



## APPENDIX

## 8

## Windows Platform Examples

---

<i>Windows: APPC Access Method</i>	<b>606</b>
<i>SAS/CONNECT</i>	<b>606</b>
<i>Local Host</i>	<b>606</b>
<i>Remote Host</i>	<b>606</b>
<i>SAS/SHARE</i>	<b>607</b>
<i>Client</i>	<b>607</b>
<i>Server</i>	<b>607</b>
<i>Windows: CPIC Access Method</i>	<b>607</b>
<i>SAS/CONNECT</i>	<b>607</b>
<i>Local Host</i>	<b>607</b>
<i>Remote Host</i>	<b>608</b>
<i>Windows: DECnet Access Method</i>	<b>608</b>
<i>SAS/CONNECT</i>	<b>608</b>
<i>Local Host</i>	<b>608</b>
<i>Remote Host</i>	<b>608</b>
<i>SAS/SHARE</i>	<b>609</b>
<i>Client</i>	<b>609</b>
<i>Server</i>	<b>609</b>
<i>Windows: EHLLAPI Access Method</i>	<b>609</b>
<i>SAS/CONNECT</i>	<b>609</b>
<i>Local Host</i>	<b>609</b>
<i>Remote Host</i>	<b>610</b>
<i>Windows: NetBIOS Access Method</i>	<b>610</b>
<i>SAS/CONNECT</i>	<b>610</b>
<i>Local Host</i>	<b>610</b>
<i>Remote Host</i>	<b>610</b>
<i>SAS/SHARE</i>	<b>611</b>
<i>Client</i>	<b>611</b>
<i>Server</i>	<b>611</b>
<i>Windows: SPX Access Method</i>	<b>611</b>
<i>SAS/CONNECT</i>	<b>611</b>
<i>Local Host</i>	<b>611</b>
<i>Remote Host</i>	<b>612</b>
<i>SAS/SHARE</i>	<b>612</b>
<i>Client</i>	<b>612</b>
<i>Server</i>	<b>612</b>
<i>Windows: TCP/IP Access Method</i>	<b>613</b>
<i>SAS/CONNECT</i>	<b>613</b>
<i>Local Host</i>	<b>613</b>
<i>Remote Host</i>	<b>613</b>
<i>SAS/SHARE</i>	<b>613</b>

Client	613
Server	614
Windows: TELNET Access Method	614
SAS/CONNECT	614
Local Host	614
Remote Host	614

---

## Windows: APPC Access Method

---

### SAS/CONNECT

#### Local Host

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

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

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

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 that is identified as REMOTELU. In this example, REMOTELU identifies a local LU that is defined at the Microsoft SNA Server. The SIGNON statement performs the sign-on process. The USER= option in the SIGNON statement specifies that a local host be prompted for a username and a password that are valid on the remote host.

#### Remote Host

The following example illustrates the statements that you specify in a Windows NT, Windows 95, or Windows 98 remote host's configuration file to prepare for a connection from a supported local host with the APPC access method:

```
-dmr
-comamid appc
-remote remotelu
-icon
-sasdmr msgqueue
-no$syntaxcheck
-noterminal
-noxwait
```

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

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

---

## SAS/SHARE

### Client

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

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

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

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

```
options comamid=appc;
libname sasdata 'c:edc.prog2.sasdata' server=share1 user=_prompt_;
```

The APPC access method is declared. The LIBNAME statement specifies the data library that is accessed through the server SHARE1. To access a server that is running on the Windows NT platform, specify *remote-LU-alias* for the server name. The USER= option in the LIBNAME statement specifies that a client be prompted for a username and a password that are valid on the server.

### Server

The following example illustrates the statements that you specify in a SAS session on the Windows NT host at which you start a server:

```
%let appcsec=_secure_;
options comamid=appc;
proc server id=share1;
run;
```

The value `_SECURE_` for the APPCSEC macro variable requires that clients specify a userid and a password that are valid on the server. The APPC access method is declared and the server SHARE1, which is the *local-LU*, is started on the Windows NT host.

---

## Windows: CPIC Access Method

---

### SAS/CONNECT

#### Local Host

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 process.

*Note:* The value for the REMOTE= option must be identical on both the local and the remote hosts.  $\Delta$

## Remote Host

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

---

# Windows: DECnet Access Method

---

## SAS/CONNECT

### Local Host

The following example illustrates the statements that you specify in a Windows NT, Windows 95, Windows 98, or Windows 32s local host SAS session to connect to a remote host with the DECnet access method:

```
%let rmthost=rhost;
options comamid=decnet remote=rmthost;
signon user=_prompt_;
```

A macro variable is used to assign the remote host name RHOST to the alias RMTHOST. The OPTIONS statement specifies the DECnet access method and the macro variable RMTHOST as the remote host. The SIGNON statement performs the sign-on process. The USER= option in the SIGNON statement specifies that a client be prompted for a username and a password that are valid on the server.

