CHAPTER

2

CMS: APPC Access Method

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

Network Administrator, System Administrator, and User

To use the APPC access method with a CMS 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 APPC access method have been defined.
3 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 SAS software is installed on both the local and remote hosts.
2 To use the APPC access method between sessions on a VM host or another type of host, the following conditions apply:
   - □ You can communicate within your local VM/ESA system without additional software.
   - □ You can communicate between VM/ESA systems that are in either the same Transparent Services Access Facility (TSAF) collection or the same Communication Services (CS) collection.
   - □ You can communicate with systems in an SNA network if you have installed the Advanced Communication Facility for the Virtual Telecommunications Access Method (ACF/VTAM), the Group Control System (GCS), and the APPC/VM VTAM Support (AVS).

Note: For SAS/CONNECT only, you will need to manage SNA session limits to use the APPC access method in an SNA network. For more information about setting up an SNA network (including setting session limits), see “System Configuration for the APPC Access Method for SAS/CONNECT” on page 29. △

Defining Resources for the APPC Access Method

Network Administrator

APPC is an IBM strategic enterprise connectivity solution. Based on a System Network Architecture (SNA) logical unit type 6.2 (LU 6.2), APPC is the foundation for distributed processing within an SNA network. In this book, APPC is used to refer to the SNA LU 6.2 distributed processing method.

Before you can use SAS/CONNECT or SAS/SHARE with the APPC access method, you must first define APPC resources for the CMS system. This enables CMS to behave as either a local or a remote host in a SAS/CONNECT session or as a SAS/SHARE server or client. See “System Configuration for the APPC Access Method for SAS/CONNECT” on page 29 for SAS/CONNECT resource configuration. See “System Configuration for the APPC Access Method for SAS/SHARE” on page 39 for SAS/SHARE resource configuration.

Setting SAS Options

To use the APPC access method with SAS/CONNECT and SAS/SHARE, you may need to set specific options.

You may specify an option in any of several forms, as follows:

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

  OPTIONS *variable-name*=*value*;

  Example:

  ```
  options appcsec=_secure_;
  ```

- □ in a SAS configuration file or at SAS invocation: *variable-name*=*value*

  Example:

  ```
  appcsec=_secure_
  ```

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

OPTIONS statement

AUTOEXEC file

SAS invocation

SAS configuration file

Setting Security for SAS/CONNECT and SAS/SHARE

There are several methods for supplying userid and password information for SAS/CONNECT and SAS/SHARE. They are:

- □ USER= and PASSWORD= options in selected statements
- □ APPCSEC option
- □ SCOMDIR NAMES or UCOMDIR NAMES file
- □ APPCPASS statement.

The person who maintains the userid and password information varies according to the method used.

For SAS/CONNECT, you must supply identifying information to sign on without a script to a remote host running a spawner program. A SAS/SHARE server, running secured, requires identification from each connecting client. The next two sections outline the version-specific methods for specifying client identification for SAS/CONNECT and SAS/SHARE.

Providing Client Identification in a Version 8 Session

In Version 8, you provide client identification to a SAS/CONNECT remote host or a SAS/SHARE server using the USER= and PASSWORD= options. These options are valid in the following statements:

SIGNON

RSUBMIT

LIBNAME

PROC SQL
Connect to Remote

PROC OPERATE
(in the PROC statement)
set server
stop server
quiesce server
start server
display server

Specifying client identification in the APPCSEC option is still accepted but is not recommended in Version 8. The USER= and PASSWORD= options take precedence over the client APPCSEC option when both are specified. For example, a SAS/SHARE client's execution of a LIBNAME statement with values assigned to the USER= and PASSWORD= options would override an APPCSEC option setting in the same client SAS session.

CAUTION:
In order to make a SAS/SHARE server secured, the APPCSEC option must still be set at a SAS/SHARE server that can run on a supported host. △

Here is the syntax and the definitions for these options:

USER | **USERNAME** | **USERID** | **UID**=*username* | _PROMPT_

PASSWORD | **PASSWD** | **PASS** | **PWD** | **PW**=*password* | _PROMPT_

Specifying these options allows a user on the local host whose username and password have been verified to access the remote host.

username
is a valid userid for the remote host and is thus host-dependent in form. If the value contains blanks or special characters, it must be enclosed in quotes.

password
is the password, if any, required for authentication of the supplied username. This value will not be echoed in the SAS log. If the value contains blanks or special characters, it must be enclosed in quotes.

PROMPT
specifies that the SAS System prompts the client for *username* and *password*.

