



CHAPTER

7

OS/390: TCP/IP Access Method

<i>Tasks That Are Common to SAS/CONNECT and SAS/SHARE</i>	108
<i>System and Software Requirements for SAS/CONNECT and SAS/SHARE</i>	108
<i>Defining Resources for the TCP/IP Access Method</i>	108
<i>Understanding TCP/IP Access Method Terminology</i>	109
<i>Setting SAS Options and Variables</i>	109
<i>Displaying SAS System Option Settings</i>	110
<i>Setting Security for SAS/CONNECT and SAS/SHARE</i>	110
<i>Providing Client Identification in a Version 8 Session</i>	110
<i>Providing Client Identification in a pre-Version 8 Session</i>	111
<i>Providing Userid-Based Security for a SAS/SHARE Server</i>	112
<i>SAS/CONNECT and SAS/SHARE Options</i>	112
<i>SAS/CONNECT Only Options and Variables</i>	113
<i>SAS/SHARE Only Option</i>	114
<i>SAS/CONNECT</i>	115
<i>Local Host Tasks</i>	115
<i>Remote Host Connection Considerations</i>	115
<i>Configuring the Spawner in the SERVICES File</i>	116
<i>Setting Security for Local Hosts</i>	116
<i>Specifying the TCP/IP Access Method</i>	116
<i>Specifying the Remote Node Name</i>	116
<i>Identifying a Script File for Signing On and Signing Off</i>	117
<i>Signing On to the Remote Host</i>	118
<i>Local Host Example</i>	118
<i>Remote Host Tasks</i>	118
<i>Installing the Logon Procedure on the Remote Host</i>	119
<i>Remote Host Example</i>	119
<i>SAS/SHARE</i>	119
<i>Client Tasks</i>	119
<i>Configuring the Server in the SERVICES File</i>	120
<i>Setting Security for Connecting Clients</i>	120
<i>Specifying the TCP/IP Access Method</i>	120
<i>Specifying a Server Name</i>	121
<i>Client Example</i>	121
<i>Server Tasks</i>	122
<i>Installing the SAS SVC Routine</i>	122
<i>Configuring the Server in the SERVICES File</i>	122
<i>Setting Server Security</i>	123
<i>Enforcing Server Userid and Password Encryption</i>	123
<i>Specifying the TCP/IP Access Method</i>	123
<i>Specifying a Server Name</i>	124
<i>Server Example</i>	124

<i>System Configuration for the TCP/IP Access Method</i>	124
<i>Installing the SAS/C Transient Library</i>	125
<i>Understanding the Function of Network Configuration Files</i>	126
<i>IBM TCP/IP Network Configuration Files</i>	126
<i>Setting SAS/C Environment Variables with the SASCTCPV DDname</i>	126
<i>Interlink TCP/IP Network Configuration Files</i>	126
<i>Understanding the Search Order for Locating Network Configuration Files</i>	127
<i>Understanding the Search Order for Locating Host Names and Internet Addresses</i>	127
<i>Knowing about the HOSTS File</i>	128
<i>HOSTS File Search Order</i>	128
<i>Knowing about the SERVICES File</i>	128
<i>SERVICES File Search Order</i>	128
<i>Knowing about the RESOLV.CONF File</i>	129
<i>RESOLV.CONF File Search Order</i>	129
<i>Knowing about Configuration File Naming Conventions</i>	129
<i>Using the ETC High-Level Qualifier</i>	130
<i>Using the TCPIP Default Prefix</i>	130
<i>Applying a Zap to Change the Default TCPIP Prefix</i>	130
<i>Setting the TCPIPREF Option to the TCPIP Prefix Value</i>	130
<i>Redirecting Error Output with the SASCTCPE DDname</i>	130
<i>Reference</i>	131

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

System Administrator or User

To use the TCP/IP access method with an OS/390 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 TCP/IP access method have been defined.
- 3 Verify that you know how to set SAS options.
- 4 Set the SAS/CONNECT and SAS/SHARE options that you want.

System and Software Requirements for SAS/CONNECT and SAS/SHARE

SAS/CONNECT and SAS/SHARE require the SAS/C Transient Library when running SAS Release 6.08 TS405 and subsequent releases. (The SAS/C Transient Library is provided with SAS/CONNECT and SAS/SHARE.)

Note: If your site has installed Release 5.50 of the SAS/C Transient Library, you must replace it with the transient library that is included with SAS/CONNECT and SAS/SHARE. See “Installing the SAS/C Transient Library” on page 125 for details. △

SAS/CONNECT and SAS/SHARE also require the Interlink SNSTCP, Version 4.1 or a subsequent release and IBM TCP/IP, Version 3.2 or a subsequent release.

Defining Resources for the TCP/IP Access Method

System Administrator, SAS Site Representative, Applications Programmer, or User
 Before you can use SAS/CONNECT or SAS/SHARE with the TCP/IP access method, you must first define TCP/IP resources for the OS/390 system. See “System Configuration for the TCP/IP Access Method” on page 124 for the tasks that you perform to define resources for SAS/CONNECT and SAS/SHARE.

Understanding TCP/IP Access Method Terminology

Familiarity with these terms will help you when you set SAS options:

name resolution

The process of mapping a host name to an address. The domain name system provides a facility for naming hosts in which programs use remote name servers to resolve host names into IP addresses for those hosts.

name server

The server program that supplies name-to-address translation, mapping from domain names to IP addresses. The server processor often runs on a dedicated processor, and the host itself is referred to as the name server.

name resolver

The client software that uses one or more name servers when translating a host name.

For a complete discussion of TCP/IP, see *Internetworking with TCP/IP, Volume I: Principles, Protocols, and Architecture, Second Edition* from IBM.

Setting SAS Options and Variables

You may need to set specific SAS options and variables to establish the desired connections with SAS/CONNECT and SAS/SHARE when using the TCP/IP communications access method.

Consult with your network administrator to determine what options must be set and what values to assign to them.

