

Paper CC-07

Locally Visible, Remote Data and Format

Hsiwei Yu, Custom Software Systems; Leesburg, VA

Brian Deitch, FDIC; Arlington VA

Kamau Njuguna, Lockheed Martin; Arlington, VA

ABSTRACT

When in an a workplace environment with mixed platforms, sooner or later one will come across the need to work in or access files in a host that is in a different environment . This paper shows how to make remote data available in the local environment. This way one can perform interactive tasks such as browsing and querying as if everything is local, at one's finger tip, for more productive work.

PROBLEM STATEMENT

Say, you have SAS® data on Unix or z/OS; do you have to be on the remote host to invoke SAS for viewing and working?

Answer: No, you don't have to. With SAS/Connect or SAS/Share, you can use PC SAS to view and work with SAS data from different hosts and environments. This paper gives step-by-step instructions.

BACKGROUND

The authors have worked in organizations having a mixture of machines, PC, Unix, and z/OS, and SAS is licensed on each platforms. Without SAS/Connect or SAS/Share, a user must login to one platform and work on SAS data. To transfer data between hosts, use FTP and Proc Cport, Cimport etc.

Typically SAS/Connect is licensed. PC SAS user can connect to other platform to work on SAS data on the remote host and transfer SAS data between the PC and remote hosts.

However if SAS/Share is licensed, PC SAS users can work with remote SAS data without the need to login to remote host. For non-SAS clients such as Excel, through ODBC, can view SAS data from remote host.

Recently the authors were very delighted to find out that in SAS 9.2 the PC SAS session can not only access remote data but also remote formatted data. This feature definitely was not available with older versions. We didn't bother to determine when this feature first becomes available. We are only too happy to use it for productivity gains.

SOLUTION OUTLINE 1:

```
/* On PC SAS session when only SAS/Connect is licensed. */
%let sassrv=remote_host TCP_IP_port_number;
```

```
options remote=sassrv comamid= tcp;
filename rlink '.. .. .';
signon;
```

```
rsubmit;
```

Where there is a table and a column having a format associated with it.
See Appendix on how to create such a table.

```
libname datalib '... / ...';
endrsubmit;
```

```
libname local server= sassrv slibref= datalib;
```

The above PC SAS libref, *local*, is actually provided by the remote session, *sassrv*, on Unix or z/OS etc and its libref known to remote session is called *datalib*.

```
options insert= ( fmtsearch= ( local )) /* Insert is a new 9.2 option. */
```

The *local* library has format in it and we want to use it to see the decoded, i.e. formatted data values!

```
  fmterr /* reports an error when format NOT available then. */
;
```

Method 1 to show format is resolved and available.

```
proc print data= local.demo_fmt_available_to_PC;
  var comments; /* The column having a format associated with it. */
run;
```

Method 2 to view the column value decoded.

```
dm `viewtable data= local.demo_fmt_available_to_PC( keep= comments );'
  viewtable;
```

Method 3 to invoke SAS Explorer to view available data libraries.

```
dm `explorer;';
```

SOLUTION OUTLINE 2, ON SAS/SHARE SERVER ITSELF :

You must first start up the SAS/Share server!

```
%let server= remote_host.__TCP_IP_port_number;

libname rmt_data '.. ..';
libname rmt_fmt '.. ..';

proc server id= &server;
run;
```

Solution Outline 2, PC SAS client using SAS/Share server:

On the PC SAS session with SAS/Share installed

```
%let sassrv=remote_host.__TCP_IP_port_number;
libname lcl_data server= &sassrv slibref= rmt_data;
libname lcl_fmt server= &sassrv slibref= rmt_fmt;
options comamid= tcp
  insert= ( fmtsearch= ( lcl_fmt ))
  fmterr
;
```

CONCLUSION:

The paper showed how to interactively view remote SAS data and its formatted data values on a PC SAS session if they are simply local. The standard warning is network traffic; the PC SAS session may request huge amount of data across the network. Warnings aside, the benefit of convenience and productivity gains is tremendous.

REFERENCE:

Note 1: [SAS/Connect online documentation](#).

Note 2: Training info [SAS/CONNECT for Distributed Processing: Accessing a Remote SAS Data Library](#)

ABOUT THE AUTHORS:

Hsiwei Yu programmed in SAS for many years and has published papers in SGF and NESUG. He is currently working on-site at the FDIC in Arlington, Virginia.

Brian Deitch is a programmer with FDIC in Arlington, Virginia.

Kamau Njuguna is a project manager and SAS developer with Lockheed Martin in Arlington, Virginia. He has over 16 years of SAS experience and has published papers in NESUG.

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.

APPENDIX:

The code below creates a format and a table with a column that is associated with that format. Although the table exists on Unix, it can be made visible to your PC SAS session.

```
libname datalib '... / ...';

options fmtsearch= ( datalib )
;

proc format lib= datalib;
value E_desc /* E_desc for English description. */
1= 'Good'
2= 'Bad'
;
run;

data datalib.demo_fmt_available_to_PC;
attrib comments format= E_desc.;

id= 'a'; comments= 1;
output;

id= 'b'; comments= 2;
output;
run;
```