



CHAPTER

19

UNIX: APPC Access Method

<i>Tasks That Are Common to SAS/CONNECT and SAS/SHARE</i>	266
<i>System and Software Requirements for SAS/CONNECT and SAS/SHARE</i>	266
<i>AIX System Requirements</i>	266
<i>HP-UX System Requirements</i>	266
<i>Solaris System Requirements</i>	266
<i>Configuring the APPC Access Method</i>	267
<i>Understanding APPC Communications Terminology</i>	267
<i>Setting Variables in SAS</i>	267
<i>Setting Security for SAS/CONNECT and SAS/SHARE</i>	268
<i>Providing Client Identification in a Version 8 Session</i>	268
<i>Providing Client Identification in a pre-Version 8 Session</i>	269
<i>SAS/CONNECT and SAS/SHARE Variables</i>	270
<i>SAS/CONNECT Only Variable</i>	271
<i>SAS/CONNECT</i>	271
<i>Local Host Tasks</i>	271
<i>Specifying the APPC Communications Access Method</i>	271
<i>Specifying the Remote Host Name</i>	272
<i>Signing On to the Remote Host</i>	273
<i>Local Host Example</i>	273
<i>Remote Host Example</i>	274
<i>SAS/SHARE</i>	274
<i>Client Tasks</i>	274
<i>Setting Security for Connecting Clients</i>	274
<i>Specifying the APPC Access Method</i>	274
<i>Specifying a Server Name</i>	275
<i>Client Example</i>	276
<i>Server Example</i>	276
<i>System Configuration for the APPC Communications Environment</i>	276
<i>Configuring the APPC Communications Environment for AIX</i>	276
<i>Sample AIX Configuration File</i>	277
<i>References</i>	280
<i>Configuring the APPC Communications Environment for HP-UX</i>	280
<i>Configuring SNAplusLink</i>	280
<i>Configuring SNAplusAPI</i>	281
<i>Sample HP/UX Configuration File</i>	281
<i>References</i>	286
<i>Configuring the APPC Communications Environment for Solaris</i>	286
<i>Installing and Configuring Link Components</i>	287
<i>Adding Gateways to Your Network</i>	287
<i>Ensuring that the Appropriate Daemons Are Running</i>	287
<i>Sample Solaris Configuration File</i>	287

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

System Administrator or User

To use the APPC access method with a UNIX 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 variables in SAS software.
- 4 Set the desired SAS/CONNECT and SAS/SHARE variables.

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 software is installed on both the local and remote hosts.

The system requirements for the APPC access method in the UNIX environments are based on which vendor's type of UNIX operating system that you are using. SAS Institute supports the APPC access method on the following types of UNIX systems:

AIX
 HP-UX
 Solaris

AIX System Requirements

To use the APPC access method under AIX, your site must meet the following requirements:

- run Version 3.2.5 or subsequent version of the AIX operating system
- run AIX SNA Server/6000 Version 2.1.1 or subsequent version
- run eNetwork Communications Server for AIX.

HP-UX System Requirements

To use the APPC access method under HP-UX, your site must meet the following requirements:

- run Version 10.20 of the HP-UX operating system
- run HP-UX SNAplusAPI and SNAplusLink Release 3 or a subsequent release. Recommended patches at Release 3/Dart 16 are PHNE_4773, PHNE_5314, and PHNE_5374.

Solaris System Requirements

Note: The Solaris platform is not supported in Version 7 or later. Δ

To use the APPC access method under Solaris, your site must meet the following requirements:

- run Version 2.6 of the Solaris operating system

- run SunLink SNA Peer-to-Peer Version 8.0 or subsequent version. Recommended patches for version 8.0 are 102146, 102147, and 102229.

Configuring the APPC Access Method

Network Administrator

Before you can use SAS/CONNECT and SAS/SHARE with the APPC access method, you must first configure the SNA LU6.2 APPC communications environment for your UNIX systems. Separate tasks are provided for each type of supported UNIX system: AIX, HP-UX, and Solaris. See “System Configuration for the APPC Communications Environment” on page 276 for more information.

Understanding APPC Communications Terminology

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

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 will connect to a SAS/CONNECT remote host or with a client that will access a SAS/SHARE server.

partner 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 Communications Environment” on page 276.

Setting Variables in SAS

You may need to set specific variables 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 variables must be set and what values to assign to them.

You may specify a variable in any of several forms, as follows:

Note: In these examples, the variable name varies according to the form used. APPC_SECURE is the environment variable form; APPCSEC is the macro variable form. Δ

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

`-SET variable-name value`

Example:

`-set appc_secure _none_`

- as a SAS macro variable:

`%LET variable-name=value;`

Example:

```
%let appcsec=_none_;
```

- as a UNIX environment variable in a shell or a profile file:

Korn shell: `export VARIABLE-NAME=value`

C shell: `setenv VARIABLE-NAME value`

Examples:

```
export APPC_SECURE=_NONE_
setenv APPC_SECURE _NONE_
```

Values for these variables may 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 variable, here is the order of precedence that is followed:

SAS macro variable

SAS invocation

SAS configuration file

UNIX 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` variable is still accepted but is not recommended in Version 8. The `USER=` and `PASSWORD=` options take precedence over the client `APPCSEC` variable 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 variable 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. Δ

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 assigns 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

Both the environment variable and SAS macro variable forms are provided, as appropriate. Use the SAS macro variable for run-time specification.

