CHAPTER

13

OS/2: APPC Access Method

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

System Administrator or User

To use the APPC access method with an OS/2 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 environment variables in SAS software.
4. Set the desired SAS/CONNECT and SAS/SHARE environment variables 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 both the local and remote hosts at your site.
2. SAS is installed on both the local and remote hosts.
3. Release 2.11 or a subsequent release of OS/2 has been installed.
4. One of the following packages has been installed:
   - □ the IBM CM/2 Version 1.11 or subsequent version.
   - □ the eNetwork Communications Server for OS/2 WARP, Version 4.1 or a subsequent release.
   - □ the eNetwork Personal Communications, Version 4.2 or higher.
   - □ Any product that offers APPC and runs on OS/2 2.11 or subsequent version.

Defining Resources for the APPC Access Method

Network Administrator

Note: The following describes how to configure an IBM Communications Server. However, other communication products may be used and configured according to their documentation. △

An IBM communications network enables an OS/2 host to provide client and server functionality for both SAS/CONNECT and SAS/SHARE using the APPC communications access method. Before you can use SAS/CONNECT and SAS/SHARE with the APPC access method, you must configure the network through the IBM Communications Manager utility or by editing directly the Network Description File (NDF). Regardless of the configuration method that you use, in this chapter, we refer to the Communications Manager utility. See “System Configuration for the APPC Access Method” on page 204 for the procedures to define network resources.

Note: At present, only a single SAS/CONNECT remote session or a single SAS/SHARE server may be running on an OS/2 workstation at a time because of the global characteristics in the RECEIVE_ALLOCATE interface with Communications Manager. △

Understanding IBM Network Terminology

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

LU (logical unit)
: a device or program by which a user (LU6.2 applications program) gains access to a network.

local LU
: a named LU that is associated with a local host that connects to a SAS/CONNECT remote host or with a client that accesses a SAS/SHARE server.

remote LU
: a named LU that is associated with the SAS/CONNECT remote host or with a SAS/SHARE server to which a local host or a client will attach.

LU alias
: an alternative name assigned to an LU (local or remote).

For more information about this terminology, see “System Configuration for the APPC Access Method” on page 204.

Setting SAS Options and Variables

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

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

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

OPTIONS SET=*variable-name value*;

Example:

```
options set=appc_luname remotelu;
```

□ in a SAS configuration file or at a SAS invocation:

-SET *variable-name value*

Example:

```
-set appc_luname remotelu
```

□ as a DOS operating system environment variable:

SET *variable-name=value*

Example:

```
set appc_luname=remotelu
```

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 allowed:

OPTIONS statement

AUTOEXEC file

SAS invocation

SAS configuration file

DOS environment variable.

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 several sections outline the alternatives for storing security information 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

The APPC_SECURE variable passes a remote host userid and password to a remote SAS/CONNECT host or to a SAS/SHARE server for verification. After the userid and the password have been verified, the connection to the remote SAS/CONNECT host or the SAS/SHARE server can proceed.

APPC_SECURE=_NONE_ | _PROMPT_ | *userid.password* | UPM

NONE
must be set at the SAS/CONNECT local host or the SAS/SHARE client. This is the default. Not setting the APPC_SECURE option also selects the default.

Setting _NONE_ does not establish secure sessions for connecting SAS/CONNECT local hosts or SAS/SHARE clients.

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

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

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

userid.password specifies both the userid and password. Assigning the userid and password directly to the APPC_SECURE option at the SAS/CONNECT local host or SAS/SHARE client may inadvertently publicize this information and compromise the security of the SAS/CONNECT remote host or the SAS/SHARE server. Assigning the value to the option in a file allows anyone to read it.

UPM
sets the userid and the password through the User Profile Management (UPM) facility, which is included in the APPC access method configuration process. See "Security Configuration" on page 206 for details about setting up this profile.

Ensure that the APPC_SECURE option is set before a server is started.

Examples:

```
options set=appc_secure _none_;
options set=appc_secure _prompt_;
options set=appc_secure bass.timego;
options set=appc_secure upm;
```

See "Setting SAS Options and Variables" on page 193 for examples of the forms you can use to specify APPC_SECURE.

SAS/CONNECT and SAS/SHARE Options

APPC_LUNAME
specifies the name of the local logical unit (LU) to use when connecting to a remote LU. An OS/2 local host may have multiple dependent LUs defined by means of the Communications Manager to support multiple concurrent remote sessions to which it can connect. You may use this option to override the default LU that was defined through the configuration utility.

Ask your network administrator for the name of the local LU that you can use to assign to this option or for the default local LU value.

