CHAPTER 33

OpenVMS Alpha Spawner Program

Starting the OpenVMS Alpha Spawner Program

As an alternative method to signing on through the TELNET daemon, the OpenVMS Alpha spawner program allows SAS/CONNECT sessions on OpenVMS Alpha systems without requiring that username and password pairs be passed over the network in clear text mode.

If the local SAS session is running Release 6.09E or a subsequent release or Release 6.11 TS040 or a subsequent release, all data that flows from the local host to the spawner program during sign on is encrypted.

Unlike the TELNET sign-on process, the spawner program allows sign ons to an OpenVMS Alpha system without scripts. You should run the spawner by using *syspriv* or bypass privileges in order to start the remote SAS processes with the privileges of the user who connects to the spawner.

The spawner program is stored on the remote host in the executables directory. An alias can be defined to point to the appropriate directory and executable for the spawner program by using the following DCL command:

```
SASTCPD:==$SAS$ROOT:[PROCS]SASTCPD.EXE
```

The spawner can then be invoked with options. For example:

```
SASTCPD -service spawnjob -nocleartext
```

Here is the complete syntax for the command to start the spawner program:

SASTCPD <-BACKGROUND>
<-SERVICE *service-name*>
<-HELP>
<-INHERITANCE>
<-NETENCRALG *algorithm*>
<-NETENCRYPT>
<-NETENCRKEY *n*>
<-NETMAC | -NONETMAC>
<-NOCLEARTEXT>
<-NOSCRIPT>
<-SASCMD *filename*>

<-SHELL>
<-USER>

-BACKGROUND

specifies that the spawner program run as a background process. The default specifies that the spawner program run in the foreground.

-SERVICE *service-name*

specifies the name of the service that the spawner program uses to listen for incoming requests. This value is identical to the *service* value in the REMOTE= option that the user specifies at the local host prior to sign on. Because there is no default, you must specify this value. See "Specifying the Remote Node Name" on page 177 for details.

The service name must be defined identically in the **/etc/services** file on both the local and remote hosts. See "Configuring the SERVICES File" on page 485 for more information about the **/etc/services** file.

-HELP

prints a list of valid parameters.

-INHERITANCE

causes the SAS session that is spawned to inherit the socket that was created when the spawner accepted the initial connection from the local SAS session . This option is useful if your configuration involves a firewall and you want to minimize the number of ports that you define to the firewall for use by SAS/CONNECT.

If you start a spawner with the -INHERITANCE option, you then define the port that the spawner is listening on to the firewall and map it to the server machine's port. This will enable any number of SAS/CONNECT clients to connect through this single port and SIGNON to a remote host on the inside of the firewall. Each client just opens a unique socket on the defined port. This eliminates the need to define an individual port for each client that may need to come in through the firewall. In this configuration you set your REMOTE= value to a two-level name where the first level is the name of the host running the firewall and the second level is the well-known service name of the port that you have enabled for connections.

-NETENCRALG *algorithm*

If you specify more than one algorithm, enclose the algorithm names in parenthesis and use commas to separate the names. If there are embedded blanks in the algorithm name, enclose each algorithm in quotation marks.

Set this option at the remote host and, optionally, at the local host to specify one or more encryption algorithms to use in a SAS/CONNECT session. However, the local host and the remote host must share an encryption algorithm in common. If you specify the option in the remote host session only, the local host attempts to select an algorithm that was specified at the remote host. If you also set the option at the local host and specify an algorithm that is not specified at the remote host, the local host's attempt to connect to that remote host fails when the local host assigns a library.

Valid values for this option are

RC2

RC4

DES

TripleDES

SAS Proprietary.

See the *SAS/CONNECT User's Guide* or the *SAS/SHARE User's Guide* for more information about the -NETENCRALG option.

-NETENCRYPT

Set this option at both the local host and the remote host. At the remote host, this option specifies that encryption is required for each connection from a local host SAS session. At the local host, this option specifies that the local host must connect only to a remote host that supports encryption.

The default for this option is that encryption is used if the -NETENCRALG option is set and if both the local host and the remote host are capable of encryption. If encryption algorithms were specified but either the local host or the remote host is incapable of encryption, then encryption will not be performed.

Encryption may not be supported at the local host or at the remote host for the following reasons:

- □ You are running a release of SAS (prior to Version 7) that does not support encryption.
- □ Your site has not purchased a SAS/SECURE license for a specific platform.
- □ You specified encryption algorithms in the local host and the remote host SAS sessions that are incompatible.
- □ You do not have a cryptographic service provider installed on your OpenVMS system.

See the *SAS/CONNECT User's Guide* or the *SAS/SHARE User's Guide* for more information about the -NETENCRYPT option.

-NETENCRKEY*n*

You set this option in either the local host or the remote host SAS session. It specifies the key length to be used by the encryption algorithm.

