CHAPTER 23

Windows: APPC Access Method

SAS Support for APPC on Windows

Note: The APPC communications access method can be used on Windows NT, Windows 95, Windows 98, and Windows 32s platforms. You can use the Windows NT, Windows 95, and Windows 98 platforms for SAS/CONNECT local and remote hosts and for SAS/SHARE client and server hosts. The Windows 32s platform supports both the SAS/SHARE client and the server roles, but it is restricted to the SAS/CONNECT local host role only.

However, beginning with Version 7, the Windows 32s platform is not supported. Information about Windows 32s is included for Version 6 users. △

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

System Administrator or User

To use the APPC access method with a Windows 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 both the local and remote hosts at your site.
2. SAS has been installed on both the local and remote hosts.

The APPC access method gives you access to an SNA network. SAS/CONNECT and SAS/SHARE software use the Microsoft Windows Open Services Architecture (WOSA) standard (WinAPPC). Therefore, you should be able to use software from any vendor that supports this standard. The following package has been verified by SAS Institute:

- □ the Microsoft SNA Server, Version 2.11 SP1 (Service Pack 1) or subsequent release.
- □ any program that supports the WOSA APPC standard.

Sample configuration files for Windows NT, Windows 95, Windows 98, and Windows 32s are included on the install media in *!SASROOT*\CONNECT\SASMISC and *!SASROOT*\SHARE\SASMISC.

Configuring a Microsoft Server Environment

Network Administrator

Note: The following applies to configuring the Microsoft SNA Server only. If you are using another communications product, refer to that product's configuration instructions. △

An IBM Systems Network Architecture (SNA) network enables a Windows 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 along with the APPC access method, you must install and configure the Microsoft SNA Server.

Optionally, for Windows NT only, you may want to configure Windows NT Host Security Integration features that simplify the login procedure and manage multiple passwords that you may need for accessing Windows NT and host security domains.

See "Installing and Configuring a Microsoft Server Environment" on page 345 for information about configuring Windows NT Host Security Integration features and installing an SNA server and SNA clients.

Understanding SNA Server Terminology

Familiarity with these terms will help you when you talk to your network administrator about selection options.

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

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

For more information about this terminology, see the *Microsoft SNA Server Administration Guide*.

Setting SAS Options

You may need to set specific options in SAS 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, as follows:

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

OPTIONS SET=*variable-name value*;

Example:

```
options set=appc_secure _none_;
```

- □ SAS configuration file or on SAS invocation:

  -SET *variable-name value*

  Example:

  ```
  -set appc_secure _none_
  ```

- □ DOS operating system environment variable:

  SET *variable-name*=*value*

  Example:

  ```
  set appc_secure=_none_
  ```

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

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 specifying security information for SAS/CONNECT and SAS/SHARE.

Providing Client Identification in a Version 8 Session

Note: In the Windows environment, SAS/SHARE server security is supported on the Windows NT platform only. △

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

CAUTION:

In order to make a SAS/SHARE server secured, the APPC_SECURE= option must be set at a SAS/SHARE server that can run on any host. △

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. On Windows NT only, the username can also include the domain name, which locates the specified username in a domain.

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

For SAS/CONNECT and SAS/SHARE, you must set the APPC_SECURE variable in order to pass a remote host userid and a 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* | _SAME_

NONE

must be set at the SAS/CONNECT local host or the SAS/SHARE client. This is 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 userid and password provides more security than assigning the userid and password to the variable.

userid.password

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

This option specifies both the userid and password. Assigning the userid and the password directly to the APPC_SECURE variable at the SAS/CONNECT local host or the 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 variable in a file allows anyone to read it.

If the userid or the password contains numeric or special characters, enclose the entire *userid.password* in quotation marks.

