CHAPTER

23

Starting and Stopping SAS/CONNECT Software

Establishing a Connection

SAS/CONNECT must perform these two steps in order to establish a connection from a local SAS session to a remote SAS session:

- Gain access to a remote host (for example, by means of a logon or a spawner program).
- Execute a command to invoke a remote SAS session.

Here are the methods for establishing a connection. The primary difference among these methods is how the remote SAS session is invoked.

- Remote host spawner
- Local host sign on script
- Multi-Process (MP) sign on with SASCMD option
- Access method-specific command file.

In addition, for all preceding methods, there are two methods for establishing a connection to a remote SAS session:

- Explicit execution of a SIGNON statement or command
- Autosignon feature that is built into the RUSBMIT statement or command.

Remote Host Spawner

A spawner program must be invoked on a remote SAS/CONNECT host prior to executing a SIGNON. The spawner program listens for client requests for connection to

the remote host. The advantages of using a spawner program are userid and password encryption and the optional use of a SIGNON script file.

SAS/CONNECT requires the following options for a spawner-based connection:

REMOTE=
designates the name of the remote host that you want to access.

COMAMID=
specifies the communication access method identifier that is used to pass data between the two hosts.

Script file (optional)
For details, see "Local Host-Based Sign-on Script" on page 194.

For more information about spawner programs, see *Communications Access Methods for SAS/CONNECT and SAS/SHARE Software*.

Local Host-Based Sign-on Script

Script files make use of either a remote TELNET daemon (for TELNET or the TCP/IP access method) or a remote presentation space (for the EHLLAPI access method) to establish a connection to the remote host. Script files provide a programmatic interface for navigation through remote host login and invocation to the remote SAS session.

SAS/CONNECT requires the following options for a script-based connection:

- REMOTE=
- COMAMID=

The need for a sign-on script is based on the access method that is used to communicate between hosts and the method used to invoke SAS/CONNECT. The same script that is used to sign on is usually also used at sign off.

Uses a Script	May Use a Script	Does Not Use a Script
TELNET	TCP/IP	DECnet
EHLLAPI		NetBIOS
		APPC

For a description of the statements used in the scripts, see Table 28.1 on page 247.

Multi-Process (MP) CONNECT

MP CONNECT establishes a connection to a remote SAS session that runs on the same host as the local SAS session. If you have MP/SMP hardware, this feature enables your operating system to exploit the multi-processor capabilities. Besides ease-of-use, the primary advantage of MP CONNECT is the ability to asynchronously process encapsulated units of work whose results can be subsequently merged and managed from the originating local SAS session. This method eliminates the need for executing a program on the remote host such as a spawner program or a TELNET daemon. It also eliminates the need for a script file on the local host and the need to configure the access method.

MP CONNECT requires the following options:

REMOTE= or PROCESS=
specifies the name of the remote session to which statements are submitted for execution.

COMAMID=
specifies the identifier for the communications access method that is used to establish a connection between a local SAS session and a remote SAS session. TCP/IP is the only supported access method.

SASCMD=
specifies the command to be used to invoke the remote SAS session for MP CONNECT.

Communications Access Method Specific Command File

This method of establishing a connection uses the features that are built into various protocols and is defined by the protocol vendor. These access methods provide command files that are used to establish a connection:

Communications Access Method	Type of Command File
APPC	Transaction program
DECnet	SASCONN command file

For more information about these access methods and their associated command files, see *Communications Access Methods for SAS/CONNECT and SAS/SHARE Software*.

Using Encryption Services

In Version 7 and later versions of SAS/CONNECT software, you may choose to implement encryption services to protect data that is sent between hosts across a network. See “What Are Encryption Services?” on page 487 for more information.

Using the SAS Windowing Environment to Start and Stop SAS/CONNECT

SAS software provides a windowing environment that you can use to issue commands and perform other operations during a SAS session. This section describes how to use the SIGNON and SIGNOFF pull-down menu options to start and stop a SAS/CONNECT session.

