to convert plain-text data into an unintelligible form, thus protecting data from being used by unauthorized parties.
NETENCRYPT
specifies the requirement to operate in encryption mode; for example, both the
client and the server sessions must be operating in encryption mode.

NETENCRYTALGORITHM
specifies one or more encryption algorithms to use in a SAS/SHARE client/server
session.

NETENCRYPTKEYLEN
specifies the key length - 40-bit or 128-bit - to be used by the encryption algorithm.

NETMAC
controls the use of Message Authentication Codes (MACs) on network
communications. A MAC is the equivalent of a checksum that is used to ensure
that the original message has not been modified.

The ALLOCATE SASFILE Command in a PROC SERVER Statement

The PROC SERVER statement ALLOCATE SASFILE command specifies that one or
more SAS data sets can be opened for the duration of a server session until you
terminate the server. Keeping SAS data sets open until a server terminates can
enhance server performance by reducing overhead from users opening and closing data
sets during the server’s processing. In each ALLOCATE SASFILE command, you can
specify up to 20 SAS data set names with accompanying data set options. Syntax for
the ALLOCATE SASFILE command follows:

PROC SERVER <options>;
ALLOCATE SASFILE SAS-data-set1 (data-set-options);
<ALLOCATION SAS-data-set2 (data-set-options) ...>
SAS-data-set20 (data-set-options)>;

Support for General Version 7 Enhancements

SAS/SHARE also supports the following enhancements that have been implemented
for Version 7 of base SAS software:
- long names
- concatenated libraries
- concatenated catalogs
- generation data sets
- integrity constraints.

Specifying a SAS/SHARE Server Identifier as a Port Number

On all platforms that use the TCP/IP communications access method, in the SAS/
SHARE LIBNAME statement, the SERVER procedure, and the OPERATE procedure
that include an option which specifies a server identifier, you can optionally represent
the server identifier as a port number.
Here is the syntax:

SERVER=node.server

where
node is a valid TCP/IP node name that is specified if the client and the server sessions are running on different nodes.
server is either a server-id or a port
server-id corresponds to a service name that is specified in the TCP/IP services file.
port is a unique number that is associated with the service name.

For UNIX hosts, server ports are limited to the range 256 through 1023. For all other hosts, server ports extend from 1024 to 65532. Precede the port number with two underscores.

Here is a UNIX example:

% let srvnode=mktserver.acme.com;
libname sales server=srvnode. _ _ 256;

Usually, a system administrator provides you with a server port number.