CHAPTER

28

Windows: SPX Access Method

SAS Support for SPX on Windows

Note: Beginning with Version 7, the SPX access method is not supported. Information about SPX is included here for Version 6 users. △

Tasks That Are Common to SAS/CONNECT and SAS/SHARE

System Administrator or User

To use the SPX access method with a Windows host for SAS/CONNECT and SAS/SHARE, perform these tasks:

1. Verify that you have met all your site and software requirements.
2. Verify that the resources for the IPX/SPX access method have been defined.
3. Verify that you know how to set options in the SAS software.
4. Set the SAS/CONNECT and SAS/SHARE options that you want.

System and Software Requirements for SAS/CONNECT and SAS/SHARE

Ensure that the following conditions have been met:

1. The IPX/SPX protocol has been installed at both the local and remote hosts.
2. SAS has been installed on both the local and remote hosts.

Windows NT and Windows 95 Requirements

To use the SPX access method with Windows NT or Windows 95, install and configure the following:

1. the SPX protocol which is included with both Windows NT and Windows 95.

Setting SAS Options and Variables

You may need to set specific options to establish the connections that you want with SAS/CONNECT and SAS/SHARE when using the SPX communications access method.

Consult with your network administrator to determine what options must be set and what values to assign to them.

You may specify an option in any of the following forms:

- in an OPTIONS statement in a SAS session or in an AUTOEXEC file:

  OPTIONS SET=*variable-name value*;

  Example:

  ```
  options set=spxmsgsize 500;
  ```

- SAS configuration file or at SAS invocation:

  -SET *variable-name value*

  Example:

  ```
  -set spxmsgsize 500
  ```

- SAS macro variable:

  %LET *variable-name=value*;

  Example:

  ```
  %let spxmsgsize=500;
  ```

- DOS operating system environment variable:

SET *variable-name=value*

Example:

```
set spxmsgsize=500
```

Values for these options may contain up to eight characters, consisting of alphanumeric characters, the percent sign (%), the dollar sign ($), the pound sign (#), the at sign (@), and the underscore (_).

If you set multiple forms of the same option, here is the order of precedence that is followed:

SAS macro variable

OPTIONS statement

AUTOEXEC file

SAS invocation

SAS configuration file

DOS environment variable.

SAS/CONNECT and SAS/SHARE Option

SPXMSGSIZE *size-of-input/output-buffer*
specifies the size of the SAS program input and output buffer. The range of acceptable values for this option is from 298 bytes to the maximum packet size that your network allows; for example, 1514 bytes on an Ethernet network and 4202 bytes on a Token Ring network. You must ensure that the IPX/SPX protocol is using the same packet type as the other computers that you want to communicate with. You must choose 802.2, 802.3, or similar packet types.

Example:

```
-set spxmsgsize 500
```

Ask your network administrator for advice about setting this option.

SAS/CONNECT SASUSER and SASPASS Options

SASUSER *userid*

SASPASS *password*

On the local host when the remote host is secure, you must either assign a valid userid and a password to the SASUSER and SASPASS options or supply them to SAS, when prompted.

Consult with the system administrator of the remote host at which the spawner is invoked for a valid userid and a password.

The SASUSER and SASPASS options store the userid and the password of the remote host that, when passed to the remote host, allows a local host connection.

Example:

```
options set=sasuser bass;
options set=saspass time2go;
```

See “Setting SAS Options and Variables” on page 404 for examples of the forms that you can use to specify SASUSER and SASPASS.

Also see Chapter 35, “PC Spawner Program,” on page 471 for information about the -PROTECTION and -SECURITY options in the PC spawner program, which controls the security of the remote host.

SAS/SHARE SASSECUR Option

CAUTION:
Windows NT Only SAS/SHARE server security is supported on the Windows NT platform only. △

You may set the SASSECUR option to provide security for the SAS/SHARE server, allowing access to local hosts or clients whose userids and passwords have been verified.
Values that you set at a SAS/SHARE client follow:

SASSECUR=_NONE_ | _PROMPT_ | *userid.password* | _SECURE_

NONE
must be set at the SAS/SHARE client.
Setting this value does not establish secure sessions for connecting SAS/SHARE clients.
This is the default.

PROMPT
must be set at the SAS/SHARE client.
PROMPT specifies that SAS prompt the user for userid and password information. When prompted for a password, the input field is not displayed. Choosing to prompt for a userid and a password provides more security than assigning the userid and the password to the system option.

userid.password
must be set at the SAS/SHARE client.
This value specifies both the userid and password. Assigning both the userid and password directly to the SASSECUR option at the SAS/SHARE client may inadvertently publicize this information and compromise the security of the SAS/SHARE server. Assigning the value to the option in a file allows anyone to read it.

