Support for Version 6 Only

Version 7 does not support the Windows 32s platform. However, information about Windows 32s is included here for Version 6 users.

Note: You can use the CPIC communications access method with the Windows 32s platform only.
The Windows 32s platform is restricted to the SAS/CONNECT local host and SAS/SHARE server host roles only.

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

System Administrator or User

To use the CPIC access method with a Windows 32s 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 CPIC 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. CPIC has been installed at your site.
2. SAS has been installed on both the local and remote hosts.

The CPIC access method gives you access to an SNA network. SAS/CONNECT and SAS/SHARE software can use the Microsoft Windows Open Services Architecture (WOSA) standard (WinCPIC) or the CPIC 2.0 standard. Therefore, you should be able to use software from any vendor that supports these standards. To use the CPIC access method, you must install and configure one of the following packages:

- the Microsoft SNA Server, Version 2.11.SP1 (Service Pack 1). This software package supports the WOSA standard WinCPIC.
- the Wall Data Rumba APPC engine, Version 2.0, with corrective maintenance (ZB0APC10) applied. This software package supports the CPIC 2.0 standard with WinCPIC extensions.
- any program that supports the WOSA CPIC (WinCPIC) or CPIC 2.0 standards.

Configuring the Underlying SNA Subsystem

Network Administrator

Before you can use SAS/CONNECT and SAS/SHARE with the CPIC access method, you must first install and configure the underlying SNA subsystem using either of the following products:

- the Microsoft SNA Server
- the Wall Data Rumba APPC engine.

For information about how to install and configure the Microsoft SNA Network Server, see “Installing and Configuring a Microsoft Server Environment” on page 334.

For information about how to install and configure the Wall Data APPC engine, see “Configuring the Wall Data APPC Engine” on page 356.
Optionally, you may configure a remote host's symbolic name through CPIC-side properties. See “Optionally Configuring CPIC Properties” on page 359 for this information.

---

**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 an end user (LU 6.2 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).

---

**Setting SAS Options and Variables**

You may set specific options in SAS to establish the connections that you want with SAS/CONNECT and SAS/SHARE when using the CPIC communications access method. 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=cpic_secure _prompt_;

- SAS configuration file or at SAS invocation:

  -SET variable-name value

  Example:

  -set cpic_secure _prompt_

- DOS operating system environment variable:

  SET variable-name=value

  Example:

  set cpic_secure=_prompt_

Values for these options may contain up to eight characters, consisting of alphanumerical 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.

**SAS/CONNECT and SAS/SHARE Options**

**CPIC_LU62MODE**

specifies the mode name that is associated with an LU-LU pair and determines 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 in both the local session and the remote environment (either on the remote host in a SAS/CONNECT session or on a SAS/SHARE server, as necessary).

As an alternative, you may specify the node name through CPIC-side properties. See “Optionally Configuring CPIC Properties” on page 359 for details. However, the option takes precedence over CPIC-side properties.

Consult your network administrator for advice about setting CPIC_LU62MODE.

**CPIC_SECURE _NONE_ | _PROMPT_ | userid.password**

set the CPIC_SECURE option in order to pass a remote host userid and password to a remote SAS/CONNECT host or to a SAS/SHARE server for verification. After the userid and the password have been verified, the connection to the remote SAS/CONNECT host or the SAS/SHARE server can proceed.

Values that you set at a SAS/CONNECT local host or at a SAS/SHARE client follow:

**_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 a userid and a password provides more security than assigning the userid and password to the system option.

userid.password

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

userid.password specifies both the userid and password. Assigning the userid and the password directly to the CPIC_SECURE option 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.

Examples:

- options set=cpic_secure _none_;
- options set=cpic_secure _prompt_;
- options set=cpic_secure bass.timego;
- options set=cpic_secure "bass.time2go";

See “Setting SAS Options and Variables” on page 345 for examples of the forms that you can use to specify CPIC_SECURE.

The OS/390, CMS, and VSE platforms that are used as remote hosts for SAS/CONNECT require security. However, the OS/2 platform does not require security unless you have a user profile on the OS/2 computer.