You may specify an option in several forms, as follows:

- OPTIONS statement in a SAS AUTOEXEC file or in a SAS session:

`OPTIONS variable=value;`

Example:

`OPTIONS COMAMID=TCP;`

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

`option-name=value`

Example:

`tcppipmch=tcPIP`

- SAS macro variable:

`%LET variable=value;`

Example:

`%let tcPIPmch=tcPIP;`

- CLIST variable:

Add `option-name(default value)` to the SAS CLIST. Then add `option-name(current value)` at a SAS invocation.

Example CLIST:

```
tcpipmch(tcpip)          /* TCP options */ +
```

Specify the CLIST at the SAS invocation as follows:

```
sas tcpipmch(tcpip) o('dmr comamid=tcp noterminal')
```

- environment variables in a SASCTCPV data set:

Add *environment_variable=value* to the SASCTCPV data set. See “Setting SAS/C Environment Variables with the SASCTCPV DDname” on page 126 for more information about this data set.

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 (_).

Note: If you set the same option using different forms, typically the last option setting will take precedence and override an earlier option setting. Δ

Displaying SAS System Option Settings

To display the settings of the SAS system options in the SAS log, use the OPTIONS procedure. The following statement produces a list of options with a brief explanation of what each option does:

```
proc options;
run;
```

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 two sections outline the version-specific methods for specifying client identification for SAS/CONNECT and SAS/SHARE. The third section describes how to configure your SAS/SHARE server to either require or not require connecting clients to supply user identification.

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 TCPSEC option is still accepted but is not recommended in Version 8. The USER= and PASSWORD= options take precedence over the client TCPSEC option when both are specified. For example, a SAS/SHARE client's execution of a LIBNAME statement with values assigned to the USER= and PASSWORD= options would override a TCPSEC option 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 assign to a SAS/SHARE server may require specification of the options, subsequent assigns to the same server will not need specification of these options as long as the original connection is in effect. A subsequent re-connect to the same server or connect to a different server would require re-supplying of the USER= and PASSWORD= options.

Here is a Version 8 example for SAS/SHARE:

```
libname test 'prog2 a' user=joeblue password="2muchfun" server=share1;
```

For SAS/CONNECT, these values are valid until SIGNOFF.

Here is a Version 8 example for SAS/CONNECT:

```
signon rmthost user=joeblack password=born2run;
```

As a security precaution, PASSWORD= field entries echoed in the log are replaced with Xs. If _PROMPT_ was specified for entering the password, the entry would not be displayed on the screen as it is typed.

Providing Client Identification in a pre-Version 8 Session

In Version 6 and Version 7, you provide client identification to a SAS/CONNECT remote host or a SAS/SHARE server using the TCPSEC option. TCPSEC must be

defined on the local host before you connect to the remote host (using the SIGNON statement) or access a SAS/SHARE server (using the LIBNAME statement).

Here is the syntax and description of this option.

TCPSEC=*userid.password* | PROMPT

userid.password

specifies the remote host userid and password and is thus host-dependent in form. If either the userid or password contains blanks or special characters, it must be enclosed in quotes. A period (.) is used as a delimiter between the userid and password and, therefore, is not a valid character.

PROMPT

specifies that the SAS system prompt the client for the userid and password.

Note: The values provided when prompted must NOT be quoted. Δ

This technique is especially useful when the configuration file specifying this option is shared among many users.

Examples:

```
TCPSEC=bass.time2go;
TCPSEC=PROMPT;
```

Providing Userid-Based Security for a SAS/SHARE Server

The TCPSEC option also specifies whether the TCP/IP access method performs user authentication before connecting to a SAS/SHARE server. The TCPSEC option must be set on the server before you start the SAS/SHARE server.

Here is the syntax and description of this option.

TCPSEC=SECURE | NONE

SECURE

The SECURE value for the TCPSEC option causes the TCP/IP access method to attempt to authenticate connecting SAS/SHARE clients. Each client connecting using TCP/IP is required to supply a userid and password valid for the host on which the server is running.

NONE

The NONE value for the TCPSEC option causes the TCP/IP access method to NOT attempt to authenticate connecting SAS/SHARE clients. This is the default action when TCPSEC has not been set.

Examples:

```
TCPSEC=SECURE;
TCPSEC=NONE;
```

SAS/CONNECT and SAS/SHARE Options

You may set these options at the SAS/CONNECT local and remote hosts and at the SAS/SHARE client and server.

ICSRSLV = ONLY | FIRST | LAST | NEVER

Specifies when or if the ICS name resolver is called to resolve an Internet address to a host name.

Note: This option is available for Interlink TCP/IP sites only. Δ

ONLY

specifies that only the ICS name resolver is called.

FIRST

specifies that the ICS name resolver is called first. If an error occurs, then the SAS/C resolver is called.

LAST

specifies that the SAS/C resolver is called first. If an error occurs, then the ICS name resolver is called.

NEVER

specifies that the ICS name resolver is never called. The SAS/C resolver is always called.

See “Understanding the Search Order for Locating Host Names and Internet Addresses” on page 127 for details about the resolver.

Note: You may set ICSRSLV in a SAS configuration file, at a SAS invocation, or in a CLIST variable. Δ

TCPIPMCH=*value*

identifies which version of TCP/IP to use at sites that simultaneously run multiple versions of TCP/IP.

If the TCPIPMCH option is not defined, SAS uses the name of the address space that is running TCP/IP. If the Interlink TCP/IP is used, then SAS uses the value ACSS. For all other versions of TCP/IP, SAS uses the value TCPIP.

Note: You may set TCPIPMCH in a SAS configuration file, at a SAS invocation, or in a CLIST variable. Δ

TCPIPPRF=*name*

allows you to devise a naming convention for a data set at your site by attaching a descriptive prefix to data set names. For example, to attach the descriptive prefix SYS2.VER2.TCP to the configuration file ETC.HOSTS set the TCPIPPRF option as follows:

```
TCPIPPRF=SYS2.VER2.TCP
```

This option setting produces the data set name SYS2.VER2.TCP.ETC.HOSTS.

