

Paper 110-2008

## An Introduction to SAS/Connect®

James C. Stokes, D&B, Short Hills, NJ

### ABSTRACT

Most of us today are required to process data on more than one computer platform (e.g. Mainframe, UNIX, Windows, etc.). The term “distributed processing” might be used to describe this scenario.

Wouldn't it be great if there was a way to easily move SAS datasets between computer platforms? Wouldn't it be great to submit SAS jobs from a local platform to run on a remote platform? Wouldn't it be great to be able to have local access to a SAS dataset stored on a remote platform? If you've answered yes to any or all of these questions, then SAS/Connect is something you really need. SAS/Connect will allow you to greatly simplify some of the more challenging aspects related to distributed processing.

This paper will define what SAS/Connect is, what you need to get started, and explore the three main features of SAS/Connect namely: Data Transfer Services, Remote Compute Services, and Remote Library Services.

### SAS/CONNECT DEFINED

SAS/Connect is software (i.e. middleware) that enables a user to seamlessly communicate with remote sessions of SAS running on one or more computer platforms.

There are three main features of SAS/Connect: Data Transfer Services, Remote Compute Services, and Remote Library Services.

### WHAT YOU NEED TO GET STARTED

In addition to having SAS® software on the platforms you intend to do the processing on, you will need three key pieces of information. The three key pieces of information needed to make use of SAS/Connect® are as follows: the communications access method (COMAMID), the remote node name (REMOTE), and a script file for signing on, establishing a remote session of SAS, and signing off The remote host (RLINK). (At first glance this may seem a bit overwhelming, but it's not really that bad.)

The communications access method and the remote node name are supplied to SAS as SAS system options: COMAMID and REMOTE respectively. The location of the script file is coded on a filename statement and is assigned the fileref RLINK.

COMAMID is an acronym for Communications Access Method Identification. This identifies the access method used by the local machine to communicate with the remote machine (e.g. TCP/IP, etc.).

REMOTE is where you would code either the IP Address of the remote host or the remote hostname.

The fileref RLINK points to the script you wish to use to connect to the remote host. SAS/Connect® supplies numerous signon scripts to be used to establish a connection to a remote host. So there you have it, all you need is COMAMID, REMOTE, and RLINK and you are on your way to making use of SAS/Connect®.

### DATA TRANSFER SERVICES

Data Transfer Services makes moving SAS datasets between platforms quite easy. PROC DOWNLOAD and PROC UPLOAD are the two procedures used to move data between platforms.

PROC DOWNLOAD is used to move a SAS dataset from the remote host to the local host.

PROC UPLOAD is used to move a SAS dataset from the local host to the remote host.

Consider the following scenario: you have a SAS<sup>®</sup> dataset named **TESTDAT** on OS/390 stored under DSN **myuserid.sas.data** and you want to move it to the following directory on Windows **d:\projects\companyx\data**. The access method used to connect to the remote host is TCP/IP and the name of the remote host is **MAINFR1**. The signon script is named **tcptso.scr** and is stored in the following directory on Windows C:\Program Files\SAS Institute\SAS\V8\saslink. The code to accomplish this follows:

```

1      Options comamid=tcp remote=MAINFR1;
2      Filename RLINK 'C:\Program Files\SAS Institute\SAS\V8\saslink\tcptso.scr';
3      Signon;
4      Libname out 'd:\projects\companyx\data';
5      Rsubmit;
6      Libname in 'myuserid.sas.data';
       Proc download in=in out=out;
       Select testdat;
       Run;
9      Endrsubmit;
10     Signoff;

```

Step1 – Access method (COMAMID) is identified as TCP and remote (hostname) MAINFR1

Step2 – fileref RLINK is established and points to tcptso.scr signon script.

Step3 – signon to remote host is initiated. User is prompted to enter MAINFR1 userid and password. Once signon is complete, remote session of SAS<sup>®</sup> is started.

Step4 – libref OUT is assigned to point to directory on Windows

Step5 – statements following rsubmit will be processed on the remote session of SAS<sup>®</sup>

Step6 – libref IN is assigned on MAINFR1

Step7 – PROC DOWNLOAD is executed on MAINFR1

Step8 – testdat is selected for download

Step9 – signifies the end of statements to be executed on the remote session

Step10 – connection to remote session is terminated

## REMOTE COMPUTE SERVICES

Remote Compute Services is used to run SAS code on a remote session of SAS. To run code remotely, one must first establish a SAS session on the remote host (i.e. SIGNON to the remote host). Once a SAS session has been established on a remote host, one need only to preface the code to be run on the remote host with a RSUBMIT statement and follow the block of code with a ENDRSUBMIT statement.