SECURE
must be set at the SAS/SHARE server on a Windows NT host only.
The _SECURE_ value for the SASSECUR option requires a SAS/SHARE client to supply both a valid userid and password to the remote host or the remote host on which the server is running in order to allow client access to the server.
Specify the SASSECUR option before you create a server.
Examples:

```
options set=sassecur _none_;
options set=sassecur _prompt_;
options set=sassecur bass.time2go;
options set=sassecur _secure;
```

See “Setting SAS Options and Variables” on page 404 for examples of the forms that you can use to specify the SASSECUR option.

SAS/CONNECT

Local Host Tasks

User or Applications Programmer

To connect a Windows local host to a remote host, perform these tasks at the local host:

1. Optionally, set a userid and a password to ensure security at the remote host, as necessary.
2. Specify the communications access method.
3. Specify a remote host name.
4. Sign on to the remote host.

Setting Security for Local Hosts

If the PC spawner program is running in secure mode, you must also set the remote host userid and password at the local host. Setting the -PROTECTION option in the PC spawner invocation command secures the spawner. This is valid for Windows 95 and Windows 98 only.

See Chapter 35, “PC Spawner Program,” on page 471 for information about starting the spawner on the remote host.

Specifying the SPX Communications Access Method

You must specify the SPX communications access method to make a remote host connection. Use the following syntax:

```
OPTIONS COMAMID=access-method-id;
```

where COMAMID is an acronym for Communications Access Method Identification. *access-method-id* identifies the method used by the local host to communicate with the remote host.

SPX (an abbreviation for Sequenced Packet Exchange) is an example of an access-method-id.

Example:

```
options comamid=spx;
```

Alternatively, you may specify this option at a SAS invocation or in a SAS configuration file.

Specifying the Remote Host Name

To make a connection from a Windows NT or a Windows 95 local host to a remote host, use the following syntax:

```
OPTIONS REMOTE=network-name;
```

where *network-name* is the -NETNAME option in the PC spawner program that you start on the remote host. See Chapter 35, “PC Spawner Program,” on page 471 for more information.

Example:

```
options remote=mynet;
```

Alternatively, you may specify this option at a SAS invocation or in a SAS configuration file.

Signing On to the Remote Host

To complete your sign on to the remote host, enter the SIGNON statement, as follows:

```
signon;
```

Note: Sign-on script files are not needed with the SPX access method because the PC spawner program directly invokes the remote SAS session and replaces the need for a script file. △

Although no errors are produced if you specify a script file, you do waste processing time. If you defined the RLINK fileref before establishing a connection, when you sign on, SAS/CONNECT processes and loads the script file identified by the fileref, but the SPX access method will ignore the script.

If you do not want to omit the RLINK fileref but want to prevent wasted processing time, use the NOSCRIPT option in the SIGNON and SIGNOFF statements, as shown here:

```
signon noscript;
.
.
.
signoff noscript;
```

Local Host Example

The following example illustrates the statements that you specify in a Windows NT, or Windows 95 local host SAS session to connect to a remote host with the SPX access method:

```
options set=sasuser userid set=saspass password;
options set=spxmsgsize 4202;
options comamid=spx remote=sasrem;
signon;
```

This example assumes a connection to a PC spawner that is running in secure mode. The SAS options SASUSER and SASPASS allow the userid and the password to be passed to the remote PC spawner, which permits a connection. SPXMSGSIZE is set (see “Setting SAS Options and Variables” on page 404 for details). The SPX communications access method is declared with a connection to the remote host SASREM, which is the name that is specified in the -SPXNAME option at the PC spawner invocation. The SIGNON statement performs the sign-on process.

Remote Host Tasks

System Administrator

To allow a connection from a local host perform these tasks at the remote host:

1. Start the PC spawner program.
2. Set several remote host options, as necessary.

Starting the PC Spawner Program

You must invoke the PC spawner program on the Windows remote host to enable local hosts to connect to it. The spawner program resides on a remote host, listening for SAS/CONNECT client requests for connection to the remote host. After the spawner program receives a request, it invokes the remote SAS session.

Optionally, you may set password protection through the -PROTECTION option in the PC spawner invocation command.

Note: For Windows NT only, setting the -SECURITY option in the PC spawner invocation command also secures the spawner. △

The spawner then verifies the userid and the password that are assigned to the SASUSER and SASPASS options. For information about setting security, see "SAS/CONNECT SASUSER and SASPASS Options" on page 405.

See Chapter 35, "PC Spawner Program," on page 471 for information about starting the spawner on the remote host.

