

Paper 001-2010

**Getting Connected to Your Data with SAS/CONNECT®**

Greg McLean, Statistics Canada, Ottawa, Ontario, Canada

Vecdet Mehmet-Ali, Statistics Canada, Ottawa, Ontario, Canada

**ABSTRACT**

With the introduction of Microsoft SSPI authentication in SAS® version 8, using SAS/CONNECT® to establish a server session on a remote Windows server and accessing data on the processing server or on the network has evolved. This paper explores different ways available to sign on to a remote Windows server with SAS/CONNECT® and describes the implications of accessing data.

Methods of channeling data through the client SAS® session will also be discussed.

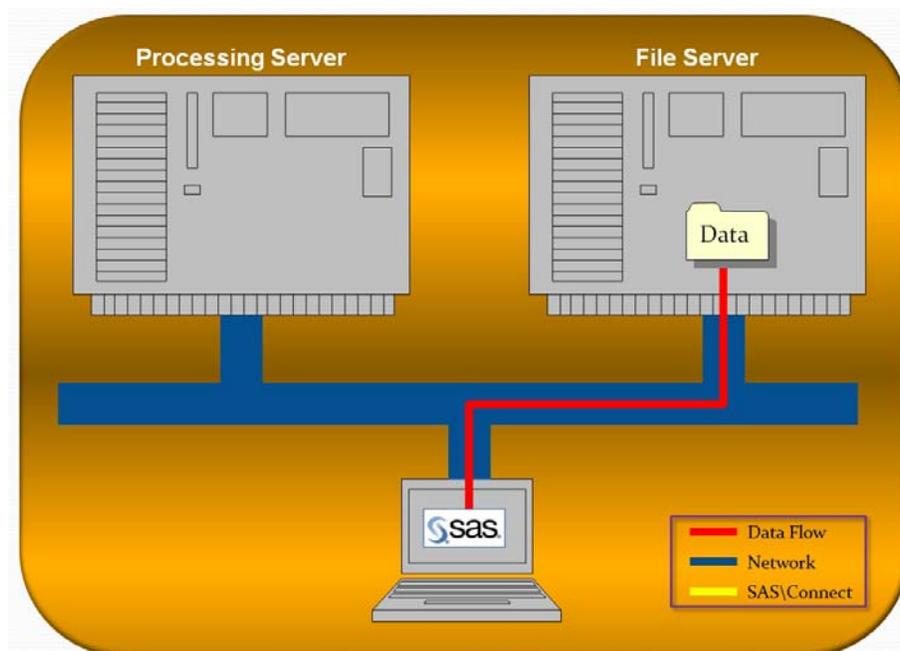
**INTRODUCTION**

Accessing data within a SAS® session is as simple as assigning a LIBNAME or FILENAME statement. However, we often take for granted what is going on behind the scenes when these types of statements are executed. Obviously the operating system plays an important role in how and where we may access our data. In the Microsoft Windows environment, it is necessary for the operating system to “authenticate” the user. In other words grant or deny access to a given machine and/or folder. The following is a more technical definition of what we mean by authentication.

“**Authentication** is the act of verifying the identity of the user who is attempting to access a machine--that is, the machine that either the client session or the server session runs on. Authentication is performed so that a machine can use the identity information to make decisions about the user's authority to access protected resources. Under Windows, the user ID, password, and access permissions make up a user context. “

(SAS® Institute Inc. 2009, SAS Online Help and Documentation)

In **Figure 1** we observe a local SAS® session connecting to a remote file server and accessing a given folder. In order to successfully connect to the folder it is necessary for the operating system to authenticate our credentials. We are either granted access or denied. And in some cases, we are granted access, but for read-only.

