CHAPTER

6

# OS/390: 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 an OS/390 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. Verify that you know how to set options in 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. APPC has been installed at your site.
2. SAS is installed on both the local and remote hosts.
3. VTAM, Version 3, Release 2 or a subsequent release is running on your system.

## Defining Resources for the APPC Access Method

*Network Administrator*

APPC is the IBM strategic enterprise connectivity solution. Based on 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, the term 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 OS/390 system which allow it 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 93 and “System Configuration for the APPC Access Method for SAS/SHARE” on page 102 for details about resource configuration.

### Understanding APPC Access Method Terminology

Familiarity with these terms will help you when you talk to your network administrator about SAS option settings.

LU (logical unit)

is a device or program by which an end user (LU6.2 applications program) gains access to an APPC subsystem.

LU pool

is a group of LUs from which a single LU is dynamically selected and assigned to a requesting local host in a SAS/CONNECT session or to a requesting client in a SAS/SHARE session. Pool size is determined by the setting of the LUFIRST and LULAST options (see “SAS/CONNECT and SAS/SHARE Options” on page 88 for information about these options). See “System Configuration for the APPC Access Method for SAS/CONNECT” on page 93 and “System Configuration for the APPC Access Method for SAS/SHARE” on page 102 for details about defining properties of the LU pool.

## Setting SAS Options

You may need to set specific SAS options to establish the connections that you want with SAS/CONNECT and SAS/SHARE when using the APPC 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 several forms:

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

OPTIONS *option-name=value*;

Example:

```
options appcsec=_secure_;
```

□ in a SAS configuration file or at SAS invocation

*option-name=value*

Example:

```
appcsec=_secure_
```

Values for these options can 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:

OPTIONS statement

AUTOEXEC file

SAS invocation

SAS configuration file.

## Setting Security for SAS/CONNECT and SAS/SHARE

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 a APPCSEC option setting in the same client SAS session.

Here is the syntax and 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;
```

For SAS/CONNECT, these values are valid until SIGNOFF.

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 Version 7, you provide client identification to a SAS/CONNECT remote host or a SAS/SHARE server using the APPCSEC option. APPCSEC must be defined on the local host before you connect to the remote host (using the SIGNON statement) or access a SAS/SHARE server (using the LIBNAME statement).

Here is the syntax and description of this option.

APPCSEC=*userid.password* | _PROMPT_

*userid.password*

specifies the remote host userid and password and is thus host-dependent in form. If either the userid or password contains blanks or special characters, it must be enclosed in quotes. A period (.) is used as a delimiter between the userid and password and, therefore, is not a valid character.

_PROMPT_

specifies that the SAS system prompt the client for the userid and password.

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

This technique is especially useful when the configuration file specifying this option is shared among many users.

Examples:

```
options appcsec=bass.time2go;
options appcsec=_prompt_;
```

## Providing Userid-Based Security for a SAS/SHARE Server

The APPCSEC option also specifies whether the APPC access method performs user authentication before connecting to a SAS/SHARE server. The APPCSEC option must be set on the server before you start the SAS/SHARE server.

Here is the syntax and description of this option.

APPCSEC=_SECURE_ | _NONE_

_SECURE_

The _SECURE_ value for the APPCSEC option causes the APPC access method to attempt to authenticate connecting SAS/SHARE clients. Each client connecting using APPC is required to supply a userid and password valid for the host on which the server is running.

_NONE_

The _NONE_ value for the APPCSEC option causes the APPC access method to NOT attempt to authenticate connecting SAS/SHARE clients. This is the default action when APPCSEC has not been set.

Examples:

```
options appcsec=_secure_;
options appcsec=_none_;
```

## SAS/CONNECT and SAS/SHARE Options

LU62MODE=*mode*

specifies the communications mode and its properties (for example, protocol and performance parameters). The default mode name is SASAPPC. Whether you assign a mode name to the option or you accept the default SASAPPC, you must define the mode in both the local session and the remote environments (on the local and the remote hosts in a SAS/CONNECT session or on the SAS/SHARE server and the clients, as necessary). Mode properties are defined through the MODEENT macro. See "Defining the Logon Mode Table Entries" on page 94 for details about defining mode properties.

Ask your network administrator for advice about setting this option.

LUPREFIX=*name*

specifies an identifying prefix to attach to a value that is in the range defined by the LUFIRST and LULAST options. Combining LUPREFIX and a value in the range defined by the LUFIRST and LULAST options forms an ACBNAME, which is assigned to each LU pool member.

You can set this option only in a SAS configuration file or at a SAS invocation.

LUFIRST=*n*

specifies the beginning (numeric) boundary of the range to form the pool of LU names.

You can set this option only in a SAS configuration file or at a SAS invocation.

LULAST=*n*

specifies the final (numeric) boundary of the range to form the pool of LU names.

You can set this option only in a SAS configuration file or at a SAS invocation.

Example 1:

If you have a maximum pool size (or depth) of 99 LUs and you want an 8-byte ACBNAME LU name, select an LUPREFIX value that is 6 bytes long.

```
luprefix=sascon
lufirst=1
lulast=99
```

In this example, ACBNAME LU names (or LU pool names) range from SASCON01 (zero-filling is required) to SASCON99 (no zero-filling is required).

Example 2:

If you have a maximum pool size (or depth) of 9 LUs and you want an 8-byte ACBNAME LU name, select an LUPREFIX value that is 7 bytes long.

```
luprefix=sascont
lufirst=1
lulast=9
```

In this example, ACBNAME LU names (or LU pool names) range from SASCONT1 to SASCONT9 (no zero-filling is required).

Example 3:

If you have a maximum pool size (or depth) of 10 LUs and you want a 6-byte ACBNAME LU name, select an LUPREFIX value that is 4 bytes long.

```
luprefix=sasc
lufirst=1
```

```
lulast=10
```

In this example, LU pool names range from SASC01 (zero-filling is required) to SASC10 (no zero-filling is required).

You may also infer the pool size from the setting of the LULAST and LUFIRST options. Use the following formula:

*pool-size*= (LULAST - LUFIRST) + 1

Example:

```
pool-size=(99-1)+1=99
```

These option settings describe the pool of LUs from which an LU is dynamically selected when a local host connects to a SAS/CONNECT remote host or when a client accesses a SAS/SHARE server. For SAS/CONNECT only, these options are used only when pool use is enabled. See "Options for SAS/CONNECT Only" on page 89 for information about the LUPOOL option.

For SAS/CONNECT only, instead of defining an LU pool, you may define a user-dedicated LU with the LUNAME option. This is explained in the next section.

*Note:* SAS/CONNECT does not support communication through the transaction program (TP) scheduler's LU that uses the APPC/MVS subsystem communication interfaces. Therefore, you must specify the LUPOOL= option. This release uses APPC/MVS subsystem services to invoke the remote SAS session. SAS/CONNECT opens a separate access method control block (ACB) and performs subsequent communication by means of the standard APPC/VTAM interface. △

## Options for SAS/CONNECT Only

LUNAME=*name*
: defines a dedicated LU name (eight characters maximum) for use in the current session instead of a dynamically selected LU from an LU pool.

LUPOOL=USER | ALL

specifies when an LU should be dynamically selected from a pool.

The value USER enables pool use for local OS/390 SAS sessions. The value ALL enables pool use for both local and remote OS/390 SAS sessions. USER is the default.

# SAS/CONNECT

## Local Host Tasks

*User or Applications Programmer*

To connect an OS/390 local host to a remote host, perform these tasks at the local host:

1. Specify the communications access method.
2. Specify the remote node name.
3. Sign on to the remote host.

## 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;
```