Note: The TCPIPPRF option initializes a data set prefix for the current SAS session. You must set this option each time you invoke a SAS session on the local and the remote hosts in a SAS/CONNECT session and at the SAS/SHARE server and client. Δ

Note: You may set TCPIPPRF in a SAS configuration file, at a SAS invocation, or in a CLIST variable. Δ

SAS/CONNECT Only Options and Variables

TCPPORTFIRST**TCPPORTLAST**

The TCPPORTFIRST and TCPPORTLAST SAS options restrict the range of TCP/IP ports through which local hosts can remotely connect to remote hosts.

These options must be set at the SAS/CONNECT remote host.

Define the range of TCP/IP ports by assigning a beginning range value to TCPPORTFIRST and an ending range value to TCPPORTLAST, within the range of 0 through 32767.

Consult with your network administrator for advice about these settings.
Use the following syntax for the configuration file:

```
TCPPORTFIRST=n
TCPPORTLAST=n
```

Use the following syntax for the AUTOEXEC file:

```
OPTIONS TCPPORTFIRST=n;
OPTIONS TCPPORTLAST=n;
```

In the following example, the local host is restricted to TCP/IP ports 4020 through 4050 when making a remote host connection:

```
options tcpportfirst=4020;
options tcpportlast=4050;
```

To restrict the range of ports to only one port, you may set the TCPPORTFIRST and TCPPORTLAST options to the same number.

Note: At the remote host, you may set TCPPORTFIRST and TCPPORTLAST in an OPTIONS statement, at a SAS invocation, in the configuration file, or in the AUTOEXEC file. Δ

TCPTN3270

TCPTN3270 is an environment variable that is set on the local host to support connections to CMS and OS/390 remote hosts that use the full-screen 3270 TELNET protocol. The following sample script files are provided:

```
CMS          TCPCMS32
OS/390       TCPTSO32
```

See “Identifying a Script File for Signing On and Signing Off” on page 117 for more information.

To set the TCPTN3270 variable, perform the following tasks at the OS/390 local host:

- Set the TCPTN3270 CLIST variable at the local host.
- Add TCPTN3270(1) to the SAS CLIST.

Example CLIST:

```
tcptn3270(1)
```

If you do not set this variable, the TCP/IP access method uses the TELNET line mode protocol by default.

SAS/SHARE Only Option

By default, a secure server accepts userids and passwords from clients in either encrypted or plain text form. The option to accept either form ensures compatibility with client sessions running older releases of SAS/SHARE.

To require only encrypted userids and passwords, you must set the AUTHENCR option as an environment variable or a SAS macro variable. Requiring encryption ensures that all clients have been upgraded to Release 6.11 or the 6.09 Enhanced Release of SAS.

Setting this option in a server session enables encryption for clients connecting to a secured server. The values for this option follow:

```
AUTHENCR=OPTIONAL | REQUIRED
```


OPTIONAL

means that a client can optionally encrypt the username and the password that it sends to the server. This is the default. When using the default, the server allows connections from clients that are capable of encryption and from clients that are incapable of using encryption because they are running earlier releases of SAS (releases prior to the 6.09E and 6.11) that do not support encryption.

REQUIRED

means that each client must encrypt the username and password that it sends to the server.

For examples of the forms you can use to specify the AUTHENCR option, see “Setting SAS Options and Variables” on page 109.

SAS/CONNECT

Local Host Tasks

User or Applications Programmer

To connect an OS/390 local host to a remote host, perform these tasks at the local host:

- 1 Consider the requirements of the remote host that you are connecting to.
- 2 If you connect to a UNIX remote host or an OS/390 remote host by means of the spawner program, configure the UNIX spawner service in the SERVICES file, as necessary.
- 3 If you connect to a remote host by means of the UNIX spawner the PC spawner, or the OS/390 spawner, set security, as necessary.
- 4 Specify the communications access method.
- 5 Specify a remote host to connect to.
- 6 Identify the script file to be used for signing on and signing off, as necessary.
- 7 Sign on to the remote host.

Remote Host Connection Considerations

If you are connecting to a Windows 95, Windows 98, or Windows NT remote host, you *must* connect by means of a spawner program that is already running on the remote host. If you are connecting to an OS/2, a UNIX, or an OS/390 remote host, you *optionally may* connect by means of a spawner program that also is already running on the remote host. A spawner program allows the encryption of userids and passwords when passed through the network. Without a spawner, readable userids and passwords are passed through the network, which may present a security risk. See Chapter 32, “Spawner Programs,” on page 457 for information about starting the spawner on the remote host.

You may also sign on to the remote host with a script file. If you do not sign on with a script file, as a security measure, set the USER= and PASSWORD= options in the SIGNON statement, which is passed to the remote host, allowing a local host connection.

Note: Setting the Version 7 TCPSEC option at the local host also works. △

If the -NOSCRIP option is set at the spawner invocation, sign on with a script is prohibited. Ask your network administrator whether the -NOSCRIP option is set at the spawner invocation.

If you sign on to a host that is not a Windows NT, a Windows 95, a Windows 98, an OS/2, a UNIX, or an OS/390 host, you will sign on with a script.

Configuring the Spawner in the SERVICES File

Before connecting to a UNIX remote host or an OS/390 remote host with the spawner program, configure the spawner service in the SERVICES file on the local host. See “Starting the OS/390 Spawner” on page 465 for information about configuring the SERVICES file.

Setting Security for Local Hosts

If you are not using a script file to sign on to the remote host, set security at the local host using either of the methods explained in “Setting Security for SAS/CONNECT and SAS/SHARE” on page 110. For Version 8 security behavior, specify the USER= and PASSWORD= options to the SIGNON statement. For details, see “Providing Client Identification in a Version 8 Session” on page 110.