As an alternative, you may specify these security features through CPIC-side properties. See “Optionally Configuring CPIC Properties” on page 359 for details.

The following option settings take precedence over CPIC-side properties:

- set cpic_secure bass.time2go
- set cpic_secure _prompt_

However, the following option setting does not take precedence over CPIC-side properties:

- set cpic_secure _none_

**CPIC_NET**

specifies the network name to be concatenated with a remote-LU to generate a fully qualified partner LU name for APPN End Node (EN) that is capable of CPIC implementations. For example, specifying both this option and the remote-LU, which is used in the REMOTE or the SERVER option, causes a fully qualified remote LU name to be used. By using a fully qualified remote LU name in an APPN capable environment, no other partner configuration is necessary. The Wall Data Rumba APPC engine is capable of APPN EN.

**CPIC_PARTNER_COUNT**

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

**CPIC_CONFORMANCE**

specifies a vendor’s CPIC implementation conformance. The default conformance is adherence to the Wall Data Rumba APPC engine implementation. Use CPIC_CONFORMANCE to override this default conformance standard. If a vendor’s CPIC implementation conforms to the WinCPIC standard (such as Microsoft SNA server), then specify it as follows:

- set cpic_conformance wincpic

Otherwise, it is assumed that the vendor’s package conforms to the CPIC 2.0 standard. Therefore, specify the optional CPIC 2.0 conformance classes (in addition to the Mandatory and Data Conversion Routines conformance classes, which are required).
Table 24.1 on page 348 contains the CPIC 2.0 optional conformance class names with brief explanations.

Table 24.1  Windows CPIC 2.0 Optional Conformance Classes

<table>
<thead>
<tr>
<th>Optional Conformance Classes</th>
<th>Explanation</th>
</tr>
</thead>
<tbody>
<tr>
<td>CNONBLOCKING</td>
<td>CPIC 2.0 Conversation-Level Non-Blocking option set</td>
</tr>
<tr>
<td>QNONBLOCKING</td>
<td>CPIC 2.0 Queue-Level Non-Blocking option set</td>
</tr>
<tr>
<td>SERVER</td>
<td>CPIC 2.0 Server option set</td>
</tr>
<tr>
<td>SECURITY</td>
<td>CPIC 2.0 Security option set</td>
</tr>
<tr>
<td>WINEXT</td>
<td>CPIC Windows extensions option set</td>
</tr>
</tbody>
</table>

If a vendor’s CPIC implementation conforms to the CPIC 2.0 standards, then specify the optional conformance classes that it supports as follows:

```
-set cpic_conformance optional-conformance-classes
```

Note:  Use commas to separate multiple option classes. Put the options inside quotes if you specify them in a SAS session.

Example:
```
options set=CPIC_CONFORMANCE
    'CNONBLOCKING,SERVER,SECURITY,WINEXT' ;
```

For more information about the CPIC 2.0 requirements, see “CPIC 2.0 Requirements” on page 349.

---

**SAS/CONNECT Only Option**

**CPIC_SURROGATE_LUNAME**

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

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

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

---

**SAS/SHARE Only Option**

**CPIC_USER**

specifies the user context to be used by a SAS/SHARE server when referring to your user session in its SAS log and in output from the OPERATE procedure.
Otherwise, your user session log entries would not be easily identified in the server's log.

**CPIC 2.0 Requirements**

See the IBM publication Common Programming Interface CPIC 2.0 Specification to determine whether a particular software vendor is CPIC 2.0 compatible.

SAS uses only a subset of operations that are provided by the CPIC-C 2.0 standard. Refer to the following list of CPIC 2.0 conformance classes and SAS requirements that are put upon these classes.

**Mandatory (Conversations, LU 6.2)**

are required.

**Data Conversion Routines**

are required.

