

Paper 39-28

UNIX Meet PC: Version 8 to The Rescue

Yadong Zhang, Oxford Health Plans, Trumbull, CT

ABSTRACT

In this information age, options often seem to be unlimited. You often need to make some tough decisions: Should I develop it on UNIX or PC? Should I deliver it through Web or Microsoft®? Fortunately some techniques available in SAS® V8 can help us to ease the pain of choosing. This paper shows how to use such techniques like ODS, FTP Access Method, Socket Access Method and DDE to integrate the processing power of SAS on UNIX and the flexibility of SAS on PC for dynamic web publishing.

INTRODUCTION

In our company, we have license for both SAS for PC and SAS for Unix. But unfortunately, we do not have license for SAS/CONNECT® or SAS/SHARE®. Which means that the two sides cannot talk to each other easily. Since our tables are pretty large (Our claims table is about 80 G now), so many times we have to stick with the horsepower of the UNIX server. But you know what, no surprise that all users want their data in Excel format. So, before we can unite together and drive Microsoft to bankrupt, we have to work with it.

PROBLEM

Publish the demographic information of our top 50 contracts to our internal website. Users want the report in a Excel template.

TECHNIQUES**1. FTP ACCESS METHOD**

Allows you to access remote files using the FTP protocol

Syntax

```
FILENAME fileref FTP 'external-file' <ftp-options>;
```

Examples

1: retrieves a directory listing from a host named magsvr2 for user yzhang, and prompts yzhang for a password:

```
filename dir ftp ' ls user='yzhang'
             host='magsvr2' prompt;
```

```
data _null_;
  infile dir;
  input;
  put _INFILE_;
run;
```

2: importing a transport dataset.

```
filename inp ftp
  '/finance/finance/yzhang/demo' user= 'yzhang'
  pass="#####" host='magsvr2';
```

```
proc cimport library=work file=inp;
run;
```

2. SOCKET ACCESS METHOD

Allows you to read from or write to a TCP/IP socket

Syntax

```
FILENAME fileref SOCKET 'hostname:portno'
```

Example: Communicating between Two SAS Applications over a TCP/IP Socket

```
**-- Runs on Unix --**;
Filename local socket ':5478' server reconn
= 0;
  data tcpip;
    infile local eov = v missover;
    input x $ ;
    put x;
    output;
  run;
proc print data = tcpip;
run;
**-- Runs on PC --**
filename remote socket 'magsvr2:5478';

data _null_;
  file remote;
  put '1234';
run;
```

3. DYNAMIC DATA EXCHANGE (DDE)

Dynamic Data Exchange (DDE) is a method of dynamically exchanging information between Windows applications. DDE uses a client/server relationship to enable a client application to request information from a server application. In Version 8, the SAS System is always the client. In this role, the SAS System requests data from server applications, sends data to server applications, or sends commands to server applications.

To use DDE in SAS, issue a FILENAME statement with the following syntax:

```
FILENAME fileref DDE 'DDE-triplet' <DDE-options>
```

Examples

1. Suppose you want to use SAS to populate the first 4 rows and 2 columns of the Microsoft Excel spreadsheet named Sales Data stored in C:\EXCEL\SALES.XLS. You would use the following code

```
filename test dde `excel|c:\excel\
                [Sales.xls]Sales Data!R1C1:R4C2';
```

2: Using DDE and the SYSTEM Topic to Invoke Commands in an Application Using Excel.

```
filename cmds dde `excel|system';
data _null_;
  file cmds;
  put '[SELECT("R1C1:R20C3")]';
  put '[SORT(1,"R1C1",1)]';
  put '[SAVE()]';
  put '[QUIT()]';
run;
```

4. OUTPUT DELIVERY SYSTEM (ODS)

Beginning with Version 7, procedure output became much more flexible. The Output Delivery System (ODS) has been designed to overcome the limitations of traditional SAS output and to make it easy to make new formatting options available to users. ODS is a method of delivering output in a variety of formats and of making the formatted output easy to access.

Example

```

/* Create HTML files. */
ods html file= 'odshtml-body.htm'
      contents = odshtml-contents.htm'
      page = 'odshtml-page.htm'
      frame = 'odshtml-frame.htm';
proc univariate data =statepop mu0= 3.5;
  var citypop_90 noncitypop_90;
  title;
run;
/* Close the HTML destination.          */
ods html close;

```

WRAPPING THEM ALL TOGETHER**❖ UNIX SIDE: SOCKETSERVER.SAS**

Build a Socket server on Unix

```

%macro demo;
/** Set Process Flag **/
...
/** Crunch data **/
...
...
/**-- Write summary to Transport File --**/
/** Reset process flag **/
%mend;

filename local socket ':5478' server reconn = 0;
data ctl;
infile local eov = v missover;
input project $;
if upcase(project) in (&prolist) then do;
call execute('%|| project|:');
end;
else do;
...
end;
run;

/ **Recursive call, keep the server running **/
systask command "sas ./socketserver.sas -log
./socket&sysdate%sysfunc(time())" nowait;

```

❖ PC SIDE : DEMOGRAPHIC.SAS

```

1. Call Unix Server to crunch data
filename remote socket 'magsvr2:5478';

data _null_;
file remote;
put 'demo';
run;

/** Sleep control , wait for Unix process to finish**/
...
...
2. FTP the summarized data from Unix to PC
filename inp ftp '/finance/finance/yzhang/demo' user = 'yzhang'
pass = "#####" host = 'magsvr2';

proc cimport library = work file = inp;
run;
3. Write to Excel
%macro contract(contract);
data a;

```

```

set demo;
where contract = &contract;
run;

/** Open Excel **/
filename cmds dde 'excel|system';
data _null_;
file cmds;
cmd = 'open("c:\data\template.xls")';
put cmd;
run;

/** DDE **/
filename t3 dde 'excel|Demographic!r9c7:r9c7' notab;

data _null_;
file t3;
set a;
put comname;
run;
...
...
/** Save **/
filename cmds dde 'excel|system';
data _null_;
file cmds;
put '[save.as("c:\data\demo\new.xls")]';
run;

systask command "move /y c:\data\demo\new.xls
c:\data\demo\&contract.xls";
%mend;

```

4. Write index.html
options pagesize = 5000;
ods html body = 'c:\data\demo\index.html' nogtitle nogfootnote;

```

data _null_;
file print;
title 'Demographic Report';
set control;
if _n_ = 1 then do;
put @1 'Contract ID' @20 'Company Name' /;
end;
put "<a href = './" contract +(-1) ".xls">" contract
comname"</a>";
run;

```

ods html close;

```

5. Publish
systask command "copy c:\data\demo
\\Nwkfile_oxford\yzhang\demo"

```

CONCLUSION

SAS software provides a rich set of tools for application development. The author hopes this paper will give you an idea how versatile and flexible SAS is. Sky is the limit.

REFERENCES

Vyverman K. "Using Dynamic Data Exchange to Export your SAS data to MS Excel" SUGI 27

SAS Language Reference , V8

ACKNOWLEDGMENTS

The author would like to thank his colleagues of Health Care Economics department, which gave him the opportunity to prepare this paper.

CONTACT INFORMATION

Your comments and questions are valued and encouraged.

Contact the author at:

Yadong Zhang
Oxford Health Plans
48 Monroe Tpk
Trumbull, CT 06611
Email: yzhang@oxhp.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.