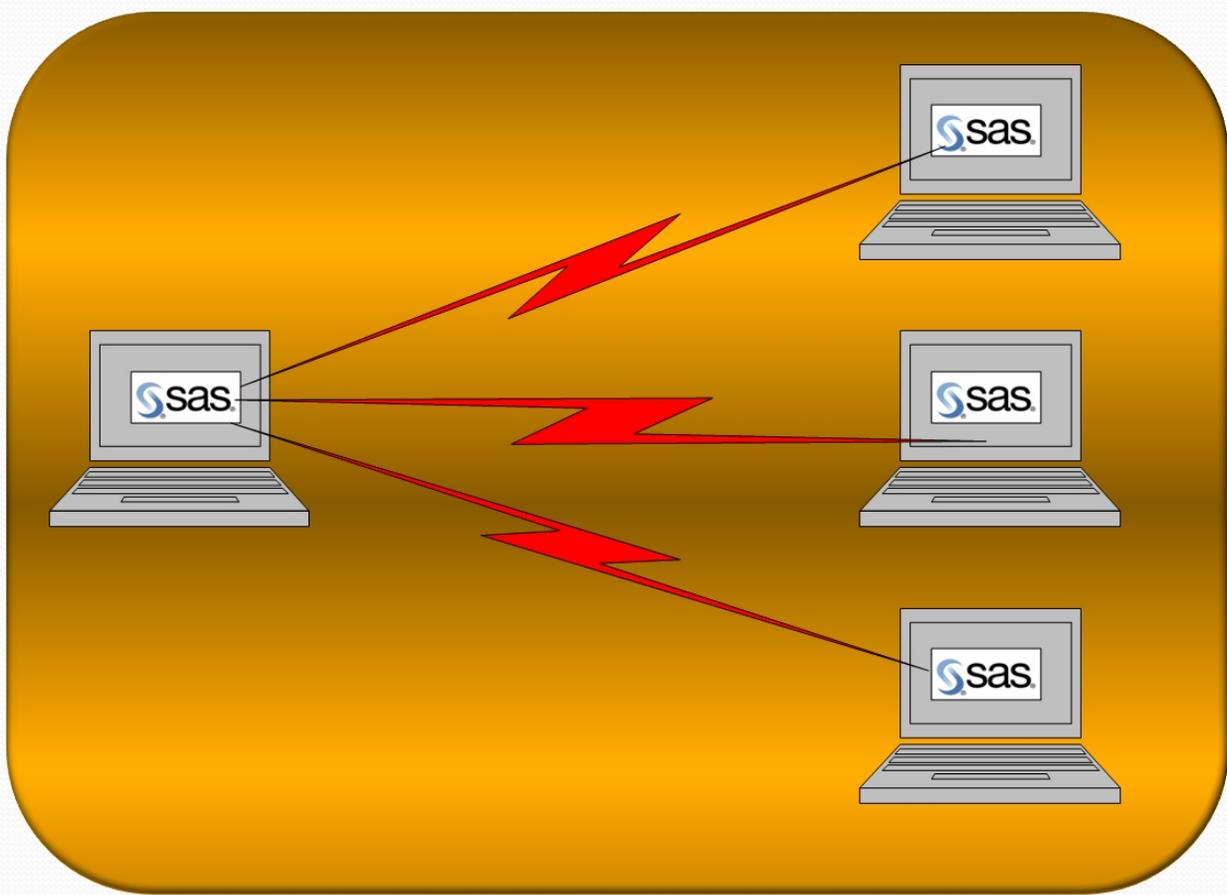


**Figure 1**

However, when we introduce SAS/CONNECT<sup>®</sup> into this mix, this concept of authentication becomes much more complex and deserves a more detailed discussion. In many instances processing data using SAS/CONNECT<sup>®</sup> has a few advantages worth mentioning:

- Faster Processing Time (due to proximity of data)
- Faster Processing Time (due to advance server hardware)
- Better security (built into server OS)

SAS/CONNECT<sup>®</sup> allows a SAS<sup>®</sup> session on a client machine to connect and “spawn” SAS<sup>®</sup> sessions on one or more remote machines. These remote machines may or may not be the same platform type.



**Figure 2**

When connecting to a remote machine using SAS/CONNECT<sup>®</sup>, there are basically two different methods that can be used to sign on within a Microsoft Windows environment. And as we will see later, the method used, will impact how and where we may access data.

1. Connecting by explicitly providing credentials (Script File or User ID / Password on SIGNON statement). Also referred to as “Simulated Logon” in SAS<sup>®</sup>.
2. Connecting by allowing OS to pass client credentials (Microsoft SSPI)

## TERMS

Before we begin discussing the various means of connecting to our data, we first need to define a few terms and concepts:

### Microsoft SSPI

SSPI (Security Support Provider Interface) enables transparent authentication for connections between Windows machines. Users that are members of a "trusted" domain are authenticated automatically, and user context information is transferred to the server.