APPC_LU62MODE
specifies the communications mode to use. The default mode name is SASAPPC. Whether you assign a mode name to the option or you accept the default SASAPPC, you must identify the mode and use the same mode name value in both the local and remote environments for SAS/CONNECT and SAS/SHARE.

Ask your network administrator for advice about setting this option.

APPC_SURROGATE_LUNAME
applies only when connecting to an OS/390 remote host.

APPC_SURROGATE_LUNAME specifies an LU to use for a SAS/CONNECT remote session on an OS/390 host. If this option is not defined, the OS/390 remote session dynamically selects an LU from the pool of LUs that is defined on the OS/390 host for this purpose.

Ask your network administrator for the name of the remote LU for the OS/390 host that you can use to assign to this option.

Note: Do not use this option's value for the value of the REMOTE= option in SAS/CONNECT . △

SAS/SHARE Option Only

APPC_USER
identifies individual output in the server's output log.

SAS/CONNECT

Local Host Tasks

User or Applications Programmer
To connect an OS/2 local host to a remote host, perform these tasks at the local host:

1. Specify the APPC 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 OS/2 local host to connect to 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 set this option at a SAS invocation or in a SAS configuration file.

Specifying the Remote Host Name

To make a connection from an OS/2 local host to a remote host, use the following syntax:

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

where *remote-session-id* specifies the remote host that you are connecting to. Types of valid values are shown in the following table.

Table 13.1 OS/2 APPC SAS/CONNECT REMOTE= Values

Type of Remote Host	Remote Session Identifiers
OS/390	name of APPC/MVS scheduler LU
OS/390 (with TSO)	long or short 3270 terminal emulation session identifier
CMS	name of private gateway LU for VM system

Type of Remote Host	Remote Session Identifiers
VSE	name of VTAM APPL ID (ACBNAME) that was set up for APPC LU6.2 communications
OS/2	name of control-point LU or other OS/2 locally defined LU
Windows NT, Windows 95, and Windows 98	name of control-point LU or other SNA server locally defined LU

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

```
options remote=remotelu;
```

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

Signing On to the Remote Host

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

```
signon user=_prompt_;
```

Sign-on script files are not needed on an OS/2 local host that connects to an OS/390 remote host that uses the APPC access method. APPC has the ability to communicate with the APPC/MVS subsystem to initiate the remote session. 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 194.

If you specify a script file through the RLINK fileref before establishing a connection, when you sign on, SAS/CONNECT processes and loads the script file that is identified by the fileref. The APPC access method attempts to use the script file but fails, leading to undesirable results.

If you do not want to free the RLINK fileref but want to prevent failure, you must use the NOSCRIPT option in the SIGNON and SIGNOFF statements, shown as follows:

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

Local Host Example

The following example illustrates the statements that you specify in an OS/2 local host SAS session to connect to a remote host with the APPC access method.

```
options comamid=appc remote=remotelu;
options set=appc_lu62mode appcmode;
signon user=_prompt_;
```

The APPC communications access method is declared with a connection to the remote host REMOTELU, which uses a mode name of APPCMODE. 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. Specify the remote host name.
2. Optionally, set several remote host options.

Specifying the Remote Host Name

Specify the remote host name in the configuration file of the OS/2 remote host. Use the following syntax:

```
-REMOTE remote-host-id
```

where *remote-host-id* takes the form of the *control-point-LU-of-the-OS/2-remote-host*.

You must specify the remote host identifiers at both the local and remote hosts. These remote host identifiers must be identical.

Example:

```
-remote rmtnode;
```

Setting Options at the Remote Host

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

NO$SYNTAXCHECK

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

NO$SYNTAXCHECK 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 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.

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

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

NOXWAIT

applies to OS/2 or Windows remote hosts only.

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

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

See *SAS Companion for the OS/2 Environment, Second Edition* for details about this option.

Remote Host Example

The following example illustrates the statements that you specify in an OS/2 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
-no$syntaxcheck
-noterminal
```

The APPC communications access method is declared with a connection to an OS/2 remote host that is identified as *control-point LU* REMOTELU.

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

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 the server name.

Setting Security for Connecting Clients

Requiring connecting clients to supply a valid userid and a 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 194.

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 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 an OS/2 client connecting to a Windows NT server follows:

```
-comamid appc
-comaux1 tcp
-comaux2 netbios
```

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

Specifying the Server Name

The server name that you specify in the PROC SERVER, PROC OPERATE, and LIBNAME statements must be defined as the *local-LU* at the SAS/SHARE server and as a *remote-LU* at the SAS/SHARE client. For complete information about defining appropriate LUs for use with SAS/SHARE, see “System Configuration for the APPC Access Method” on page 204.

The server name must meet the criteria for a valid SAS name. See *SAS Language Reference: Dictionary* for details about SAS naming rules.

An example of specifying the server name follows:

```
options comamid=appc;
libname demo 'C:\' server=remote-lu;
```

In this example, at the client, the server name is expressed as a *remote-LU*.

For details about creating LIBNAME, PROC OPERATE, and PROC SERVER statements, see *SAS/SHARE User's Guide*.

Client Example

The following example illustrates the statements that you specify in an OS/2 client configuration file to access a server with the APPC access method:

```
-comamid appc
-set appc_lu62mode appcmode
```

The APPC communications access method is declared. The APPC_LU62MODE option is set to APPCMODE.

```
options comamid=appc;
libname sasdata 'c:\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

Network Administrator
You can configure the APPC access method to authenticate connecting clients.

Authenticating Connecting Clients

Authenticate users at the server by setting up APPC conversation security within the SASTP62 TP (transaction program) definition.

For further details about setting up TP definitions, see "System Configuration for the APPC Access Method" on page 204.

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

1 Set server security through the SASTP62 transaction program.
2 Specify the APPC access method.
3 Specify the server name.

Setting Server Security

To run the server in secure mode, set the CONV_SECURE_RQD parameter in the SASTP62 transaction program to YES. See "Defining the SASTP62 Transaction Program" on page 206 for information about setting server security.

Specifying the APPC Access Method at the Server

Specify the APPC communications access method before you create and access 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 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 by using the COMAUX1 and COMAUX2 options. See Table 1.3 on page 10 for the supported access methods by host.

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

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

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

An example of configuration file entries for a server that is running on an OS/2 host follows:

```
-comamid appc
-comaux1 tcp
-comaux2 netbios
```

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

Specifying the Server Name

The server name that you specify in the PROC SERVER statement must be defined as the *local-LU* at the SAS/SHARE server and as a *remote-LU* at the SAS/SHARE client. For complete information about defining appropriate LUs for use with SAS/SHARE, see “System Configuration for the APPC Access Method” on page 204.

The server name must meet the criteria for a valid SAS name. See *SAS Language Reference: Dictionary* for details about SAS naming rules.

An example of specifying the server name follows:

```
options comamid=appc;
libname demo 'C:/\' server=remote-lu;
```

In this example, at the client, the server name is expressed as a *remote-LU*.

For details about the PROC SERVER statement, see *SAS/SHARE User's Guide*.

Server Example

The following example illustrates the statements that you specify in a configuration file on the OS/2 host at which you start a server:

```
-comamid appc
```

The APPC communications access method is declared.

The following example illustrates the statements that you specify in a SAS session on the OS/2 remote host at which you start a server:

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

The server SHARE1 is started on the OS/2 remote host.

System Configuration for the APPC Access Method

Network Administrator

This section highlights the general tasks that you perform to configure the OS/2 host for the APPC access method.

To configure each SAS/CONNECT local and remote host and each SAS/SHARE client and server, define the following properties:

1. Local control point and logical units
2. Communications modes.

To configure a SAS/CONNECT remote host and a SAS/SHARE server, define the following properties:

1. Default communications parameters
2. Transaction program SASTP62
3. Transaction program SASRMT for SAS/CONNECT only
4. Attach Manager.

If you are using IBM Communications Server, you may use either of these methods to configure the OS/2 system:

1. Interact with the configuration utility, supplying answers to menu prompts that are captured automatically in the Network Description File (NDF).
2. Edit directly the NDF.

The instructions and examples in this section assume the direct editing of the Network Description File (NDF). If you use the Communications Manager utility, verify your selections with the NDF files that are provided in the configuration example in this section.

Defining the Local Control Point and Logical Units

You must define the local control point by using the DEFINE_LOCAL_CP definition statement and the local logical units (LU) by using the DEFINE_LOCAL_LU definition statement. You must define one control point per workstation, and you may define one or more LUs.

An LU can be independent or dependent. An LU for another OS/2 workstation is always independent. The communications software that is used in your network determines whether an LU for a SAS/CONNECT remote host on an OS/390 or on a CMS host is capable of performing as an independent or dependent LU. Such an LU may be defined as dependent because of the level of the control program running in the communications controller that connects the OS/2 workstation to the mainframe. Ask your SNA support personnel whether you are limited to a dependent LU capability.

