

CHAPTER

1

SAS/CONNECT: Definitions and Services

<i>SAS/CONNECT Terminology</i>	3
<i>SAS/CONNECT</i>	3
<i>The Client/Server Relationship</i>	4
<i>Single-User Server</i>	4
<i>Multi-User Server</i>	4
<i>Communications Access Method</i>	4
<i>Programming Services</i>	5
<i>Compute Services and MP CONNECT</i>	5
<i>Compute Services That Use RSUBMIT</i>	5
<i>Compute Services That Use Remote SQL Pass-Through</i>	7
<i>Data Transfer Services</i>	7
<i>Remote Library Services</i>	9

SAS/CONNECT Terminology

SAS/CONNECT

SAS/CONNECT software is a SAS client/server toolset that provides scalability through parallel SAS processing. SAS/CONNECT provides users and applications developers the ability to manage, access, and process data in a distributed and parallel environment by enabling you to

- Achieve SAS interoperability across architectures and SAS releases
 - directly process to a remote data source and get results back locally
 - transfer disk copies of data
 - develop local graphical user interfaces that process remote data sources.
- Develop scalable SAS solutions
 - run multiple independent processes asynchronously and coordinate the results from each task execution in your client SAS session
 - scale up to fully use the capabilities of SMP hardware, and scale out to fully use the features of distributed processors
 - use pipeline processing (TCP/IP ports) to run multiple dependent processes asynchronously
 - collect the resources of multiple machines that work in parallel, which produces a powerful, yet inexpensive processing solution.
- Manage distributed resources

- perform daily or nightly automated backups
- initiate transaction processing to a master database at a specified time each day
- centralize and automate data and report distribution to workstations in a network
- centralize and automate data collection from workstations in a network.

The Client/Server Relationship

SAS/CONNECT links a SAS client session to a SAS server session. The terms SAS/CONNECT *client* and *server* depict a relationship between two SAS sessions.

The client session is the initial SAS session that creates and manages one or more server sessions. The server sessions can run either on the same machine as the client (for example, an SMP machine) or on a remote machine across a network.

Single-User Server

SAS/CONNECT provides the single-user server functionality for Remote Library Services (RLS), which

- provides transparent access to remote data.
- gives single-user access to a dedicated server.
- enables full, unrestricted access to DBMS data via a SAS/ACCESS engine.
- enables you to connect to the server by using a SIGNON statement and a LIBNAME statement that specifies the REMOTE engine.

```
SIGNON server-ID;
LIBNAME libref REMOTE 'datalib' SERVER=server-ID;
```

The LIBNAME statement implicitly starts the single-user server.

Multi-User Server

SAS/SHARE provides the multi-user server functionality for Remote Library Services (RLS), which

- gives concurrent, multi-user access to a server
 - Note:* The ability to access DMBS data through a multi-user server is controlled by a specific SAS/ACCESS engine. \triangle
- is explicitly started and controlled by a system administrator.

```
PROC SERVER server=server-ID;
```

- enables you to connect to the server by using a LIBNAME statement that specifies the REMOTE engine.

```
LIBNAME libref REMOTE 'datalib' SERVER=server-ID;
```

The LIBNAME statement causes a connection to a pre-existing server.

Communications Access Method

A *communications access method* is the interface between SAS/CONNECT and the network protocol that you use to connect two SAS sessions. You must specify a communications access method for SAS/CONNECT.

TCP/IP is the supported access method on all SAS 9.1 operating environments. The XMS access method is used to connect client and server sessions that both run under z/OS.

Before any meaningful work can be accomplished between a client and a server, the access method must be configured in the client and the server environments. For details, see *Communications Access Methods for SAS/CONNECT and SAS/SHARE*.

