CHAPTER

31

DOMAIN Server

Support for Version 6 Only

The protocol gateway service provided by the DOMAIN server is not supported in Version 7 and later releases. However, information about this service is included here for Version 6 users.

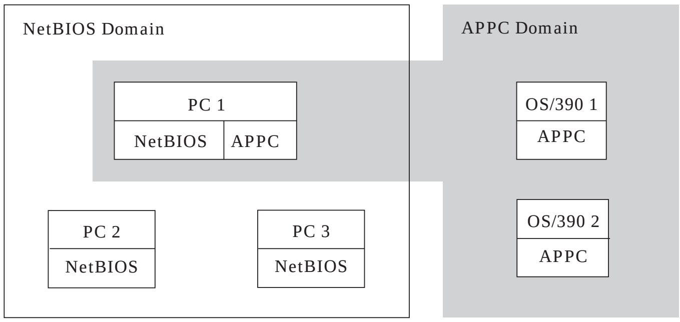
Using a DOMAIN Server for Cross-Access Method Communication

The DOMAIN server is a protocol gateway service that connects two SAS sessions that do not use a common communications access method in a networked environment. A network comprises one or more logical domains, each of which maps a topological area according to the communications protocol or access method that it supports.

A SAS session can communicate with any other SAS session that runs in the same logical domain because both sessions use the same access method. Furthermore, SAS sessions from different logical domains can communicate by means of the DOMAIN server if at least one domain supports multiple access methods, one of which must be used in both domains. The DOMAIN server, therefore, eliminates the need to configure multiple protocols in a given domain.

Consider a network that supports NetBIOS and APPC domains, as illustrated in Figure 31.1 on page 444.

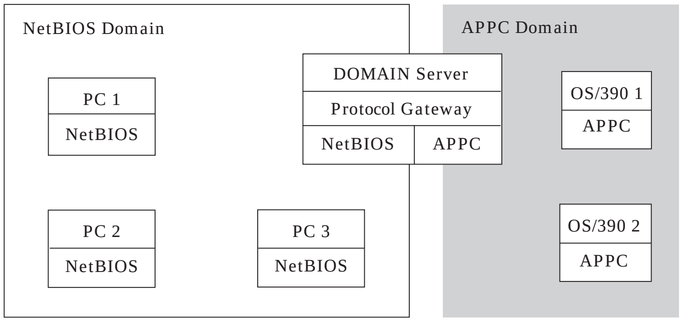
Figure 31.1 Example of a Multiple Protocol Network



Within the NetBIOS domain, PC 1, PC 2, and PC 3 can intercommunicate using the NetBIOS access method. Within the APPC domain, PC 1, OS/390 1, and OS/390 2 can intercommunicate using the APPC access method. However, PC 2 and PC 3 cannot intercommunicate with OS/390 1 or OS/390 2 because they do not share a common access method.

The DOMAIN server provides the gateway that allows communication between SAS sessions that are running in different logical domains. Incorporating the DOMAIN server has the result that is illustrated in Figure 31.2 on page 444.

Figure 31.2 Example of a Multiple Protocol Network with DOMAIN Server



All sessions in both domains can intercommunicate by using the DOMAIN server's set of common access methods. A message flows over the originator's native access method to the DOMAIN server. This access method is called the inbound access method. The server redirects a message to the cross-domain destination using the destination's native access method. Conversely, the outbound access method is the access method that is used on the target host side of the protocol gateway.

Supported Access Methods

The OS/2, Windows NT, Windows 95, Windows 32s, Macintosh, and UNIX (AIX, HP-UX, and Solaris 2) platforms support the following access methods:

Inbound Access Methods
APPC, DECnet, NetBIOS, SPX, TCP/IP

Outbound Access Methods
APPC, TCP/IP

DOMAIN Server System Requirements

The DOMAIN server requires a dedicated SAS session running in an OS/2 or in a Windows NT environment.

Configuring and Using the DOMAIN Server

You will perform tasks at both the DOMAIN server and the SAS/CONNECT local host or the SAS/SHARE client when you connect to a SAS/CONNECT remote host or a SAS/SHARE server.