Valid values for this option are

128	specifies 1024-bit RSA and 128-bit RC2 and RC4 key algorithms.
40	specifies 512-bit RSA and 40-bit RC2 and RC4 key algorithms.
0	no value is set. This is the default.

If you require extra security, then set the -NETENCRKEY option to 128. If you prefer to save CPU, then set the -NETENCRKEY option to 40.

By default, if you try to connect a host that is capable of only a 40-bit key length with a host that is capable of both a 40-bit and a 128-bit key length, then the connection is made using the lesser key length. If both hosts are capable of 128-bit key lengths, then a 128-bit key length is used.

See the *SAS/CONNECT User's Guide* or the *SAS/SHARE User's Guide* for more information about the -NETENCRKEY option.

-NETMAC | -NONETMAC

Set this option to control the use of Message Authentication Codes (MACs) on network communications. A Message Authentication Code is the equivalent of a checksum that is used to ensure that the original message has not been modified.

This option may be set at either the local host or the remote host. The default is -NETMAC.

See the *SAS/CONNECT User's Guide* or the *SAS/SHARE User's Guide* for more information about the -NETMAC option.

-NOCLEARTEXT

prevents a sign on from a local host that does not support username and password encryption. This option prevents local hosts in a SAS session that are running releases prior to 6.09E and 6.11 TS040 from signing on to the spawner program. The default is to accept both encrypted and clear-text userids and passwords. This allows local hosts in a SAS session that are running releases prior to 6.09E and 6.11 TS040 to sign on to the spawner program.

-NOSCRIPT

prevents sign ons from local hosts that use scripts, and allows sign ons only from local hosts that do not use scripts.

This option requires that the user set security in the local SAS session during sign on. For details about setting security (for example, by means of the USER= and PASSWORD= options in an appropriate statement), see “Setting Security for SAS/CONNECT and SAS/SHARE” on page 171.

If you use the -NOSCRIPT option, you must also use the -SASCMD option.

-SASCMD *filename*

specifies the command that starts a SAS session when you sign on without a script. If the RLINK fileref is not defined in the local host SAS session, then the user is signing on without a script. In this case, the -SASCMD option must be specified.

Here is an example of the content of an executable file that starts a SAS session:

```
$!
$! mystartup
$!
sas /DMR/NOTERMINAL/NO$SYNTAXCHECK/DEVICE=GRLINK/COMAMID=TCP
$ exit
```

-SHELL

allows the SAS session that is invoked by the spawner program to create a shell. A shell is necessary for the remote host to execute commands.

-USER

allows the spawner program to run without root privileges. SAS processes assume the security status of the user or the administrator who started the spawner program. The default action is to assume the privileges of the user whose username and password are given to the spawner program.

Note: Because Open/VMS systems require group or system privilege in order to validate passwords, this option may not work unless the user has the correct privilege, or the user that starts the spawner passes his or her own uername and password to the spawner for access verification. △

Examples of Starting and Connecting to the OpenVMS Alpha Spawner Program

The following examples illustrate how to start the spawner program and how to connect to it.

Example 1:

The following command starts the spawner program at the remote OpenVMS Alpha host with the *service-name* VMSSPAWN and allows connections only from local hosts that support username and password encryption.

```
sastcpd -service vmsspawn -nocleartext
```

At the local host host, the following statements specify a script file named TCPVMS.SCR that makes a connection to the spawner program named MONARCH.VMSSPAWN. The value MONARCH for REMOTE= is the name of the OpenVMS Alpha node, or it can be a macro variable that contains the Internet address of the OpenVMS Alpha node where the spawner program is running.

```
options comamid=tcp;
options remote=monarch.vmsspawn;
```

```
filename rlink "$SAS$ROOT:[TOOLS]tcpvms.scr";
signon;
```

Example 2:

From the OpenVMS Alpha node that will be the remote side of a SAS/CONNECT session, the following command starts the spawner program with the service name VMSSPAWN, which supports only sign ons without scripts.

```
sastcpd -service vmsspawn -noscript
        -sascmd $SAS$ROOT:[TOOLS]mystartup.com
```

The MYSTARTUP file starts the remote SAS session. See the -SASCMD option for an example of the content of the executable file MYSTARTUP.

In the following example, the TCP/IP access method connects the local host to the remote host RMTHOST on which the spawner program VMSSPAWN runs. The value RMTHOST must be either the node name of the OpenVMS Alpha node or a macro variable that contains the Internet address of the OpenVMS Alpha node where the spawner program is running. The USER= option to SIGNON causes a prompt for userid and password during sign on.

```
options comamid=tcp;
signon user=_prompt_ rmthost.vmsspawn;
```

Ending the OpenVMS Alpha Spawner Program

To end the spawner program, enter the interrupt signal, which typically is CTRL-C. If the OpenVMS Alpha spawner is running in the background, kill its process.

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

The Institute is a private company devoted to the support and further development of its software and related services.