Programming Services

Compute Services and MP CONNECT

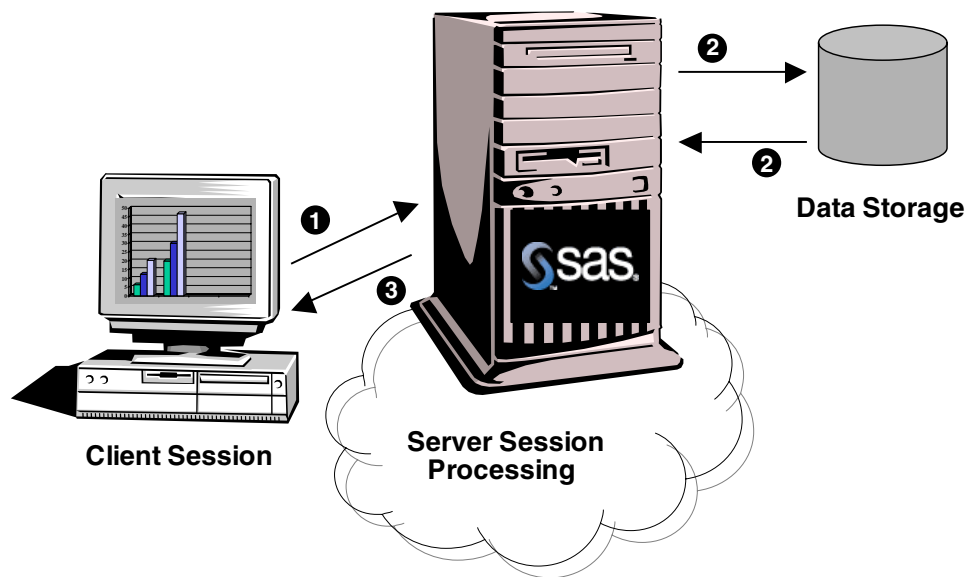
Compute Services That Use RSUBMIT

Compute Services provides access to all of the computing resources on your network by enabling you to direct the execution of SAS programs to one or more server sessions. The results and any output that is generated by the remote execution are returned to the client session. For short-running tasks, remote submits can be processed synchronously. This means that control is returned after the remote processing is complete. For longer-running tasks, remote submits can be processed asynchronously. This means that control is returned immediately, and you can continue local processing or remote processing to another server session.

Note: Asynchronous Compute Services is commonly referred to as MP (Multi-Process) CONNECT. △

The following figure shows that these services enable you to move some or all portions of an application's processing to a remote machine.

Figure 1.1 Model of Compute Services



Legend

- ❶ Send SAS statements to the server session for processing
- ❷ Use server data in the server session
- ❸ Receive results at the client

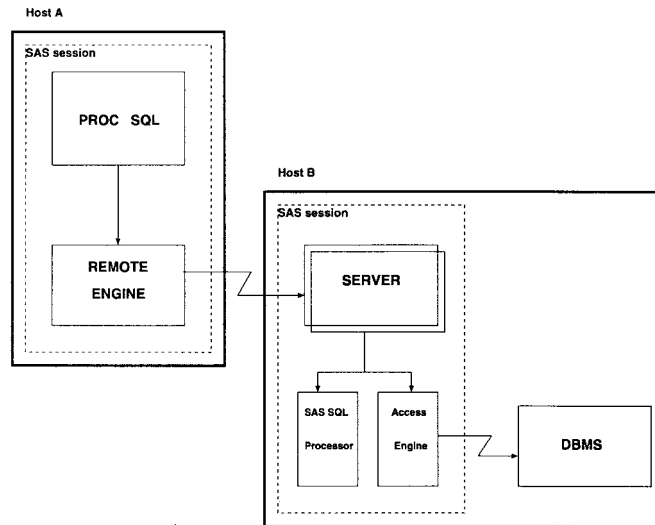
Compute Services enables you to:

- Achieve scalability for your SAS applications
 - perform remote tasks in the background (asynchronously) while processing locally.
 - run multiple SAS processes asynchronously and coordinate the results from each task execution in your client SAS session.
 - use pipeline processing to overlap execution of multiple dependent SAS DATA steps or procedures.
 - use processors on an SMP machine, which is referred to as “scaling up”; and using idle processors across a network, which is referred to as “scaling out.”
- Access remote resources
 - take advantage of server hardware and software resources.
 - access mainframe and other legacy systems, for example, by building a single SAS program that contains statements that run locally and statements that execute on multiple remote legacy machines.
 - execute against the remote copy of the data.
 - remote submit macro steps to the server, and then pass return code information about the server process to the client.
 - execute graphics programs on the server, and display the graphics locally by using the graphics capabilities of the local workstation, plotter, or printer.

Compute Services That Use Remote SQL Pass-Through

Remote SQL Pass-Through (RSPT) gives you control of where SQL processing occurs. RSPT enables you to pass SQL statements to a remote SAS SQL processor by passing them through a remote SAS server. You can also use RSPT to pass SQL statements to a remote DBMS by passing them through a remote SAS server and a REMOTE access engine that supports pass-through.

Figure 1.2 Remote SQL Pass-Through Services



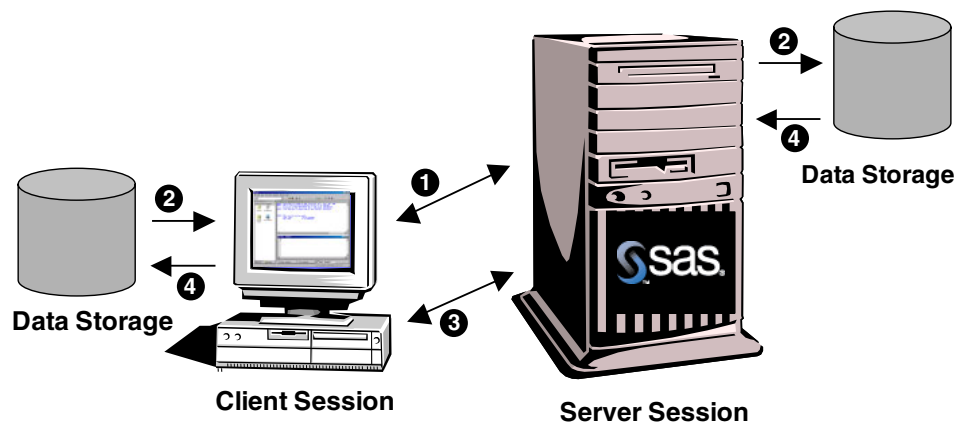
RSPT enables you to:

- pass SQL statements to a remote DBMS in order to select data or to execute statements in order to modify, manipulate, and manage data. This includes creating DBMS views.
- pass SQL statements to SAS SQL in order to select data or to execute statements in order to modify, manipulate, and manage data. This includes creating SAS SQL views.