APPC_SECURE (*HP-UX, Solaris, and AIX*)
is an environment variable.

APPCSEC
is a SAS macro variable.

NONE

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

Setting this value 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 a password provides more security than assigning the userid and password to the environment variable or the macro variable.

userid.password

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

This value specifies both the userid and password. Assigning the userid and the password directly to the APPCSEC variable at the SAS/CONNECT local host or at 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.

Note: You must specify security for OS/390, CMS, and VSE remote hosts. However, security is not required for OS/2 or Windows hosts unless session security has been explicitly defined. Δ

Macro variable examples:

```
%let appcsec=_none_;
%let appcsec=_prompt_;
%let appcsec='bass.time2go';
```

SAS/CONNECT and SAS/SHARE Variables

APPC_GATEWAY (*Solaris only*)

is an environment variable.

APPCGATE

is a SAS macro variable.

These variables specify the name of the peer-to-peer gateway that you will attach to.

The gateway provides the SNA stack for connecting to an SNA network. If you are working in a stand-alone environment and your gateway has the same name as your machine, do not specify this variable. By default, SAS uses the **gethostname** UNIX function to obtain the machine name on which SAS is running and then uses that name as the gateway name.

In a SunLink SNA Peer-to-Peer configuration, specify the gateway name either in a local **/etc/appcs** file or in the NIS/NIS+ database.

APPC_LU (*HP-UX only*)

is an environment variable.

APPCLU

is a SAS macro variable.

These variables specify the name of the local LU alias to use.

This name must match an LU alias that is established during configuration. This variable is required unless a default local APPC LU has been defined.

APPC_MODE (*HP-UX* and *AIX*)
is an environment variable.

APPCMODE

is a SAS macro variable.

These variables specify the communication mode that represents the set of networking characteristics that are defined during configuration.

The default communications mode name is SASAPPC. This name must be defined in both the local and remote environments regardless of whether you specify it explicitly or you allow it to default to SASAPPC.

APPC_NET (*HP-UX* and *AIX*)
is an environment variable.

APPCNET

is a SAS macro variable.

These variables specify the network name that is used when forming the fully-qualified remote LU name in APPN environments.

This name is required to exploit APPN connections in the absence of explicitly configured remote LU profiles.

APPC_PARTNER_COUNT (*HP-UX* and *Solaris*)
is an environment variable.

This variable specifies the number of simultaneous partners that this local session will have at one time. This estimate improves allocation of memory resources for internal control block usage.

SAS/CONNECT Only Variable

APPC_SURROGATE_LUNAME (*HP-UX*, *Solaris* and *AIX*) - is an environment variable

This variable specifies which LU to use for a SAS/CONNECT remote session on an OS/390 host.

If this variable 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.

SAS/CONNECT

Local Host Tasks

User or Applications Programmer

To connect a UNIX 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. Use the following syntax:

```
OPTIONS COMAMID=access-method-id;
```

where COMAMID is an acronym for Communications Access Method Identification. *access-method-id* identifies the method used by the local host to communicate with the remote host. APPC (an abbreviation for Advanced Program-to-Program Communication) is an example of *access-method-id*.

Example:

```
options comamid=appc;
```

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

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, use the following syntax:

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

where the *remote-host-id* that you specify at the local host is based on the type of UNIX system that you are running. The following table lists what the remote host identifiers mean on each of the supported UNIX local host types.

Table 19.1 UNIX APPC SAS/CONNECT REMOTE= Values Interpreted at the UNIX Local Host

Type of UNIX Local Host	Remote Host Identifier
AIX	value of partner LU name if APPC_NET is defined. Otherwise, remote LU alias.
HP-UX	partner LU alias.
Solaris	unique session name that is defined in the MODE definition and is a combination of mode name, partner LU, and local LU.

The following table lists what the remote host identifiers mean on each of the supported remote hosts.

Table 19.2 UNIX APPC SAS/CONNECT REMOTE= Values Interpreted at the Remote Host

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

Note: The remote host identifiers that are provided at both the local and remote hosts must be identical. Δ

Example:

```
options remote=remotelu;
```

The remote host identifier that you use is based on the remote host that you connect to.

Alternatively, you may specify the *remote-host-id* in the REMOTE= 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 a UNIX local host that uses the APPC access method because APPC has the ability to initiate a 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 268.

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 that is identified by the fileref, but the APPC access method will ignore 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 UNIX AIX local host configuration file to connect to a remote host with the APPC access method.

```
-set appc_gateway mygate
-set appc_mode appcmode
```

The APPC_GATEWAY environment variable specifies MYGATE as the name of the peer-to-peer gateway to which the local host will attach. The APPC_SECURE variable specifies that connecting local hosts be prompted for a userid and a password that are

valid on the remote host. The APPC_MODE variable specifies the communications mode APPCMODE.

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

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

The APPC communications access method is declared with a connection to the remote host REMOTELU. 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.

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

Remote Host Example

SAS Institute does not provide support for connections to the UNIX remote host with the APPC access method.