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

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

Example:

```
options comamid=appc;
```

## Specifying the Remote Host Name

To connect an OS/390 local host to a remote host, use the following syntax:

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

where *remote-session-id* specifies the logical unit of the remote host that you are connecting to. Types of valid values for the REMOTE= option follow:

**Table 6.1** OS/390 APPC SAS/CONNECT REMOTE= Values

| Type of Remote Host | Remote Host Identifier |
|---|---|
| OS/390 (with ESA) | logical unit name that is configured to the APPC (OS/390) scheduler |
| CMS | name of the AVS gateway |
| OS/2 | name of a control-point LU or other OS/2 locally defined LU |
| Windows NT, Windows 95 , and Windows 98 | name of a control-point LU or other SNA Server locally defined LU |
| OS/390 (with TSO) | for a scripted sign on to TSO, if the local host is OS/2, specify the 3270 EHLLAPI short or long session identifier of the remote 3270 session. (You define the short and long session identifiers in your emulation program.) |
| VSE | name of the VTAM APPL ID (ACBNAME) that was set up for APPC LU6.2 communications |

Example:

```
options remote=NO2SV01;
```

The remote node NO2SV01 is the LU name of the remote host that you want to connect to.

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

Ask your network administrator for the *remote-session-id* .

## 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 85.

## Local Host Example

The following example illustrates the statements that you specify in an OS/390 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 the remote host REMOTELU. The USER= option in the SIGNON statement 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 these tasks at the remote host :

1. Know the conditions under which an OS/390 host can be used as the remote host.
2. Specify the remote host name.
3. Optionally, set several remote options at the remote host.

## Using the OS/390 Host as a Remote Host

To use the OS/390 host as a remote host, make sure that it meets the following requirements:

- □ If the OS/390 remote host has Version 4, Release 2 or later of MVS/ESA with APPC/MVS configured, you can connect from a local OS/390, CMS, VSE, OS/2, Windows NT, Windows 95, Windows 98, Windows 32s, or UNIX host.
- □ If the OS/390 remote host does not have Version 4, Release 2 or later of MVS/ESA with APPC/MVS configured, you can use a script to connect from a local OS/2 host to the remote OS/390 host.

## Specifying the Remote Host Name

You must declare a remote host name at both the local host and the remote host in a SAS/CONNECT session. At both hosts, you specify an OPTIONS statement with 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. See Table 6.1 on page 90 for valid values.

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 remote host 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 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 recommended 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 SAS 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 an OS/390 remote host's configuration file to prepare for a connection from a supported local host with the APPC access method.

```
dmr
comamid=appc
remote=remotelu
luprefix=sascon
lufirst=1
lulast=10
lu62mode=appcmode
lupool=all
no$syntaxcheck
noterminal
```

The APPC communications access method is declared with a connection to REMOTELU, which is the LU name of the APPC/MVS scheduler. LUPREFIX SASCON is specified with the LUFIRST and LULAST values to form ACBNAME LU names ranging from SASCON01 to SASCON10. The LU62MODE name is APPCMODE, and an LUPOOL value of ALL specifies pool use for both local and remote OS/390 SAS sessions.

*Note:* The value for the REMOTE= option that is specified in the local and the remote sessions must be identical. △

# System Configuration for the APPC Access Method for SAS/CONNECT

*VTAM Systems Personnel*

Configure resources that enable the OS/390 system to behave as either a local host or a remote host in a SAS/CONNECT session when using the APPC access method.

To configure the APPC system perform the following tasks:

1. Define an LU pool.
2. Add entries to the logon mode table.
3. Define the OS/390 host as a remote host, if necessary.

This section highlights the tasks that you perform to configure a system for the APPC access method. For full details about configuring the APPC subsystem, see "References" on page 97 for a list of related documents. For full details about configuring an OS/390 remote host, see *MVS/ESA Planning: APPC Management (GC28-1110)*.

