



CHAPTER

5

Macintosh: TCP/IP Access Method

<i>SAS Support for TCP/IP on Macintosh</i>	69
<i>Tasks That Are Common to SAS/CONNECT and SAS/SHARE</i>	70
<i>System and Software Requirements for SAS/CONNECT and SAS/SHARE</i>	70
<i>Setting SAS Options</i>	70
<i>SAS/CONNECT TCPSEC Option</i>	71
<i>Configuring Services in the CONFIG.SAS6xx File</i>	71
<i>SAS/CONNECT</i>	72
<i>Local Host Tasks</i>	72
<i>Remote Host Sign-on Considerations</i>	72
<i>Configuring the UNIX Spawner Service in the CONFIG.SAS File</i>	73
<i>Setting the TCPSEC Variable</i>	73
<i>Specifying the TCP/IP Access Method</i>	73
<i>Specifying the Remote Node Name</i>	73
<i>Identifying a Script File for Signing On and Signing Off</i>	74
<i>Signing On to the Remote Host</i>	75
<i>Local Host</i>	75
<i>Remote Host</i>	75
<i>SAS/SHARE</i>	76
<i>Client Tasks</i>	76
<i>Configuring the Server in the CONFIG.SAS File</i>	76
<i>Specifying the TCP/IP Communications Access Method</i>	76
<i>Specifying a Server Name</i>	76
<i>Client Example</i>	77
<i>Server Tasks</i>	77
<i>Specifying the TCP/IP Communications Access Method</i>	78
<i>Specifying a Server Name</i>	78
<i>Configuring the Server in the CONFIG.SAS File</i>	79
<i>Specifying a Server Name</i>	79
<i>Server Example</i>	79

SAS Support for TCP/IP on Macintosh

Note: Beginning with Version 7, the Macintosh platform is not supported. Information about Macintosh is included here for Version 6 users. △

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

System Administrator and User

To use the TCP/IP access method with a Macintosh 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 you know how to set options in SAS software.
- 3 Set the TCPSEC option.
- 4 Configure services for the TCP/IP access method.

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

Ensure that the following conditions have been met:

- 1 The Macintosh TCP/IP package is installed on the local host.
- 2 SAS is installed on both the local host and the remote host.
- 3 The remote host is attached to a TCP/IP network.

Using the TCP/IP access method, you can connect to any supported platform that is on the TCP/IP network and is running a SAS release that has the corresponding access method support with SAS/CONNECT and SAS/SHARE properly licensed.

Setting SAS Options

You may need to set specific options in SAS to establish the connections that you want with SAS/CONNECT and SAS/SHARE when using the TCP/IP communications access method. Ask your network administrator for advice about these settings.

You may specify an option in any of the following ways:

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

OPTIONS *variable-name=value*;

Example:

options tcpsec=_prompt_;

- in a SAS configuration file:

-SET *variable-name value*

Example:

-set tcpsec _prompt_

- in a SAS macro variable:

%LET *variable-name=value*;

Example:

%let tcpsec=_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, this is the order of precedence that is followed:

- SAS macro variable
- OPTIONS statement

AUTOEXEC file
SAS configuration file.

SAS/CONNECT TCPSEC Option

For SAS/CONNECT, to sign on to a UNIX, an OS/2, a Windows NT, or a Windows 95 remote host through a spawner program instead of a script file, you must set the TCPSEC option. Setting the TCPSEC option allows local hosts whose userids and passwords have been verified to access the desired remote host. See Chapter 32, “Spawner Programs,” on page 457 for more information.

Values for TCPSEC that you may set at a local host and a remote host are

TCPSEC= *_PROMPT_* | *userid<.password>*

PROMPT

must be set at the SAS/CONNECT local host.

PROMPT causes SAS to 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 the password to the system option.

userid<.password>

must be set at the SAS/CONNECT local host.

This value specifies both the userid and password. Assigning the userid and the password to the TCPSEC option in a file may inadvertently publicize this information and compromise the security of the spawner program that is used with SAS/CONNECT. Assigning the value to the option in a file allows anyone to read it.

Optionally, you may assign *userid* to TCPSEC and prompt only for *password*.

Examples:

```
%let tcpsec=_prompt_;
%let tcpsec=bass.time2go;
```

Configuring Services in the CONFIG.SAS6xx File

You can configure TCP/IP services by editing the CONFIG.SAS. *version.release* file, where *version.release* reflects the current version and release of SAS that you are using; for example, CONFIG.SAS612.

Entries in the CONFIG.SAS file are specified as follows:

```
-set gsbn_ service-name port-number
```

where GSBN is a command that specifies the service name and port number.

service-name may have either of two values: *protocol* or *server-id*.

For SAS/CONNECT, enter either of the following in the CONFIG.SAS file:

- the TELNET protocol and the associated port number.

Example:

```
-set gsbn_telnet 23
```

- the UNIX spawner service and the associated port number.