Tasks to Perform at the DOMAIN Server

To connect two SAS sessions using the protocol gateway of the DOMAIN server, perform the following tasks at the DOMAIN Server:

1. If you use the TCP/IP inbound access method, configure the DOMAIN server in the SERVICES file.
2. Initialize the DOMAIN server.
3. Specify the inbound and outbound communications access methods.
4. Put outbound sign-on scripts for SAS/CONNECT on the DOMAIN server, if necessary.

Configuring the DOMAIN Server in the SERVICES File

Verify that the DOMAIN server identifier is specified in the SERVICES file at the DOMAIN server.

Example:

```
domsrv 5010/tcp #DOMAIN server
```

See "Configuring the SERVICES File" on page 485 for details about customizing the SERVICES file.

Starting the DOMAIN Server

You must start the DOMAIN server before a SAS/CONNECT local host or a SAS/SHARE client can use it.

To start the DOMAIN server, use the following syntax:

```
PROC DOMAIN PROTOCOL SERVERID=domain-server-id;
```

where PROC DOMAIN is a procedure to create a DOMAIN service. PROTOCOL is always specified for the DOMAIN gateway service. SERVERID=*domain-server-id* specifies the access method network name.

Example:

```
proc domain protocol serverid=domsrv;
```

This statement declares DOMSRV as the DOMAIN server.

Specifying the Outbound Communications Access Method at the DOMAIN Server

At the DOMAIN server, specify the communications access method that will redirect the inbound message that was received from the SAS/CONNECT local host or the SAS/SHARE client. 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 that is used by the remote host to communicate with the SAS/CONNECT local host or the SAS/SHARE client by means of the domain server.

You may specify the COMAMID option in an OPTIONS statement, at a SAS invocation, or in a SAS configuration file. An example of an OPTIONS statement follows:

```
options comamid=tcp;
```

The DOMAIN server attempts to locate the destination partner with the TCP/IP access method.

You must specify at least one auxiliary method for the DOMAIN server. Specifying a second auxiliary method is optional. Auxiliary methods are specified by using the COMAUX options.

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

COMAUX options can be specified only at a SAS invocation or in a SAS configuration file. An example of specifying COMAMID, COMAUX1, and COMAUX2 options in a SAS configuration file follows:

```
-comamid tcp
-comaux1 appc
-comaux2 netbios
```

To minimize the number of failure-retry operations required to establish a connection to the remote host, specify the most heavily used outbound access methods, in descending order.

For example, if NetBIOS is specified as the inbound access method at the local host, the NetBIOS domain is searched first for the named service. If the service is not found, the search is redirected to the DOMAIN server, which attempts to locate the named service using the access methods specified in the COMAMID and COMAUX options; in this example, TCP/IP, APPC, and NetBIOS.

Storing SAS/CONNECT Outbound Sign-on Scripts on the DOMAIN Server

The DOMAIN server provides script support for outbound access methods that require a script file to establish a SAS/CONNECT connection to a remote host.

You may store scripts in either of the following locations:

!SASROOT\DOMAIN\SCRIPT
Store default scripts here. Name the default script DEFAULT.SCR and any additional scripts REMOTE.SCR, where REMOTE is the value specified in the REMOTE= option on the SAS/CONNECT local host. See "Specifying the Remote Node Name" on page 451 for information about specifying the REMOTE= option.

!SASROOT\DOMAIN\SCRIPT\user
Store customized scripts here. The DOMAIN server identifies the user by the userid and the password that are specified in the security macro variable (whose name varies by access method). The server then searches the user tree for the DEFAULT.SCR and REMOTE.SCR scripts, in that order.

For example, suppose the DOMAIN server node contained the following scripts:

```
!SASROOT\DOMAIN\SCRIPT\DEFAULT.SCR
                       SDCMVS.SCR
```

and

```
!SASROOT\DOMAIN\SCRIPT\user\DEFAULT.SCR
                            SDCMVS.SCR
```

If the TCPSEC macro variable is set to *userid.password*, one of the script files in *!SASROOT*\DOMAIN\SCRIPT\user is used. Otherwise, the default script is used.

Prompt windows cannot be passed from the DOMAIN server to the local system. Therefore, INPUT and NOTIFY statements should not be used in the script files. The userid and password values normally obtained by an INPUT statement are managed through the two global macro variables SYSGWID and SYSGWPW. These macro variables are initialized with the value that is obtained from the appropriate security variable (the TCPSEC or VQSECURE macro variable or the APPC_SECURE environment variable). The global macro variables are referenced in the script file in TYPE statements as follows:

```
waitfor 'login';
type "&SYSGWID" LF;
waitfor 'Password';
type "&SYSGWPW" LF;
```

See "Setting SAS Macro Variables" on page 447 for details about the SYSGWID and SYSGWPW macro variables.