SAS/CONNECT can use either a dependent or an independent LU. However, if you are using dependent LUs, you must have one dependent LU defined for each concurrent remote session that is established by the local session. A single independent LU allows multiple concurrent SAS/CONNECT sessions.

If multiple local LUs are defined (for example, to support concurrent dependent LU environments), you may specify the LU_ALIAS that is associated with the LU through the option APPC_LUNAME. See “Setting SAS Options and Variables” on page 193 for information about APPC_LUNAME.

Note: SAS/SHARE and the SAS/CONNECT Remote Library Services (RLS) require an independent LU. △

Defining Communications Modes

If site-naming conventions permit, you should specify the mode name as SASAPPC. The APPC access method uses this default name if the option APPC_LU62MODE is not defined.

You specify a mode definition using the DEFINE_MODE verb, which provides two classes of information:

□ maximum session constraints

□ performance metrics.

The MAX_NEGOTIABLE_SESSION_LIMIT parameter sets the upper boundary on concurrent session requests from a partner logical unit. The PLU_MODE_SESSION_LIMIT parameter sets the upper boundary on concurrent session requests that are initiated by the local LU within the defined MODE_NAME.

Specify MAX_NEGOTIABLE_SESSION_LIMIT(32767) and, at minimum, PLU_MODE_SESSION_LIMIT(16382). See “Configuration Example” on page 208 for an example of how these parameters are used.

You must also set the MIN_CONWINNERS_SOURCE parameter because only contention-winner sessions are used for locally-initiated communication. Unless you are using Remote Library Services (RLS), communication between SAS/CONNECT local and remote hosts requires only one contention-winner session. RLS requires a minimum of four contention-winner sessions. This limit affects the number of data sets that can be accessed concurrently through RLS and SAS/SHARE. Therefore, you should specify one contention-winner session from each connection and one for each data set that is open.

Your local SNA administrators may have defined multiple classes of service within the network to control performance and security. The COS_NAME parameter is used to associate a class of service with a mode. You reference these classes of service through a unique MODE_NAME parameter and the option APPC_LU62MODE. See “Setting SAS Options and Variables” on page 193 for information about APPC_LU62MODE. Keep in mind that mode names must be consistently defined across the distributed domain. In addition to defining a mode name locally and presenting it to a remote control point, you must also define the mode name at the remote control point. For example, all mode names that you defined locally through DEFINE_MODE that may be presented to a host VTAM control point must also be defined to that VTAM through the assembly of corresponding MODEENT macros.

If naming conventions at your site prevent you from using the default communications mode name, SASAPPC, include DEFAULT_MODE_NAME in the DEFINE_DEFAULTS specification for the workstation in order to avoid having to define APPC_LU62MODE for each SAS/CONNECT local host or SAS/SHARE client.

Defining Default Communications Parameters

If your environment supports APPN, specify IMPLICIT_INBOUND_PLU_SUPPORT (YES) in a DEFINE_DEFAULTS statement. Doing so eliminates a need to define partner LUs for clients that will establish a SAS/CONNECT remote session or will connect to a SAS/SHARE server on this workstation.

If restrictions at your site force you to specify IMPLICIT_INBOUND_PLU_SUPPORT(NO), you must define a partner LU and a partner LU location for each workstation control point or LU that is associated with either a SAS/CONNECT local host that will connect to a remote host or a SAS/SHARE client that will access a server on your workstation.

Defining the SASTP62 Transaction Program

You must define transaction program SASTP62, which handles inbound communications, to allow a SAS/CONNECT remote host or a SAS/SHARE server to run on a workstation. Typical content of SASTP62 follows:

```
DEFINE_TP TP_NAME(SASTP62)
DESCRIPTION(Internal SAS Service TP)
FILESPEC(D:\SAS\SASTP62.EXE)
CONVERSATION_TYPE(EITHER)
CONV_SECURITY_RQD(NO)
SYNC_LEVEL(EITHER)
TP_OPERATION(QUEUED_OPERATOR_PRELOADED)
PROGRAM_TYPE(BACKGROUND)
INCOMING_ALLOCATE_QUEUE_DEPTH(255)
INCOMING_ALLOCATE_TIMEOUT(120)
RECEIVE_ALLOCATE_TIMEOUT(60);
```

You must supply the appropriate SAS root path to the FILESPEC parameter. The CONV_SECURITY_RQD parameter specifies whether the SAS/CONNECT remote session or the SAS/SHARE server should run in a secure mode. If you specify YES for this parameter, the TP will require a userid and a password for the workstation. The timeout parameters are specified in seconds. See the next section for more information about security. See “Setting SAS Options and Variables” on page 193 for information about the APPC_SECURE option.