## Defining an LU Pool

Specify APPL statements to define a pool of LUs from which an LU is dynamically selected for a connecting local host. Use a comma to separate each entry.

A sample APPL definition follows:

```
N02SVnnn  APPL   APPC=YES,
                 AUTOSES=1,
                 DDRAINL=NALLOW,
                 DLOGMOD=mode-table-entry,
                 DMINWNL=16384,
                 DMINWNR=0,
                 DRESPL=NALLOW,
                 DSESLIM=32767,
                 EAS=3,
                 MODETAB=mode-table,
                 SECACPT=CONV,
                 VPACING=n
```

An explanation of each entry follows:

NO2SV*nnn*
: specifies the LU name, which is formed by the values of the LUPREFIX, LUFIRST, and LULAST options. These options are used to form the name of the pool of LUs from which an LU is dynamically selected. See "SAS/CONNECT and SAS/SHARE Options" on page 88 for more information about these options and how they are used to name an LU pool.

APPL
: declares an APPL definition statement.

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

AUTOSES=1
: tells the APPC subsystem how many sessions to start automatically after CNOS (Change Number of Sessions) negotiation.

DDRAINL=NALLOW
: does not allow the application to drain its allocation requests during shutdown.

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

DMINWNL=16384
specifies the initial negotiation value for local contention winner sessions.

DMINWNR=0
specifies that the remote partner's contention winner sessions request be used.

DRESPL=NALLOW
requires the remote partner to deactivate sessions.

DSESLIM=32767
defines the maximum session limits.

EAS=3
specifies the estimated number of sessions that are active with this logical unit at any given time.

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

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

VPACING=*n*
sets network requirements per site.

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

## Defining the Logon Mode Table Entries

A logon mode table contains one or more sets of session properties that contain session protocols, which characterize how the session is conducted. These entries 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 *Technical Reference 5, VTAM Customization* for complete information.

A typical logon mode table, which contains three sets of session properties, is shown here.

```
SNASVCMG MODEENT LOGMODE=SNASVCMG,
                 FMPROF=X'13',
                 TSPROF=X'07',
                 PRIPROT=X'B0',
                 SECPROT=X'B0',
                 COMPROT=X'50B1',
                 RUSIZES=X'xxxx',
                 PSERVIC=X'060200000000000000000300',
                 TYPE=0

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

MAPPCDEP MODEENT LOGMODE=MAPPCDEP,
                 FMPROF=X'13',
                 TSPROF=X'07',
                 PRIPROT=X'B0',
                 SECPROT=X'B0',
                 COMPROT=X'50B1',
                 RUSIZES=X'xxxx',
                 PSERVIC=X'060200000000000000102C00',
                 COS=name,
                 TYPE=0
```

*Note:* The SNASVCMG entry is required to support SNA services that are initiated through a boundary NCP. Although the MAPPCIND entry specifies a parallel session and CNOS support for independent LUs, the MAPPCDEP entry for dependent LUs does not. Both examples specify access security field acceptance. The RUSIZES and COS settings are site-specific. △

If you do not intend to use an OS/390 host as a remote host, then you have completed the APPC system configuration.

## Configuring an OS/390 Remote Host

You must perform additional tasks if you plan to use the OS/390 host as a remote host in a SAS/CONNECT session with a local host that is one of the following types:

CMS

OS/390

OS/2 (without TSO logon)

UNIX (AIX, HP-UX, Solaris)

VSE

Windows NT, Windows 95, Windows 98, Windows 32s

These types of connections depend on SAS/CONNECT MVS/ESA APPC/MVS subsystem capabilities.

To configure an OS/390 remote host, perform the following tasks:

1. Define the VSAM transaction program profile data set to be run when a local host attempts to connect to an OS/390 remote host.
2. Define the transaction program profile.
3. Define a class name for the SASRMT transaction program.
4. Define and activate a target LU.