SAS/SHARE

Client Tasks

The APPC access method on the UNIX platform supports the SAS/SHARE client only.

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 Know how to specify a server name.

Setting Security for Connecting Clients

Requiring connecting clients to supply both a valid userid and password enforces server security. At the client, set the preferred security method for relaying a userid and a password that are valid on the server host. For details, see “Setting Security for SAS/CONNECT and SAS/SHARE” on page 268.

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 a UNIX client connecting to a Windows NT server follows:

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

If the server cannot be reached using the APPC access 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 and the LIBNAME statements must be defined at the SAS/SHARE server and the client.

The form of the server name is based on the type of host on which the server is running. For the correct form of the server name, ask the network administrator of the appropriate remote host on which the server runs.

The following table specifies server names by host type.

Table 19.3 Remote SAS/SHARE Server Name Types

Type of Server Host	Server Identifier
OS390	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
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

For complete information about defining appropriate LUs for use with SAS/SHARE, see “System Configuration for the APPC Communications Environment” on page 276.

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 a server name follows:

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

In this example, you might specify the name of the server that is running on a Windows NT system (for example, the SNA server LU).

Note: If the server is running on a CMS system that is connected to your system by means of a 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. \triangle

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

Client Example

The following example illustrates the statements that you specify in a UNIX client session to access a Windows NT 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.

Server Example

SAS Institute does not provide support for connections to a server that runs on a UNIX host with the APPC access method.

System Configuration for the APPC Communications Environment

Network Administrator

The procedure that you use to configure the APPC communications environment is based on the type of UNIX system that you are using. See the appropriate section according to the UNIX system type.

AIX	“Configuring the APPC Communications Environment for AIX” on page 276.
HP-UX	“Configuring the APPC Communications Environment for HP-UX” on page 280.
Solaris	“Configuring the APPC Communications Environment for Solaris” on page 286.

Configuring the APPC Communications Environment for AIX

Note: The following applies to configuring the SNA Server/6000 software only. If you are using another communications product, refer to that product's configuration instructions. \triangle

You must perform several SNA Server/6000 configuration tasks before the APPC access method can be used with SAS/CONNECT and SAS/SHARE. Configuration may be unnecessary if other applications at your site already use SNA APPC.

The particular configuration tasks that you must perform are based on the capabilities of your SNA network. If your SNA network supports APPN, you do not have to configure partner logical unit, location, and side information profiles. At a minimum, however, you must configure the following components:

node

defines a set of default parameters that establish operational controls that you can modify. The node profile is created automatically when the SNA Server/6000 product is installed.

control point

defines local SNA aspects regarding PU/LU functionality, including APPN characteristics.

data link control profile and link station profile

defines transport layer attributes, such as network interface type and local control and remote link station.

mode profile

specifies that the mode must be defined on both the local and remote hosts.

The profile definitions are used during session setup to establish flow-control parameters, such as request unit sizes and pacing limits, and to control maximum session limits. If your SNA network does not support APPN, partner logical unit, location, and side information, then profiles must be configured for each potential partner.

Sample AIX Configuration File

A sample AIX configuration file follows:

```
sna:
prof_name           = "sna"
max_sessions       = 200
max_conversations   = 200
restart_action      = once
rrm_enabled        = no
dynamic_inbound_partner_lu_definitions_allowed = yes
standard_output_device = "/dev/console"
standard_error_device = "/var/sna/sna.stderr"
nmvt_action_when_no_nmvt_process = reject
trusted_group_ids  = {system}
comments           = ""

control_pt:
prof_name           = "node_cp"
xid_node_id        = "*"
network_name       = "SASNET01"
control_pt_name_alias = "POCP1001"
control_pt_name     = "POCP1001"
control_pt_node_type = appn_end_node
max_cached_trees   = 500
max_nodes_in_topology_database = 500
route_addition_resistance = 128
comments           = ""
```

```

    partner_lu6.2:
prof_name                = "P0LU2001"
fq_partner_lu_name       = "SASNET01.P0LU2001"
partner_lu_alias         = "P0LU2001"
session_security_supp    = no
parallel_session_supp    = yes
conversation_security_level = none
comments                 = ""

    partner_lu6.2_location:
prof_name                = "P0LU2001"
fq_partner_lu_name       = "SASNET01.P0LU2001"
partner_location_method  = owning_cp
fq_partner_owning_cp_name = "SASNET01.P00U1000"
local_node_is_network_server_for_len_node = no
fq_node_server_name      = "SASNET01.P00U1000"
local_lu_name            = ""
link_station_profile_name = ""
comments                 = ""

    side_info:
prof_name                = "P0LU2001"
local_lu_or_control_pt_alias = "P0CP1001"
partner_lu_alias         = ""
fq_partner_lu_name       = "SASNET01.P0LU2001"
mode_name                = "MODE001"
remote_tp_name_in_hex    = no
remote_tp_name           = ""
comments                 = ""

    link_station_token_ring:
prof_name                = "TR3174"
use_control_pt_xid       = yes
xid_node_id              = "*"
sna_dlc_profile_name     = "TR3174"
stop_on_inactivity       = no
time_out_value           = 0
LU_registration_supported = no
LU_registration_profile_name = ""
link_tracing             = no
trace_format             = long
access_routing_type      = link_address
remote_link_name         = ""
remote_link_address      = 0x400000000001
remote_sap                = 0x04
call_out_on_activation   = yes
verify_adjacent_node     = no
net_id_of_adjacent_node  = "SASNET01"
cp_name_of_adjacent_node = "P00U1000"
xid_node_id_of_adjacent_node = "*"
node_type_of_adjacent_node = learn
solicit_sscp_sessions    = yes