Note: SASTP62.EXE is an internal transaction program that cannot be located in the SAS directory. △

Security Configuration

You can require users who are connecting to a SAS/CONNECT remote host or to a SAS/SHARE server to supply a userid and a password by specifying CONV_SECURITY_RQD(YES) when you define transaction program SASTP62. If you choose to run in a secure mode, you must configure userid and password profiles for each workstation.

You may create userid and password profiles for connecting users through a series of menus that are provided with the Communications Manager utility. The userid and password that you define will be stored in an encrypted format in the .sec file for the associated configuration.

Alternatively, you can instruct the Communications Manager utility to interact with User Profile Management, the IBM OS/2 access control system.

Defining the SASRMT Transaction Program

The SASRMT transaction program, which handles inbound communications, applies to a SAS/CONNECT remote host only.

You must define transaction program SASTP62 to allow a SAS/CONNECT remote host to run on a workstation. The content of SASRMT follows:

```
DEFINE_TP TP_NAME(SASRMT)
DESCRIPTION(Remote SAS Initiation TP)
FILESPEC(D:\SAS\SAS.EXE)
PARM_STRING(-DMR -COMAMID APPC -REMOTE local-LU
-NOTERMINAL -NO$SYNTAXCHECK -CONFIG D:\SAS\CONFIG.SAS)
```

```
CONVERSATION_TYPE(EITHER)
CONV_SECURITY_RQD(NO)
SYNC_LEVEL(EITHER)
TP_OPERATION(NONQUEUED_AM_STARTED)
PROGRAM_TYPE(PRESENTATION_MANAGER)
RECEIVE_ALLOCATE_TIMEOUT(INFINITE);
```

You must specify

- □ the full pathname for SAS.EXE to the FILESPEC parameter.
- □ the full pathname for the SAS configuration file to the PARM_STRING parameter for the SAS/CONNECT remote host.
- □ the SAS system options COMAMID= and REMOTE= in the CONFIG.SAS file. If you enabled security for SASTP62, you should also specify CONV_SECURITY_RQD(YES) for SASRMT.

Note: You may also create a SAS configuration file to contain the required options. △

Defining the Attach Manager

The Attach Manager agent is required to enable users to start a SAS/CONNECT remote session or to access a SAS/SHARE server on a workstation. Typically, you should have the Attach Manager agent started automatically when the Communications Manager utility initializes.

Configuring Sessions on Other Workstations

In environments other than APPN, in order to connect to a SAS/CONNECT remote host or to a SAS/SHARE server that is running on another workstation, you must define a partner LU using the DEFINE_PARTNER_LU statement. The definition for a SAS/SHARE server should specify PARALLEL_SESSION_SUPPORT(YES).

Note: A partner LU definition requires supporting partner LU location and logical link definitions in the Network Description File (NDF). See “Configuration Example” on page 208 for examples of partner LU, partner LU location, and logical link definitions. △

A SAS/CONNECT remote session on an OS/390 host can use either a predetermined LU or a dynamically selected LU from a pool of LUs. The advantage of using a specific LU is that only one partner LU definition is required in environments other than APPN. The disadvantage is that each workstation must specify a different partner LU definition. Pooling avoids this requirement. However, environments other than APPN require a partner LU definition on the workstation for each pooled LU on an OS/390 host.

Thus, pooling requires that more partner LUs be defined, but the set of partner LUs is the same for all workstations. Pooling is also easier if only a relatively small fraction of your users are simultaneously connected to remote sessions on an OS/390 host. For example, if only 10 of 100 users will be connected simultaneously, a pool of only 10 LUs is sufficient and is preferable to defining 100 reserved LUs. If pooling is enabled, each user must specify a dynamically assigned LU from an LU pool through APPC_SURROGATE_LUNAME. See “Setting SAS Options and Variables” on page 193 for information about setting this option.

Configuration Example

SAS Institute developed the following working example of an NDF, which defines a Token Ring network that is connected to an OS/390 system through an IBM 3174 Establishment Controller, Configuration Support-C, with the APPN LIC feature.

This example does not take advantage of APPN network node dynamic directory services because an APPN-capable VTAM (V4R1) was not available to verify the configuration. The availability of these dynamic directory services should reduce the workstation configuration burden by eliminating partner LU and associated definitions.