SAME

CAUTION:

Windows NT only This value is supported on Windows NT only. △

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

SAME offers the convenience of not having to specify a userid and password to APPC_SECURE. Setting _SAME_ automatically obtains the appropriate host userid and password from the NT Host Security Integration system. In order to take advantage of this feature, you must have installed and configured the NT Host Security Integration feature in your Windows NT environment. For details about installing and configuring this feature, see “Configuring Windows NT Host Security Integration Features” on page 345.

Examples:

```
options set=appc_secure _none_;
options set=appc_secure _prompt_;
options set=appc_secure bass.timego;
options set=appc_secure "apex\bass.timego";
options set=appc_secure bass.time2go;
options set=appc_secure _same_;
```

See “Setting SAS Options” on page 333 for examples of the forms that you can use to specify APPC_SECURE.

SAS/CONNECT and SAS/SHARE Options

APPC_LUNAME

specifies the name of the local LU alias to use. You must declare APPC_LUNAME unless a default local APPC LU has been defined. A default APPC LU is defined when setting up the Microsoft SNA Server. You use the value assigned to the APPC_LUNAME when making a remote host connection with SAS/CONNECT or when accessing a SAS/SHARE server.

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

APPC_LU62MODE

specifies the mode name that is associated with an LU-LU pair and determines the session properties for that pair.

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 and use the same mode name value in both the local session and the remote environments (either on the remote host in a SAS/CONNECT session or on a SAS/SHARE server, as necessary).

APPC_PARTNER_COUNT

specifies the maximum number of simultaneous remote hosts that this local session has at any one time. This estimate permits better allocation of memory for resources for internal control block usage.

SAS/CONNECT Only Options

APPC_SURROGATE_LUNAME

specifies an LU to use for a SAS/CONNECT remote session on an OS/390 host. If APPC_SURROGATE_LUNAME 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.

Note: This option applies only when connecting to an OS/390 remote host. △

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

APPC_DEPLU

enables dependent LU processing support by releasing unused sessions.

In order for APPC_DEPLU to work properly, when specifying the connection name during Microsoft SNA Server configuration, you must configure the remote LU as a dependent host connection. See “Configuring the Server” on page 347 for the context for this requirement.

SAS/SHARE Only Option

APPC_USER

identifies user output in the server output log.

Note: Must be set at the SAS/SHARE client. △

SAS/CONNECT

Local Host Tasks

User or Applications Programmer

To connect a Windows local host to a remote host, perform these tasks at the local host:

1 Specify the APPC communications access method.

2 Specify the remote host 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, using 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*.

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 a Windows local host to a remote host, use the following syntax:

```
OPTIONS REMOTE=remote-LU-alias;
```

where *remote-LU-alias* specifies the logical unit of the remote host that you are connecting to. Ask your network administrator for the *remote-LU-alias*. Types of valid values follow:

Table 23.1 Windows APPC SAS/CONNECT REMOTE= Values

Type of Remote Host	Remote Host Identifier
OS/390	name of APPC/MVS scheduler LU
CMS	name of AVS (APPC/VM VTAM Support) private gateway LU for VM system
VSE	name of VTAM APPL ID (ACBNAME) that was set up for APPC LU6.2 communications

Type of Remote Host	Remote Host Identifier
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

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 used on a Windows local host that uses the APPC access method because APPC has the ability to communicate directly with the remote host. 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 334.

Although no errors are produced if you specify a script file, you do waste processing time. If you defined the RLINK fileref before establishing a connection, when you sign on, SAS/CONNECT processes and loads the script file identified by the fileref, but the APPC access method ignores the script.

If you do not want to omit the RLINK fileref but you want to avoid wasting processing time, 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 a Windows local host configuration file to connect to a remote host with the APPC access method:

```
-set appc_luname locallu
-set appc_lu62mode appcmode
```

LOCALLU is the name of the *local-LU-alias* that is defined at the Windows NT SNA Server. APPCMODE is the *mode-name* that is defined in the Windows NT SNA server.