```

```

activate_link_during_system_init      = yes
activate_link_on_demand               = no
cp_cp_sessions_supported              = yes
cp_cp_session_support_required        = no
adjacent_node_is_preferred_server     = yes
initial_tg_number                     = 0
restart_on_normal_deactivation         = no
restart_on_abnormal_deactivation       = no
restart_on_activation                  = no
TG_effective_capacity                 = 4300800
TG_connect_cost_per_time              = 0
TG_cost_per_byte                      = 0
TG_security                           = nonsecure
TG_propagation_delay                   = lan
TG_user_defined_1                     = 128
TG_user_defined_2                     = 128
TG_user_defined_3                     = 128
comments                              = ""

    sna_dlc_token_ring:
prof_name                             = "TR3174"
datalink_device_name                  = "tok0"
force_timeout                         = 120
user_defined_max_i_field              = no
max_i_field_length                    = 30729
max_active_link_stations              = 100
num_reserved_inbound_activation       = 0
num_reserved_outbound_activation      = 0
transmit_window_count                 = 8
dynamic_window_increment               = 1
retransmit_count                      = 8
receive_window_count                  = 8
priority                              = 0
inact_timeout                         = 48
response_timeout                      = 4
acknowledgement_timeout               = 1
link_name                             = ""
local_sap                             = 0x04
retry_interval                        = 60
retry_limit                           = 20
dynamic_link_station_supported         = no
trace_base_listen_link_station        = no
trace_base_listen_link_station_format = long
dynamic_lnk_solicit_sscp_sessions     = yes
dynamic_lnk_cp_cp_sessions_supported  = yes
dynamic_lnk_cp_cp_session_support_required = no
dynamic_lnk_TG_effective_capacity     = 4300800
dynamic_lnk_TG_connect_cost_per_time  = 0
dynamic_lnk_TG_cost_per_byte          = 0
dynamic_lnk_TG_security                = nonsecure
dynamic_lnk_TG_propagation_delay       = lan
dynamic_lnk_TG_user_defined_1         = 128
dynamic_lnk_TG_user_defined_2         = 128
dynamic_lnk_TG_user_defined_3         = 128

```

```

comments                    = ""

mode:
prof_name                   = "MODE001"
mode_name                   = "MODE001"
max_sessions                = 5000
min_conwinner_sessions     = 5000
min_conloser_sessions      = 0
auto_activate_limit        = 0
max_adaptive_receive_pacing_window = 16
receive_pacing_window      = 7
max_ru_size                 = 1024
min_ru_size                 = 256
class_of_service_name      = "#CONNECT"
comments                    = ""

```

References

For complete details about how to install and configure the SNA server, see the following documents:

AIX SNA Server/6000 Command Reference (SC31-7100)

AIX SNA Server/6000 Diagnosis Guide and Messages (SC31- 7101)

AIX SNA Server/6000 User's Guide (SC31-7002)

AIX SNA Server/6000 Transaction Program Reference (SC31- 7003)

AIX SNA Server/6000 Configuration Reference (SC31-7014)

Contact IBM for information about obtaining this documentation.

Configuring the APPC Communications Environment for HP-UX

You must install and configure these products in order to configure the APPC communications environment for the HP-UX platform:

- SNAplusLink
- SNAplusAPI.

Configuring SNAplusLink

The SNAplusLink product allows three types of connectivity:

- Synchronous Data Link Control (SDLC),
- Qualified Logical Link Control (QLLC), and
- Token Ring (TR).

To use SNAplus over QLLC link, either X.25/9000 Link for the Series 700 or X.25/9000 Link for the Series 800 must be installed and configured before you can install and configure SNAplus.

To use SNAplus over a Token Ring link, either HP Token Ring/9000 for the Series 700 or HP Token Ring/9000 for the Series 800 must be installed and configured before you can install and configure SNAplus.

Before you install SNAplus, decide whether SNAplus software will function in a stand-alone or a client/server environment. In a stand-alone environment, all functionality is isolated to a single HP workstation. In a client/server environment, client HP workstations that run SNAplus Presentation Services products (SNAplusAPI)

can access server HP workstations that run SNAplusLink where the physical link resides.

Use the **snapconfig** program to configure SNAplusLink. This program allows you to configure the following:

- link (SDLC, QLLC, TR)
- connection (logical path)
- local node (PU 2.1).

The **snapconfig** installation script automatically creates these configuration files:

```
sna.ini
com.cfg
com.sec
sna.net( for client/server environment only)
```

Configuring SNAplusAPI

You also use the **snapconfig** program to configure SNAplusAPI APPC. This program allows you to configure the following:

- APPC modes
- remote APPC LUs
- local APPC LUs.