Tasks to Perform at the SAS/CONNECT Local Host and at the SAS/SHARE Client

Perform these tasks at the SAS/CONNECT local host and at the SAS/SHARE client:

1. Set the necessary macro variables.
2. Specify the inbound communications access method.
3. Set up remote host or server security, if applicable.

Setting SAS Macro Variables

You must set macro variables at the SAS/CONNECT local host or at the SAS/SHARE client to identify the DOMAIN server node name and server identifier and to provide

DOMAIN server security for connecting local hosts or clients. The macro variables that are set depend on the inbound access method that is used.

Macro variables are provided for the following inbound access methods:

APPC

NetBIOS

SPX

TCP/IP.

TCP/IP Access Method Macro Variables

The macro variables for the TCP/IP access method are

gateway-host

specifies the fully qualified TCP/IP node name if it has not already been defined in the local system's HOSTS file. If the node name of the DOMAIN server has already been defined, you do not have to define a macro variable. Set the *gateway-host* macro variable by using the following syntax:

```
%LET GWHOST=domain-server-nodename;
```

Example:

```
%let gwhost=star.abc.xyz;
```

Macro variable GWHOST stores the fully qualified TCP/IP node name of the DOMAIN server STAR.ABC.XYZ.

TCP-gateway

specifies the DOMAIN server's server identifier that you use to initialize the DOMAIN server in the PROC DOMAIN statement. You must also specify the same DOMAIN server identifier in the SERVICES file. See "Configuring the SERVICES File" on page 485 for details about customizing the SERVICES file. Set the macro variable using the following syntax:

```
%LET TCPGW=GWHOST.domain-server-id;
```

Example:

```
%let tcpgw=gwhost.domsrv;
```

DOMSRV is the server identifier of the DOMAIN server. If the DOMAIN server node has been defined as STAR in the system's HOSTS file, you omit the GWHOST macro variable and assign the macro variable as follows:

```
%let tcpgw=star.domsrv;
```

TCPSEC

is required by the target host. The DOMAIN server supplies the specified value to the outbound access method.

%LET TCPSEC=_NONE | _PROMPT_|
    userid.password;

Select the appropriate value for TCPSEC according to the measure of security that you want. For no security, either do not set the TCPSEC variable or set it to _NONE_. The default is _NONE_

If you set the TCPSEC variable to _PROMPT_, you will be prompted for a userid and a password.

Example:

```
%let tcpsec=_prompt_;
```

If you choose to assign the userid and the password directly to the TCPSEC variable at the SAS/CONNECT local host or at the SAS/SHARE client, you inadvertently may publicize this information and compromise the security of the DOMAIN server. Hard-coding the value to the variable in a file allows anyone to read it.

Example:

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

In this example, BASS is the userid and TIME2GO is the password on the server or on the remote host.

NetBIOS and SPX Access Method Macro Variables

The macro variables for the NetBIOS and SPX access methods are

VQPROTGW

specifies the access method's network name, which you use to initialize the DOMAIN server in the PROC DOMAIN statement. Set the VQPROTGW macro variable using the following syntax:

%LET VQPROTGW=*domain-server-id*;

Example:

```
%let vqprotgw=domsvr;
```

DOMSVR is the access method's network name.

VQSECURE

is sometimes required by the target host (if the server is secure). The DOMAIN server supplies the specified value to the outbound access method.

%LET VQSECURE=_NONE_| _PROMPT_| *userid.password*;

Select the appropriate value for VQSECURE according to the measure of security that you want. The default is no security. For no security, either do not set the VQSECURE variable or set it to _NONE_.

If you set the VQSECURE variable to _PROMPT_, you will be prompted for a userid and a password.

Example:

```
%let vqsecure=_prompt_;
```

If you choose to assign the userid and the password directly to the VQSECURE variable at the SAS/CONNECT local host or at the SAS/SHARE client, you inadvertently may publicize this information and compromise the security of the remote host. Hard-coding the value to the variable in a file allows anyone to read it.