### Defining the VSAM Transaction Program Profile Data Set

Run the IDCAMS utility to define the VSAM transaction program (TP) profile data set. This program creates a KSDS VSAM cluster, which defines the TP profile. An example follows:

```
DEFINE CLUSTER  ( NAME ( vsam.profile.dataset ) -
                 VOLUME ( volser ) -
```

```
                KEYS ( 112 0 ) -
                RECORDSIZE ( 248 4000 ) -
                RECORDS ( 100 100 ) -
                SHAREOPTIONS ( 3 3 ) -
                INDEXED REUSE ) -
        DATA    ( NAME ( vsam.profile.data ) ) -
        INDEX   ( NAME ( vsam.profile.index ) )
```

The data set member ATBTPVSM, which is located in the SYS1.SAMPLIB data set, contains other examples for creating this type of data set.

## Defining the Transaction Program Profile

Run the APPC/MVS administrative utility ATBSDFMU to define the TP profile. An example of a TP profile follows:

```
//UPDTTPSI EXEC PGM=ATBSDFMU
//SYSPRINT DD SYSOUT=*
//SYSSDLIB DD DSN=vsam.profile.dataset,DISP=SHR
//SYSSDOUT DD SYSOUT=*
//SYSIN    DD DATA,DLM=$$
TPADD
     TPNAME(SASRMT)
     SYSTEM
     ACTIVE(YES)
     TPSCHED_DELIMITER(SCHEND)
     TAILOR_SYSOUT(YES)
     TAILOR_ACCOUNT(YES)
     CLASS(xxx)
     JCL_DELIMITER(JCLEND)
//SASRMT  JOB 'REMOTE SAS',MSGLEVEL=(1,1)
//   JCLLIB ORDER=(your.proc.library)
//   EXEC  copySAS
JCLEND
     KEEP_MESSAGE_LOG(ALWAYS)
SCHEND
$$
```

Note that the utility SYSIN input includes embedded JCL. The APPC/MVS transaction scheduler uses this JCL to invoke the SASRMT TP profile, which starts a remote SAS application on an OS/390 host by executing a SAS cataloged procedure. You must specify the DSN of the cataloged procedure library where your SAS procedure resides in the JCLLIB statement of this JCL.

If users on local hosts do not need to specify SAS options when invoking a remote SAS session, you can specify your standard SAS cataloged procedure in the preceding EXEC statement in the SASRMT JCL.

However, if users on local hosts do need to specify options when invoking a remote SAS session, you must set up a separate cataloged procedure, such as the following sample APPCSAS procedure. The cataloged procedure refers to a user-specific configuration file that has the system symbolic parameter &SYSUID.

```
//APPCSAS  PROC USER=&SYSUID
//SAS  EXEC PGM=SASHOST,REGION=4096K
//STEPLIB  DD  DISP=SHR,DSN=prefix.LIBRARY
//CONFIG   DD  DISP=SHR,dsn=&USER..SASRMT.CONFIG
//         DD  DISP=SHR,DSN=prefix.CNTL(BATCH)
//SASHELP  DD  DISP=SHR,DSN=prefix.SASHELP
```

```
//SASMSG   DD  DISP=SHR,DSN=prefix.SASMSG
//WORK     DD  UNIT=3380,SPACE=(CYL,(1,1)),
//
DCB=(RECFM=FS,LRECL=6144,BLKSIZE=6144,DSORG=PS)
//SASLOG   DD SYSOUT=A,DCB=(BLKSIZE=141,LRECL=137,
                            RECFM=VBA)
//SASLIST  DD SYSOUT=A,DCB=(BLKSIZE=141,LRECL=137,
RECFM=VBA)
//SYSIN    DD  DUMMY
```

## Defining a Class Name for the SASRMT Transaction Program

Create an ASCHPM*xx* member in the SYS1.PARMLIB data set, where *xx* is a two-character identifier for the member that defines the CLASS name for the SASRMT transaction program (TP). The MAX() parameter determines how many SAS SASRMT TPs can be initiated at a given time. Set MAX() to the maximum number of sessions that may be simultaneously initiated.