The SNAplus control daemon controls the SNAplusLink product (local nodes, links, and connections) and manages communication among SNAplus products and product components. You must start the SNAplus control daemon on each host on the LAN (and on each stand-alone computer) before you can use any of the SNAplus products that are installed on that host.

To start the SNAplus control daemon, issue the following command at the HP-UX command prompt:

```
snapstart daemon
```

Subsequently, you will use the **snapmanage** program to start and stop SNAplus products (SNAplusLink local nodes, links, and connections), to view the status and monitor the use for SNAplus products, and to control logging and tracing.

Sample HP/UX Configuration File

The following HP/UX configuration file excerpt was created by using the **snaptextcfg** command. The excerpt is limited to APPC information with Token Ring connection.

```
;
*****

; SNAplus Binary to Text Configuration Utility
; Copyright (C) 1993 Hewlett-Packard Company
; Binary Configuration      = /usr/lib/sna/com.cfg
; Security File             = /usr/lib/sna/com.sec
; File version              = 100.20
;
*****

;
*****
```

```
; Diagnostics Record (Mandatory)
```

```
;
```

```
*****
```

```
[DIAGNOSTICS]
```

```
connection          = ""                ; Name of network mgt connection
UCF_user            = ""                ; User ID for UCF commands
error_log           = "/usr/lib/sna/sna.err" ; Error log file
audit_log           = "/usr/lib/sna/sna.aud" ; Audit log file
audit_level         = 6                 ; Detailed problem analysis
send_overfl         = No                ; Send RTM when response counter max
send_end            = No                ; Send RTM at end of session
stop_timer          = screen            ; Data first reaches the screen
boundary_1          = 0.5              ; RTM histogram time boundaries
boundary_2          = 1.0
boundary_3          = 2.0
boundary_4          = 5.0
pc_error_log        = "sna.err"         ; PC client error log file
pc_audit_log        = "sna.aud"         ; PC client audit log file
```

```
;
```

```
*****
```

```
; Local Node Record
```

```
;
```

```
*****
```

```
[NODE]
```

```
name                = "NODE1"          ; Local Node
Name
description          = "Node for APPC"  ; Description of Local Node
network              = "SASNET01"      ; Node Network Name
```

```
;
```

```
*****
```

```
; APPC Local LU Record
```

```
;
```

```
*****
```

```
[APPC_LOCAL_LU]
```

```
alias               = "LOCLU001"       ; LU Alias
node                 = "NODE1"         ; Local Node
description          = "Local LU/CP"   ; Text description of LU
net_name             = "SASNET01"      ; LU Network Name
LU_name              = "LOCLU001"     ; Name of LU
LU_number            = 0               ; LU Number
```

```

session_lim           = 254                ; Session Limit
default_LU           = Yes                 ; LU in pool of Default LUs
local_use            = Yes                 ; LU can be used locally
syncpoint           = No                  ; LU supports syncpoint sessions
conv_sec            = No                  ; LU uses conversation level security
partner_LU           = "PARTLU01,4"       ; List of Partner LUs and Modes
partner_LU           = "PARTLU02,4"
partner_LU           = "PARTLU03,4"
partner_LU           = "PARTLU04,4"

```

```

;
*****

```

; Token Ring Connection Record

```

;
*****

```

[TR_CONN]

```

name                 = "TRCON"             ; Name of connection
node                 = "NODE1"            ; Name of node
description           = "Token Ring connection" ; Description
remote_end           = peer                ; Remote end is peer
link_role            = negotiable         ; Station behaves as negotiable
activation           = initially          ; Initially active
node_send            = "05D.FF815"        ; Node id to send
node_rcv             = "017.00000"        ; Node id to receive
control_point        = "SASNET01.LOCALCP" ; Fully qualified control point name
remote_address       = 400031740001       ; Address of remote TR network
remote_sap           = 04                 ; Remote SAP address
retry_limit          = 10                 ; Retry limit
rcv_ack_limit        = 10                 ; Receive acknowledgment threshold
send_ack_limit       = 10                 ; Unacknowledged send threshold
max_btu              = 4096               ; Maximum BTU length
link                 = "TOKEN"           ; link

```

```

;
*****

```

; APPC Mode Record

Note: Be generous in defining session limits. Define enough sessions so that session limits will never be reached. As an APPC API limitation, if session limits are reached, the next time a session is requested, the APPC layer will not return to the application layer until a session is available. This indefinite waiting condition may cause you to think that SAS is not responding or that it is in a loop when, in fact, the underlying APPC layer is waiting for a session to become available.

```

;