**Conversation-Level Non-Blocking**

is recommended, if Queue-Level Non-Blocking is not supported. Conversation-Level Non-Blocking and Queue-Level Non-Blocking are mutually exclusive. Set_Processing_Mode and Specify_Windows_Handle are used to set up a nonblocking callback mechanism. CPIC is expected to call PostMessage to post a message of "WinAsyncCPIC-C" to the window that is specified by the Specify_Windows_Handle call. The Specify_Windows_Handle call is not part of the CPIC 2.0 standard, but it is part of the WOSA CPIC (WinCPIC) standard and is required for SAS to use conversation-level nonblocking. SAS also uses the Cancel-Conversation conversation-level nonblocking operation. This type of nonblocking mechanism is used by the Wall Data Rumba APPC engine. Either conversation-level nonblocking or queue-level nonblocking is required for server function and is highly recommended for client function as well.

**Queue-Level Non-Blocking**

is recommended, if Conversation-Level Non-Blocking is not supported. Conversation-Level Non-Blocking and Queue-Level Non-Blocking are mutually exclusive. The Callback Function conformance class is included with the Queue-Level Non-Blocking conformance class. Set_Queue_Processing_Mode and Set_Queue_Callback_Function are used to process calls in a nonblocking mode. The Set_Queue_Callback_Function sets up a window posting mechanism in the Microsoft Windows environment. SAS also uses the Cancel-Conversation queue-level nonblocking operation. This type of nonblocking mechanism is used with a software package that supports CPIC 2.0 nonblocking without additional WinCPIC extensions. Either conversation-level nonblocking or queue-level non-blocking is required for server function and is highly recommended for client function.

**Server**

is required for server function. Initialize_For_Incoming and Accept_Incoming calls are used to accept incoming sessions in a nonblocking mode of operation.

**Security**

is optional. If not available, the SAS option CPIC_SECURE will have no effect; therefore, security must be specified in a CPIC symbolic destination table entry. See “Optionally Configuring CPIC Properties” on page 359 for details.
Windows Extensions

is optional. This conformance class is not defined by the CPIC 2.0 standard. Specify it if the vendor’s software package supports the WinCPICStartup and WinCPICCleanup WinCPIC operations. The Wall Data Rumba APPC engine is an example.

Because SAS requires the LU 6.2 conformance class rather than the OSI TP conformance class, the term "mandatory" is used to refer to the Conversations and LU 6.2 conformance classes. SAS also requires the Data Conversion Routines conformance class to be supported by the underlying CPIC software package. By definition, any package that provides a CPIC 2.0 implementation is required to support the Mandatory Conversations conformance class, as well as either the LU 6.2 or the OSI TP conformance class.

---

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

---

**Specifying the CPIC Communications Access Method**

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

```plaintext
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. CPIC (an abbreviation for Common Programming Interface for Communications) is an example of access-method-id.

Example:

```plaintext
options comamid=cpic;
```

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

---

**Specifying the Remote Node Name**

To make a connection from a Windows 32s local host to a remote host, use the following syntax:
OPTIONS REMOTE=symbolic-destination-name
   | remote-LU
   | remote-LU-alias;

where symbolic-destination-name is defined to the underlying CPIC subsystem using side-information. The remote-LU specifies the logical unit of the remote host that you are connecting to. The remote LU name is concatenated with the network name (which is defined by the CPIC_NET variable) and forms a fully qualified partner LU name for an APPN EN capable environment, such as the Wall Data Rumba APPC engine. The remote-LU-alias specifies a remote LU that has been defined to the underlying CPIC subsystem.

Types of valid values are shown in the following table:

<table>
<thead>
<tr>
<th>Type of Remote Host</th>
<th>Remote Host Identifier</th>
</tr>
</thead>
<tbody>
<tr>
<td>OS/390</td>
<td>name of APPC/MVS scheduler LU</td>
</tr>
<tr>
<td>CMS</td>
<td>name of AVS (APPC/VM VTAM Support) private gateway LU for VM system</td>
</tr>
<tr>
<td>OS/2</td>
<td>name of control-point LU or other locally defined LU</td>
</tr>
<tr>
<td>VSE</td>
<td>name of VTAM APPL ID (ACBNAME) that was set up for APPC LU6.2 communications</td>
</tr>
<tr>
<td>Windows NT Windows 95</td>
<td>name of control-point LU or other SNA Server locally defined LU</td>
</tr>
</tbody>
</table>

Whether you use a symbolic destination name, a remote LU name, or a remote LU alias, the value of the REMOTE= option must be identical in the local and the remote sessions.

Ask your network administrator for advice about what value to assign to the REMOTE= option.

Example:

    options remote=remotelu;

Alternatively, you may specify 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;

Note: Sign-on script files are not used on a Windows local host that uses the CPIC access method because CPIC communicates directly with the remote host.

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

\begin{verbatim}
  signon noscript;
  .
  .
  .
  signoff noscript;
\end{verbatim}

**Local Host Example**

The following example illustrates the statements that you specify in a Windows 32s local host configuration file to connect to a remote host with the CPIC access method:

\begin{verbatim}
  -set cpic_lu62mode cpicmode
  -set cpic_conformance wincpic
  -set cpic_secure _prompt_ \\
  CPICMODE is the mode-name that is defined on the underlying CPIC subsystem. The value WINCPIC specifies a conformance to the WINCPIC standard. The CPIC_SECURE option specifies that connecting local hosts be prompted for a username and a password that are valid on the remote host.
  The following example shows the statements that you specify in a local SAS session:

  options comamid=cpic remote=remotelu;
  signon;

  The CPIC communications access method is declared with a connection to REMOTELU, which is the symbolic-destination-name, the remote-LU, or the remote-LU-alias. The SIGNON statement performs the sign-on procedure.

  Note: The value for the REMOTE= option must be identical on both the local and the remote hosts. ▲
\end{verbatim}

**Remote Host Example**

SAS Institute does not provide support for connections to the Windows 32s remote host with the CPIC access method.

---

**SAS/SHARE**

**Client Tasks**

System Administrator or User

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

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

Setting Security for Connecting Clients

If you specified session security in the SASTP62 TP definition when configuring the SNA subsystem, clients must have secure userids and passwords.

Requiring connecting clients to supply a valid userid and password enforces server security. At the client, set the CPIC_SECURE option to store a userid and password that are valid on the server's host. See “SAS/CONNECT and SAS/SHARE Options” on page 346 for more information about setting the CPIC_SECURE option.

Specifying the CPIC Access Method

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

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

```plaintext
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. CPIC (an abbreviation for Common Programming Interface for Communications) is an example of an access-method-id.

Example:

```plaintext
options comamid=cpic;
```

The server is accessed using the CPIC access method.

You may specify the COMAMID option in an OPTIONS statement, on a SAS invocation, or in a SAS configuration file.

Additionally, you may use the COMAUX1 and COMAUX2 options to designate auxiliary communications access methods in an ordered list. See Figure 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:

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

An example of configuration file entries for a Windows 32s client connecting to a Windows 32s server follows:

```plaintext
-comamid cpic
-comaux1 tcp
-comaux2 netbios
```

If the server cannot be reached using the CPIC method, a second attempt is made with the TCP/IP access method, and then with the NetBIOS access method.
Specifying the Server Name

The server name that you specify in the PROC OPERATE statement and the LIBNAME statement must be defined as the symbolic-destination-name, remote-LU, or remote-LU-alias at the client computer. For complete information about defining appropriate LUs for use with SAS/SHARE, see “Configuring the Wall Data APPC Engine” on page 356 and “Installing and Configuring a Microsoft Server Environment” on page 334.

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:

```sas
options comamid=cpic;
libname demo 'remote.data' server=remote-LU-alias
```

In this example, at the client host, the server name is specified as a remote-LU-alias, which is a name that refers to a remote LU that is defined at the server.

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:

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

where gateway is defined to the CMS system and is locally defined as a symbolic-destination-name, remote-LU, or remote-LU-alias.