Example:

```
-set gsbn_unxspawn 5020
```

For SAS/SHARE, enter the following in the CONFIG.SAS file:

- the server's identifier and the associated port number.

Example:

```
-set gsbn_sasshare 5018
```

- if the server is to be accessed by Java clients, then you will also need to set:

```
-set gsbn_5018 sasshare
```

The person responsible for the SAS applications at your site can provide you with the names of the services that you can use. Before you configure a service, you can find out whether it has already been configured by issuing the following:

```
%PUT %SYSGET(gsbn_service);
```

Examples:

```
%put %sysget(gsbn_telnet);
23
```

```
%put %sysget(gsbn_mktserve);
5015
```

In these examples, the port numbers are reported for the TELNET protocol and the MKTSERV server, respectively. If the service has not been configured, an error message is returned.

SAS/CONNECT

Local Host Tasks

User and Applications Programmer

To connect a Macintosh 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 Configure the UNIX spawner service in the CONFIG.SAS file, as necessary.
- 3 Optionally, set the TCPSEC option, 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 Sign-on Considerations

If you are connecting to a Windows 95 remote host, you *must* connect by means of a spawner program that is already running on the remote host. If you are connecting to a Windows NT, an OS/2 or a UNIX remote host, you optionally, may connect by means of a spawner program that 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, you must set the TCPSEC option at the local host. The TCPSEC option stores the userid and the password of the remote host that, when passed to the remote host, allows a local host connection.

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

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

Configuring the UNIX Spawner Service in the CONFIG.SAS File

Before connecting to a UNIX remote host with the UNIX spawner program, configure the spawner service in the CONFIG.SAS file on the local host. See “Configuring Services in the CONFIG.SAS6xx File” on page 71 for more information.

Setting the TCPSEC Variable

If you are not using a script file to sign on to the remote host, you must set the TCPSEC option at the local host. See “SAS/CONNECT TCPSEC Option” on page 71 information about setting TCPSEC.

Specifying the TCP/IP Access Method

Note: TCP/IP is the default communications access method on the CMS platforms. You may omit specifying the access method in a COMAMID option and the TCP/IP access method is assumed, by default. △

If you 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 an *access-method-id*.

Example:

```
options comamid=tcp;
```

Alternatively, you may set this option in the SAS configuration file.

Specifying the Remote Node Name

To make a connection from a Macintosh local host to a remote host, use the following syntax:

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

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

- If you are connecting to a Windows NT, a Windows 95 or an OS/2 remote host that is running the PC spawner program, use the name of the node that the PC

spawner is running on. See Chapter 35, “PC Spawner Program,” on page 471 for more information.

- If you are connecting to a UNIX host that is running the UNIX spawner program, use a two-level name in the form of *node-name.service-name*, where *node-name* specifies the node on which the UNIX spawner program is running and *service-name* specifies the port on which the UNIX spawner is listening for a connection request.

See “Starting the UNIX Spawner Program” on page 479 for more information about the UNIX spawner program. See “Configuring the SERVICES File” on page 485 for information about configuring the UNIX spawner service in the SERVICES file.

- If you are connecting to any other remote host platform that uses 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 name server.

The value of the REMOTE= option must be a valid SAS name. See *SAS Language: Reference, Version 6, First Edition* 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 as the value of the REMOTE= option.

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 the following example, MYNODE).

```
%let mynode=172.20.10.200;
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, depending on the remote host that you are connecting to. The fileref format follows:

```
filename rlink '!sasroot:misc: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 5.1 Macintosh TCP/IP SAS/CONNECT Sign-on Scripts

Remote Host		Script Name
CMS		TCPCMS.SCR
CMS (using full-screen 3270	TELNET protocol)	TCPCMS32.SCR
TSO under OS/390		TCPTSO.SCR

Remote Host	Script Name
OS/390 (without TSO)	TCPMVS.SCR
OS/390 (using full-screen 3270 TELNET protocol)	TCPTSO32.SCR
OpenVMS	TCPVMS.SCR
OS/2	TCPOS2.SCR
UNIX	TCPUNIX.SCR
Windows NT, Windows 95	TCPWIN.SCR

You must customize the sample scripts to accurately reflect your site's logon process; failure to do so will produce errors. (Your SAS Software Consultant may have altered the sample sign-on scripts as needed.) If someone has altered an automatic logon sample script for users at your installation, you need to load a copy of the modified file onto your local host.

Signing On to the Remote Host

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

```
signon;
```

Local Host

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

```
filename rlink '!sasroot:misc:tcpos2.scr';
options comamid=tcp remote=rmtnode;
signon;
```

The first line identifies the script file that you use to sign on to a remote host. The script includes a prompt for a userid and a password that are valid on the remote host. The TCP/IP access method is declared with a connection to the remote host RMTNODE. The SIGNON statement performs the sign-on process.

Remote Host

SAS Institute does not provide support for connections to the Macintosh remote host with the TCP/IP access method.