```

```
*****
```

```
[APPC_MODE]
```

```
name           = "SASAPPC"           ; Mode name
mode_ID        = 4                   ; Unique Mode ID
description     = "APPC Mode"        ; Description
connection     = "TRCON"            ; Connection used by this mode
priority       = high                ; Mode is High Priority
session_limit   = 12                 ; Mode Session Limit
MCW            = 12                 ; Min Conwinner Sessions
partner_MCW    = 0                   ; Partner Min Conwinner Sessions
auto_act       = 0                   ; Auto activated sessions
min_sendRU     = 256                 ; Min Send RU size
max_sendRU     = 4096                ; Max Send RU size
send_pace     = 10                   ; Send Pacing count
min_rcvRU     = 256                 ; Min Receive RU size
max_rcvRU     = 4096                ; Max Receive RU size
rcv_pace      = 10                   ; Receive Pacing count
```

```
;
```

```
; APPC Remote LU Record
```

```
;
```

```
[APPC_REMOTE_LU]
```

```
alias          = "PARTLU01"         ; LU Alias
description    = "MVS remote LU"    ; Text description of LU
net_name       = "SASNET01"         ; LU Network Name
LU_name        = "PARTLU01"         ; Name of LU
SSCP_Alias     = "PARTLU01"         ; SSCP LU Alias
parallel_sess  = Yes                 ; Parallel Sessions supported
conv_sec       = Yes                 ; LU uses conversation level security
preval_sec     = No                  ; LU can prevalidate security
session_sec    = none                ; No Session Level Security
```

```
;
```

```
; APPC Remote LU Record
```

```
;
```

```
[APPC_REMOTE_LU]
```

```
alias          = "PARTLU02"         ; LU Alias
description    = "MVS Surrogate LU"; Text description of LU
net_name       = "SASNET01"         ; LU Network Name
LU_name        = "PARTLU02"         ; Name of LU
```

```
SSCP_Alias      = "PARTLU02"      ; SSCP LU Alias
parallel_sess   = Yes             ; Parallel Sessions supported
conv_sec        = Yes             ; LU uses conversation level security
preval_sec      = No              ; LU can prevalidate security
session_sec     = none            ; No Session Level Security
```

```
;
*****
```

```
; APPC Remote LU Record
```

```
;
*****
```

```
[APPC_REMOTE_LU]
```

```
alias           = "PARTLU03"      ; LU Alias
description     = "Windows remote LU" ; Text description of LU
net_name        = "SASNET01"      ; LU Network Name
LU_name         = "PARTLU03"      ; Name of LU
SSCP_Alias      = "PARTLU03"      ; SSCP LU Alias
parallel_sess   = Yes             ; Parallel Sessions supported
conv_sec        = Yes             ; LU uses conversation level security

preval_sec      = No              ; LU can prevalidate security
session_sec     = none            ; No Session Level Security
```

```
;
*****
```

```
; APPC Remote LU Record
```

```
;
*****
```

```
[APPC_REMOTE_LU]
```

```
alias           = "PARTLU04"      ; LU Alias
description     = "OS/2 remote LU" ; Text description of LU
net_name        = "SASNET01"      ; LU Network Name
LU_name         = "PARTLU04"      ; Name of LU
SSCP_Alias      = "PARTLU04"      ; SSCP LU Alias
parallel_sess   = Yes             ; Parallel Sessions supported
conv_sec        = Yes             ; LU uses conversation level security
preval_sec      = No              ; LU can prevalidate security
session_sec     = none            ; No Session Level Security
```

```
;
*****
```

```
; APPC Remote LU Record
```

```
;
*****
```

```

; Token Ring Link Record

;
*****

[TR_LINK]
name           = "TOKEN"           ; Name of LU
description    = "Token Ring link"; Text description of LU
device_name    = "sna_TR"          ; Name of device file for link
port_number    = 0                  ; Adapter port

;
*****

; Token Ring Link Usage Record

;
*****

[TR_USAGE]
node           = "NODE1"           ; Node name
link           = "TOKEN"           ; Link name
incoming       = Yes                ; Incoming calls accepted
max_conn       = 1                  ; Maximum number of connections
local_sap      = 04                 ; Local SAP address

```

References

For complete details about how to install and configure SNAplusLink and SNAplusAPI, see the following documents:

HP-UX SNAplus Installation Guide (J2220-61021)

HP-UX SNAplusLink Administrator's Guide (J2220-61023)

HP-UX SNAplusAPI Administrator's Guide (J2223-61008)

HP-UX SNAplus Diagnostics Guide (J2220-61022)

Installing and Administering X.25/9000 (36940-90018)

Installing and Administering Token Ring/9000 (J21625-61001)

Contact Hewlett-Packard for information about obtaining this documentation.

IBM SNA: Technical Overview (CC30-#073)

IBM SNA: Formats(CA27-3136)

Contact IBM for information about obtaining this documentation.

Configuring the APPC Communications Environment for Solaris

Note: The Solaris platform is not supported in Version 7 and later. Δ

You must perform the following procedures to configure the APPC communications environment for the Solaris platform:

- Install and configure link components.
- Add gateways to your network.
- Ensure that the appropriate daemons are running.

Installing and Configuring Link Components

You must install and configure link components, such as SDLC or LLC drivers. For example, if a Token Ring connection is desired, you must install and configure the SunLink Token Ring Interface/SBus software.

Adding Gateways to Your Network

To update or add gateways to your network, you must run the `install.maps` script, which produces output that identifies workstations that are running SunLink SNA Peer-to-Peer gateways. Output is sent to the `/etc/appcs` file, which has the following format:

```
gateway_name host_name:host_gateway_name
```

Use the `gateway_name` value to set the `APPC_GATEWAY` environment variable or the `APPCGATE` SAS macro variable. See “SAS/CONNECT and SAS/SHARE Variables” on page 270 for information about these variables. `gateway_name` is used for attaching the SAS local transaction program to the specified peer-to-peer gateway.