### Remote Host

In order to allow a local host to connect to a Windows NT, Windows 95, Windows 98, or Windows 32s remote host, a PC spawner program must be invoked from the remote host. The spawner program is invoked with the DECnet access method using:

```
\sas\connect\sasexe\spawner -comamid decnet -file spawnsas.bat;
```

The SPAWNSAS.BAT file is used to set the configuration on the PC. The SPAWNSAS.BAT file contents is:

```
@echo off
sas -config config.sas %1 %2 %3 %4 %5 %6 %7 %8
```

The following example illustrates the configuration file entries for a Windows NT, Windows 95, Windows 98, or Windows 32s remote host:

```
-no$syntaxcheck
-noterminal
-noxwait
```

---

## SAS/SHARE

### Client

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

```
options comamid=decnet;
libname sasdata 'edc.prog2.sasdata' server=rhost.share1 user=_prompt_;
```

The COMAMID option specifies the DECnet access method. The LIBNAME statement specifies the data library that is accessed through the *node.server-id* RHOST.SHARE1. The USER= option in the LIBNAME statement specifies that a client be prompted for a userid and a password that are valid on the server.

### Server

The following example illustrates the statements that you specify in a SAS session on the Windows NT host at which you start a server:

```
%let sassecur=_secure_;
options comamid=decnet;
proc server id=share1;
run;
```

The value `_SECURE_` for the SASSECUR macro variable requires that clients specify a userid and a password that are valid on the server. The DECnet access method is declared, and the server SHARE1 is started on the Windows NT host.

---

## Windows: EHLLAPI Access Method

---

### SAS/CONNECT

#### Local Host

The following example illustrates the statements that you specify in a Windows local host SAS session to connect to a remote host with the EHLLAPI access method:

```
filename rlink '!sasroot\connect\saslink\tso.scr';
options comamid=ehllapi remote=a;
signon;
```

The first line identifies the script file that you use to sign on to an OS/390 remote host. The script file contains a prompt for a userid and a password that are valid on the remote host. The EHLLAPI communications access method is declared with a connection to the remote host A, which is the remote session identifier that was specified when the emulation package was configured on your local host. The SIGNON statement performs the sign-on process.

## Remote Host

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

---

## Windows: NetBIOS Access Method

---

### SAS/CONNECT

#### Local Host

The following example illustrates the statements that you specify in a Windows local host SAS session to connect to a remote host with the NetBIOS access method:

```
options set=vqmlinks 3 set=vqmconvs 3;
options comamid=netbios remote=sasrem;
signon;
```

This example assumes a connection to a PC spawner that is running in secure mode. Two options are set (see “SAS/CONNECT and SAS/SHARE Options” on page 392 for details). The NetBIOS communications access method is declared with a connection to the remote host SASREM. SASREM is the name that is specified in the -NETNAME option that the PC spawner uses to communicate with the local host. The USER= option to SIGNON specifies that the connecting local host be prompted for a userid and a password that are valid on the remote host. The SIGNON statement performs the sign-on process.

#### Remote Host

The following example illustrates the statements that you specify in a Windows NT, a Windows 95, or a Windows 98 remote host configuration file to prepare for a connection from a supported local host with the NetBIOS access method:

```
-no$syntaxcheck
-noterminal
-noxwait
```

An example follows of how the PC spawner is invoked on a Windows NT, a Windows 95, or a Windows 98 remote host:

```
c:\sas\connect\sasexe\spawner -comamid netbios -netname sasrem
                               -file mysas.cmd
```

The spawner is invoked and the NetBIOS access method is specified. The -NETNAME option specifies the name of the network (SASREM) that the PC spawner program uses to communicate with the local host. The -FILE option executes the MYSAS.CMD file, which invokes a SAS session.

See “Starting the PC Spawner Program” on page 237 for information about the contents of a command file and executing the PC spawner. Options that are set through the spawner may override options that are set in a remote host configuration file.

---

## SAS/SHARE

### Client

The following example shows the statements that are specified in a Windows NT client session:

```
options comamid=netbios;
libname sasdata 'c:\edc\prog2\sasdata' user=_prompt_ server=share1;
```

The NetBIOS access method is declared. The LIBNAME statement specifies the data library that is accessed through the server SHARE1. The USER= option in the SIGNON statement specifies that the client be prompted for a userid and a password that are valid on the server SHARE1.

### Server

Specify the following statements in a SAS session on the Windows NT remote host to start a server:

```
%let sassecur=_secure_;
options comamid=netbios;
proc server id=share1;
run;
```

The first line uses the SAS macro variable SASSECUR to prompt clients for a userid and a password that are valid on the server. The NetBIOS access method is declared for the server SHARE1 that is started on a Windows NT remote host.