For Version 7 security behavior, if you set the TCPSEC option at the local host, either specify a userid and a password that are valid on the remote host or specify PROMPT to supply the userid and password when connecting to a remote host. For information about setting the TCPSEC option, see “Providing Client Identification in a pre-Version 8 Session” on page 111.

Specifying the TCP/IP Access Method

You must specify the TCP/IP 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. TCP (short for TCP/IP, which is an abbreviation for Transmission Control Protocol/Internet Protocol) is an example of *access-method-id*.

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

Example:

```
options comamid=tcp;
```

Specifying the Remote Node Name

To make a connection from an OS/390 local host to a remote host, use the following syntax:

```
OPTIONS REMOTE=node-name<.service-name>;
```

The value of *node-name* is based on the type of remote host that you are connecting to.

- If you are connecting to a Windows NT, a Windows 95, a Windows 98, or an OS/2 remote host that is running the PC spawner program, use the name of the node on which the PC spawner is running. See Chapter 32, “Spawner Programs,” on page 457 for more information.
- If you are connecting to a UNIX or an OS/390 host that is running a spawner program, use a two-level name in the form of *node-name.service-name*, where

node-name specifies the node on which the spawner program is running and *service-name* specifies the port on which the spawner is listening for a connection request.

See Chapter 32, “Spawner Programs,” on page 457 for information about the spawner program. See “Starting the UNIX Spawner Program” on page 479 for information about configuring the spawner service in the SERVICES file.

- If you are connecting to any other remote host platforms that use a sign-on script, use the node name of the remote host. The remote host must be defined in a local HOSTS file or in a domain server.

The value of the REMOTE= option must be a valid SAS name. See *SAS Language Reference: Dictionary* for details about SAS naming rules.

Example:

```
OPTIONS REMOTE=node-name;
```

If you use an Internet address (or some other invalid SAS name), you must assign the address to a macro variable and specify the macro variable for the value of the REMOTE= option. Here is an example:

```
%let node=Internet-address;
options remote=node;
```

Do not choose a macro name that is also a valid host name on your network. SAS first attempts to reach a network host with the value of the REMOTE= option (in this example, MYNODE).

```
%let mynode=149.999.228.6;
options remote=mynode;
```

Identifying a Script File for Signing On and Signing Off

To use one of the sample script files that is supplied with SAS/CONNECT for signing on and signing off, assign the RLINK fileref to the appropriate script file, based on the remote host that you are connecting to. The sample script files are installed in your *prefix*.CTMISC data set. You must customize the sample scripts to accurately reflect the logon processes at your site. Failure to do so will produce errors.

The fileref format follows:

```
FILENAME RLINK '&prefix.CTMISC(script-name)';
```

where *script-name* identifies the script that corresponds to the remote host that you want to connect to.

The following table lists the scripts that are supplied by SAS Institute:

Table 7.1 OS/390 TCP/IP SAS/CONNECT Sign-on Scripts

Remote Host	Script Name
CMS	TCPCMS
CMS (using full-screen 3270 TELNET protocol)	TCPCMS32
OS/390 (with TSO)	TCPTSO
OS/390 (without TSO)	TCPMVS
OS/390 (using full-screen 3270 TELNET protocol)	TCPTSO32
OpenVMS	TCPVMS

Remote Host	Script Name
OS/2	TCPOS2
UNIX	TCPUNIX
Windows NT, Windows 95 , and Windows 98	TCPWIN

Example:

```
filename rlink '&prefix.CTMISC.(tcpunix)';
```

Signing On to the Remote Host

To complete your sign on to the remote host, enter the SIGNON statement, as follows:

```
signon user=_prompt_;
```

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 110.

Local Host Example

The following example illustrates the statements that you specify in an OS/390 local host SAS session to connect to a remote host with the TCP/IP access method.

```
filename rlink 'sas.ctmisc.tcpvms';
options comamid=tcp remote=rmthost;
signon user=_prompt_;
```

The first line identifies the script file that you use to sign on to the OpenVMS remote host. The script file contains a prompt for a userid and a password that are valid on the remote host. The TCP/IP communications access method is declared with a connection to the remote host RMTHOST. The USER= option in the SIGNON statement specifies that the connecting local host be prompted for a userid and a password that are valid on the remote host.

Remote Host Tasks

System Administrator

You have two options:

- 1 To allow a connection from a local host to an OS/390 remote host that is running an OS/390 spawner, start the spawner program at the remote OS/390 host. For information about how to start the spawner, see Chapter 32, “Spawner Programs,” on page 457.

2

To allow a connection from a local host to an OS/390 remote host without running a TSO terminal monitor program, install the logon procedure on the remote OS/390 host.

If you are signing on to an OS/30 remote host with TSO, there are no remote host tasks.

Installing the Logon Procedure on the Remote Host

For OS/390 remote host connections, you can eliminate the need for TSO by replacing the terminal monitor program (also called logon procedure) with a procedure that invokes SAS with the options that you want. The benefits of this method are that signing on and signing off an OS/390 remote host is much faster than running with TSO, and you eliminate the overhead consumed by running TSO. However, a disadvantage of running without TSO is that you cannot execute any X commands or TSO commands.