Note: The values provided when prompted must NOT be quoted. △

Specifying USER=_PROMPT_ and omitting the PASSWORD= specification will cause SAS to prompt you for both userid and password.

This is especially useful for allowing the SAS statements containing the USER= and PASSWORD= options to be copied and otherwise effectively reused by others.

For SAS/SHARE, the values supplied for the USER= and PASSWORD= options are valid for the duration of the remote host connection. Additional accesses of the remote host while the connection to that host is still in effect do not require re-supplying of the USER= and PASSWORD= options. For example, while the first connecting library assign to a SAS/SHARE server may require specification of the options, subsequent assigns to the same server will not need specification of these options as long as the original connection is in effect. A subsequent re-connect to the same server or connect to a different server would require re-supplying of the USER= and PASSWORD= options.

Here is a Version 8 example for SAS/SHARE:

```
libname test 'prog2 a' user=joeblue password="2muchfun" server=share1;
```

Here is a Version 8 example for SAS/CONNECT:

```
signon rmthost user=joeblack password=born2run;
```

As a security precaution, PASSWORD= field entries echoed in the log are replaced with Xs. If _PROMPT_ was specified for entering the password, the entry would not be displayed on the screen as it is typed.

Providing Client Identification in a pre-Version 8 Session

In Version 6 and 7, the APPCSEC option is used to specify how users are authenticated when connecting between hosts using the APPC access method. On the local host, you may set the APPCSEC option to allow local hosts or clients whose userids and passwords have been verified to access a SAS/CONNECT remote host or a SAS/SHARE server. On the remote host, you must specify the APPCSEC option before you start a server.

The valid values for the APPCSEC option are:

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

NONE

has different meanings, depending on whether it is set at the local host or the remote host.

SAS/CONNECT local host or at the SAS/SHARE client

NONE specifies that the userid and password are to be obtained from the UCOMDIR NAMES, SCOMDIR NAMES, or APPCPASS CP directory entries instead of from the APPCSEC option. _NONE_ is the default.

SAS/CONNECT remote host or at the SAS/SHARE server

NONE specifies an unsecured remote host, which does not require the local host to supply a verified userid and password.

PROMPT

must be set at the SAS/CONNECT local host or at the SAS/SHARE client.

PROMPT specifies that SAS prompt the user for userid and password information. If the communications directory file entry contains SECURITY.NONE, no prompting is performed.

When prompted for a userid, if you press the ENTER key without supplying one, then SAS uses the local userid. The userid is not obtained from UCOMDIR NAMES, SCOMDIR NAMES, or an APPCPASS CP directory statement as it is when _NONE_ is specified.

When prompted for a password, the input field is not displayed. If you press the ENTER key without supplying a password, one is obtained from UCOMDIR NAMES, SCOMDIR NAMES, or an APPCPASS CP directory statement. The behaviors of the _PROMPT_ and _NONE_ values are different.

userid.password

must be set at the SAS/CONNECT local host or at the SAS/SHARE client.

This value optionally specifies the userid and the password. If you do not specify a userid, SAS uses the local userid. The userid is not obtained from UCOMDIR NAMES, SCOMDIR NAMES, or an APPCPASS CP directory statement as it is when _NONE_ is specified.

SECURE
: must be set at the SAS/SHARE server only.

The _SECURE_ value for the APPCSEC option requires the SAS/SHARE client to supply a valid userid and password to the remote host on which the server is running in order to allow client access to the server.

APPCSEC is maintained by the user. If you assign the *userid.password* or *password* to the APPCSEC option and store the option in a disk file, you should make the file secure, for example, by using a read password on the disk. If you are running SAS/CONNECT or SAS/SHARE interactively, you can assign the userid and password to the APPCSEC option without a need for file security.

If you assign _PROMPT_ to the APPCSEC option, the *userid* and *password* cannot be revealed by writing it to either SASLOG or a console spool file.

You may use the APPCSEC option as the means to override the userid and password information in the UCOMDIR NAMES or SCOMDIR NAMES file, or the APPCPASS statement.

SCOMDIR NAMES or UCOMDIR NAMES File

The SCOMDIR NAMES or UCOMDIR NAMES file can be used to specify userid and password security information. For more information about storing the userid and password in either of these files, for SAS/CONNECT, see "Creating a Communications Directory File" on page 30; for SAS/SHARE, see "Creating a User Communications Directory File" on page 42.

The UCOMDIR NAMES file is maintained by the user. If you store passwords in the file you should secure it, for example, by using a disk password.

For information about the UCOMDIR NAMES file, see "System Configuration for the APPC Access Method for SAS/CONNECT" on page 29. For information about the SCOMDIR NAMES file, see "System Configuration for the APPC Access Method for SAS/SHARE" on page 39.