Local Control Point and Single Logical Unit

The following definitions establish the local control point and a single logical unit for the workstation:

```
DEFINE_LOCAL_CP FQ_CP_NAME(USNETI01.NETCP801)
  DESCRIPTION(INDEPENDENT CP NODE DEFINITION)
  CP_ALIAS(N01P0001)
  NAU_ADDRESS(INDEPENDENT_LU)
  NODE_TYPE(EN) NODE_ID(X'12345')
  HOST_FP_SUPPORT(YES);

DEFINE_LOCAL_LU LU_NAME(N01P0001)
  DESCRIPTION(Independent LU)
  LU_ALIAS(APPCIND )
  NAU_ADDRESS(INDEPENDENT_LU);
```

Logical Link Definitions

The following logical link definitions provide data link control details to the associated symbolic link name. The first definition is for the 3174 host sub-area gateway; the second is for a peer workstation on the Token Ring network. The symbolic link names NETCP800 and NETCP802 are used in the partner logical unit location definitions shown in “Partner Logical Unit Location Definitions” on page 210.

```
DEFINE_LOGICAL_LINK LINK_NAME(NETCP800)
  DESCRIPTION(3174 Host Gateway)
  FQ_ADJACENT_CP_NAME(USNETI01.NETCP800)
  ADJACENT_NODE_TYPE(NN)
  PREFERRED_NN_SERVER(NO)
  DLC_NAME(IBMTRNET)
  ADAPTER_NUMBER(0)
  DESTINATION_ADDRESS(X'400031740001')
  CP_CP_SESSION_SUPPORT(YES)
  ACTIVATE_AT_STARTUP(YES)
  LIMITED_RESOURCE(NO)
  LINK_STATION_ROLE(USE_ADAPTER_DEFINITION)
  SOLICIT_SSCP_SESSION(NO)
  EFFECTIVE_CAPACITY(USE_ADAPTER_DEFINITION)
  COST_PER_CONNECT_TIME(USE_ADAPTER_DEFINITION)
  COST_PER_BYTE(USE_ADAPTER_DEFINITION)
  SECURITY(USE_ADAPTER_DEFINITION)
  PROPAGATION_DELAY(USE_ADAPTER_DEFINITION)
  USER_DEFINED_1(USE_ADAPTER_DEFINITION)
  USER_DEFINED_2(USE_ADAPTER_DEFINITION)
  USER_DEFINED_3(USE_ADAPTER_DEFINITION);

DEFINE_LOGICAL_LINK LINK_NAME(NETCP802)
  DESCRIPTION(Token Ring Peer Node)
  FQ_ADJACENT_CP_NAME(USNETI01.NETCP802)
  ADJACENT_NODE_TYPE(NN)
  PREFERRED_NN_SERVER(NO)
  DLC_NAME(IBMTRNET)
  ADAPTER_NUMBER(0)
  DESTINATION_ADDRESS(X’400000314003’)
  CP_CP_SESSION_SUPPORT(YES)
  ACTIVATE_AT_STARTUP(NO)
  LIMITED_RESOURCE(NO)
  LINK_STATION_ROLE(USE_ADAPTER_DEFINITION)
  SOLICIT_SSCP_SESSION(NO)
  EFFECTIVE_CAPACITY(USE_ADAPTER_DEFINITION)
  COST_PER_CONNECT_TIME(USE_ADAPTER_DEFINITION)
  COST_PER_BYTE(USE_ADAPTER_DEFINITION)
  SECURITY(USE_ADAPTER_DEFINITION)
  PROPAGATION_DELAY(USE_ADAPTER_DEFINITION)
  USER_DEFINED_1(USE_ADAPTER_DEFINITION)
  USER_DEFINED_2(USE_ADAPTER_DEFINITION)
  USER_DEFINED_3(USE_ADAPTER_DEFINITION);
```

Partner Logical Unit Definitions

The following partner logical unit definitions provide the aliases that are associated with communication partners, among other attributes. The first group, which contains partner LUs N02SV601 through N02SV605, corresponds to the set of potential OS/390 SAS/CONNECT remote sessions. The LU for an OS/390 remote session is selected dynamically from a pool of LUs that is defined on an OS/390 host for this purpose. See Chapter 6, “OS/390: APPC Access Method,” on page 83 for details about defining pool size with the SAS options LUPOOL, LUFIRST, LULAST, and LUPREFIX. Each LU in the pool must be defined as a partner LU on the workstation.