In the following example, the logon procedure invokes SAS with the DMR and the COMAMID=TCP options. When you log on to the OS/390 remote host, this procedure is immediately run so that the current OS/390 account is limited to running SAS each time that the current OS/390 account user logs on.

```
//JOBDDL PROC ENTRY=SASHOST,
//          OPTIONS=,
//          WORK='500,200'
//JOBDDL EXEC PGM=&ENTRY,
// PARM='&OPTIONS DMR COMAMID=TCP',REGION=4096K
//STEPLIB DD DISP=SHR,DSN=&prefix.TS450.LIBRARY
//CONFIG DD DISP=SHR,DSN=&prefix.TS450.CNTL(TSOXA)
//SASAUTOS DD DISP=SHR,DSN=&prefix.TS450.AUTOLIB
//SASHELP DD DISP=SHR,DSN=&prefix.TS450.SASHELP
//SASMSG DD DISP=SHR,DSN=&prefix.TS450.SASMSG
//WORK DD UNIT=SYSDA,SPACE=(6144,(&WORK),,ROUND),
//      DCB=(RECFM=FS,DSORG=PS,LRECL=6144,BLKSIZE=6144)
//SASPARM DD UNIT=SYSDA,SPACE=(400,(100,300)),
//      DCB=(RECFM=FB,LRECL=80,BLKSIZE=400,BUFNO=1)
```

A script file is still required at the local host for sign on. However, a SAS invocation is not included in the script file because the logon procedure already performs SAS invocation.

See "TCPMVS.SCR Script" on page 520 for the content of the script file.

Remote Host Example

You may set the following variables to restrict port access in the AUTOEXEC file:

```
options tcpportfirst=5020;
options tcpportlast=5050;
```

These statements restrict access to ports 5020 through 5050.

SAS/SHARE

Client Tasks

User and Applications Programmer

To prepare for creating and accessing a SAS/SHARE server, make sure that you know how to:

- 1 Configure the server in the client SERVICES file.
- 2 Set security.
- 3 Specify the TCP/IP access method.
- 4 Specify a server name.

Configuring the Server in the SERVICES File

Each server must be defined as a service in the SERVICES file on each host node from which a client session will access the server. This file usually is located in the directory in which the TCP/IP software is installed. See “Configuring the SERVICES File” on page 485 for information about editing the SERVICES file.

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 password that are valid on the server host. For details, see “Setting Security for SAS/CONNECT and SAS/SHARE” on page 110.

Specifying the TCP/IP Access Method

You must specify the TCP/IP communications access method at the server before you can create a server.

Use the following syntax to specify the TCP/IP 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. TCP (short for TCP/IP, which is an abbreviation for Transmission Control Protocol/Internet Protocol) is an example of an *access-method-id*.

Example:

```
options comamid=tcp;
```

The server is accessed using the TCP/IP 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 an OS/390 client connecting to an OS/390 server follows:

```
comamid=tcp
comaux1=appc
```

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

Specifying a Server Name

If the client and server sessions are running on different network nodes, you must include the node in the server identifier in the LIBNAME and PROC OPERATE statements as follows:

```
SERVER=node.server
```

This representation is known as a two-level server name.

node must be a valid TCP/IP node name. If the server and the client sessions are running on the same node, you may omit the node name.

server can represent either a *server-id* or a *port* number.

- server-id* must be identical to the service name specified in the SERVICES file. See “Configuring the Server in the SERVICES File” on page 122 for more information on specifying the *server-id* in the SERVICES file.
- port* is the location for passing data to and receiving data from the server. The port number is specified with two preceding underscore (_) characters. For example, you can specify the server port as 5000 using the SERVER= option in a LIBNAME statement:

```
libname mylib '.' server=srvnode.__5000;
```

If the TCP/IP node name is not a valid SAS name, assign the name of the server's node to a SAS macro variable, then use the name of the macro variable for *node* in the two-level server name.

The access method evaluates the node name, in this order of precedence:

- acceptable node name
- SAS macro variable.

The following example shows the assignment of a SAS macro variable to a server's node name:

```
%let srvnode=mktserve.acme.com;
libname sales server=srvnode.server1;
```

Note: Do not use an ampersand (&) in a two-level server name. An ampersand would cause the macro variable to be resolved by the SAS parser prior to syntactic evaluation of the SERVER option. The access method evaluates the node name in a two-level server name. Δ

See *SAS Language Reference: Dictionary* for details about SAS naming rules. See *SAS/SHARE User's Guide* for details about the LIBNAME and PROC OPERATE statements.

Client Example

The following example illustrates the statements that you specify in an OS/390 client SAS session to access a server with the TCP/IP access method:

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

The first line declares the TCP/IP access method. The LIBNAME statement specifies the data library that is accessed through the server, which is specified by the two-level server name RMTNODE.SHARE1, by means of a prompt for a username and a password that are valid on the server.

Server Tasks

Server Administrator

Your primary tasks are

- 1 Verify that the SAS SVC routine has been installed.
- 2 Configure SAS/SHARE servers in the SERVICES file.
- 3 Set server security through the TCPSEC option.
- 4 Set the AUTHENCR option to enforce userid and password encryption, as necessary.
- 5 Specify the TCP/IP access method.
- 6 Specify the server name.

Installing the SAS SVC Routine

The SAS SVC control program routine is an interface between the OS/390 operating system and a specific request, such as "third-party checking." This facility provides verification in the form of calls for authentication of userids and passwords and of library authority.

- 1 Install the SAS SVC routine, if necessary.

If you have already installed the SAS SVC routine for Release 6.09 of SAS software, do not repeat the step here. If you need to perform the installation, see the *Installation Instructions and System Manager's Guide, The SAS System under MVS* for details.

Because SAS SVC in Release 6.09 is backward compatible, it replaces the SAS SVC routines from previous releases. You may continue using previous releases of base SAS and SAS/SHARE with the Release 6.09 SAS SVC that is installed on your system.

- 2 Verify the SVC routine SAS system options.

Verify that the SAS system options for the SVC routine accurately reflect the way that the SAS SVC is installed. The SAS system option SVC0SVC should be set to the number at which the SAS SVC is installed (for example, 251 or 109). If the SAS SVC is installed at 109 as an ESR SVC, the SAS system option SVC0R15 should be set to the ESR code (for example, 4).

- 3 Verify installation on all CPUs, as needed.

If you have more than one CPU, verify that the SAS SVC is installed on the systems that will be running SAS/SHARE at your site.

Configuring the Server in the SERVICES File

Each server must be defined as a service in the TCP/IP SERVICES file on each remote host node on which a server runs and on each node from which a client session

will access the server. This file usually is located in the directory in which the TCP/IP software is installed. See “Configuring the SERVICES File” on page 485 for more information.