Windows attempts to use SSPI for authentication whenever a user ID is not explicitly supplied. SSPI is available only when the client and the server sessions both run on Windows machines, and the user who runs the client machine is a member of a domain that is "trusted" at the server machine.

In order to use SSPI for authentication, the "-SECURITY" setting must be used when the SAS® Spawner is initially setup up.

### Script File

A script is a SAS® program that is stored in a file on the client machine. The programming statements in a script file are not the usual SAS® programming statements. Script files use a specialized set of SAS® statements called script statements. Scripts are executed to start or to stop SAS/CONNECT® sessions. A FILENAME statement using the reserved name of "RLINK" is used to define the location and filename of the script file. A SIGNON statement is then used to actually start the connection process.

When using a script file, the user's credentials (User ID and Password) are explicitly passed to the remote machine for authentication. In other words, when using a script file, the user will be required to provide a User Id and Password which will be either hard-coded within script file (not recommended) or prompted.

```
FILENAME RLINK "c:\temp\tcpwin.scr";
SIGNON serv1;
```

### SIGNON Statement

New in SAS® Version 9 is the ability to provide a User ID and Password on the SIGNON statement. Additionally, it is possible to have the user prompted for the User Id and/or password using the \_PROMPT\_ command. As in the case of using a script file, user credentials are explicitly acquired on the client and sent to the remote machine for authentication.

```
SIGNON serv1 USERID=gsmith PASSWORD=_PROMPT_;
```

### "Log on as a batch job" Windows Domain User Right

This is a policy or security setting that allows a user to be logged on by means of a batch-queue facility. It basically allows a user to be logged in as a job rather than an interactive user. This policy can be accessed from the Control Panel under Administrative Tools, Local Security Settings. In the list of policies you will find one called "Log on as a batch job".

When user credentials are supplied to the server explicitly (script or prompt), they are compared against the "Log on as batch job" user rights. If the user id is found then a remote SAS® session is able to be started.

## ACCESSING DATA ON PROCESSING SERVER

It is quite common to have one's data located on the same machine where it will get processed. There are many advantages to this setup. One of the most common reasons for this type of processing is efficiency. Besides the fact that server equipment is typically more advanced and more powerful, data access time is greatly improved as it does not have to travel across the network. Therefore, overall processing time tends to be dramatically improved.