APPCPASS Statement

The APPCPASS statement is used to specify userid and password security information in the local user's CP directory. See the IBM publication *VM/ESA Connectivity Planning Administration and Operation (SC24-5448)* for more information about APPCPASS.

The system administrator maintains an APPCPASS statement for each userid. It is secure because users must have privileged authority to access the CP directories of other users.

SAS/CONNECT

Local Host Tasks

User or Applications Programmer

To connect a CMS local host to a remote host, perform these tasks:

1. Set security for local hosts.
2. Specify the communications access method.
3. Specify the remote host name.
4. Sign on to the remote host.

Setting Security for Local Hosts

Set security using either of the methods explained in "Setting Security for SAS/CONNECT and SAS/SHARE" on page 23. For Version 8 security behavior, specify the USER= and PASSWORD= options in the SIGNON statement. For details, see "Providing Client Identification in a Version 8 Session" on page 23.

For Version 7 security behavior, if you set the APPCSEC option at the local host, either specify a userid and a password that are valid on the remote host or specify PROMPT to supply the userid and password when connecting to a remote host. For information about setting the APPCSEC option, see "Providing Client Identification in a pre-Version 8 Session" on page 25.

Specifying the APPC Communications Access Method

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

OPTIONS COMAMID=*access-method-id*;

access-method-id identifies the method used by the local host to communicate with a remote host. APPC (an abbreviation for Advanced Program-to-Program Communication) is an example of *access-method-id*.

Example:

```
options comamid=appc;
```

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

Specifying the Remote Host Name

To connect a CMS local host to a remote host, use the following syntax:

OPTIONS REMOTE=*remote-session-id*;

where *remote-session-id* specifies an entry in a communications directory file.

The following example shows a connection to an MVS/ESA remote host. The remote node N02SV01 is the LU name that is defined in a communications directory file.

Example:

```
options remote=N02SV01;
```

Alternatively, you may set this option at a SAS invocation or in the 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 user=_prompt_;
```

To set security at the remote host, specify valid values for the USER= and PASSWORD= options in the SIGNON statement. For details, see "Providing Client Identification in a Version 8 Session" on page 23.

You do not need to use a script file because APPC has the ability to interface with the APPC/CMS subsystem to initiate a remote session. If you previously identified a script file in an RLINK fileref statement, you will receive an error message when you attempt to make a connection. If you do not want to omit the RLINK fileref but want to prevent the error, 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 CMS local host SAS session to connect to a remote host with the APPC access method.

```
options comamid=appc remote=remotelu;
signon user=_prompt_;
```

The APPC communications access method is declared with a connection to a remote host that is identified by the LU name REMOTELU. The SIGNON statement performs the sign-on process to the remote host. The USER= option to SIGNON specifies that the connecting local host be prompted for a userid and a password that are valid on the remote host.

Remote Host Tasks

System Administrator

To allow a connection from a local host, perform this task at the remote host:

1 Specify the remote host name.

2 Optionally, set several remote host options.

Specifying the Remote Host Name

You must declare a remote host name at the local host and the remote host in a SAS/CONNECT session. At both hosts, specify an OPTIONS statement. Use the following syntax:

```
OPTIONS REMOTE=remote-host-id;
```

where the *remote-host-id* that you specify at the remote host is based on the type of remote host that you are connecting to.

The remote host identifiers that you specify at both the local and the remote hosts must be identical.

Example:

```
options remote=remotelu;
```

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

Setting Options at the Remote Host

Although sign-on script files are not used for the APPC access method, you may set these options at the remote host:

NO$SYNTAXCHECK

allows the continuation of statement processing at the remote host regardless of syntax error conditions.

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

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

Remote Host Example

The following example illustrates the statements that you specify in a CMS remote host's configuration file to prepare for a connection from a local host to a remote host with the APPC access method.