Setting Server Security

You may use file permissions to restrict a user’s access to libraries and files through a server. A secured server allows connections only from those clients that provide valid userids and passwords for the host on which the server is running. A secured server uses a validated userid and password to verify a user’s authority to access a SAS library or a SAS file.

Requiring connecting clients to supply both a valid userid and a password enforces server security. From a server session, set the TCPSEC option to the value `_SECURE_`. For information about setting the TCPSEC option, see “Providing Client Identification in a pre-Version 8 Session” on page 111.

Enforcing Server Userid and Password Encryption

As a security measure, you may set the AUTHENCR option to enforce the encryption of userids and passwords when they are passed from the client to the server. For information about the AUTHENCR option, see “SAS/SHARE Only Option” on page 114.

Specifying the TCP/IP Access Method

You must specify the TCP/IP communications access method at the server before a client can access it.

Use the following syntax to specify the TCP/IP 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. TCP (short for TCP/IP, which is an abbreviation for Transmission Control Protocol/Internet Protocol) is an example of an *access-method-id*.

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

Example:

```
options comamid=tcp;
```

The server will be available only to SAS/SHARE sessions that use the TCP/IP 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. See Table 1.3 on page 10 for the supported access methods by host.

All of the access methods initialize when the server initializes. The activation of multiple access methods makes a server available to several groups of clients, each using a different communications access method simultaneously.

COMAUX options can be specified only at a SAS invocation or in a SAS configuration file. The syntax for the COMAUX options follows:

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

An example of configuration file entries for a server that is running on an OS/390 host follows:

```
comamid=tcp
comaux1=appc
comaux2=xms
```

When the server starts, all of the communications access methods are initialized. The server is simultaneously available to client sessions that use the TCP/IP access method as well as to clients that use the APPC and XMS access methods.

Specifying a Server Name

You must specify the server name in the PROC SERVER statement. Use the following syntax:

```
SERVER=server
```

server can represent either a *server-id* or a *port* number.

- *server-id* corresponds to the service that was configured in the SERVICES file. See “Configuring the Server in the SERVICES File” on page 122 for more information.
- *port* is the location for passing data to and receiving data from the server. The port number is specified with two preceding underscore () characters. For example, you can specify the server port as 5000 using the SERVER= option in a LIBNAME statement:

```
libname mylib '.' server=__5000;
```

See *SAS Language Reference: Dictionary* for details about SAS naming rules. See *SAS/SHARE User's Guide* for details about the PROC SERVER statement.

Server Example

The following example illustrates the statements that you specify in the server configuration file on the OS/390 host:

```
tcpsec=_secure_
```

The `_SECURE_` value for TCPSEC requires clients to supply a userid and a password that are valid on the server.

The following example illustrates the statements that you specify in a SAS session on the OS/390 host at which you start a server:

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

The TCP/IP access method is declared and the server SHARE1 is started on the OS/390 host.

System Configuration for the TCP/IP Access Method

System Administrator, SAS Site Representative, Applications Programmer, or User
The TCP/IP access method for the OS/390 host uses routines in the SAS/C Transient Library for communications services. This library relies on an

underlying layer of TCP/IP communications software such as IBM TCP/IP Version 2, Release 2.1 or a subsequent release for OS/390.

You must perform the following tasks or have knowledge of the following issues to configure the TCP/IP access method for SAS/CONNECT and SAS/SHARE:

- Install the SAS/C Transient Library.
- Understand the function of network configuration files.
- Understand the search order for locating network configuration files.
- Understand the search order for locating host names and Internet addresses.
- Know about configuration file-naming conventions.
- Redirect error output with the SASCTCPE DDname.

Installing the SAS/C Transient Library

You should have already installed the SAS/C Transient Library. If you have not, the installation steps are provided in this section.

The TCP/IP access method requires the SAS/C Transient Library, which is included with both SAS/CONNECT and SAS/SHARE.

If your site is running Release 5.50 of the SAS/C Transient Library, either you must re-install it, thereby replacing it with a later version, or you must apply zap Z5500974 to your installed library. The zap is included on the install tape and is described in usage notes.

Steps for installing the SAS/C Transient Library follow:

- 1 With the install tape mounted, install the SAS/C Transient Library.

The CNTL data set member ULCTRANL contains JCL that is updated with job, *&prefix*, and allocation information that you supplied in the SASEDITP member. If you run this job as supplied, it unloads the SAS/C Transient Library Release 5.50 or a subsequent release from your install tape to a data set named *&prefix*.SASC.TRANSLIB.

- 2 If you want to change the library name (for compatibility with a particular SAS release), you can do so by setting library name DSN= in the SASCTRAN DD statement in the IEFBR14 step, and in the DDOT0 DD statement in the IEBCOPY step.
- 3 Submit the job and verify successful completion.
- 4 To allocate the SAS/C Transient Library for SAS execution, you may use either of the following options:

- Change your SAS CLIST to add a CTRANS allocation as follows:

```
ALLOC F(CTTRANS) DA('&prefix.SASC.TRANSLIB')
      SHR &SU11
```

- Change your SAS cataloged procedure to allocate the SAS/C Transient Library as follows:

```
// CTRANS DD DSN=&prefix.SASC.TRANSLIB,
          DISP=SHR
```

- 5 For SAS/SHARE only, change the started task JCL for your server.

Note: If you choose to install your SAS/C Transient Library to a link-list data set, you do not need to allocate it in your CLIST and in your cataloged procedure. Δ

Understanding the Function of Network Configuration Files

The operation of the TCP/IP access method depends on its ability to access network configuration information from particular site-specific files. The specific filenames are based on the particular vendor TCP/IP product.