Setting Options at the Remote Host

Although sign-on script files are not used for the SPX access method, you may set remote host options at the remote host. It is recommended that you set these options:

NO$SYNTAXCHECK
: allows the continuation of statement processing at the remote host regardless of syntax error conditions.

  This option is valid as part of a configuration file, at SAS invocation, or in an OPTIONS statement.

NOTERMINAL
: specifies whether a terminal is attached at SAS invocation. If NOTERMINAL is specified, requestor windows are not displayed.

  Setting NOTERMINAL at the remote host is advisable so that no terminal is associated with the remote session. This option prevents SAS from displaying error messages and dialog boxes on the remote host, which requires user intervention.

  This option is valid as part of a configuration file or at SAS invocation.

  See *SAS Language Reference: Dictionary* for details about this option.

NOXWAIT
: applies to OS/2 or Windows remote hosts only.

  specifies whether you have to type EXIT at the DOS prompt before the DOS shell closes. Setting NOXWAIT at the remote host is recommended to prevent SAS from displaying a dialog box on the remote host. Such a display requires that you explicitly type EXIT at the remote host and gives the appearance that the REMOTE SUBMIT command is hung.

  This option is valid as part of a configuration file, at SAS invocation, or in an OPTIONS statement.

  See *SAS Companion for the Microsoft Windows Environment* for details about this option.

Remote Host Example

The following example illustrates the statements that you specify in a Windows NT or a Windows 95 remote host configuration file to prepare for a connection from a supported local host with the SPX access method:

```
-dmr
-comamid spx
-no$syntaxcheck
-noterminal
-noxwait
```

The following example shows how to invoke the PC spawner on a Windows NT remote host:

```
c:\sas\connect\sasexe\spawner -comamid spx -protection
                              -spxname sasrem -file mysas.cmd
```

The PC spawner is invoked, and the SPX access method is specified. The -PROTECTION option verifies the userids and the passwords of connecting clients. The -SPXNAME option specifies the name that the PC spawner program uses to communicate with the local host. The -FILE option executes the MYSAS.CMD file, which invokes a SAS session.

See Chapter 35, "PC Spawner Program," on page 471 for information about the contents of a command file and executing the PC spawner. Options that are set by means of the spawner may override options that are set in a remote host configuration file.

SAS/SHARE

Client Tasks

User or Applications Programmer

To prepare for accessing a SAS/SHARE server, perform the following tasks:

1. For Windows NT only, assign the appropriate rights to each connecting client.
2. For Windows NT only, set security for connecting clients.
3. Specify the SPX access method.
4. Know how to specify a server name.

Assigning the Appropriate Rights for Connecting Clients

CAUTION:

Windows NT only Server security is supported on the Windows NT platform only. △

The account in which a connecting client runs must have the appropriate rights. To assign these rights

1. Click on the Administrative Tools icon.
2. Click on the User Manager icon.
3. From the Policies pull-down menu, select "User Rights."
4. Click the "Show Advanced User Rights" box.
5. Assign "Log on as a batch job" rights to the appropriate users.

Setting Security for Connecting Clients

CAUTION:
Windows NT only Server security is supported on the Windows NT platform only. △

If you set the SASSECUR option at the client, specify a userid and a password that are valid on the server. For information about setting the SASSECUR option, see "SAS/SHARE SASSECUR Option" on page 406.

Specifying the SPX Access Method

You must specify the SPX access method at each connecting client before you can access a server. Use the following syntax:

```
OPTIONS COMAMID=access-method-id;
```

where COMAMID is an acronym for Communications Access Method Identification. *access-method-id* identifies the method used by the client to communicate with the server. SPX (an abbreviation for Sequenced Package Exchange) is an example of an *access-method-id*.

Example:

```
options comamid=spx;
```

The server is accessed using the SPX access method.

You may specify the COMAMID option in an OPTIONS statement, at a SAS invocation, or in a SAS configuration file.

Additionally, you may use the COMAUX1 and COMAUX2 options to designate auxiliary communications access methods. See Table 1.3 on page 10 for the supported access methods by host.

If the first method fails to access a server, the second method is attempted, and so on. You can specify up to two auxiliary access methods, depending on the number of methods that are supported between client and server hosts.

COMAUX options can be specified only at a SAS invocation or in a SAS configuration file. The syntax for the COMAUX option follows:

```
-COMAUX1 alternate-method
-COMAUX2 alternate-method
```

An example of configuration file entries for a Windows NT client connecting to a Windows NT server follows:

```
-comamid spx
-comaux1 tcp
-comaux2 netbios
```

If the server cannot be reached with the SPX access method, a second attempt is made with the TCP/IP access method, and then with the NetBIOS method.