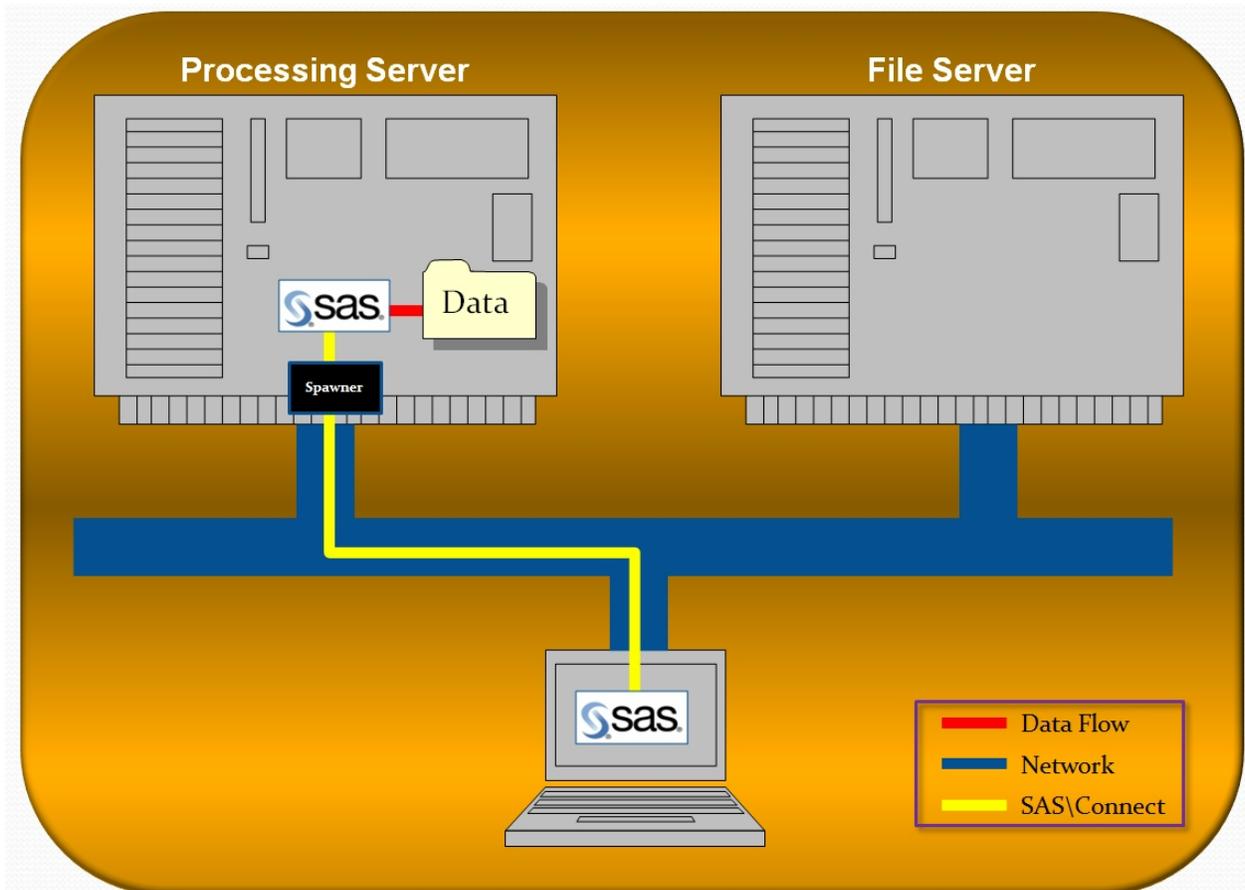


Figure 3

In this instance it is **not** necessary to connect using a script file or User Id / Password on the SIGNON statement. Instead, we can allow Microsoft SSPI to automatically forward our credentials to the remote server for authentication. It will also be necessary for the user to have the correct security credentials defined on the desired data folder on the processing machine (i.e. Read / Write).

## ACCESSING DATA ON FILE SERVER

Although it is common to store one's data on the processing machine, many organizations actually differentiate the use of their servers. Some servers may be dedicated and optimized for processing only, whereas other servers are optimized for storage and data access. In these situations we still may want to make use of a remote SAS® session for processing.