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

Client Example

The following example illustrates the statements that you specify in a Windows 32s client configuration file to access a server with the CPIC access method.

```bash
-set cpic_lu62mode cpicmode
-set cpic_secure _prompt_
-set cpic_conformance wincpic
```

CPICMODE is the mode-name that is defined on the underlying CPIC subsystem.
The value WINCPIC specifies a conformance to the WINCPIC standard.
The CPIC_SECURE options specifies that connecting clients be prompted for a username and a password that are valid on the server.

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

```bash
options comamid=cpic;
libname sasdata 'c:\edc\prog2\sasdata’ server=share1;
```

The CPIC access method is declared. The LIBNAME statement specifies the data library that is accessed through the server SHARE1. The server can be specified as either a symbolic-destination-name, a remote-LU, or a remote-LU-alias.
Server Tasks

Server Administrator

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

1. Configure CPIC session security.
2. Specify the CPIC access method.
3. Specify the server name.

Configuring CPIC Session Security

You can authenticate users at the server, but you cannot control file authorization. For the CPIC access method on Windows 32s, you can authenticate users at the server by setting up CPIC session security within the SASTP62 TP (transaction program) definition.

However, securing user authorization for file access is not possible because the underlying CPIC subsystem performs the authentication and prevents 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 a CPIC implementation limitation. Under Windows 32s, user contexts do not exist.

See “SASTP62 Transaction Program” on page 339 for details about setting up TP definitions.

Specifying the CPIC Access Method at the Server

You must specify the CPIC communications access method before you can create and access a SAS/SHARE server.

Use the following syntax to specify the CPIC 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. CPIC (an acronym for Common Programming Interface for Communications) is an example of an access-method-id.

For a server that is running on a host on which only one communications access method is available, use the COMAMID option.

Example:

```
options comamid=cpic;
```

The server will be available only to SAS/SHARE sessions that use the CPIC access method.

You may specify the COMAMID option in an OPTIONS statement, at a SAS invocation, or in a SAS configuration file.

However, if the host on which a server is running supports multiple access methods, you may specify up to two auxiliary access methods by which clients may access the server by using the COMAUX1 and COMAUX2 options. See Figure 1.3 on page 10 for the supported access methods by host.
All of the access methods initialize when the server initializes. The activation of multiple access methods makes a server available to several groups of clients, each using a different communications access method simultaneously. COMAUX options can be specified only at a SAS invocation or in a SAS configuration file. The syntax for the COMAUX options follows:

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

An example of configuration file entries for a server that is running on an Windows 32s host follows:

- `comamid cpic`
- `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 CPIC access method as well as to clients that use the TCP/IP and NetBIOS access methods.

---

**Specifying the Server Name**

The server name that you specify in the PROC SERVER statement must be defined as the local-LU at the SAS/SHARE server. For complete information about defining appropriate LUs for use with SAS/SHARE, see “Configuring the Wall Data APPC Engine” on page 356 and “Installing and Configuring a Microsoft Server Environment” on page 334.

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 32s host at which you start a server:

```sas
options comamid=cpic;
proc server id=share1;
run;
```

The CPIC access method is declared, and the server SHARE1 is started on the server that is running on the Windows 32s host. The server-id is a local LU that is defined to the underlying CPIC subsystem.

---

**Configuring the Wall Data APPC Engine**

Network Administrator

Because the Wall Data APPC engine connects directly to the SNA network rather than to an SNA server by means of a router, it is considered a stand-alone product. This engine supports CPIC API level 2.0 with WinCPIC calls and APPN as an intelligent End Node (EN).
Perform the following tasks to set up the Wall Data APPC engine:

1. Configure network properties.
2. Configure connections, logical units, and modes.
3. Optionally, configure for a SAS/SHARE server.
4. Optionally, configure CPIC properties.

After you have completed the installation and configuration of this product, users of SAS/CONNECT and SAS/SHARE can make connections from their SAS/CONNECT local hosts or SAS/SHARE clients transparently by means of the SNA network to the selected remote host or SAS/SHARE server.

This section highlights the general procedures that you must perform to connect to an SNA network. For details about how to install and configure the Wall Data APPC Engine with the Rumba APPC Configuration Utility, see Wall Data product literature.

### Configuring Network Properties

Define the network properties using the Rumba APPC Configuration Utility.