```
dmr
comamid=appc
remote=remotelu
no$syntaxcheck
noterminal
```

The APPC communications access method is declared with a connection to a remote host that is identified as the LU name that is configured to the name of the AVS private gateway. In this example, REMOTELU identifies the AVS private gateway.

System Configuration for the APPC Access Method for SAS/CONNECT

VTAM Systems Personnel

Configure CMS userids for the CMS system that enable it to behave as either a local or a remote host in a SAS/CONNECT session when using the APPC access method.

Perform the following tasks to configure CMS userids to use with the APPC access method:

1. At a CMS local host, create a communications directory.
2. At a CMS remote host, set up a $SERVER$ NAMES directory.
3. At a CMS remote host, edit the PROFILE EXEC file, as necessary.
4. At both a CMS local host and a remote host, define a VTAM gateway.
5. At both a CMS local host and a remote host, define the logon mode table entries.
6. At a CMS local host, set session limits and contention values.

This section highlights the general tasks that you must perform to configure the system to use with the APPC access method. For full details about configuring the APPC access method, see “References” on page 35.

Creating a Communications Directory File

To connect a CMS local host to a remote host, create a communications directory file for the connecting user. The file contains an entry that is used for the value of the SAS option REMOTE=, at the local host.

The communications directory file can reside at the system level, the user level, or both. The default system-wide communications directory file is named SCOMDIR NAMES, and the default user communications directory file is named UCOMDIR NAMES. The format of a CMS communications directory entry follows:

```
:NICK.LU-name   :LUNAME.gateway target-LU
                :TPN.SASRMT
                :MODENAME.modename
                :SECURITY.level
                :USERID.userid
                :PASSWORD.password
```

where

NICK.*LU-name*
: is short for nickname. It specifies the eight-character symbolic destination name of the resource.

You will use *LU-name* as the value for the REMOTE= option at the local host.

A CMS user cannot take advantage of the aliasing support that is implicit in the CMS communications directory structure when accessing a host that is not a CMS host. Instead, the NICK value in the communications directory must be identical to the *target LU* value in the :LUNAME definition.

LUNAME.*gateway target-LU*
: is composed of two eight-character names. The first name defines the name of the gateway for connections outside the TSAF collection; the second is the name of the partner LU.

For connections within the TSAF collection the first name can be USERID and the second name the remote host userid.

TPN.SASRMT
: indicates the transaction program name as it is known to the target LU. For SAS/CONNECT, the transaction program name is always SASRMT.

MODENAME.*modename*
: specifies the mode name of the SNA session that connects the gateway to the target LU.

SECURITY.*level*
: specifies whether to use security in a SAS/CONNECT session. The two values for *level* are PGM, which indicates program security, or NONE. Specify PGM if the server is running secured (requiring a userid and password from each connecting user). Specify NONE if the server is running unsecured (not requiring a userid and password from each connecting user). See “Setting Security for SAS/CONNECT and SAS/SHARE” on page 23 for more information about userid and password security.

USERID.*userid*
: indicates the access security userid that is presented to the target LU for verification.

PASSWORD.*password*
: indicates the access security password that is presented to the target LU.

Note: You can omit specifying the USERID. **userid** and PASSWORD. **password** parameters, if you do one of the following:

□ use the Version 8 USER= and PASSWORD= options

□ add a fully qualified APPCPASS statement to your CP directory

□ assign _PROMPT_ to the APPCSEC option.

Both the APPCPASS statement and the APPCSEC option are more secure methods for presenting userid and password information to a target LU than the USERID and PASSWORD parameters. Refer to *VM/ESA Connectivity Planning, Administration and Operation (SC24-5448)* for more details about the APPCPASS statement. △

Examples:

The first example allows a CMS-to-CMS connection. In this example, the NICK value (CONNVM) is different from the *target-LU* value (N01SASPG).

```
:NICK.CONNVM   :LUNAME.N01SASOG N01SASPG
               :TPN.SASRMT
               :MODENAME.SASAPPC
               :SECURITY.PGM
               :USERID.bass
               :PASSWORD.time2go
```

The next example allows a CMS-to-OS/390 connection. In this example, the NICK value (N01TGT62) is identical to the *target-LU* value (N01TGT62).

```
:NICK.N01TGT62 :LUNAME.N01SASOG N01TGT62
               :TPN.SASRMT
               :MODENAME.SASAPPC
               :SECURITY.PGM
               :USERID.bass
               :PASSWORD.time2go
```

Creating a $SERVER$ NAMES Directory

To connect to a CMS remote host, you must create the $SERVER$ NAMES directory at the remote CMS host. This directory defines the EXEC to be run when the connection is made to the remote CMS system. The format of a CMS $SERVER$ NAMES directory entry follows:

```
:NICK.SASRMT   :LIST.userid1 ...  useridn|.*
               :MODULE.exec-name
```

where

NICK.SASRMT
: specifies the eight-character symbolic destination name of the resource. SASRMT is the transaction program name as it is known to the target LU. For SAS/CONNECT, the transaction program name is always SASRMT.

LIST.*userid1 ... useridn*
: enables you to limit the number of users that are allowed to connect to this system.

LIST.*
: specifies that all users are allowed to connect to this system.

MODULE.*exec-name*
: specifies the EXEC to be run when a connection is made to the remote CMS system. Specifying this module eliminates the need for a sign-on script. The

primary purpose of this EXEC is to invoke the remote SAS session with the SAS options that you want.

A sample $SERVER$ NAMES file follows:

```
 $SERVER$ NAMES
:NICK.SASRMT   :LIST.*
               :MODULE.RMTBOOT
```

The RMTBOOT EXEC that was specified in the previous example might be structured as follows:

```
/*   This is the BOOTSTRAP EXEC */
/*   for the remote CMS host */
say 'Remote Bootstrap in Progress'
say 'Invoking the SAS System     '
'EXEC SAS (COMAMID=APPC DMR REMOTE=N01SASPG
           NOTERMINAL NO$SYNTAXCHECK)'
queue 'CP logoff'
exit
```

Note: The SAS options COMAMID, DMR, and REMOTE that are included in this example are required to invoke SAS/CONNECT on the remote host. △

Customizing the PROFILE EXEC File

You must ensure that the appropriate CMS SET commands are specified in the remote CMS virtual machine to allow the virtual machine to function as a remote host. Edit the PROFILE EXEC file on the remote CMS userid and add the following commands:

```
/*   Make sure that we are set up to accept */
/*   connections if we get autologged       */
if substr(diagrc(24,-1),11,1) = '2' then do
   'SET SERVER ON'
   'SET FULLSCREEN OFF'
   'SET AUTOREAD OFF'
end
```

Because these commands are part of the PROFILE EXEC file, these CMS commands are automatically issued during login if the CMS userid is being autologged through a SAS/CONNECT sign on.

Defining a VTAM Gateway

A VTAM gateway provides a path for local host users to reach desired remote hosts. Use APPL statements to define separate VTAM gateways for

□ a CMS local host connecting to the desired remote host

□ a CMS remote host to which a local host connects.

To connect a CMS local host to a remote host, you must define local-domain VTAM application minor node identifiers (outbound gateway) using APPL statements.

Use a comma to separate each entry.

APPL statements for a CMS local host follow:

```
N01SASOG  APPL   ACBNAME=N01SASOG,

                APPC=YES,
                AUTHL=(ACQ),
                AUTHEXIT=YES,
                AUTOSES=0,
                DLOGMOD=mode-table-entry,
                DMINWNL=16384,
                DMINWNR=0,
                DSESLIM=32767,
                EAS=30,
                MODETAB=mode-table,
                PARSESS=YES,
                SECACPT=CONV,
                SONSCIP=YES,
                VPACING=n
```

To connect to a VM/CMS system, you must define a non-dedicated private gateway to provide SAS/CONNECT users with a pathway for reaching the VM/CMS system. APPL statements for a CMS remote host follow:

```
N01SASPG  APPL   ACBNAME=N01SASPG,

                APPC=YES,
                AUTHL=(ACQ),
                AUTHEXIT=YES,
                AUTOSES=0,
                DLOGMOD=mode-table-entry,
                DMINWNL=0,
                DMINWNR=16384,
                DSESLIM=32767,
                EAS=30,
                MODETAB=mode-table,
                PARSESS=YES,
                SECACPT=CONV,
                SONSCIP=YES,
                VPACING=n
```

Note: The only differences between the two sets of APPL statements are the values assigned to the DMINWNL and DMINWNR parameters. △

An explanation of each entry follows:

ACBNAME
: defines the minor node name assigned to this application program.

APPC=YES
: tells APPC that the application program can issue APPCCMD macros.

APPL
: declares an APPL definition statement.

AUTH=(ACQ)
: enables the application to acquire a session with a particular logical unit.

AUTHEXIT=YES
: allows the application's exit routines to run in supervisor state.

AUTOSES=0
defines the number of contention winner sessions to activate automatically.

DLOGMOD=*mode-entry*
defines the default session parameter mode table entry.

DMINWNL=*x*
specifies the initial negotiation value for local contention winner sessions.

DMINWNR=*y*
specifies that the remote partner's contention winner sessions request be used.

DSESLIM=32767
defines the maximum session limits.

EAS=30
specifies the estimated number of sessions that will be active with this logical unit at any given time.

MODETAB=*mode-table*
defines the session logon mode table.

PARSESS=YES
allows multiple concurrent sessions with another application program.

SECACPT=CONV
indicates that the FMH5 security subfield information is accepted.

SONSCIP=YES
allows the application to receive UNBIND RUs in its SCIP exit routine.

VPACING=*n*
sets network requirements per site.

See *VTAM Installation and Resource Definition (SC23-0111)* for more information about the VTAM gateway parameters.

Defining Logon Mode Table Entries

A logon mode table contains one or more sets of session properties that support session binding to a secondary LU that resides within the local VTAM domain.

Refer to the BIND RU description in *Technical Reference 3, SNA Formats* and the MODEENT discussion in *VTAM Resource Definition Reference (SC32-6412)* for complete information.

The following sample logon mode table entry contains a single set of session properties.