Starting SAS/CONNECT by Using the Signon Window

To start a SAS/CONNECT session:

1 Select **Run->Signon** from the SAS Program Editor window and menu bar.

2 Complete the entry fields in the Signon... window. Use these guidelines:

Script file name:
If your communications access method does not require a script file, leave this field blank. If your access method requires a script file, type the full path and

the name of the script file. For example, if you are connecting to a remote OS/390 host by using the TCP/IP access method, type:

pathname/tcptso.scr

For more information about the names and locations of scripts used with a specific access method, see *Communications Access Methods for SAS/CONNECT and SAS/SHARE Software.*

Remote session name:
Type the name of the remote node that you want to access. For example, use the *tcp-host-name* for the TCP/IP access method. For more information, see "CONNECTREMOTE=" on page 219.

Communications access method ID:
Type the value for the COMAMID= option. For example, for the TCP/IP access method, type:

tcp

For more information about the values that can be used in the COMAMID= option for a specific host and access method, see "COMAMID=" on page 213.

Transmission buffer size:
Type in the value of the buffer size that SAS/CONNECT will use for transmitting data. For more information, see "TBUFSIZE=" on page 223.

Remote session macro variable name:
Type in the name of the macro variable that you want to use to associate with the remote session. For more information, see the CMACVAR= option in the SIGNON Command and Statement on page 228.

Display transfer status (yes/no):
Type **yes** or **no** to determine if the status window is displayed during transfers.

YES status window *will* be displayed during file transfers. This is the default.

NO status window *will not* be displayed during file transfers.

For more information, see "CONNECTSTATUS=" on page 221.

Execute remote submit synchronously (yes/no):
Type **yes** or **no** to specify whether remote submits are to be executed serially (synchronously).

YES indicates synchronous remote submits, which means that you will wait until the remote submit is finished processing before regaining control in the local SAS session. This is the default.

NO indicates asynchronous remote submit, which means that after the processing begins on the remote host, you will immediately regain control of your local SAS session to continue processing.

For more information, see "CONNECTWAIT=" on page 222.

3 Select **OK** to sign on, or select **Cancel** to return to the Program Editor window without signing on.

Stopping SAS/CONNECT by Using the Signoff Window

1 To stop a SAS/CONNECT session by signing off, from the pull-down menu , select **Run -> Signoff** and select **OK**.

2 After you sign on to a SAS/CONNECT session, the values that you type in the Signon window remain in effect during your current SAS session. If you sign off and later during that session want to sign on again, you can leave the fields blank. Simply select **OK** to sign on to the same host that has the same communications access method that you used previously.

After you end your SAS session, the values in the Signon window are deleted, and you must type them in again to sign on during a new SAS session.

Using the Program Editor Window to Start and Stop SAS/CONNECT

Sign On from the Program Editor Window

1 Enter an OPTIONS statement in the Program Editor window of the local SAS session.

Use the SUBMIT command, statement, or function key to execute the OPTIONS statement. The OPTIONS statement must specify the COMAMID= system option and may specify the REMOTE= system option. For example:

```
options comamid=communications-method
   remote=remote-session-id;
```

See “COMAMID=” on page 213 and “CONNECTREMOTE=” on page 219 for more information about specifying values for these options.

2 Issue the SIGNON command or type the SIGNON statement in the local SAS session. Specify the appropriate sample script (if necessary) for your remote operating system:

```
signon cscript='external-file-name-of-script';
```

Note: Automatic signon sample scripts should be modified with installation-specific information before you can use them to start the connection. △

An example of signing on a remote host that is running a spawner program includes:

```
options comamid=communications-method
   remote=nodename.servicename;
signon;
```

After the SIGNON command executes successfully, a message in the Log window indicates that the connection is established.

Now you are ready to use SAS/CONNECT. You can remote submit SAS programs to execute on the host, access a single-user server with RLS, and upload and download SAS files and external files.

Sign Off from the Program Editor Window

Issue the SIGNOFF command, or type the SIGNOFF statement in the local SAS session:

```
signoff cscript='external-file-name-of-script'
```

Note: If you used a script to sign on, the same script can be used to stop the connection. △

After the SIGNOFF command executes successfully, a message in the Log window indicates that the connection is terminated.