An example follows:

```
CLASSADD CLASSNAME(xxx) MIN(1) MAX(10) RESPGOAL(1)
```

where *xxx* specifies the class name that is in the site-specific transaction program profile.

## Defining and Activating an LU

To define and activate an LU, perform the following tasks:

1 Create an APPCPM*xx* member in the SYS1.PARMLIB data set, where *xx* is a two-character identifier for the member that defines the LU for connecting local hosts.

   This LU name is the value supplied to the REMOTE= option in both the local and remote SAS/CONNECT sessions.

2 Create a statement to specify the target LU name. An example follows:

```
LUADD ACBNAME(N01APL00)
TPDATA(vsam.profile.dataset)
TPLEVEL(SYSTEM)
```

3 In addition to defining an LU pool using the example in “Defining an LU Pool” on page 93, you must also create another APPL statement to define a VTAM application node for an LU that is dedicated to APPC/MVS .

   An APPL statement is used for each member within the pool and for the APPC/MVS dedicated LU. The dedicated LU is also referred to as a *scheduler LU* and as the *APPC/MVS LU*.

4 To activate the LU to APPC/MVS, use a START or SET APPC command to specify the *xx* identifier.

You have completed the OS/390 remote host configuration.

## References

For complete details about how to install and configure the APPC system, refer to the following documents:

*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 or 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 both a valid userid and password enforces server security. At the client, set the preferred security method for relaying 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 85.

## Specifying the APPC Communications Access Method

You must specify the APPC communications access method at the connecting 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 that is 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 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 options follows:

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

An example of configuration file entries for a Windows 95 client connecting to an OS/390 server follows:

```
comamid=appc
comaux1=tcp
```

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

## Specifying a Server Name

You must specify the server name in the LIBNAME and the PROC OPERATE statements as follows:

```
SERVER=server-id
```

where *server-id* is defined in an APPL statement when configuring the OS/390 system for use with the APPC access method. The APPC syntax allows a server name that contains up to eight characters. Because APPC and SAS naming requirements are compatible, there should be no naming conflict. See "System Configuration for the APPC Access Method for SAS/SHARE" on page 102 for details about defining an LU for the server.

An example of specifying a server name from a client follows:

```
libname demo 'SAS.DATA.LIB' server=server-id;
```

where *server-id* is defined when configuring the OS/390 host for the APPC access method. See "System Configuration for the APPC Access Method for SAS/SHARE" on page 102 for information about specifying a server identifier.

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 an OS/390 client configuration file to access a server with the APPC access method:

```
luprefix=sascon
lufirst=1
lulast=10
lu62mode=appcmode
```

LUPREFIX SASCON is specified with the LUFIRST and LULAST options to form ACBNAME LU names ranging from SASCON01 to SASCON10. The LU62MODE name is APPCMODE.

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

The APPC access method is declared. 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

*Server Administrator*

Perform the following tasks at the server:

1. Verify that the SAS SVC routine has been installed.
2. Set server security by means of the APPCSEC option.
3. Specify the APPC access method.
4. Specify the server name.

## Installing the SAS SVC Routine

The SAS SVC control program routine is an interface between the OS/390 operating system and a specific request, such as "third-party checking." This facility provides verification in the form of calls for authentication of both the userid and password and of library authority.

1. Install the SAS SVC routine, if necessary.

   If you have already installed the SAS SVC routine for Release 6.09E of SAS software, do not repeat the step here. If you need to perform the installation, see the *Installation Instructions and System Manager's Guide, The SAS System under OS/390* for details.

   Because SAS SVC in Release 6.09E is backward compatible, it replaces the SAS SVC routines from previous releases. You may continue using previous releases of base SAS and SAS/SHARE with the Release 6.09E SAS SVC that is installed on your system.