The following example shows the statements that you specify in a local SAS session:

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

The APPC communications access method is declared with a connection to the remote host REMOTELU. In this example, REMOTELU identifies a local LU that is defined at the Microsoft SNA Server. The SIGNON statement performs the sign-on process. 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 the local host to make a remote host connection, 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

You must declare a remote host name at both the local host and the remote host in a SAS/CONNECT session. At both hosts, specify an OPTIONS statement, using 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 23.1 on page 338 for valid values.

The remote host identifiers that you specify at both the local and 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 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 a 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. NOTERMINAL 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

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 requires that you explicitly type EXIT at the remote host, and the display gives the appearance that the REMOTE SUBMIT command is hung.

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

See *SAS Companion for the Microsoft Windows Environment* for details about this option.

Remote Host Example

The following example illustrates the statements that you specify in a Windows NT, a Windows 95, or a Windows 98 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
-icon
-sasdmr msgqueue
-no$syntaxcheck
-noterminal
-noxwait
```

The APPC communications access method is declared with a connection to a *local-LU-alias* REMOTELU.

Note: The value for 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 For Windows NT only, set security for connecting clients.

2 Specify the APPC access method.

3 Know how to specify a server name.

Setting Security for Connecting Clients

CAUTION:

Windows NT only Server security is supported on the Windows NT platform only. △

If the network administrator specified session security in the SASTP62 TP definition, clients must have secure userids and passwords.

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 334.

Specifying the APPC Communications 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 a Windows NT client follows:

```
-comamid appc
-comaux1 tcp
```

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

Specifying a Server Name

The server name that you specify in the PROC OPERATE statement and the LIBNAME statement must be defined as the *local-LU* at the SAS/SHARE server and as a *remote-LU-alias* at the client computer. For complete information about defining appropriate LUs for use with SAS/SHARE, see “Setting SAS Options” on page 333 and “Installing and Configuring a Microsoft Server Environment” on page 345.

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

Examples of specifying the server name follow:

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

In this example, at the client computer, the server name is expressed as a *remote-lu-alias*, which is a name that refers to a *remote-LU*.

If the server is running on a CMS system that is connected to your system through a global VTAM AVS gateway, you must use a two-level server name specification as follows:

```
libname demo 'demo a' server=gateway.server;
```

where *gateway* is defined to the CMS system as the AVS-gateway LU.

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

Client Example

The following example illustrates the statements that you specify in a Windows NT client configuration file to access a server with the APPC access method.

```
-set appc_luname locallu
-set appc_lu62mode appcmode
```

LOCALLU is the name of a *local-LU-alias* and APPCMODE is the mode name that are defined at the Windows NT SNA server.

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

```
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. To access a server that is running on the Windows NT platform, specify *remote-LU-alias* for the server name.

Server Tasks

Server Administrator

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

1. Configure APPC conversation security.
2. Specify the APPC access method.
3. Specify the server name.

Configuring APPC Conversation Security

You can authenticate users at the server, but you cannot control the file authorization.

For the APPC access method on Windows NT, Windows 95, Windows 98, and Windows 32s, you can authenticate users at the server by setting up APPC session security within the SASTP62 TP (transaction program) definition.

However, securing user authorization for file access is not possible because the underlying APPC subsystem performs the authentication, preventing the server application from obtaining user password information.

Without user password information, it is impossible to switch user contexts for validating user authorization. This is an APPC implementation limitation.

For further details about setting up TP definitions, see “SASTP62 Transaction Program” on page 350.

A secure server allows connections only from those clients that provide valid userids and passwords for the host on which the server is running. Requiring connecting clients to supply a valid userid and password enforces server security.

Specifying the APPC Access Method at the Server

You must specify the APPC communications access method before you can start 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 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 a Windows NT 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 a 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-alias* at the client computer. For complete information about defining appropriate LUs for use with SAS/SHARE, see “Setting SAS Options” on page 333 and “Installing and Configuring a Microsoft Server Environment” on page 345.