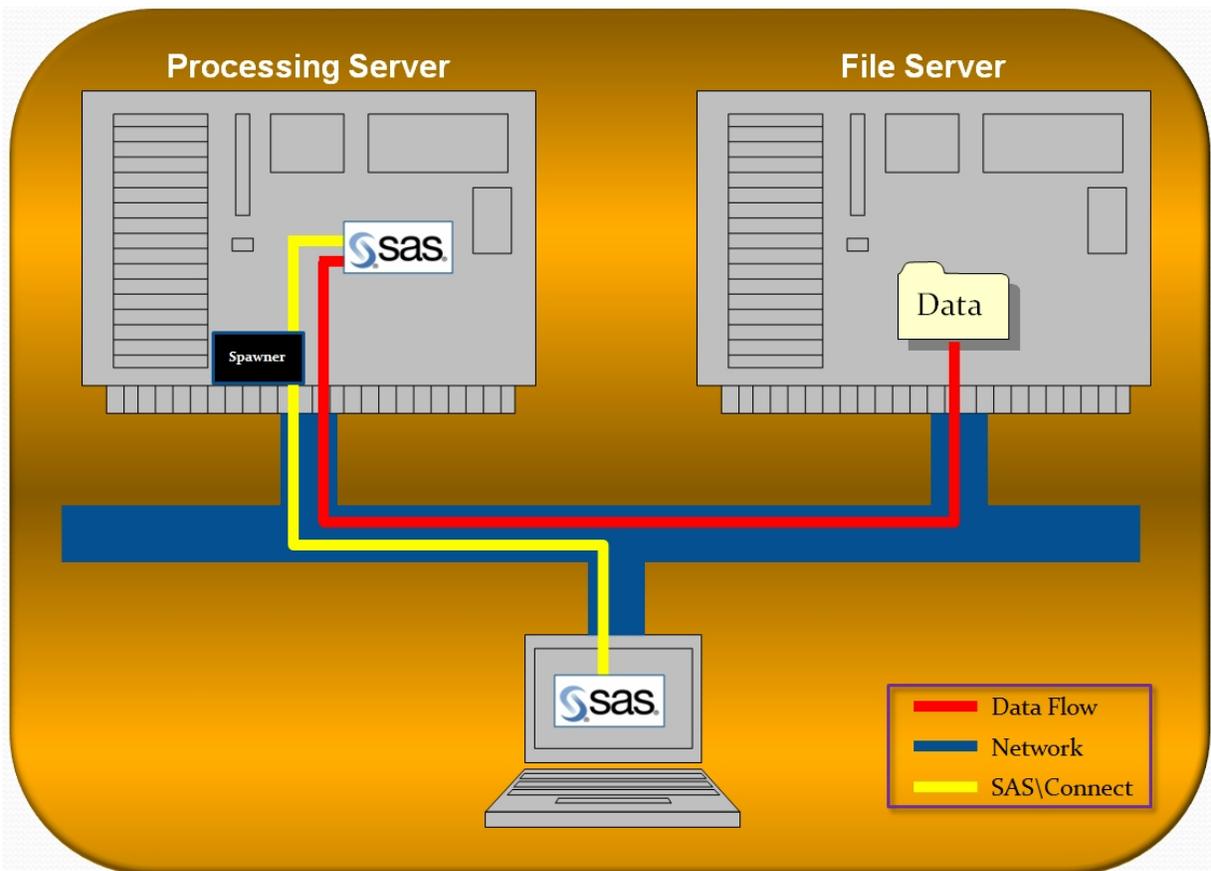


Figure 4

In this instance it is imperative that we use a script file or provide User Id and Password on the SIGNON statement when remote connecting. Instead of having SSPI automatically forwarding the user credentials, we must acquire the user credentials from the user before remote connection. The user must be defined with the **“Logon as a batch job”** domain user right policy on the processing server as well as having the correct security credentials on the desired data folder on the file server machine (i.e. Read / Write).

Of course using this method will allow the remote SAS® session direct access to data on the network. However, it is also possible to connect the remote SAS® session to the data on the network through the client SAS® session. This data access method will be described in greater detail in the next section.

## ACCESSING DATA ON FILE SERVER (THROUGH CLIENT SESSION)

In some instances it may not be possible for the remote SAS<sup>®</sup> session to directly access data from the network. This will greatly depend on how the SAS<sup>®</sup> connect session was started. As we learned in the previous section, we must use a script file or provide user credentials on the SIGNON statement if we intend for the remote SAS<sup>®</sup> session to access data directly on the network.

However, there may be occasions when it is not possible or not desired to explicitly provide credentials using a script file or SIGNON statement. For example, if we wish to use the Microsoft Windows Scheduler to start a SAS<sup>®</sup> job during off-peak hours, obviously it is not possible to type a User ID and Password when the job starts (i.e. 1:00AM). And it is not recommended to hardcode a User ID or Password in the script file or on the SIGNON statement for obvious security reasons. Therefore we will need to use other methods to get the data from the file server to the remote processing server. Using the client SAS<sup>®</sup> session as a medium, it is possible to connect the data to the remote SAS<sup>®</sup> session on the processing server.