Specifying a Server Name

You must specify the server name in the LIBNAME and the PROC OPERATE statements. Use the following syntax:

```
SERVER=server-id
```

Follow standard SAS naming rules when defining a server name. See *SAS Language Reference: Concepts* for details about SAS naming rules. See *SAS/SHARE User's Guide* for details about the LIBNAME and PROC OPERATE statements.

Client Example

The following example illustrates the statements that you specify in a Windows client session that are used to access a server with the SPX access method:

```
options comamid=spx;
libname sasdata 'c:\edc\prog2\sasdata' server=share1;
```

The SPX access method is declared. The LIBNAME statement specifies the data library that is accessed through the server SHARE1.

Server Tasks

Server Administrator

CAUTION:
Windows NT Only Server security is supported on the Windows NT platform only. △

To set up a secure server and to make it accessible to a client, perform the following tasks:

1. Assign the appropriate rights for a secure server for Windows NT only.
2. Allow only validated clients to access a secure server for Windows NT only.
3. Set the SPX access method security for Windows NT only.
4. Specify the SPX access method.

Assigning the Appropriate Rights for a Secure Server

CAUTION:
Windows NT only This process is supported on the Windows NT platform only. △

The account in which a secure server runs must have the appropriate rights. To assign these rights

1. Click on the Administrative Tools icon.
2. Click on the User Manager icon.
3. From the Policies pull-down menu, select "User Rights."
4. Click the "Show Advanced User Rights" box.
5. Assign "Act as part of the operating system" rights to the appropriate users.

Setting SPX Access Method Security

CAUTION:
Windows NT only This process is supported on the Windows NT platform only. △

Before you can create a secure SAS/SHARE server, you must make the access method secure by assigning the _SECURE_ value to the SASSECUR option. See "SAS/SHARE SASSECUR Option" on page 406 for information about setting the SASSECUR option.

Specifying the SPX Access Method at the Server

You must specify the SPX communications access method before you can create and access a SAS/SHARE server.

Use the following syntax to specify the SPX access method at the server:

```
OPTIONS COMAMID=access-method-id;
```

where COMAMID is an acronym for Communications Access Method Identification. *access-method-id* identifies the method used by the server to communicate with the client. SPX (an abbreviation for Sequenced Packet Exchange)is an example of an *access-method-id*.

For a server that is running on a host on which only one communications access method is available, use only the COMAMID option.

Example:

```
options comamid=spx;
```

The server will be available only to SAS/SHARE sessions that use the SPX access method.

You may specify the COMAMID option in an OPTIONS statement, at a SAS invocation, or in a SAS configuration file.

However, if the host on which a server running supports multiple access methods, you may specify up to two auxiliary access methods by which clients may access the server using the COMAUX1 and COMAUX2 options. See Table 1.3 on page 10 for the supported access methods by host.

All of the access methods initialize when the server initializes. The activation of multiple access methods makes a server available to several groups of clients, each using a different communications access method simultaneously.

COMAUX options can be specified only at a SAS invocation or in a SAS configuration file. The syntax for the COMAUX option follows:

```
-COMAUX1 alternate-method
-COMAUX2 alternate-method
```

An example of configuration file entries for a server that is running on an Windows NT host follows:

```
-comamid spx
-comaux1 tcp
-comaux2 netbios
```

When the server starts, all of the communications access methods are initialized. The server is simultaneously available to client sessions that use the SPX access method as well as to clients that use the TCP/IP and NetBIOS access methods.

Specifying a Server Name

You must specify the server name in the PROC SERVER statement. Use the following syntax:

```
SERVER=server-id
```

Follow standard SAS naming rules when defining a server name. See *SAS Language Reference: Concepts* for details about SAS naming rules. See *SAS/SHARE User's Guide* for details about the PROC SERVER statement.

Server Example

The following example illustrates the statements that you specify in a configuration file on the Windows host at which you start a server:

```
-set spxmsgsize 4202
```

See “SAS/CONNECT and SAS/SHARE Option” on page 405 for details about this option.

The following statements issued in a SAS session on the Windows remote host illustrate how to start a server:

```
options comamid=spx;
proc server id=share1;
run;
```

The SPX access method is declared for the server SHARE1 that is started on the Windows NT remote host. The additional options in the PROC SERVER statement allow only validated clients to access the server.

The correct bibliographic citation for this manual is as follows: SAS Institute Inc., *Communications Access Methods for SAS/CONNECT and SAS/SHARE Software, Version 8*, Cary, NC: SAS Institute Inc., 1999. pp. 643.

Communications Access Methods for SAS/CONNECT and SAS/SHARE Software, Version 8

ISBN 1–58025–479–9

SAS Institute Inc., SAS Campus Drive, Cary, North Carolina 27513.

1st printing, September 1999