The sample scripts that are used for automatic log on are used for log off from your remote system session.

Setting the Autoexec File for SAS/CONNECT

You can simplify the process of starting and stopping the connection by following these recommendations:

- Review the contents of your SAS autoexec file. The *autoexec file* is a file of SAS statements that may be executed automatically when you begin a local session. The autoexec file could include a FILENAME statement that defines the fileref RLINK. Be sure that it gives the correct file specification for the script that you use to start SAS/CONNECT. See Chapter 26, "Syntax for the SIGNON and the SIGNOFF Commands and Statements," on page 227 for more information about the autoexec file.

  By assigning the fileref RLINK to your script, you can start the connection without using the SIGNON command to specify the script name and you can stop it without using the SIGNOFF command to specify the script name. This is because RLINK is the reserved fileref for script files.

  When SAS executes a SIGNON or a SIGNOFF command without a fileref, SAS automatically searches for a file that is defined with RLINK as the fileref. If RLINK has been defined, SAS executes the corresponding script.

- Include an OPTIONS statement in your autoexec file to specify the COMAMID= system option and the REMOTE= system option. For example, under OS/2 your OPTIONS statement might include these values:

  ```
  options comamid=tcp
     remote=tcp-host-name;
  ```

  By including an OPTIONS statement that contains the REMOTE= and COMAMID= system options in the autoexec file, you avoid having to execute an OPTIONS statement in each SAS session when using SAS/CONNECT.

Modifying your autoexec file as recommended eliminates a step in the process of starting the connection, and you can use the short form of the SIGNON and SIGNOFF commands.

For example, to start a connection from a SAS session that was invoked by using a modified autoexec file, simply issue the SIGNON command or submit the SIGNON statement:

```
signon
```

or

```
signon;
```

When you have completed your remote processing, you need only to issue the SIGNOFF command or submit the SIGNOFF statement to stop the connection:

```
signoff
```

or

```
signoff;
```

Starting Multiple SAS/CONNECT Sessions

An easy way to start-up multiple SAS/CONNECT sessions is by using your autoexec file to invoke multiple sessions when you start a local SAS session.

For example, if you want to connect to a TSO host by means of the APPC access method and a UNIX host by means of the TCP/IP access method, your autoexec file would contain the following:

```
   /*************************************/
   /* Start a TSO session.              */
   /*************************************/
options comamid=appc;
remote=LU_NAME;
signon;

   /*************************************/
   /* Start a UNIX session.             */
   /*************************************/
cscript='!sasroot\connect\saslink\tcpunix.scr';
options comamid=tcp;
remote=unxnode;
signon;
```

After your autoexec processing is complete, you can direct statements to either remote host from a single SUBMIT block in your local SAS session. You do this by preceding each section of statements with an RSUBMIT statement that specifies the remote host's session id. You end the block with an ENDRSUBMIT statement. For example, enter the following to remote-submit to the TSO host:

```
rsubmit LU_NAME;
   statements...

endrsubmit;
```

Enter the following to remote-submit to the UNIX host:

```
rsubmit unxnode;
   statements...

endrsubmit;
```

Issue the SIGNOFF command or submit the SIGNOFF statement when you are finished. Note that you must have the appropriate script file to sign off from each host. A simple, direct method for ensuring that you are using the correct script file when signing off multiple remote sessions is to define the fileref RLINK before each sign-off as in the following example.

Submit these statements to the local host:

```
signoff LU_NAME;
signoff unxnode cscript=
   '!sasroot\connect\saslink\tcpunix.scr';
```

For more information, see "SIGNON Command and Statement" on page 227, "SIGNOFF Command and Statement" on page 233, and "RSUBMIT Command and RSUBMIT Statement" on page 21.

The correct bibliographic citation for this manual is as follows: SAS Institute Inc., *SAS/CONNECT User's Guide, Version 8*, Cary, NC: SAS Institute Inc., 1999. pp. 537.

SAS/CONNECT User's Guide, Version 8

ISBN 1–58025–477–2

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

1st printing, September 1999