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

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 SAS session on the Windows NT host at which you start a server:

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

The APPC access method is declared, and the server SHARE1, which is the *local-LU*, is started on the Windows NT host.

Installing and Configuring a Microsoft Server Environment

Two features of the Microsoft server environment are:

Windows NT Host Security Integration
For Windows NT only, may be optionally configured.

SNA Server Gateway
must be installed and configured.

Configuring Windows NT Host Security Integration Features

Windows NT Host Security Integration provides a simplified logon procedure and maintains multiple passwords that you may need for accessing Windows NT and host security domains. This feature provides

- secure storage for user host accounts
- tools for maintaining the same password in all environments
- automatic logon to host computers in an enterprise computing environment.

This approach coordinates existing host security databases, rather than forcing a migration to a single, shared security database.

Windows NT Host Security Integration contains these components, which must be installed and configured:

Host Account Synchronization Service
supports third-party interfaces to various host security databases in order to coordinate password changes between Windows NT security and the respective hosts. For example, this service sends information about password changes in the Windows NT domain to host computers or sends notification of host password changes to the Windows NT Account Synchronization Service.

You install this service during configuration of the Microsoft SNA Server. This feature is installed by selecting the Security Integration Service option of SNA Server Setup.

Windows NT Account Synchronization Service
synchronizes passwords between the host and the Windows NT domain. When passwords are changed in either the Windows NT domain or on the host, this service updates the Host Account Cache and the other host domains through the Host Account Synchronization service. Users may then access files, printers, databases, messaging systems, and other applications throughout the network using the same password. The Windows NT Account Synchronization service

coordinates internal operation of the other services and must be installed even if automatic password synchronization is not installed.

You install this service on the Primary Domain Controller. Installation software and instructions are located in the SNA Server HOSTSEC folder.

Host Account Cache

contains a database that maps host user IDs and passwords to Windows NT usernames and passwords. The Host Account Cache service runs on primary domain controllers of Windows NT domains in which the SNA Server is running. One Host Account Cache can be shared among many SNA Server subdomains.

You install this service on the Primary Domain Controller. Installation software and instructions are located in the SNA Server HOSTSEC folder.

For complete details about installing these services, see *Microsoft SNA Server Getting Started* and *Installing and Configuring Host Security Integration*. This document is available from the Microsoft Corporation.

Configuring a Microsoft SNA Server

Central to the SNA network is the SNA server, which is responsible for performing connections between local and remote hosts on a local area network (LAN).

Note: An unqualified reference to Windows means all Windows platforms - Windows NT, Windows 95, Windows 98, and Windows 32s. △

Network Administrator

To set up an SNA network perform the following tasks:

1 Install the SNA server.

2 Configure the SNA server.

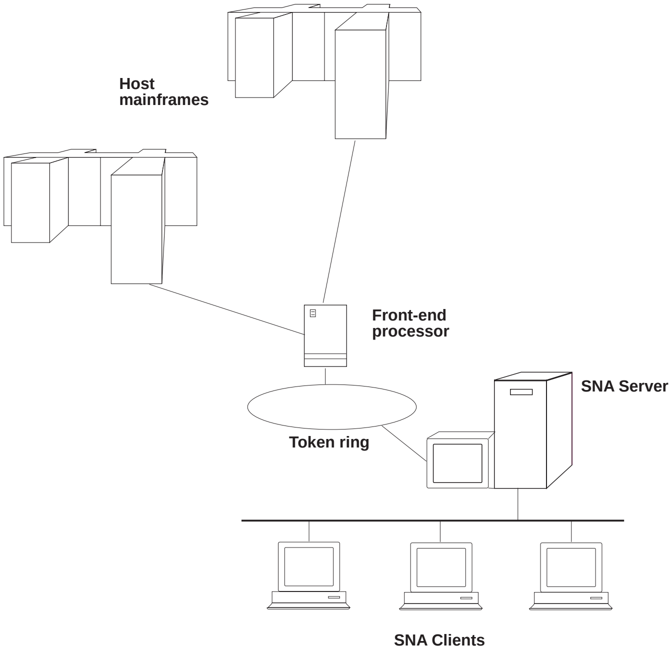
3 Configure SNA clients.