Example:

```
%let vqsecure=bass.time2go;
```

In this example, BASS is the userid and TIME2GO is the password on the server or on the remote host.

APPC Access Method Variables

The variables for the APPC access method are

APPC_GATEWAY
: specifies the access method's network name, which you use to initialize the DOMAIN server in the PROC DOMAIN statement. Set the APPC_GATEWAY variable using the following syntax:

```
SET=APPC_GATEWAY domain-server-id;
```

Example:

```
set=appc_gateway domsvr;
```

DOMSVR is the access method's network name.

APPC_SECURE
: is required by the target host. The DOMAIN server supplies the specified value to the outbound access method.

SET=APPC_SECURE _NONE_| _PROMPT_| *userid.password*;

Select the appropriate value for APPC_SECURE according to the measure of security that you want. For no security, either do not set the APPC_SECURE variable or set it to _NONE_. The default is _NONE_.

If you set the APPC_SECURE variable to _PROMPT_, you will be prompted for the userid and the password.

Example:

```
set=appc_secure _prompt_;
```

If you choose to assign the userid and the password directly to the APPC_SECURE variable at the SAS/CONNECT local host or at the SAS/SHARE client, you inadvertently may publicize this information and compromise the security of the remote host. Hard-coding the value to the variable in a file allows anyone to read it.

Example:

```
set=appc_secure bass.time2go;
```

In this example, BASS is the userid and TIME2GO is the password on the server or on the remote host.

Specifying the Inbound Communications Access Method

You must specify the communications access method to be used by the local host. Use the following syntax:

```
OPTIONS COMAMID=access-method-id;
```

where COMAMID is an acronym for Communications Access Method Identification and *access-method-id* identifies the communications access method that is used.

For example, if you configure a NetBIOS inbound access method at the local host, then the DOMAIN server may redirect the connection request through the TCP/IP or APPC outbound communications access methods, which are active in the same logical domain.

```
options comamid=netbios;
```

The DOMAIN server will switch an inbound NetBIOS request for connection to a remote host to some other outbound access method that is contained in the same logical domain.

You may specify this OPTIONS statement at a SAS invocation or in a SAS configuration file.

Setting Up Security

You may use the TCPSEC, VQSECURE, or APPC_SECURE variable, if applicable, to specify the userid and the password that are required by the target host (see "Setting SAS Macro Variables" on page 447). The DOMAIN server supplies the specified value to the outbound access method, which negotiates security with the target host.

If you are using SAS/CONNECT and your outbound access method is TCP/IP, the DOMAIN server attempts to provide a userid and a password through a script. If your SAS/CONNECT script contains a hard-coded userid and password, the DOMAIN server does not alter these values.

Otherwise, security is handled by the DOMAIN server.

Tasks to Perform at the SAS/CONNECT Local Host Only

Perform these tasks at the SAS/CONNECT local host only:

1. Specify the remote node to connect to.
2. Put sign-on scripts, as necessary, on the DOMAIN server.
3. Sign on to the remote host.

Specifying the Remote Node Name

If using SAS/CONNECT to make a connection from a local host to a remote host, use the following syntax:

```
OPTIONS REMOTE=remote-session-id;
```

where *remote-session-id* specifies the remote host that you are connecting to.

Note: The remote host is not the host that is running the DOMAIN server. △

The value that you specify will depend on the platforms that are to be connected. See the appropriate part and chapter in this book for the connection that you want. For example, to connect a Windows NT local host to an OS/390 remote host through an inbound TCP/IP access method, specify the OS/390 node name. See Table 1.1 on page 8 and Table 1.2 on page 9for information about access methods between all remote host and local host pairs.

Example:

```
options remote=rmtnode;
```

You may specify this OPTIONS statement at SAS invocation or in a SAS configuration file.

Putting Sign-on Scripts on the DOMAIN Server

Supplying a sign-on script depends on the requirement of the remote host that you are connecting to. If you use sign-on scripts, put them in the following location on the DOMAIN server:

```
!sasroot\domain\script
```

See "Storing SAS/CONNECT Outbound Sign-on Scripts on the DOMAIN Server" on page 446 for details about locating scripts at the DOMAIN server.