If you are running NIS/NIS+, you can also run the `install.maps` program to update the NIS/NIS+ databases with your new gateway configurations.

Ensuring that the Appropriate Daemons Are Running

If the `SUNLINK_MAPPER` daemon and the `SNACOMMD` daemon are not already running or do not start automatically, then you must start them. The `SUNLINK_MAPPER` daemon allows client transaction programs (TPs) to find the SunLink SNA Peer-to-Peer gateway on the network. The `snacommmd` daemon manages and controls access to the DLC drivers protocol stacks.

Sample Solaris Configuration File

A sample Solaris configuration file follows:

```
*****
# PU definition *
# When this verb is not specified, a system default is used. *
*****

:DEFINE_PU:

pu_name = XXXPU000, network_name = SASNET01, contents_id = 01234567

*****
# Node definition *
# When this verb is not specified, a system default is used. *
# Note: *
# node_id is the control point name for my workstation -- it is *
# important since it will be used during XID negotiation *
*****

:DEFINE_NODE:

pu_name = XXXPU000; node_id = LOCALCP

*****
```

```

# Local LU definition (1 for each local lu) *
# This LU corresponds to LOCLU001 defined in VTAM. *
# 'pu_name' is sifted down from ':DEFINE_NODE:'. *
#*****
:DEFINE_LOCAL_LU:

fql_lu_name      = SASNET01.LOCLU001

lu_local_address = 1          # must be non-zero even for independent
lu_name          = LOCLU001
lu_session_limit = 512       # session limit

#*****
# Partner LU definition *
# This is actually one of the subsystems of VTAM. *
# A local LU cannot communicate with the subsystem of VTAM unless*
# the partner LU (subsystem) is defined for the local LU. *
# 'pu_name' and 'lu_local_address' are sifted down from *
# ':DEFINE_LOCAL_LU:' *
#*****
:DEFINE_PARTNER_LU:

fql_plu_name     = SASNET01.PLUNAM01
u_plu_name       = PLUNAM01
parallel_session = yes
lu_is_dependent  = no
initiate_type    = INITIATE_ONLY

#*****
# Mode definition (1 for each mode) *
# This is actually one of the MODENAME in VTAM or CICS. *
# A local LU cannot communicate with the subsystem *
# of VTAM over a specific mode name unless *
# (1) the partner LU (subsystem) is defined for the local LU and*
# (2) the mode name is defined for the partner LU. *
# A transaction program uses (unique_session_name) in the *
# allocate verb to establish a session between the local *
# LU and the partner LU over the mode name. *
# 'pu_name', 'lu_local_address', and 'fql_plu_name' are sifted *
# down from ':DEFINE_PARTNER_LU:'. *
#*****
:DEFINE_MODE:
mode_name          = SASAPPC
unique_session_name = PLUNAM01 # remote partner name
snd_pac_window     = 0
rcv_pac_window     = 0
snd_max_ru_size    = 256
rcv_max_ru_size    = 256
sync_level         = none
auto_activate_limit = 0
session_limit      = 30

```



```

min_conwinner_limit      = 15
min_conloser_limit      = 15

```

```

#*****
# Partner LU definition *
# This is actually one of the subsystems of VTAM. *
# A local LU cannot communicate with the subsystem of VTAM unless*
# the partner LU (subsystem) is defined for the local LU. *
# 'pu_name' and 'lu_local_address' are sifted down from *
# ':DEFINE_LOCAL_LU:'. *
#*****

```

```

:DEFINE_PARTNER_LU:

```

```

fql_plu_name      = SASNET01.PLUNAM02
u_plu_name        = PLUNAM02
parallel_session  = yes
lu_is_dependent   = no
initiate_type     = INITIATE_ONLY

```

```

#*****
# Mode definition (1 for each mode) *
# This is actually one of the MODENAME in VTAM or CICS. *
# A local LU cannot communicate with the subsystem of VTAM over *
# a specific mode name unless *
# (1) the partner LU (subsystem) is defined for the local LU and *
# (2) the mode name is defined for the partner LU. *
# A transaction program uses (unique_session_name) in the *
# allocate verb to establish a session between the local *
# LU and the partner LU over the mode name. *
# 'pu_name', 'lu_local_address', and 'fql_plu_name' are sifted *
# down from ':DEFINE_PARTNER_LU:'. *
#*****

```

```

:DEFINE_MODE:

```

```

mode_name          = SASAPPC
unique_session_name = PLUNAM02
snd_pac_window     = 0
rcv_pac_window     = 0
snd_max_ru_size    = 256
rcv_max_ru_size    = 256
sync_level         = none
auto_activate_limit = 0
min_conloser_limit = 15

```

```

#*****
# Partner LU definition *
# This is actually one of the subsystems of VTAM. *
# A local LU cannot communicate with the subsystem of VTAM unless*
# the partner LU (subsystem) is defined for the local LU. *
# 'pu_name' and 'lu_local_address' are sifted down from *
# ':DEFINE_LOCAL_LU:'. *
#*****