4 Optionally configure Host Account Synchronization Service. For details about installing this service, see *Microsoft SNA Server Getting Started* and *Installing and Configuring Host Security Integration*.

You install and configure the SNA server on a Windows NT computer, and you configure SNA clients on any of three Windows platforms: Windows NT, Windows 32s, Windows 95, and Windows 98. After the SNA server and the SNA clients are installed and configured, users of SAS/CONNECT and SAS/SHARE can make connections transparently from their local hosts to the remote hosts that they want by using the SNA server.

The following figure shows an SNA network.

Figure 23.1 Typical SNA Network Configuration



Installing the SNA Server

It is assumed that you already have completed installation of the SNA server product. Before you can configure the SNA server, verify that the following tasks have been completed.

1 Install the appropriate drivers (for example, 802.2 Token Ring, Ethernet, or X.25).

2 Select the appropriate networking protocols (for example, DLC for LAN).

3 Install the SNA Server link services.

   Link services define the protocol that is used between the SNA Server software and the communications adapters installed in your computer (for example, 802.2 Token Ring, Ethernet, SDLC, or X.25).

   If the setup program detects more than one network operating system on your computer, you must specify which systems you are using (for example, Microsoft LAN Manager, Novell Netware, or both).

Configuring the SNA Server

After you have installed the SNA server and link supports, use the SNA Server Administration Program (SNA Server Admin) to perform the following tasks:

1 Configure the server.

2 Specify connections.

3 Configure Logical Units (LU) (local and remote).

4 Define LU-to-LU pairs and modes.

Configuring the Server

Configure the SNA server by specifying the local Network Name and Control Point Name.

Specifying the Connection Name

Specify the connection name (for example, 802.2 Token Ring, SDLC, or X.25) and other properties that are appropriate to your configuration.

Connection properties are the software components of the SNA server that communicate through the device driver to a particular communications adapter.

Configuring Logical Units

Perform the following steps to configure the desired number of logical units:

1 Configure the desired number of local logical units (LUs) to be used.

   An LU may be dependent or independent. An LU's ability to perform dependently or independently in a SAS/CONNECT remote host session is based on the communications software that your network uses.

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

   SAS/SHARE requires an independent LU. When using Remote Library Services (RLS), SAS/CONNECT also requires an independent LU.

2 Configure the desired number of remote logical units (LUs) to be used.

   You must define all remote LUs to the SNA Server because the Microsoft SNA Server does not support end node (EN) Advanced Peer-to-Peer Networking (APPN) capabilities. You must also define remote (or partner) LUs to connect to a remote host with SAS/CONNECT or to access a SAS/SHARE server.

   When defining a remote LU for an MVS remote host in a SAS/CONNECT session, you must also account for the LU pooling capabilities of the remote MVS SAS session so that appropriate surrogate LUs are defined to the SNA Server.

   Your options for defining remote LUs follow:

   □ If using the APPC access method, you may assign a specific LU to the APPC_SURROGATE_LUNAME variable. For information about the APPC_SURROGATE_LUNAME variable, see "SAS/CONNECT Only Options" on page 337. In this case, you must define this remote LU to the SNA server.

   □ You may allow the remote OS/390 session to supply an LU from a pool of LUs. Although you do not need to assign a value to APPC_SURROGATE_LUNAME, you must define to the SNA server all possible MVS LU names that reside in this surrogate LU pool.

   □ Instead of defining surrogate LUs to the SNA server, you may configure the local LU to accept Implicit Incoming Remote LUs.