1. If you are using the Token Ring or Ethernet connection interfaces, install the IBM LAN support program.
2. Modify the CONFIG.SYS file to include the appropriate driver: Token Ring or Ethernet.

   **Token Ring:**
   
   device=dxma0mod.sys  
   device=dxmc0mod.sys

   **Ethernet:**
   
   device=dxme0mod.sys  
   device=dxmt0mod.sys  ES=X  EST=X

3. If you are using the Token Ring connection interface, modify the SYSTEM.INI file, as follows:

   
   [386enh]
   
   device=c:\path\vwddlc.386

   Be sure to specify the appropriate path.

### Configuring Connections, Logical Units, and Modes

Perform these tasks using the Rumba APPC Configuration Utility:

1. Configure connections.
2. Configure local LUs.
3. Configure remote LUs.
4. Configure modes.

### Configuring Connections

Connection properties are the software components of the SNA server that communicate by means of the device driver to a particular communications adapter.
Specify the connection name (for example, Token Ring, SDLC) and other properties that are appropriate to your configuration.

**Configuring Local LUs**

Configure the 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 that is 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.

**Configuring Remote LUs**

You must specify a remote LU to connect to a SAS/CONNECT remote session or to a SAS/SHARE server that is executing on a remote system. You may define a remote LU using any of the following three forms:

- configured remote LU and a network name that is defined by the CPIC_NET option
- configured remote LU alias
- symbolic destination name.

For example, to specify a fully qualified remote LU in a SAS/CONNECT session, specify the partner LU name in the REMOTE= option and specify the network identifier in the CPIC_NET option (see “SAS/CONNECT and SAS/SHARE Options” on page 346). In a SAS/SHARE session, specify the remote LU alias in the SERVER option.

**Configuring Modes**

After you have configured and paired local LUs and remote LUs, specify the properties of the communications mode that is to be used between each pair.

If site-naming conventions permit, specify the mode name SASAPPC. The CPIC access method uses this mode name if CPIC_LU62MODE has not been defined. See “SAS/CONNECT and SAS/SHARE Options” on page 346 for information about setting the CPIC_LU62MODE option.

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 through a specific SAS/CONNECT remote host or 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 by using 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, a loop or no response from SAS), the underlying APPC layer is waiting for a session to become available.

If you intend to use the Windows 32s computer as a SAS/SHARE server, you must perform additional configuration steps. Otherwise, you have completed the configuration of the Wall Data APPC engine on a Windows 32s computer.
**Configuring a SAS/SHARE Server**

Additionally, you must define the SASTP62 TP (transaction program) with the Rumba APPC Configuration Utility. Perform the following tasks:

1. Define an SASTP62 TP that has these properties:
   - Queued
   - Infinite Receive Allocate Timeout (0).
   - Reasonable Incoming Allocate Timeout (about 120 seconds).

2. Modify the WIN.INI file, as follows:
   
   ```ini
   [RUMBA]
   askoperator=0
   ```

3. When configuring connections (see “Configuring Connections” on page 357), make sure that you define the link as permanent so that, after the Rumba APPC engine is started, the engine will connect initially to the network node and be ready for incoming connections (for example, ATTACHes).

You have completed the configuration of the Wall Data APPC engine for a Windows 32s computer.

**Optionally Configuring CPIC Properties**

Network Administrator

You may define properties that automatically supply remote host information when signing on in a SAS/CONNECT session or when accessing a SAS/SHARE server. Use the Wall Data APPC engine configuration utility program to define these properties.

Properties include a remote host’s symbolic name, a partner LU, a partner LU alias, a mode, and security features. You may select these properties instead of setting the CPIC_LU62MODE and CPIC_SECURE options. See “SAS/CONNECT and SAS/SHARE Options” on page 346 for more information.

**References**

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

- IBM SNA: Technical Overview (GC30-#073)
- IBM SNA: Formats (GA27-3136)
- IBM: Multiplatform APPC Configuration Guide (GG24-4485-00)

Contact IBM for information about obtaining this documentation.