---

## Windows: SPX Access Method

### CAUTION:

**Version 6 Only** Beginning with Version 7, the SPX access method is not supported. However, information about SPX is included here for Version 6 users. △

---

## SAS/CONNECT

### Local Host

The following example illustrates the statements that you specify in a Windows NT or a Windows 95 local host SAS session to connect to a remote host with the SPX access method:

```
options set=sasuser userid set=saspass password;
options set=spxmsgsize 4202;
options comamid=spx remote=sasrem;
signon;
```

This example assumes a connection to a PC spawner that is running in secure mode. The SAS options SASUSER and SASPASS allow the userid and the password to be passed to the remote PC spawner, which permits a connection. SPXMSGSIZE is set (see “Setting SAS Options and Variables” on page 404 for details). The SPX

communications access method is declared with a connection to the remote host SASREM, which is the name that is specified in the -SPXNAME option to the PC spawner invocation. The SIGNON command performs the signon procedure.

## Remote Host

The following example illustrates the statements that you specify in a Windows NT or a Windows 95 remote host configuration file to prepare for a connection from a supported local host with the SPX access method:

```
-no$syntaxcheck
-noterminal
-noxwait
```

The following example shows how to invoke the PC spawner on a Windows NT remote host:

```
c:\sas\connect\sasexe\spawner -comamid spx -spxname sasrem -file mysas.cmd
```

The PC spawner is invoked, and the SPX access method is specified. The -SPXNAME option specifies the name that the PC spawner program uses to communicate with the local host. The -FILE option executes the MYSAS.CMD file, which invokes a SAS session.

See “Starting the PC Spawner Program” on page 237 for information about the contents of a command file and executing the PC spawner. Options that are set by means of the spawner may override options that are set in a remote host configuration file.

---

## SAS/SHARE

### Client

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

```
options comamid=spx;
libname sasdata 'c:\edc\prog2\sasdata' server=share1;
```

The SPX access method is declared. The LIBNAME statement specifies the data library that is accessed through the server SHARE1.

### Server

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

```
-set spxmsgsize 4202
```

See “Setting SAS Options and Variables” on page 388 for details about this option.

The following statements issued in a SAS session on the Windows remote host illustrate how to start a server:

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

The SPX access method is declared for the server SHARE1 that is started on the Windows NT remote host.



---

## Windows: TCP/IP Access Method

---

### SAS/CONNECT

#### Local Host

The following example illustrates the statements that you specify in a Windows local host SAS session to connect to a remote host with the TCP/IP access method:

```
filename rlink '!sasroot\connect\saslink\tcpcms.scr';
options comamid=tcp remote=rmtnode;
signon user=_prompt_;
```

The first line identifies the script file that you use to sign on to a CMS remote host. The TCP/IP communications access method is declared with a connection to the remote host RMTNODE. The SIGNON statement performs the sign-on process. The USER= option in the SIGNON statement specifies that a local host be prompted for a username and a password that are valid on the remote host.

#### Remote Host

You may set the following options in the Windows NT, the Windows 95, or the Windows 98 remote host SAS invocation to restrict port access:

```
-tcpportfirst=5020;
-tcpportlast=5050;
```

These statements restrict access to ports 5020 through 5050.

The following example shows how the PC spawner is invoked on a Windows NT, a Windows 95, or a Windows 98 remote host:

```
c:\sas\connect\sasexe\spawner -comamid tcp -file mysas.cmd
```

The spawner is invoked and the TCP/IP access method is specified. The -FILE option executes the MYSAS.CMD file, which invokes a SAS session.

See “Starting the PC Spawner Program” on page 237 for information about the contents of a command file and executing the PC spawner. Options that are set by means of the spawner may override options that are set in a remote host configuration file.

---

### SAS/SHARE

#### Client

The following example illustrates the statements that you specify in a Windows NT client SAS session to connect to a server with the TCP/IP access method:

```
options comamid=tcp;
libname sasdata 'c:edc\prog2\sasdata' server=rmtnode.share1 user=_prompt_;
```

The LIBNAME statement specifies the data library that is accessed through the server that is represented by the two-level name RMTNODE.SHARE1. The USER= option in the LIBNAME statement specifies that a client be prompted for a userid and a password that are valid on the server.

## Server

The following example illustrates the statements that you specify in the server's configuration file on a Windows NT host:

```
-set tcpsec _secure_
-set authencr required
```

The value `_SECURE_` for the TCPSEC option specifies that clients supply a userid and a password that are valid on the server. The value `REQUIRED` for the AUTHENCRCR option specifies that only encrypted userids and passwords from clients are accepted.

The following example illustrates the statements that you specify in a SAS session on the Windows NT host at which you start a server:

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

The TCP/IP access method is declared and the server SHARE1 is started on the Windows NT host.

---

## Windows: TELNET Access Method

---

### SAS/CONNECT

#### Local Host

The following example illustrates the statements that you specify in a Windows local host SAS session to connect to a remote host with the TELNET access method:

```
filename rlink '!sasroot\connect\saslink\telcms.scr';
options comamid=telnet remote=rmtnode;
signon;
```

The first line identifies the script file that you use to sign on to a CMS remote host. The script file prompts for a userid and a password that are valid on the remote host. The TELNET communications access method is declared with a connection to the remote host RMTNODE. The SIGNON command performs the sign-on process.

#### Remote Host

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

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.