SAS/SHARE

Client Tasks

User or Applications Programmer

To prepare for creating a SAS/SHARE server and accessing it, perform the following tasks:

- 1 Configure the server in the CONFIG.SAS file.
- 2 Specify the TCP/IP access method.
- 3 Specify a server name.

Configuring the Server in the CONFIG.SAS File

Each SAS/SHARE server must be defined as a service in the CONFIG.SAS file on each remote host node on which a server runs and on each node from which a user session will access the server. See “Configuring Services in the CONFIG.SAS6xx File” on page 71 for more information.

Specifying the TCP/IP Communications Access Method

You must specify the TCP/IP communications access method at the connecting client before you access 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 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*.

Because the Macintosh platform supports only one communications access method, you can use only the COMAMID option. See Table 1.3 on page 10 for the supported access methods by host.

Example:

```
options comamid=tcp;
```

The server is accessed using the TCP/IP access method.

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

An example of a configuration file entry for a Macintosh client connecting to a UNIX server follows:

```
-comamid tcp
```

Specifying a Server Name

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


```
SERVER=node.server-id
```

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

node must be a valid TCP/IP node name. *server-id* identifies the service that has been configured in the CONFIG.SAS file. See “Configuring Services in the CONFIG.SAS6xx File” on page 71 for more information.

If the TCP/IP node name is not a valid SAS name, you can 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:

- SAS macro variable
- valid node name.

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

```
%let srvnode=mktserve.acme.com;
libname sales 'machd:sasdata' server=srvnode.server1;
```

Note: Do not use an ampersand (&) in a two-level server name. An ampersand causes 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. △

See *SAS Language: Reference, Version 6, First Edition* for details about SAS naming rules. See *SAS/SHARE Software: Usage and Reference, Version 6, First Edition* for details about the PROC OPERATE and LIBNAME statements.

Client Example

The following example illustrates the statement that you specify in a Macintosh client CONFIG.SAS612 file to configure the server service.

```
-set gsbn_server1 5015
```

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

```
options comamid=tcp;
libname sasdata 'edc.prog2.sasdata'
      server=rmthost.share1;
```

The first line declares the TCP/IP access method. The LIBNAME statement specifies the name of the data library that is accessed through the server, which is identified by the two-level name RMTHOST.SHARE1.

Server Tasks

User or Applications Programmer

To prepare for starting a SAS/SHARE server, perform the following tasks:

- 1 Specify the TCP/IP access method.
- 2 Specify a server name.

Specifying the TCP/IP Communications Access Method

You must specify the TCP/IP communications access method at the server before you start a server. Use the following syntax to do this:

```
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*.

Because the Macintosh platform supports only one access method, you can use only the COMAMID option. See Table 1.3 on page 10 for the supported access methods by host.

Example:

```
options comamid=tcp;
```

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

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

An example of a configuration file entry for a server that is running on a Macintosh host follows:

```
-comamid tcp
```

Specifying a Server Name

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

```
SERVER=node.server-id
```

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-id* identifies the service that has been configured in the CONFIG.SAS file. See “Configuring Services in the CONFIG.SAS6xx File” on page 71 for more information.

If the TCP/IP node name is not a valid SAS name, you can 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:

- ☐ valid 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 'machd:sasdata' server=srvnode.server1;
```

Note: Do not use an ampersand (&) in a two-level server name. An ampersand causes 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, Version 6, First Edition* for details about SAS naming rules. See *SAS/SHARE Software: Usage and Reference, Version 6, First Edition* for details about the PROC SERVER statement.

Configuring the Server in the CONFIG.SAS File

Server Administrator

To prepare for starting a SAS/SHARE server, perform the following task:

- 1 Configure servers in the CONFIG.SAS file.

Each SAS/SHARE server must be defined as a service in the CONFIG.SAS file on each remote host node on which a server runs, and on each node from which a user session will access the server. See “Configuring Services in the CONFIG.SAS6xx File” on page 71 for more information.

Specifying a Server Name

If the client and server sessions are running on different network nodes, you must include the node name in the server identifier in the PROC SERVER statement as follows:

```
SERVER=node.server-id
```

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

node must be a valid TCP/IP node name. *server-id* identifies the service that has been configured in the CONFIG.SAS file. See “Configuring Services in the CONFIG.SAS6xx File” on page 71 for more information.

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

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

- ☐ SAS macro variable
- ☐ valid node name.

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

```
%let srvnode=mktserve.acme.com;
libname sales 'machd:sasdata' server=srvnode.server1;
```

Note: Do not use an ampersand (&) in a two-level server name. An ampersand causes 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. △

See *SAS Language: Reference, Version 6, First Edition* for details about SAS naming rules. See *SAS/SHARE Software: Usage and Reference, Version 6, First Edition* for details about the PROC SERVER statement and about starting and accessing a server.

Server Example

The following example illustrates the statement that you specify in a Macintosh server CONFIG.SAS612 file to configure the server:

```
-set gsbm_server1 5015
```

The following example illustrates the statements that you specify in a SAS session on the Macintosh 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 Macintosh host.

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.