The final entry, N01P0002, is a peer partner on the Token Ring network on which a user may connect to a SAS/CONNECT remote host or to a SAS/SHARE server.

```
DEFINE_PARTNER_LU
  FQ_PARTNER_LU_NAME(USNETI01.N02SV601)
  PARTNER_LU_ALIAS(N02SV601)
  PARTNER_LU_UNINTERPRETED_NAME(N02SV601)
  MAX_MC_LL_SEND_SIZE(32767)
  CONV_SECURITY_VERIFICATION(NO)
  PARALLEL_SESSION_SUPPORT(YES);

DEFINE_PARTNER_LU
  FQ_PARTNER_LU_NAME(USNETI01.N02SV602)
  PARTNER_LU_ALIAS(N02SV602)
  PARTNER_LU_UNINTERPRETED_NAME(N02SV602)
  MAX_MC_LL_SEND_SIZE(32767)
  CONV_SECURITY_VERIFICATION(NO)
  PARALLEL_SESSION_SUPPORT(YES);

DEFINE_PARTNER_LU
  FQ_PARTNER_LU_NAME(USNETI01.N02SV603)
  PARTNER_LU_ALIAS(N02SV603)
  PARTNER_LU_UNINTERPRETED_NAME(N02SV603)
  MAX_MC_LL_SEND_SIZE(32767)
  CONV_SECURITY_VERIFICATION(NO)
  PARALLEL_SESSION_SUPPORT(YES);

DEFINE_PARTNER_LU
  FQ_PARTNER_LU_NAME(USNETI01.N02SV604)
  PARTNER_LU_ALIAS(N02SV604)
  PARTNER_LU_UNINTERPRETED_NAME(N02SV604)
  MAX_MC_LL_SEND_SIZE(32767)
  CONV_SECURITY_VERIFICATION(NO)
  PARALLEL_SESSION_SUPPORT(YES);

DEFINE_PARTNER_LU
  FQ_PARTNER_LU_NAME(USNETI01.N02SV605)
  PARTNER_LU_ALIAS(N02SV605)
  PARTNER_LU_UNINTERPRETED_NAME(N02SV605)
  MAX_MC_LL_SEND_SIZE(32767)
  CONV_SECURITY_VERIFICATION(NO)
  PARALLEL_SESSION_SUPPORT(YES);

DEFINE_PARTNER_LU
  FQ_PARTNER_LU_NAME(USNETI01.N01P0002)
  PARTNER_LU_ALIAS(N01P0002)
  PARTNER_LU_UNINTERPRETED_NAME(N01P0002)
  MAX_MC_LL_SEND_SIZE(32767)
  CONV_SECURITY_VERIFICATION(NO)
  PARALLEL_SESSION_SUPPORT(YES);
```

Partner Logical Unit Location Definitions

The following partner LU location definitions associate an owning control point name, FQ_OWNING_CP_NAME, with the previously defined partner LUs. The first definition routes any LU name that is prefixed with N02 to the Token Ring node NETCP800, which was previously defined by the DEFINE_LOGICAL_LINK specification. In this example, the DEFINE_LOGICAL_LINK specification defines the 3174 host sub-area gateway. The second definition matches peer LU N01P0002 with the previously defined link definition NETCP802. See “Logical Link Definitions” on page 208 for the link definition NETCP802.

```
DEFINE_PARTNER_LU_LOCATION
  FQ_PARTNER_LU_NAME(USNETI01.N02 )
  WILDCARD_ENTRY(PARTIAL)
  FQ_OWNING_CP_NAME(USNETI01.NETCP800)
  LOCAL_NODE_NN_SERVER(NO);

DEFINE_PARTNER_LU_LOCATION
  FQ_PARTNER_LU_NAME(USNETI01.N01P0002)
  WILDCARD_ENTRY(NO)
  FQ_OWNING_CP_NAME(USNETI01.NETCP802)
  LOCAL_NODE_NN_SERVER(NO);
```

Mode Definitions

The following mode definition provides two classes of information: maximum session constraints and performance tuning metrics.

```
DEFINE_MODE MODE_NAME(MAPPCIND)
  COS_NAME(#CONNECT)
  DEFAULT_RU_SIZE(YES)
  RECEIVE_PACING_WINDOW(4)
  MAX_NEGOTIABLE_SESSION_LIMIT(32767)
  PLU_MODE_SESSION_LIMIT(12)
  MIN_CONWINNERS_SOURCE(6);
```

Default Communication Specifications

The default communication specifications are established with DEFINE_DEFAULTS. Setting IMPLICIT_INBOUND_PLU_SUPPORT to YES avoids having to configure any potential partner LU that may initiate communication with the workstation. The DEFAULT_MODE_NAME definition specifies a MODE_NAME to use if the one presented on an allocation request is undefined.