IBM TCP/IP Network Configuration Files

SAS software uses the IBM TCPIP.DATA file or TCPPARMS file to determine resolver configuration and IBM defaults. SAS searches the following locations for configuration data sets in this order:

- 1 Data set identified by the DDname SYSTCPD
- 2 *tsoprefix*.TCPIP.DATA
- 3 SYS1.TCPPARMS (TCPDATA)
- 4 TCPIP.TCPIP.DATA

If your configuration file is not in one of the default locations, you may specify its location by setting the TCPIPPRF SAS option to the TCP/IP prefix of your IBM files. For example, if your TCPIP.DATA file is installed in SYS2.VER2.TCP.TCPIP.DATA, you should set the TCPIPPRF option to SYS2.VER2.TCP.

If you do not use IBM defaults for resolver configuration or you do not maintain a name server, see “Understanding the Search Order for Locating Network Configuration Files” on page 127 for information about the rules for searching for configuration files.

Note: The SAS/C Transient Library does not support the IBM TCP/IP files HOSTS.LOCAL and HOSTS.SITEINFO. The previous SAS TCP/IP access method used these files through the Pascal Transient Library Services (Release 6.07 and 6.08 TS404). \triangle

Setting SAS/C Environment Variables with the SASCTCPV DDname

You can set the SAS/C environment variables that are used in the configuration data set selection by including a data set with the special ddname SASCTCPV. If a data set with this ddname is allocated to the SAS session, then its contents are read during the TCP access method initialization. Each logical record is assumed to contain an environment variable assignment of the form:

```
environment_variable_name=value
```

For example, if all your configuration files existed under the TCPIP_PREFIX names, you could include the following in your SASCTCPV data set:

```
//SASCTCPV DD *
TCPIP_PREFIX=TCPCURR
/*
```

If any of the SAS options controlling configuration data sets are specified, they will override the values that are set in the SASCTCPV data set. For example, the SAS option TCPIPPRF will take precedence over a TCPIP_PREFIX assignment in the SASCTCPV data set.

Interlink TCP/IP Network Configuration Files

Interlink provides an LSCNCOM interface to the SAS/C Transient Library. You should place the LSCNCOM routine and its alias L\$CNCOM, in the link-list search order before the module that is supplied by SAS. The routine is dynamically loaded

when the first call to the SAS socket library function is made. SAS searches the following locations for configuration data sets in this order:

- 1 STEPLIB
- 2 LLA
- 3 LPA
- 4 link list.

SAS uses the Interlink name resolver and the ICSRSLV option to determine the order in which the Interlink name resolver (DNR) and the SAS/C resolver are used. For details about the LSCNCOM interface, see the Interlink documentation.

Understanding the Search Order for Locating Network Configuration Files

An explanation of the search logic that is used by the SAS/C Transient Library is contained in the following sections.

The file that contains configuration information usually has a name that is derived from the equivalent UNIX filename; for example, the OS/390 data set name ETC.HOSTS is derived from the UNIX filename `/etc/hosts`.

The SAS/C Transient Library uses the following search logic to locate the file that contains configuration information:

- 1 Under TSO, it searches for a data set name that is composed of the user userid prefix and the name that is derived from the UNIX filename.
- 2 It searches for a data set name that is derived from the UNIX filename, `hosts`.

Understanding the Search Order for Locating Host Names and Internet Addresses

The SAS/C Transient Library uses the following search logic to locate host names and Internet addresses:

- 1 RESOLV.CONF file using the rules given in “RESOLV.CONF File Search Order” on page 129. If the SAS/C Transient Library finds the RESOLV.CONF file, it performs the requested queries through the SAS/C resolver, and it returns any answer it receives. If attempts to connect to name servers are refused, it uses the logic in step 3.
- 2 Data set in the format of the IBM TCP/IP file TCPIP.DATA. The search rules for this data set are the same as the search rules used by IBM TCP/IP.
 - Data set identified by the DDname SYSTCPD. If found, the filename is passed to FOPEN.
 - `tso-prefix.TCPIP.DATA` under TSO.
 - SYS1.TCPPARMS (TCPDATA) data set.
 - TCPIP.TCPIP.DATA.
 - Default value of TCPIPPRF and searches for `default-value.TCPIP.DATA`.
- 3 If attempts to connect to name servers that are defined in TCPIP.DATA are refused, the library looks for a HOSTS file by using the rules given in “HOSTS File Search Order” on page 128. If the library finds a HOSTS file, it returns the result, including failure.

Determining the domain name in name-server queries follows the same logic as the UNIX operating system follows when it uses the domain statement of the

RESOLV.CONF file. Name-server addresses are also determined from the RESOLV.CONF file.

The SAS/C Transient Library recognizes only the first three name servers specified in this file. Both the UNIX operating system and the SAS/C environment have a limit of three name servers.

Knowing about the HOSTS File

The HOSTS file contains Internet address and host-name pairs with an optional alias. You use either the HOSTS file or the resolver program to locate host name and Internet address equivalents.

Use the following format to create a HOSTS file:

Internet-address host-name optional-alias

Example:

```
127.0.0.1      localhost      loopback
151.174.130.8 bigserver.yourcompany.com bigserver
151.174.130.27 copland.yourcompany.com copland
151.174.130.28 ives.yourcompany.com ives
151.174.130.29 byrd.yourcompany.com byrd
```

If you have access to a UNIX system, see the **hosts(4)** manual page for more information.

If you have access to a UNIX system, you are advised to copy the **hosts** file from the UNIX system to the OS/390 system and customize it as necessary. Doing this prevents typographical errors that may occur if you create a new file.

HOSTS File Search Order

The SAS/C Transient Library searches for the HOSTS file, in the following order:

- value of ETC_HOSTS environment variable, if defined in the SASCTCPV file
- tso-prefix*.ETC.HOSTS under TSO or *userid*.ETC.HOSTS under batch execution
- ETC.HOSTS
- TCPIP.ETC.HOSTS
- tcpip-prefix*.ETC.HOSTS, if TCPIPREF is set to a value.