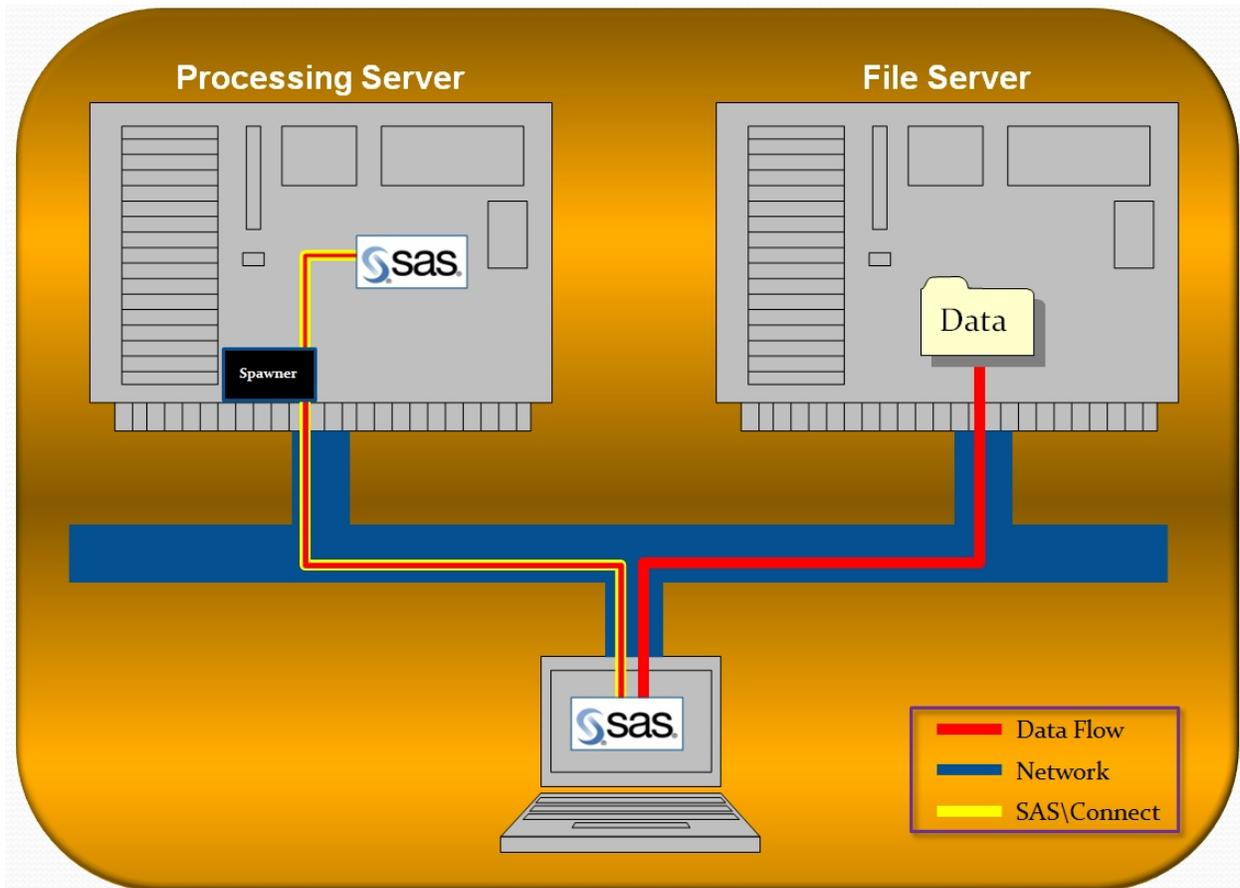


Figure 5

The following 3 methods may be used to perform this type of connection.

### DATA TRANSFER SERVICES

Data Transfer Services is a component of the SAS/CONNECT<sup>®</sup> software that allows the movement of data between the client SAS<sup>®</sup> session and the remote SAS<sup>®</sup> session. A simple LIBNAME or FILENAME statement on the client SAS<sup>®</sup> session will allow access to the desired folder on the file server. We can then use PROC UPLOAD and PROC DOWNLOAD to move data between the file server and the remote SAS<sup>®</sup> session using the client SAS<sup>®</sup> session as the medium.

```
LIBNAME dataloc "\\serv2\data";
RSUBMIT;
  PROC UPLOAD in=dataloc /* LIBREF Defined on client */
             out=WORK; /* LIBREF Defined on server */
             SELECT ds1 ds2 (memtype=data);
  RUN;
  /* PROCESS DATA THAT HAS BEEN UPLOADED */
  ...
  ...
  ...
  PROC DOWNLOAD in=WORK /* LIBREF Defined on server */
               out=dataloc; /* LIBREF Defined on client */
               SELECT ds1 ds2 (memtype=data);
  RUN;
ENDRSUBMIT;
```

### REMOTE LIBRARY SERVICES

Remote Library Services is also a component of the SAS/CONNECT<sup>®</sup> software. The purpose of this module is to allow a direct connection from the client SAS<sup>®</sup> session to a defined LIBREF on the remote SAS<sup>®</sup> session. Within the client SAS<sup>®</sup> session, we can then merely refer to a folder on the processing machine (often WORK) by a simple defined LIBREF.