Signing On to the Remote Host

To complete your sign on to the remote host, enter the SIGNON statement. An example follows:

```
options set=tcpsec _prompt_;
options comamid=tcp  remote=rmtnode;
signon;
```

To allow a connection to the remote host, the TCPSEC variable specifies that the remote host will prompt for a username and a password. The TCP/IP access method is declared with a remote connection to a remote host identified as RMTNODE, which is the *remote-session-id*. The remote session identifier that you use is determined by the type of remote host that you are connecting to.

Note: The remote host is not the host that is running the DOMAIN server. △

The SIGNON statement performs the sign-on process.

Note: Sign-on script files are not needed on an OS/2 local host that uses the APPC access method because APPC has the ability to communicate directly with the remote host. △

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

If you do not want to omit the RLINK fileref but you want to avoid wasting processing time, use the NOSCRIPT option in the SIGNON and SIGNOFF statements, shown as follows:

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

Specifying a Server Name at the SAS/SHARE Client Only

The server name that you specify in the PROC SERVER, the PROC OPERATE, and the LIBNAME statements must be defined as the SAS/SHARE server-id.

The server name must meet the criteria for a valid SAS name. See *SAS Language Reference: Dictionary* for details about SAS naming rules.

An example of specifying the server name follows:

```
options comamid=appc;
libname demo 'C:/\' server=remote-server-id;
```

In this example, at the client, the server name is specified as a *remote-server-id*.

For details about creating LIBNAME, PROC OPERATE, and PROC SERVER statements, see *SAS/SHARE User's Guide*.

The access method that you specified at the SAS/SHARE client first attempts to locate the remote session or the SAS server within its domain. If the search for that named server fails and a gateway has been configured through an environment variable, the access method forwards a setup request to the DOMAIN server node. The protocol gateway service of the DOMAIN server attempts to locate the named server in one of its connected domains through the access methods specified by the COMAMID, COMAUX1, and COMAUX2 statements. Server names must be unique throughout the multi-domain network. All subsequent messages for that access method and server are routed through the DOMAIN server and redirected to the appropriate destination.

Examples

Example 1:

The following statements illustrate a NetBIOS inbound access method and an APPC outbound access method to an OS/390 system.

Local SAS session:

```
OPTIONS COMAMID=NETBIOS
        REMOTE=LUname-of-APPC/MVS-scheduler;
%LET VQPROTGW=DOMSRV;
%LET VQSECURE=_PROMPT_;
SIGNON NOSCRIPT;
```

DOMAIN server configuration file:

```
-COMAMID APPC
-COMAUX1 NETBIOS
```

DOMAIN server syntax:

```
PROC DOMAIN PROTOCOL SERVER=DOMSRV;
RUN;
```

Note: DOMSRV is the NetBIOS resource name. △

Example 2:

The following statements illustrate a NetBIOS inbound access method and a TCP/IP outbound access method.

Local SAS session:

```
OPTIONS COMAMID=NETBIOS
        REMOTE=tcp-name-of-remote-host;
%LET VQPROTGW=DOMSRV;
%LET VQSECURE=_PROMPT_;
SIGNON NOSCRIPT;
```

DOMAIN server Configuration file:

```
-COMAMID NETBIOS
-COMAUX1 TCP
```

DOMAIN server syntax:

```
PROC DOMAIN PROTOCOL SERVER=DOMSRV;
RUN;
```

Note: DOMSRV is the TCP/IP resource name. The *server-id* DOMSRV must be defined in the SERVICES file on the server host. △

Example 3:

The following statements illustrate a TCP/IP inbound access method and an APPC outbound access method.

Local SAS session:

```
OPTIONS COMAMID=TCP REMOTE=LUname-of-APPC/MVS-scheduler;
%LET GWHOST=Internet-address of DOMAIN server machine;
%LET TCPGW=GWHOST.DOMSRV;
%LET TCPSEC=_PROMPT_;
SIGNON NOSCRIPT;
```

DOMAIN server configuration file:

```
-COMAMID APPC
-COMAUX1 TCP
```

DOMAIN server syntax:

```
PROC DOMAIN PROTOCOL SERVER=DOMSRV;
RUN;
```

Note: The *server-id* DOMSRV must be defined in the SERVICES file on the server host and on the local 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

ISBN 1–58025–479–9

SAS Institute Inc., SAS Campus Drive, Cary, North Carolina 27513.

1st printing, September 1999

The Institute is a private company devoted to the support and further development of its software and related services.