```

```

:DEFINE_PARTNER_LU:

fql_plu_name      = SASNET01.PLUNAM03
u_plu_name        = PLUNAM03
parallel_session  = yes
lu_is_dependent   = no
initiate_type     = INITIATE_ONLY

#*****
# Mode definition (1 for each mode) *
# This is actually one of the MODENAME in VTAM or CICS. *
# A local LU cannot communicate with the subsystem of VTAM over *
# a specific mode name unless *
# (1) the partner LU (subsystem) is defined for the local LU and *
# (2) the mode name is defined for the partner LU. *
# A transaction program uses (unique_session_name) in the *
# allocate verb to establish a session between the local *
# LU and the partner LU over the mode name. *
# 'pu_name', 'lu_local_address', and 'fql_plu_name' are sifted *
# down from ':DEFINE_PARTNER_LU:'. *
#*****

:DEFINE_MODE:

mode_name          = SASAPPC
unique_session_name = PLUNAM03 # remote partner name
snd_pac_window     = 0
rcv_pac_window     = 0
snd_max_ru_size    = 256
rcv_max_ru_size    = 256
sync_level         = none
auto_activate_limit = 0
session_limit      = 30
min_conwinner_limit = 15
min_conloser_limit = 15

#*****
# Partner LU definition *
# This is actually one of the subsystems of VTAM. *
# A local LU cannot communicate with the subsystem of VTAM unless *
# the partner LU (subsystem) is defined for the local LU. *
# 'pu_name' and 'lu_local_address' are sifted down from *
# ':DEFINE_LOCAL_LU:'. *
#*****

:DEFINE_PARTNER_LU:

fql_plu_name      = SASNET01.PLUNAM04
u_plu_name        = PLUNAM04
parallel_session  = yes
lu_is_dependent   = no
initiate_type     = INITIATE_ONLY

```

```

*****
# Mode definition (1 for each mode) *
# This is actually one of the MODENAME in VTAM or CICS. *
# A local LU cannot communicate with the subsystem of VTAM over *
# a specific mode name unless *
# (1) the partner LU (sub_system) is defined for the local LU and *
# (2) the mode name is defined for the partner LU. *
# A transaction program uses (unique_session_name) in the *
# allocate verb to establish a session between the local *
# LU and the partner LU over the mode name. *
# 'pu_name', 'lu_local_address', and 'fql_plu_name' are sifted *
# down from ':DEFINE_PARTNER_LU:'. *
*****

:DEFINE_MODE:
mode_name           = SASAPPC
unique_session_name = PLUNAM04 # remote partner name
snd_pac_window      = 0
rcv_pac_window      = 0
snd_max_ru_size     = 256
rcv_max_ru_size     = 256
sync_level          = none
auto_activate_limit = 0
session_limit       = 30
min_conwinner_limit = 15
min_conloser_limit = 15
*****

#*****
# DLC definition (1 for each dlc) *
# Currently, only one DLC per APPC gateway. *
#*****

:DEFINE_DLC:

dlc_name           = DLC0
dlc_driver_name     = /dev/llc2
port_driver_name    = tr0
dlc_type            = llc # logical link control
maxdata            = 265 # MAXDATA value
retries            = 32
local_sap           = 04
block_number        = 05D # MUST be first of xid parameters
id_number           = FF813
role                = negotiable # or primary, negotiable
tx_rx_capability    = alternating # or simultaneous
include_control_point = yes # xid control vector
include_link_station_name = no # xid control vector
linkid              = 0
xtwait             = 10

#*****

```

```

# ALS definition (1 for each als) *
# Currently, only one ALS per APPC gateway. *
#*****
:DEFINE_ALS:

dlc_name      = DLC0
pu_name       = XXXPU000
als_name      = XXALS000
remote_mac_addr = 400000000001
remote_sap    = 04

#*****
# DB_MSG definition *
# No field is necessary. Defaults (shown below) are used for *
# those fields that are unspecified. *
#*****

:DB_MSG:

db_pc          = no
db_mail        = no
db_buf         = no
db_dev         = no
db_api_verb    = no
db_character_set = iso
db_record_size = long
file_mode      = create
file_name      = '/local/u/saspad/appc.trc'
db_tp_info     = yes
db_max_trc_sz  = 0      # unlimited (in MB)

```

References

For complete details about how to install and configure Solaris 2.x/Sun Link see the following documents:

*SunLink SNA Peer-to-Peer 8.0 and SunLink SNA Peer-to-Peer RunTime 8.0
Installation Guide*

*SunLink SNA Peer-to-Peer 8.0 and SunLink SNA Peer-to- Peer RunTime 8.0
Administrator's Guide*

SunLink HSI/S 2.0 Installation and Administration Guide

SunLink TRI/S 3.0 Installation Guide

SunLink FDDI/S 2.0 Installation Guide

Contact Sun Microsystems, Inc. for information about obtaining this documentation.

IBM SNA: Technical Overview (CC30-#073)

IBM SNA: Formats (CA27-3136)

Contact IBM for information about obtaining this documentation.

The correct bibliographic citation for this manual is as follows: SAS Institute Inc., *Communications Access Methods for SAS/CONNECT and SAS/SHARE Software, Version 8*, Cary, NC: SAS Institute Inc., 1999. pp. 643.

Communications Access Methods for SAS/CONNECT and SAS/SHARE Software, Version 8

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.