In order to move data between the file server and the remote SAS<sup>®</sup> session, two LIBNAME statements are required on the client SAS<sup>®</sup> session. One LIBREF is required to access the data folder on the file server and the other LIBREF is used to link to an existing LIBREF on the remote SAS<sup>®</sup> session using the SLIBREF statement. From then on, any valid SAS<sup>®</sup> procedure or data step (on the client SAS<sup>®</sup> session) can be used with the two defined Library references.

```
LIBNAME dataloc "\\serv2\data"; /* LIBREF to Data Folder */
LIBNAME R_WORK Server=Serv1 SLIBREF=WORK; /* LIBREF to Server WORK */
/* MOVE DATA UP TO SERVER */
PROC COPY in=dataloc
          out=R_WORK;
          Select ds1 ds2 (memtype=data);
Run;
/* PROCESS DATA ON SERVER */
RSUBMIT;
  ...
  ...
  ...
ENDRSUBMIT;
/* MOVE DATA DOWN FROM SERVER */
PROC COPY in=R_WORK
          out=dataloc;
          Select ds1 ds2 (memtype=data);
Run;
```

## INHERITLIB

INHERITLIB is a new statement in SAS® version 9 that can be used on either the SIGNON or RSUBMIT statements. It is similar to the SLIBREF statement as mentioned in the previous section. This statement also allows a link to a predefined LIBREF statement on the client SAS® session. However, it is in the opposite direction. It allows the remote SAS® session to link to a LIBREF defined in the client SAS® session. Remote submitted code can then refer to the INHERITLIB LIBREF to directly access data through the client SAS® session.

```
LIBNAME dataloc "\\serv2\data"; /* LIBREF of Data */
RSUBMIT INHERITLIB(dataloc);
  DATA dataloc.ds1;
    SET dataloc.ds1;
    ...
    ...
  RUN;
ENDRSUBMIT;
```

## SUMMARY

As we have seen throughout this paper, there are two different ways to connect to a remote SAS® session:

- Connecting by explicitly providing credentials (Script File or SIGNON statement)
- Connecting by allowing OS to pass client credentials (Microsoft SSPI)

Depending on which method is used to connect, there will be an impact on how and where we can access our data.

The following tables summarize the various scenarios discussed in this paper. It details the necessary requirements to access data in the specified location.

Location of Data	Connection Method	Log on as a batch job	Establish Server Session	Access Data From Remote Session
Processing Server	Script / Sign On	Yes	Yes	Yes
Processing Server	Script / Sign On	No	No	N/A
Processing Server	SSPI	Yes / No	Yes	Yes
File Server	Script / Sign On	Yes	Yes	Yes
File Server	Script / Sign On	No	No	N/A
File Server	SSPI	Yes / No	Yes	No ***

**Table 1**

**Note: \*\*\***

If you use SSPI (User Id and Password not explicitly provided with a Script or on SIGNON statement) you may be able to access data on the File Server from the Processing Server if you use any of the three methods described in this paper:

- Data Transfer Services
- Remote Library Services
- INHERITLIB (on SIGNON or RSUBMIT)

## REFERENCES

SAS® Institute Inc. 2009, SAS Online Help and Documentation

## CONTACT INFORMATION

For information on topics covered in this paper please contact:

	Statistics Canada	Statistique Canada
---	----------------------	-----------------------

**Greg McLean**

Project Leader  
SAS® Technology Centre  
System Development Division

R.H. Coats Building, 14<sup>th</sup> Floor, Section Q  
Ottawa, Ontario, Canada K1A 0T6  
(613) 951-2396 Fax (613) 951-0607

[Greg.Mclean@statcan.gc.ca](mailto:Greg.Mclean@statcan.gc.ca)

**Canada**

	Statistics Canada	Statistique Canada
---	----------------------	-----------------------

**Vecdet Mehmet-Ali**

Project Leader  
SAS® Technology Centre  
System Development Division

R.H. Coats Building, 14<sup>th</sup> Floor, Section A  
Ottawa, Ontario, Canada K1A 0T6  
(613) 951-2390 Fax (613) 951-0607

[Vecdet.Mehmet-Ali@statcan.gc.ca](mailto:Vecdet.Mehmet-Ali@statcan.gc.ca)

**Canada**

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.

Other brand and product names are trademarks of their respective companies.