Defining LU-LU Pairs and Communications Mode Properties

Define the local LU-remote LU pairs, and specify the properties of the communications mode to be used between each pair.

If site-naming conventions permit, specify the mode name SASAPPC. The APPC access method uses this mode name if the APPC_LU62MODE variable has not been defined. See "SAS/CONNECT and SAS/SHARE Options" on page 337 for information about setting APPC_LU62MODE.

Specify the minimum contention-winner parameter, which is relevant for SAS software because only contention-winner sessions are used for locally initiated communication.

Communication between SAS/CONNECT local and remote hosts requires only one contention-winner session. However, this limit affects the number of data sets that can be accessed concurrently by means of the SAS/CONNECT Remote Library Services or a SAS/SHARE server.

When defining session limits, define enough sessions so that session limits will never be reached. If session limits are reached, the next time a user attempts to connect to a remote host with SAS/CONNECT or a client host attempts to access a SAS/SHARE server, the APPC layer will not return to the application layer until a session is available. Although a lengthy wait may seem like an error condition (such as no response from SAS or a loop), the underlying APPC layer is waiting for a session to become available.

You have completed the configuration of a Windows NT SNA server.

Configuring a Windows SNA Client

Client configuration tasks are based on the platform on which the client is running:

- □ Windows NT
- □ Windows 95
- □ Windows 98
- □ Windows 32s.

Tasks for configuring a Windows NT, a Windows 95, and a Windows 98 client are identical.

Configuring a Windows NT, a Windows 95, or a Windows 98 Host as a Local Host or a SAS/SHARE Client

Use the SNA Server Client Setup Program to configure Windows NT, Windows 95, and Windows 98 clients.

This section highlights the general tasks that you perform to configure an SNA client. For complete details, see the *Microsoft SNA Server Installation Guide* and the *Microsoft SNA Server Administration Guide*. (Contact the Microsoft Corporation for information about this documentation.)

Perform the following tasks:

1. Identify the transport (for example, client/server protocols) for communication with the SNA server (e.g., TCP/IP, Named Pipes, or IPX/SPX).
2. Specify the network domain in which a server can be located so that data can be routed to it over a local area network (LAN).
3. Specify client mode (local or remote).

If you intend to use the Windows NT, Windows 95, or Windows 98 computer strictly as a local host connecting to a remote host in a SAS/CONNECT session or as a client accessing a SAS/SHARE server, you have successfully completed the configuration process.

Configuring a Windows NT, a Windows 95, or a Windows 98 Host as a Remote Host or as a SAS/SHARE Server

If you intend to use either of these computers as a remote host for a SAS/CONNECT session or as a SAS/SHARE server, you must perform additional configuration tasks.

1. Configure TPs (transaction programs) that can be invoked.

   You may execute a program named TPSETUP.EXE, which was supplied by Microsoft and enhanced by SAS Institute, to configure and modify TP properties. Running the program automatically adds entries to the registry, which is a configuration file.

For Windows NT, the TPSETUP.EXE program is located at !*sasroot*\CORE\WINNT\TPSETUP.EXE. For Windows 95 and Windows 98, the TPSETUP.EXE program is located at !*sasroot*\CORE\WIN95\TPSETUP.EXE.

The TPSETUP.EXE command with arguments follows:

TPSETUP <-TP *TP-name*>
<-EDIT>
<-LU *local-LU-alias*>
<-CMD '*SAS-command-line*'>

where:

-TP *TP-name*
specifies a transaction program. Two TPs are provided: SASRMT and SASTP62. See "SASRMT Transaction Program" on page 350 and "SASTP62 Transaction Program" on page 350 for more information about these programs.

-EDIT
allows you to modify an existing -TP entry in the registry. The TP configuration dialog window opens, showing previously entered TP properties. Omission of the -EDIT argument invokes an empty dialog window, where you define properties.

