



## CHAPTER

## 24

## Windows: CPIC Access Method

---

<i>SAS Support for CPIC on Windows</i>	353
<i>SAS/CONNECT Tasks</i>	354
<i>System and Software Requirements for SAS/CONNECT</i>	354
<i>Configuring the Underlying SNA Subsystem</i>	354
<i>Understanding SNA Server Terminology</i>	354
<i>Setting SAS Options and Variables</i>	355
<i>SAS/CONNECT Options</i>	355
<i>SAS/CONNECT Only Option</i>	358
<i>CPIC 2.0 Requirements</i>	358
<i>SAS/CONNECT</i>	359
<i>Local Host Tasks</i>	359
<i>Specifying the CPIC Communications Access Method</i>	359
<i>Specifying the Remote Node Name</i>	360
<i>Signing On to the Remote Host</i>	360
<i>Local Host Example</i>	361
<i>Remote Host Example</i>	361
<i>Configuring the Wall Data APPC Engine</i>	361
<i>Configuring Network Properties</i>	362
<i>Configuring Connections, Logical Units, and Modes</i>	362
<i>Configuring Connections</i>	362
<i>Configuring Local LUs</i>	363
<i>Configuring Remote LUs</i>	363
<i>Configuring Modes</i>	363
<i>Optionally Configuring CPIC Properties</i>	363
<i>References</i>	364

---

### SAS Support for CPIC on Windows

*Note:* The CPIC communications access method can be used with the Windows 32s platform only.

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

The Windows 32s platform is restricted to the SAS/CONNECT local host role only.

---

## SAS/CONNECT Tasks

### *System Administrator or User*

To use the CPIC access method with a Windows 32s host for SAS/CONNECT, 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 options that you want.

---

## System and Software Requirements for SAS/CONNECT

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 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 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 Wall Data APPC engine, see “Configuring the Wall Data APPC Engine” on page 361.

Optionally, you may configure a remote host’s symbolic name through CPIC-side properties. See “Optionally Configuring CPIC Properties” on page 363 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.

**remote LU**

a named LU that is associated with the SAS/CONNECT remote host to which a local host 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 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 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.

## SAS/CONNECT 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.

As an alternative, you may specify the mode name through CPIC-side properties. See “Optionally Configuring CPIC Properties” on page 363 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 for verification. After the userid and the password have been verified, the connection to the remote SAS/CONNECT host can proceed.

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

**\_NONE\_**

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

Setting **\_NONE\_** does not establish secure sessions for connecting SAS/CONNECT local hosts.

**\_PROMPT\_**

must be set at the SAS/CONNECT local host.

**\_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.

*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 may inadvertently publicize this information and compromise the security of the SAS/CONNECT remote host. 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 355 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 363 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 357 contains the CPIC 2.0 optional conformance class names with brief explanations.

**Table 24.1** Windows CPIC 2.0 Optional Conformance Classes

Optional Conformance Classes	Explanation
CNONBLOCKING	CPIC 2.0 Conversation-Level Non-Blocking option set
QNONBLOCKING	CPIC 2.0 Queue-Level Non-Blocking option set
SERVER	CPIC 2.0 Server option set
SECURITY	CPIC 2.0 Security option set
WINEXT	CPIC Windows extensions option set

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.  $\Delta$

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 358.

## SAS/CONNECT Only Option

### CPIC\_SURROGATE\_LUNAME

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

## 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 363 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:

```
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:

```
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 24.2** Windows CPIC SAS/CONNECT Values for the REMOTE= Option

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
OS/2	name of control-point LU or other locally defined LU
VSE	name of VTAM APPL ID (ACBNAME) that was set up for APPC LU6.2 communications
Windows NT Windows 95	name of control-point LU or other SNA Server locally defined LU

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.  $\triangle$



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:

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

---

## 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:

```
-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. △

---

## Remote Host Example

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

---

# 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 CPIC properties.

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

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\vwdllc.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.

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

## Configuring Remote LUs

You must specify a remote LU to connect to a SAS/CONNECT remote session. 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 Options" on page 355).

## 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 Options" on page 355 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.

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, 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.

---

## Optionally Configuring CPIC Properties

### *Network Administrator*

You may define properties that automatically supply remote host information when signing on in a SAS/CONNECT session. 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 Options” on page 355 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.

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<sup>®</sup> 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. <sup>®</sup> indicates USA registration.

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