You can invoke RSPT by using PROC SQL statements that are passed to the remote server for execution in the server SAS session, or you can store SQL pass-through statements in local SQL views.

Data Transfer Services

Data Transfer Services enables you to move a copy of the data from one machine to another machine. The data is translated between machine architectures and SAS version formats, as necessary.

Figure 1.3 Model of Data Transfer Services (UPLOAD and DOWNLOAD)**Legend**

- 1** Request upload of data records from the client to the server
- 2** Data is copied from the client and written to disk on the server
- 3** Request download of data records from the server to the client
- 4** Data is copied from the server and written to disk on the client

Data is transferred with the `UPLOAD` and `DOWNLOAD` procedures. You can transfer SAS data sets, SAS catalogs, MDDDB, SQL views, entire SAS data libraries, and external files.

Note: External files can be transferred in either text or binary format. △

The data transfer capabilities enable you to:

Customize data transfers

- transfer multiple SAS files in a single step by using the `INLIB=` and `OUTLIB=` options. This capability enables you to transfer an entire library or selected members of a library in a single `PROC UPLOAD` or `PROC DOWNLOAD` step.
- transfer collections of files (such as a partitioned data set, a `MACLIB`, or a directory) between a client and a server.
- use `WHERE` processing for dynamic data subsetting and SAS data set options when transferring individual SAS data sets.
- transfer catalog entries that contain graphics output by using a simple one-step process.

Protect data

- increase the robustness of your decision support environment by keeping a local copy of your data, which is insulated from network failure.
- back up local files to a server.

Manage data distribution

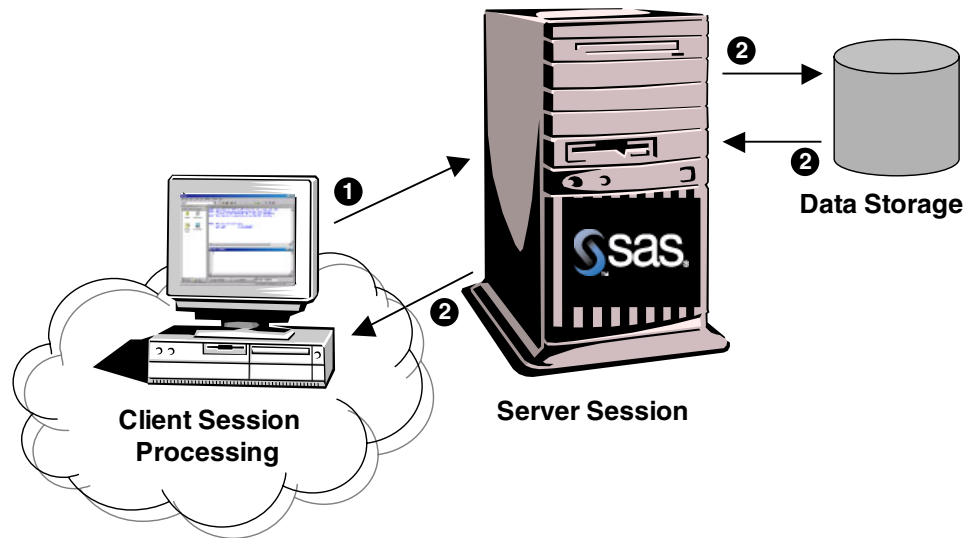
- automate both data or application distribution and centralized data collection.
- distribute files from one workstation by uploading to a server and downloading to other workstations that need the files.

- move SAS files between releases of SAS as well as across operating environments.

Remote Library Services

Remote Library Services (RLS) provides transparent access to SAS data that is located on a remote machine. The data resides in server libraries, and RLS moves the data through the network as client processing requests it. The data must again pass through the network on any subsequent use by the client session. As the following figure shows, a copy of the data is not written to the client file system.

Figure 1.4 Model of RLS Processing



Legend

- ①** Client processing requests records from the server or requests records to be written to the server
- ②** Data records written to the server or sent to the client for processing

The SAS procedures and DATA steps that run in the SAS/CONNECT client session request access via the REMOTE engine to SAS files that are located on a SAS/CONNECT server. The REMOTE engine communicates the requests for data to the server. The server administers the requests to access SAS files on behalf of the client.

RLS provides:

- transparent access to SAS data that is located on a remote machine.
- access to current SAS data because no client copy is made.
- a reduction of disk space consumption because multiple copies of the data are not created.
- the ability to run a local graphical user interface and process SAS data that is located on a remote machine.