-LU *local-LU-alias*
is the client's alias for the local-LU that you configured at the SNA server. The assignment of a *local-LU-alias* to a specific Windows NT, Windows 95, or Windows 98 client allows the SNA server to route the incoming request for attachment (ATTACH) to the appropriate client computer by alias name. The SNA server checks each client's configuration file for its *local-LU-alias* to determine where to route the request. If you are going to define SASTP62 and would like more than one remote window host, you must specify the *local-LU-alias*.

-CMD *command-line*
specifies a command that automatically executes SAS at the remote host when a local host connects to it in a SAS/CONNECT session. A command line is required only when you have defined the SASRMT transaction program.

SASTP62 Transaction Program

For Version 7 and later, the APPC access method performs dynamic TP naming which automatically generates the SASTP62 TP definition for you.

Note: Dynamic TP naming is only available when communicating between two Version 7 (or later) sessions. △

For Version 6.12 and earlier, you must still define a SASTP62 transaction program at the host where a SAS/SHARE server will be executing or where a remote SAS/CONNECT session will be established.

To define the SASTP62 transaction program in the registry, specify the following command:

```
TPSETUP -TP SASTP62
```

SASRMT Transaction Program

You must define a SASRMT transaction program at the host where a remote SAS/CONNECT session will be established.

To define the SASRMT transaction program in the registry, specify the following command:

```
TPSETUP -TP SASRMT
```

Note: For Windows NT, you must run the SASRMT transaction program as an application, not as an NT service. Therefore, make sure that you start the Microsoft program TPSTART.EXE before allowing users to establish SAS/CONNECT sessions with remote hosts. △

You have completed the configuration of the Windows NT, Windows 95, or Windows 98 host as a SAS/CONNECT remote host or a SAS/SHARE server.

Configuring a Windows 32s Client

Use the SNA Server Client Setup Program to configure a Windows 32s client.

This section highlights the general tasks that you perform to configure an SNA client. For complete details, see the *Microsoft SNA Server Installation Guide* and the *Microsoft SNA Server Administration Guide*.

Note: A Windows 32s host is supported only as a local host that connects to a remote host in a SAS/CONNECT session. △

Perform the following tasks:

1. Identify the protocol for communication with the SNA server (for example, TCP/IP, Novell, or IPX/SPX).
2. Specify the domain in which a server can be located so that data can be routed to it over a local area network (LAN).
3. To enable the client to connect automatically to the SNA server when booted, insert the following variable assignment in the [windows] section of the WIN.INI configuration file:

   ```
   [windows]
   load=wnap.exe
   ```

4. If you intend to create or to access a SAS/SHARE server on this host, you must define the SASTP62 transaction program in the WIN.INI file.

   You either may edit the WIN.INI file directly, or you may execute a program named TPSETUP.EXE, which is supplied by Microsoft, that automatically updates the configuration file.

   The appropriate entries to the WIN.INI file follow:

   ```
   [SNAServerAUTOTPs]
   SASTP62=sastp62
   ```

   Define the local-LU that you configured at the SNA server by inserting the following variable assignments in the [sastp62] section of the WIN.INI file:

   ```
   [sastp62]
   LocalLU=local-LU-alias
   Queued=operator
   ```

   The assignment of a *local-LU-alias* to a specific Windows 32s client allows the SNA server to route the incoming request for attachment (ATTACH) to the appropriate Windows 32s computer by an alias name. The SNA server checks each Windows 32s client's configuration file for its *local-LU-alias* to determine where to route the request.

You have completed the configuration of the Windows 32s client.

References

For details about how to install and configure the SNA server and SNA clients using the SNA Server Setup Program, see the *Microsoft SNA Server Installation Guide* and the *Microsoft SNA Server Administration Guide.*

For details about configuring Windows NT Host Security Integration features, see *Microsoft SNA Server Getting Started* and *Installing and Configuring Host Security Integration*.

Contact the Microsoft Corporation to obtain 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

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.