```
SASAPPC MODEENT  LOGMODE=SASAPPC,
                 FMPROF=X'13',
                 TSPROF=X'07',
                 PRIPROT=X'B0',
                 SECPROT=X'B0',
                 COMPROT=X'50B1',
                 RUSIZES=X'xxxx',
                 PSERVIC=X'060200000000000000102F00',
                 TYPE=0
```

Setting Session Limits and Contention Values

To use the SNA network for your APPC communications, you must use the CNOS (Change Number of Sessions) command to increase the session limits to greater than 0 between the AVS outbound gateway and the appropriate partner LU.

Note: CNOS is a privileged command. △

Because the IBM APPC/VM implementation does not support a programming interface to this command, SAS cannot issue it automatically. Instead, you must enter control information at an AVS console. Alternatively, you may supply this information through the Programmable Operator Facility (PROP) that is included with the CMS system.

The CNOS command remains in effect while the remote host remains in operation until the system has been rebooted. If you do not re-issue the command following a system reboot, communication between the two partner LUs cannot proceed.

It is recommended that you routinely re-issue the CNOS command each time the remote system is restarted. Re-issuing the CNOS command enables CMS users to gain access to the remote system at any time without operator intervention.

The command format for the implementation of the CMS CNOS command follows:

PROP AGW CNOS *gateway remote-LU modenameses-limit con-win con-lose*

The parameters for the CNOS command are

gateway
: specifies the name of the local LU that is the gateway.

remote-LU
: specifies the name of the remote LU for which the session limits are set.

modename
: specifies the logon mode name for which the session limit and contention values are changed.

ses-limit
: specifies the maximum number of LU-to-LU sessions that are allowed between the gateway LU and the remote LU for the logon mode name.

con-win
: specifies the number of parallel sessions for which the gateway LU is guaranteed to be the contention winner.

con-lose
: specifies the number of parallel sessions to which the remote LU is guaranteed to be the contention winner.

Example:

```
PROP AWG CNOS N01SASOG N01TGT62 NO1MOD1 100 50 50
```

Appropriate values for your site are based on the number of simultaneous users of the gateway. A general recommendation for the CNOS *ses-limit* value is to allocate three sessions per userid that will simultaneously use the gateway.

You have completed the remote and local CMS host configuration procedure for SAS/CONNECT.

References

For complete details about how to install and configure the system for use with the APPC access method, see the following IBM publications:

VM/ESA Connectivity Planning, Administration, and Operation (SC24-5448)

SNA Technical Overview (GC30-3073)

SNA Formats (GA27-3136)

VTAM Programming for LU6.2 (SC30-3400)

Contact IBM for information about obtaining this documentation.

SAS/SHARE

Client Tasks

System Administrator and User

To prepare to access a SAS/SHARE server, perform the following tasks:

1. Set security for connecting clients.
2. Specify the APPC access method.
3. Specify a server name.

Setting Security for Connecting Clients

Requiring connecting clients to supply a valid userid and password enforces server security. At the client, use the preferred security method for specifying a userid and a password that are valid on the server host. For details, see “Setting Security for SAS/CONNECT and SAS/SHARE” on page 23.

Specifying the APPC Access Method

You must specify the APPC communications access method at the client before you access a server.

Use the following syntax to specify the APPC access method at each connecting client:

```
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. APPC (an abbreviation for Advanced Program-to-Program Communication) is an example of an *access-method-id*.

Example:

```
options comamid=appc;
```

The server is accessed using the APPC 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 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 access methods that are supported between client and server hosts.

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

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

An example of configuration file entries for a CMS client connecting to a CMS server follows.

Example:

```
comamid=appc
comaux1=tcp
comaux2=iucv
```

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

Specifying a Server Name

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

```
SERVER=server-id
```

where *server-id* is defined in the communications directory file when configuring the CMS system for use with the APPC access method. See “Creating a System Communications Directory File” on page 40 for details about defining a *server-id* for the server.

See *SAS Language Reference: Dictionary* for details about SAS naming rules. See *SAS/SHARE User's Guide* for details about the PROC OPERATE and LIBNAME statements.