Consider the following scenario: you have a SAS<sup>®</sup> dataset named **TESTDAT2** on UNIX stored in the following directory **/mydir/compx/data/** and you want to run a frequency on variable KEY and return the procedure output to the output window of the session of SAS<sup>®</sup> running on Windows. The access method used to connect to the remote host is TCP/IP and the name of the remote host is **UNIX1**. The signon script is named **tcpunix.scr** and is stored in the following directory on Windows C:\Program Files\SAS Institute\SAS\V8\saslink. The code to accomplish this follows:

```

1      Options comamid=tcp remote=UNIX1;
2      Filename RLINK 'C:\Program Files\SAS Institute\SAS\V8\saslink\tcpunix.scr';
       Run;
3      Signon;
       Run;
4      Rsubmit;
5      Libname in '/mydir/compx/data/';
       Run;
6      Proc freq data=in.testdat2;
       Table key / missing;
       Run;

```

```
7      Endrsubmit;
8      Signoff;
```

Step1 – Access method (COMAMID) is identified as TCP and remote (hostname) UNIX1

Step2 – fileref RLINK is established and points to tcpunix.scr signon script.

Step3 – signon to remote host is initiated. User is prompted to enter UNIX1 userid and password. Once signon is complete, remote session of SAS® is started.

Step4 – statements following rsubmit will be processed on the remote session of SAS®

Step5 – libref IN is assigned on UNIX1

Step6 – PROC FREQ is executed on UNIX1

Step7 – signifies the end of statements to be executed on the remote session

Step8 – connection to remote session is terminated

## REMOTE LIBRARY SERVICES

Remote Library Services enables a user to gain local access to a SAS dataset stored on a remote host. To gain local access to a SAS dataset stored on a remote host, one must first establish a SAS session on the remote host (i.e. SIGNON to the remote host). Once a SAS session has been established on a remote host, a user simply needs to locally execute a LIBNAME statement using the SERVER option.

There are times when moving a SAS® dataset to the local host is not practical or may not be possible for security or other reasons. In any event, the Remote Library Services component of SAS/Connect® allows you to access data stored on a remote host from the local host. Consider the following example: you want to process a SAS dataset named **CUST** stored in the directory **/mydir/compa/data** on UNIX from a SAS® session running on Windows. The access method used to connect to the remote host is TCP/IP and the name of the remote host is **UNIX1**. The signon script is named **tcpunix.scr** and is stored in the following directory on Windows C:\ProgramFiles\SAS Institute\SAS\V8\saslink. Here is the code:

```
1      options remote=UNIX1 comamid=tcp;
2      Filename RLINK 'C:\Program Files\SAS Institute\SAS\V8\saslink\tcpunix.scr';
3      signon;
4      libname inremote "/mydir/compa/data/" server=UNIX1;
      run;
```

Step1 – Access method (COMAMID) is identified as TCP and remote (hostname) UNIX1

Step2 – fileref RLINK is established and points to tcpunix.scr signon script.

Step3 – signon to remote host is initiated. User is prompted to enter UNIX1 userid and password. Once signon is complete, remote session of SAS is started.

Step4 – libref INREMOTE is assigned locally but by coding the server= option it points to UNIX1

Once the libname is executed, you now have local access to the SAS® dataset on the remote host.

For example, if you want to run a PROC CONTENTS on the members in the libref inremote, you would simply need to code the following:

```
PROC CONTENTS DATA=INREMOTE._ALL_;
RUN;
```

Please note: it is **not** necessary to construct the code around a rsubmit/endrsubmit statements since the data is being processed locally.

## CONCLUSION

SAS/Connect® is a great way to easily conquer some of the obstacles we face related to distributed processing. While this paper in no way represents a comprehensive study of SAS/Connect® and its capabilities, it should at a minimum form the basis of a solid introduction to SAS/Connect® and some of its uses.

## CONTACT INFORMATION

Your comments and questions are valued and encouraged. Contact the author at:

Author Name: James C. Stokes  
Company: D&B  
Address: 103 John F. Kennedy Parkway  
City state ZIP: Short Hills, NJ 07078  
Work Phone: (973) 921-5909  
Fax: (866) 772-7168  
Email: [stokesj@dnb.com](mailto:stokesj@dnb.com)  
Web: [www.dnb.com](http://www.dnb.com)

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.