Thus, if you do not define the default mode name, SASAPPC, and you do not define the APPC_LU62MODE environment variable, the DEFAULT_MODE_NAME is used. A DEFAULT_LOCAL_LU_ALIAS may also be defined. If it is absent, the local control point LU is the active default. See CP_ALIAS of DEFINE_LOCAL_CP in “Local Control Point and Single Logical Unit” on page 208.

If the APPC_LUNAME environment variable is not set, the active default LU is used.

```
DEFINE_DEFAULTS IMPLICIT_INBOUND_PLU_SUPPORT(YES)
  DEFAULT_MODE_NAME(MAPPCIND)
  MAX_MC_LL_SEND_SIZE(32767)
  DIRECTORY_FOR_INBOUND_ATTACHES(*)
  DEFAULT_TP_OPERATION(NONQUEUED_AM_STARTED)
  DEFAULT_TP_PROGRAM_TYPE(BACKGROUND)
  DEFAULT_TP_CONV_SECURITY_RQD(NO)
  MAX_HELD_ALERTS(10);
```

SASRMT and SASTP62 Transaction Programs

The following two transaction programs are required to allow a SAS/CONNECT remote session to run on the OS/2 workstation. For SAS/SHARE, the transaction program SASTP62 is required.

```
DEFINE_TP TP_NAME(SASRMT)
  DESCRIPTION(Remote SAS Initiation TP)
  FILESPEC(C:\SAS\SAS.EXE)
  PARM_STRING(-DMR -CONFIG C:\SAS\CONFIG.DMR)
  CONVERSATION_TYPE(EITHER)
  CONV_SECURITY_RQD(NO)
  SYNC_LEVEL(EITHER)
  TP_OPERATION(NONQUEUED_AM_STARTED)
  PROGRAM_TYPE(PRESENTATION_MANAGER)
  RECEIVE_ALLOCATE_TIMEOUT(INFINITE);

DEFINE_TP TP_NAME(SASTP62)
  DESCRIPTION(Internal SAS Service TP)
  FILESPEC(C:\SAS\SASTP62.EXE)
  CONVERSATION_TYPE(EITHER)
```

```
    CONV_SECURITY_RQD(NO)
    SYNC_LEVEL(EITHER)
    TP_OPERATION(QUEUED_OPERATOR_PRELOADED)
    PROGRAM_TYPE(BACKGROUND)
    INCOMING_ALLOCATE_QUEUE_DEPTH(255)
    INCOMING_ALLOCATE_TIMEOUT(120)
    RECEIVE_ALLOCATE_TIMEOUT(60);
```

Attach Manager Start

The final statement in this example NDF starts the Attach Manager agent. The Attach Manager is needed to enable SAS/CONNECT users to sign on to the workstation or for SAS/SHARE users to connect to a server that is running on the workstation.

```
START_ATTACH_MANAGER;
```

References

For complete details on how to configure the OS/2 system before you can use the APPC access method, refer to the following documents:

SNA Technical Overview (GC30-3073)

SNA Formats (GA27-3136)

VTAM Programming for LU6.2 (SC30-3400)

Extended Services for OS/2 Communications Manager User's Guide (S04G-1015)

Extended Services for OS/2 Communications Manager Configuration Guide (S04G-1002)

Extended Services for OS/2 Problem Determination Guide for the Service Coordinator (S04G-1006)

Extended Services for OS/2 Programming Services and Advanced Problem Determination for Communications (S04G-1007)

Extended Services for OS/2 Communications Manager System Management Programming Reference (S04G-1116)

Extended Services for OS/2 APPC Programming Reference (S04G-1025)

Contact IBM for information about 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

Copyright © 1999 by SAS Institute Inc., Cary, NC, USA.

ISBN 1–58025–479–9

All rights reserved. Printed in the United States of America. No part of this publication may be reproduced, stored in a retrieval system, or transmitted, in any form or by any means, electronic, mechanical, photocopying, or otherwise, without the prior written permission of the publisher, SAS Institute Inc.

U.S. Government Restricted Rights Notice. Use, duplication, or disclosure of the software by the government is subject to restrictions as set forth in FAR 52.227–19 Commercial Computer Software-Restricted Rights (June 1987).

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

1st printing, September 1999

SAS® and all other SAS Institute Inc. product or service names are registered trademarks or trademarks of SAS Institute Inc. in the USA and other countries.® indicates USA registration.

IBM®, ACF/VTAM® , AIX® , APPN® , MVS/ESA® , OS/®2® , OS/390® , VM/ESA® , and VTAM® are registered trademarks or trademarks of International Business Machines Corporation. ® indicates USA registration.

Other brand and product names are registered trademarks or trademarks of their respective companies.

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