2. Verify the SAS options for the SVC routine.

   You must verify that the SAS options for the SVC routine accurately reflect the way that the SAS SVC is installed. The SAS option SVC0SVC should be set to the number at which the SAS SVC is installed (for example, 251 or 109). If the SAS SVC is installed at 109 as an ESR SVC, the SAS option SVC0R15 should be set to the ESR code (for example, 4).

3. Verify installation on all CPUs, as needed.

   If you have more than one CPU, verify that the SAS SVC routine is installed on the systems that will be running SAS/SHARE at your site.

## 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 both valid userids and passwords for the host at which the server is running. A secured server uses the validated userid and password to verify a user's authority to access a SAS library or a SAS file.

Requiring connecting clients to supply both a valid userid and password enforces server security. At the server, set the APPCSEC option to the value _SECURE_. For information about setting the APPCSEC option, see "Providing Client Identification in a pre-Version 8 Session" on page 87.

## 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. Use the COMAUX1 and COMAUX2 options. See Table 1.1 on page 8 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 an OS/390 host follows:

```
comamid=appc
comaux1=tcp
comaux2=xms
```

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 XMS access methods.

See *SAS/SHARE User's Guide* for details about starting 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 an APPL statement when configuring the OS/390 system for use with the APPC access method. The APPC syntax allows a server name that may contain up to eight characters. Because APPC and SAS naming requirements are compatible, there should be no naming conflict. See “System Configuration for the APPC Access Method for SAS/SHARE” on page 102 for details about defining an LU for the server.

An example of specifying a server name from a client follows:

```
libname demo 'SAS.DATA.LIB' server=server-id;
```

where *server-id* is defined when configuring the OS/390 host for the APPC access method. See "Defining an LU Pool" on page 93 for information about specifying a server identifier.

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 configuration file on the OS/390 host at which you start a server:

```
appcsec=_secure_
comamid=appc
```

The _SECURE_ value for the APPCSEC option requires clients to supply a userid and a password that are valid on the server.

The following statements in a SAS session on the OS/390 remote host start a server:

```
proc server id=share1 authenticate=req;
run;
```

The server SHARE1 is started on the OS/390 remote host.

# System Configuration for the APPC Access Method for SAS/SHARE

*Network Administrator*

Configure resources for the OS/390 system that allow it to behave as either a client or a SAS/SHARE server when using the APPC access method.

This section highlights the general tasks that you perform to set up the APPC access method and to configure the OS/390 host as a remote host for specific local host connections.

## Configuring the SAS/SHARE Server

To configure the APPC access method for a SAS/SHARE server, perform the following tasks:

1 Define an LU for the server.

2 Define the server as a cross-domain resource, if necessary.

### Defining an LU for the Server

You must define an LU for the server with an APPL statement in the APPC domain in which the server will run. All APPL statements must follow a VBUILD TYPE=APPL statement. Use a comma to separate each entry. The form of an APPL statement for a server is:

```
server-id  APPL      APPC=YES,
                     DDRAINL=NALLOW,
                     DLOGMOD=mode-table-entry,
                     DMINWNL=0,
                     DMINWNR=16384,
                     DSESLIM=32767,
            MODETAB=mode-table,
            SECACPT=CONV
```

An explanation of each entry follows:

*server-id*
: specifies the name of the server, which is the value for the SERVER= option in the PROC SERVER, PROC OPERATE, and LIBNAME statements.

APPL
: declares an APPL definition statement.

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

DDRAINL=NALLOW
: does not allow the application to drain its allocation requests during shutdown.

DLOGMOD=*mode-table-entry*
: specifies the default name of the logon mode table entry. If you supply neither the MODETAB nor the DLOGMOD variables, the first entry in the logon mode table is used. A logon mode entry determines which entry in the logon mode table is used for providing a set of session parameters to the application program. See “Defining the Logon Mode Table Entries” on page 94 for details about creating the logon mode table.