Client Example

The following example illustrates the statements that you specify in a CMS client SAS session to connect to a server with the APPC access method:

```
options comamid=appc;
libname sasdata 'prog2 a' user=_prompt_ server=share1;
```

The COMAMID option specifies the APPC access method. The LIBNAME statement specifies the name of the data library that is accessed through the server SHARE1 by means of a prompt for a username and a password that are valid on the server.

Server Tasks

System Administrator

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

1 Set server security.

2 Specify the APPC access method.

3 Specify the server name.

Setting Server Security

You may use file permissions to restrict a user's access to libraries and files through a server. A secured server allows connections only from those clients that provide valid userids and passwords for the host at which the server is running. A secured server uses a validated userid and password pair to verify a user's authority to access a SAS library or a file. You must provide a user exit to verify authority to access a SAS library or file. This user exit is optional. The default is to allow access to all files to any client that provides a valid userid and password. Documentation for user exits is provided in the CMS Installation Guide. For details about setting security, see "Setting Security for SAS/CONNECT and SAS/SHARE" on page 23.

Specifying the APPC Access Method

You must specify the APPC communications access method at the server before you create a SAS/SHARE server.

Use the following syntax to specify the APPC 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. APPC (an abbreviation for Advanced Program-to-Program Communication) 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=appc;
```

The server will be available only to SAS/SHARE sessions that use the APPC 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 is running supports multiple access methods, you may specify up to two auxiliary access methods by which clients may access the server. 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 options follows:

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

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

```
comamid=appc
comaux1=tcp
comaux2=iucv
```

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

See *SAS/SHARE User's Guide* for details about starting and accessing a server.

Specifying a Server Name

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

SERVER=*server-id*

where *server-id* is defined in the VM directory entry when configuring the CMS system for use with the APPC access method. See "Creating the Server VM Directory Entry for the Server Virtual Machine" on page 39 for details about defining a *server-id* for the server.

See *SAS Language Reference: Dictionary* 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 SAS session on the CMS host at which you start a server:

```
options appcsec=_secure_ comamid=appc;
proc server id=share1;
run;
```

The _SECURE_ value for the APPCSEC option requires clients to supply a userid and a password that are valid on the server. The APPC access method is declared and a server with the *server-id* SHARE1 is started on the CMS host.

System Configuration for the APPC Access Method for SAS/SHARE

VTAM Systems Personnel

To configure resources for the CMS system that will enable it to behave as either a SAS/SHARE server or a client using the APPC access method, perform the following tasks:

1 At a CMS server, create a VM directory entry for the server's virtual machine.
2 At a CMS client, create an entry in the VM directory for each user who will access the server.
3 At a CMS client, modify a system communications directory file with an entry for each server that users will access.
4 At a CMS client, set security for connecting clients.
5 At both a CMS client and a server, define a VTAM gateway.
6 At both a CMS client and a server, define logon mode table entries.

Creating the Server VM Directory Entry for the Server Virtual Machine

You must include the following statements in the VM directory entry for the server virtual machine:

```
USER SASSHARE XXXXXXX 20M 20M G 100
MACHINE XA
IPL CMS PARM AUTOCR
OPTION MAXCONN 1024
```

```
IUCV *IDENT SHR1 GLOBAL
IUCV ALLOW
CONSOLE 009 3215
SPOOL   00C 2540 READER *
SPOOL   00D 2540 PUNCH A
SPOOL   00E 1403 A
LINK MAINT 19E 19E RR
LINK MAINT 19D 19D RR
LINK MAINT 190 190 RR
MDISK 191 3380 707 5 VM0800 MR XXXXXXX
MDISK 192 3380 501 3 VM0450 MR XXXXXXX
```

Several lines that are specific to the APPC access method are explained in more detail.

```
IUCV *IDENT SHR1 GLOBAL
IUCV ALLOW
OPTION MAXCONN 1024
```

The first line specifies the name of the server that will run in the virtual machine. The server name, shown as SHR1 in this example, is specified by the SERVER= option of the SERVER procedure in the SAS program that creates the server. The *server-id* is typically the name of the server virtual machine (the VM userid). If you specify *server-id* as RESANY, any valid server name can be specified for the SERVER= option.

The second line allows users to establish IUCV connections to the server virtual machine.

The third line specifies the maximum number of simultaneous connections to the server that you want to allow. Generally, you should allow four to five connections for each user who accesses data through the server. The number shown is only a guideline. The default MAXCONN value is 64. The maximum MAXCONN value is 65535.

SAS/SHARE does not specifically limit the number of simultaneous connections to a server.

Modifying an Entry in the VM Directory for Each User

Modify the VM directory entry by including the following statement in the VM directory entry for each user who will access a server:

```
OPTION MAXCONN 128
```

This statement specifies the maximum number of simultaneous connections to a server that you want to allow. Generally, you should allow four to five connections for each server. The default MAXCONN value is 64. The maximum MAXCONN value is 65535.

Creating a System Communications Directory File

You must create a system communications directory file with an entry for each server that your users will access. The system communications directory file is named SCOMDIR NAMES by default. It should reside on a system minidisk that is accessible to all CMS users.

Specify an entry for a server within the TSAF collection in the following form:

```
:NICK.server-id :LUNAME.*IDENT
                :TPN.server-id
                :SECURITY.level
                :MODENAME.modename
```

where

server-id
: is the *server-id* as specified in the IUCV *IDENT VM directory entry record for the server virtual machine.

level
: specifies whether to use security in a SAS/SHARE session. The two values for *level* are PGM, which indicates program security, or NONE. Specify PGM if the server is running secured (requiring a userid and password from each connecting user). Specify NONE if the server is running unsecured (not requiring a userid and password from each connecting user). See "Setting Security for SAS/CONNECT and SAS/SHARE" on page 23 for more information about userid and password security.

Specify an entry for a server outside the TSAF collection in the following form:

```
:NICK.server-id :LUNAME.gateway server-id
                :TPN.SASTP62
                :MODENAME.modename
                :SECURITY.level
```

where

server-id
: is the name of the LU for the server.

gateway
: is the global gateway to the SNA network.

TPN.SASTP62
: indicates the transaction program name as it is known to the target LU.

modename
: is the communications mode for the SNA session that connects the gateway to the LU that is defined for the server.

level
: specifies whether to use security in a SAS/SHARE session. The two values for *level* are PGM, which indicates program security, or NONE. Specify PGM if the server is running secured (requiring a userid and password from each connecting user). Specify NONE if the server is running unsecured (not requiring a userid and password from each connecting user). See "Setting Security for SAS/CONNECT and SAS/SHARE" on page 23 for more information about userid and password security.

See the IBM publication *VM/ESA Connectivity Planning, Administration, and Operation (SC24-5448)* for more information about creating and processing communications directories. Contact IBM for information about obtaining this documentation.

Setting Security for Connecting Clients

Each user who connects to a server that is running in secured mode must specify a userid and a password that are valid on the system on which the server is running. A secured server requires a userid and password from each user, which it validates on the system where it is running.

You can specify a user's userid and password for the server's system in any of the following:

- □ an APPCPASS statement in the user's VM directory
- □ a user communications directory file (UCOMDIR NAMES) on the user's A-disk
- □ the SAS option APPCSEC
- □ the USER= and PASSWORD= options.

Specifying an APPCPASS VM Directory Statement

You can specify both a user's userid and password for the server's system in an APPCPASS statement in the user's VM directory as follows:

```
APPCPASS LU-name userid password
```

where

LU-name
specifies the gateway and server identifier that is defined in the :LUNAME specification in the communications directory. Gateway and server identifier names are restricted to eight characters each.

userid
specifies a user's userid for the system where the server is running.

password
specifies the password for the userid.

Creating a User Communications Directory File

You can specify both a user's userid and password for the server's system in a user communications directory file on the user's A-disk. The file, named UCOMDIR NAMES by default, should contain an entry for each secure server that the user will connect to. An entry in a user communications directory file has the same format as one in a system communications directory file with the addition of these two fields:

```
:USERID.userid
:PASSWORD.password
```

This method is less secure than an APPCPASS statement because any other user who can read the UCOMDIR NAMES file can obtain the user's userid and password for the server's system. You can limit this exposure by restricting access to the file, the minidisk, or the SFS directory in which the UCOMDIR NAMES file resides (for example, by putting a read password on the minidisk where the file resides).

A user can perform this procedure, thereby, eliminating a system administrator's support. Users can modify their own communications directories if they subsequently change passwords for the server's systems. This procedure also requires that users maintain other fields such as *server-id*, *gateway*, and *modename*.

See the IBM publication *VM/ESA Connectivity Planning, Administration, and Operation (SC24-5448)* for more information about the creation and processing of communications directories. Contact IBM for information about obtaining this documentation.

Defining a VTAM Gateway

See "Defining a VTAM Gateway" on page 32 for information about defining a VTAM outbound gateway (N01SASOG) for CMS clients to reach specific servers, as well as for defining a VTAM inbound gateway (APPL statements similar to N01SASPG) for connecting inbound to a specific CMS server.

Defining Logon Mode Table Entries

See “Defining Logon Mode Table Entries” on page 34 for information about setting up a logon mode table to contain session properties.

References

See “References” on page 35 for a list of documentation references.

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.