Knowing about the SERVICES File

The SERVICES file contains a service name, port number, protocol group with an optional alias. For SAS/SHARE, you must specify server names and port assignments in the SERVICES file. For SAS/CONNECT, the default TELNET port number is used.

If you have access to a UNIX system, see the **services(4)** manual page for more information.

If you have access to a UNIX system, you are advised to copy the **services** file from the UNIX system to the OS/390 system and customize it as necessary. Doing this prevents typographical errors that may occur if you create a new file.

See “Configuring the Server in the SERVICES File” on page 122 for details about the SERVICES file.

SERVICES File Search Order

The SAS/C Transient Library searches for the SERVICES file, in the following order:

- value of ETC_SERVICES environment variable, if defined in the SASCTCPV file

- tso-prefix*.ETC.SERVICES under TSO or *userid*.ETC.SERVICES under batch execution
- ETC.SERVICES
- TCPIP.ETC.SERVICES
- tcip-prefix*.ETC.SERVICES, if TCPIPREF is set to a value.

Knowing about the RESOLV.CONF File

The RESOLV.CONF file contains Internet address and corresponding host-name pairs within the domain. This file is used by the resolver program to find addresses of name servers that are used to locate host-name and Internet address equivalents.

You will use either the HOSTS file or the resolver program to locate host name and Internet address equivalents.

Use the following format to create a RESOLV.CONF file:

```
host-name Internet-address
```

Example:

```
domain      yourcompany.com
nameserver  151.174.166.251
nameserver  151.174.52.151
nameserver  151.174.52.113
```

If you have access to a UNIX system, see the **resolver(4)** manual page for more information.

If you have access to a UNIX system, you are advised to copy the RESOLV.CONF file from the UNIX system to the OS/390 system, and customize it as necessary. Doing so prevents typographical errors that may occur if you create a new file.

RESOLV.CONF File Search Order

Note: If you are using the IBM TCP/IP, you do not need to create this file. Δ

The SAS/C Transient Library searches for the OS/390 configuration file that is equivalent to the UNIX RESOLV.CONF file, in the following order:

- value of ETC_RESOLV_CONF environment variable, if defined in the SASCTCPV file
- tso-prefix*.ETC.RESOLV.CONF under TSO or *userid*.ETC.RESOLV.CONF under batch execution
- ETC.RESOLV.CONF
- TCPIP.ETC.RESOLV.CONF
- tcip-prefix*.ETC.RESOLV.CONF, if TCPIPREF is set to a value.

Knowing about Configuration File Naming Conventions

SAS searches for configuration files by the common default prefix TCPIP. It is stored in a 26-character string array in L\$CNDBA in the SAS/C Transient Library. The SERVICES file, for example, is named TCPIP.ETC.SERVICES. You may override this default using one of the following methods:

- Designate the default high-level qualifier ETC.
- Designate the default prefix TCPIP.

- Substitute an alternate prefix for the default TCPIP prefix by using a zap.
- Set the TCPIPPRF option.

All but one of these methods sets a prefix in the system. However, setting the TCPIPPRF option, affects only the invoking user's current session.

Note: Most sites already have these files available for existing TCP/IP applications. If your site already has TCP/IP applications running under the SAS/C library, this step was probably completed when the SAS/C Transient Library was installed. Δ

Using the ETC High-Level Qualifier

The ETC high-level qualifier prepends a configuration file name with the ETC prefix; for example, ETC.HOSTS and ETC.SERVICES.

Note: When using the ETC prefix, do not set the TCPIPPRF option to override the ETC high-level qualifier and do not apply the zap to change the default value of the TCPIP prefix in the SAS/C Transient Library. See "Applying a Zap to Change the Default TCPIP Prefix" on page 130 for more information. Δ

Using the TCPIP Default Prefix

If you cannot use ETC at your site and if DS names do not conflict with the ETC high-level qualifier, you may use the default prefix to produce names such as TCPIP.ETC.HOSTS.

Note: When using the TCPIP prefix, do not set the TCPIPPRF option to override the TCPIP prefix, and do not apply the zap to change the default value of the TCPIP prefix in the SAS/C Transient Library. See the next section for details about the default TCPIP prefix. Δ

Applying a Zap to Change the Default TCPIP Prefix

To change the default TCPIP prefix, you may apply zap number Z5504151, which is included on the SAS software install tape and is documented in usage notes.

For example, to produce this SERVICES file

```
SYS.PROD.CONFIG.ETC.SERVICES
```

specify a TCPIP prefix of SYS.PROD.CONFIG by using the zap.

Setting the TCPIPPRF Option to the TCPIP Prefix Value

The TCPIPPRF option initializes a filename prefix for the current SAS session. Each session that uses the TCP/IP access method (SAS/CONNECT local and remote hosts and SAS/SHARE server and clients) must set this option at initialization time. For example, to produce this configuration file

```
SYS2.VER2.TCP.ETC.HOSTS
```

set the TCPIPPRF option to SYS2.VER2.TCP as follows:

```
sas ... options ('TCPIPPRF=SYS2.VER2.TCP')
```

Redirecting Error Output with the SASCTCPE DDname

To document the operating behavior of the SAS/C Transient Library, the SAS/C libraries write all messages (error and informational) to the SAS log, the terminal

screen, or the SASCTCPE DDname to SYSOUT. The DDname of SASCTCPE is reserved to contain the redirection of library error messages.

To prevent the writing of informational and error messages to your terminal screen, you can direct SASCTCPE to DUMMY.

If you have trouble making a TCP/IP connection, you may want to direct SASCTCPE to a SYSOUT data set or to a permanent file in the local or the remote SAS execution as a troubleshooting aid.

You have completed the TCP/IP configuration process.

Reference

For complete details about the SAS/C Socket Library for TCP/IP, see *SAS/C Library Reference, Third Edition, Volume 2, Release 6.00*.

The TCP/IP access method for OS/390 is implemented with SAS/C Socket Library functions.

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.