DMINWNL=0
: specifies the minimum number of parallel sessions for a mode name group for a given remote LU of which the application program (local LU), not the remote LU, is guaranteed to be the contention winner. A mode name group is a group of sessions for a given mode name.

DMINWNR=16384
: specifies the minimum number of parallel sessions for a mode name group for a given remote LU of which the remote LU, not the application program (local LU), is guaranteed to be the contention winner.

DSESLIM=32767
: defines the maximum session limits allowed between the local application LU and a remote LU on a given mode name group.

MODETAB=*mode-table*
: specifies the logon mode table to be used to associate each logon mode name with a set of session parameters for the application program. The name that you supply must correspond to the name of a logon mode table that you have already created. See “Defining the Logon Mode Table Entries” on page 94 for details about creating the logon mode table.

SECACPT=CONV
: specifies that this LU supports session requests that contain the access FMH5 security subfields.

## Defining the Server as a Cross-Domain Resource

If the server will be communicating with clients in a different APPC domain, you must define the server as a cross-domain resource in the clients’ domain. Use a CDRSC statement in the following form:

```
server-id      CDRSC CDRM=cdrm-name,ISTATUS=ACTIVE
```

where *server-id* specifies the name of the server. The server name must be identical to the name in the APPL statement for the server in the controlling domain.

*cdrm-name* specifies the name of the CDRM in the domain in which the server is defined as an APPL.

## Defining a Client LU Pool

You must define to APPC a pool of LUs for the connecting clients in the APPC domain in which the clients will run. Define this LU pool in a series of APPL statements. The APPC access method dynamically selects an available LU for a connecting client. The number of LUs in the pool should be greater than or equal to the maximum number of users that will be connected simultaneously to servers with the APPC access method.

All APPL statements must follow a VBUILD TYPE=APPL statement. Use a comma to separate each entry. The form of the APPL statement for a connecting client is

```
LU-name APPL       ACBNAME=LUPREFIX+length-of-LULAST,
                   APPC=YES,
                   DDRAINL=NALLOW,
                   DLOGMOD=mode-table-entry,
                   DMINWNL=conwin,
                   DMINWNR=0,
                   DSESLIM=seslimit,
                   MODETAB=mode-table,
                   SECACPT=CONV
```

An explanation of each entry follows:

*LU-name*
specifies the name of the LU pool. This name must be unique across the SNA network.

APPL
declares an APPL definition statement.

ACBNAME=LUPREFIX + *length-of-LULAST*
specifies the ACBNAME (minor node name assigned to this application program) that is associated with an LU pool. LUPREFIX is the name assigned to the LU pool whose size is determined by the value of the LULAST option. An ACBNAME results from combining the LUPREFIX with a value within the range defined by LULAST. See “SAS/CONNECT and SAS/SHARE Options” on page 88 for an explanation and examples of how to name an LU pool and its members.

Although the ACBNAME parameter is not required, it is recommended that it be specified so that the ACBNAME can be shared by more than one OS/390 system on the APPC network. ACBNAMEs must be unique only within a single domain.

If you do not define ACBNAME explicitly, an *LU-name* is inferred from the LUPREFIX and LULAST options.

*Note:* This LU pool can also be used by the VTAM LU 6.2 access method for SAS/CONNECT remote sessions on OS/390 and the VTAM LU 0 access method for SAS/SHARE client sessions. △

For definitions of all other pool entries, see “Defining an LU for the Server” on page 102.

You have completed the configuration of the APPC system for SAS/SHARE.

## References

For complete details about configuring the APPC access method, see:

*VTAM Installation and Resource Definition (SC23-0111)*

For details about configuring OS/390 as a remote host, see:

*MVS/ESA Planning: APPC Management (GC28-1110)*

Contact IBM for information about obtaining this documentation.

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.