

## Paper 065-2011

**Hands Free Data Transfer**

Teo Gamishev, SC Budget and Control Board, Columbia, SC

**ABSTRACT**

The first step in any data management process is transferring the data from the customer. There are many ways to do that and the most common one is to use proper FTP (File Transfer Protocol) software to navigate to the customer's site and get the data. While this process is fun in the beginning of your relationship with the customer, it may become just another boring step in your data management process. There is a way to reduce that step to a single click of the mouse. This article will show you how to use SAS® to transfer data, located on a different type of servers.

**INTRODUCTION**

If you are dealing with multiple projects, requiring downloading data from the customer and uploading back reports on a regular basis, you definitely are looking for ways to automate that step and make it less time consuming for you. Here you'll find several examples of how this could be done. Hopefully, they will work for you or inspire you to do your research and find a proper solution.

**PROBLEMS**

Different data projects are using different servers to store their data. In most of the cases, it is expected from you to be able to access those servers and get whatever you need for the project to be completed. That requires using various FTP applications to access those servers and transfer the data to your machine. Very often the data you need to download happens to be under different directories (folders). Navigating from one directory to another is a hassle. Finally, as any manual process, that creates an environment for errors and mishaps.

**SOLUTIONS**

There is a very simple way to eliminate all these activities and reduce them into a single click of the mouse: Delegate the entire data transfer to SAS. A SAS program can do the access, login, folder navigation and copying the data in the proper folder on your site.

The following 3 examples will give you a hint on how that could be done:

**EXAMPLE #1: Accessing DASD area on a Mainframe**

The data in the DASD area on the Mainframe is normally accessed via Hummingbird FTP. Here is one way to access the data through SAS, without SAS ACCESS:

```
*** 1. Creating a batch file;
data _null_;
file 'C:\Data\Downloads\getDATA.bat' * the batch file;
noprint notitles;
put "ftp -s:%0";
put "goto END";
put "open 144.4.40.44";
put "&tsoID";
put "&tsoPW";
put "CD 'MY.DATA' ";
put "LCD C:\Data\Downloads\MyData";
put "GET FILE138 EMPR";
put "GET FILE150 T1F1";
put "GET FILE152 T1F2";
put "GET FILE153 T1F3";
put "GET FILE157 T1F4";
```

```

put "bye";
put ":END";
run;

*** 2. Running the batch file to FTP the new data from the Mainframe;
options xwait xsync; * synchronizing the downloading process and SAS;
x " C:\Data\Downloads\getDATA.bat ";
*MONITOR THE DOWNLOADING THROUGH THE DOS PROMPT WINDOW POPPING UP *;

*** 3. Deleting the batch file;
Filename delit PIPE "del C:\Data\Downloads\getDATA.bat";
Data _null_; Infile delit; Run;

```

#### EXPLANATION NOTES

A batch file is created in the Step #1. The user ID and the password are passed to the server as macro variables. Those macro variables should be created in your AUTOEXEC.sas file or the access to the SAS program should be limited to you, only. The CD(Change Directory) command is navigating to the data folder on the Mainframe. The LCD (Local Change Directory) command is pointing to your destination folder.

"GET FILE150 T1F1" copies FILE150 from the Mainframe, renames it to T1F1 and saves it into the local data folder. Once the batch file is created, SAS can activate the file using X command (Step #2). The option XWAIT forces SAS to wait for you to type EXIT on the COMMAND window, before moving to the next step. I like to watch the COMMAND window, which pops up after the X command. When sure that everything is executed properly, I will type EXIT on the Command window and then the SAS program will move to the next step. Automation is good, however quality control needs to be applied on each step to make sure that nothing is messed up. Here comes the role of the human eye to verify that and then move forward.

The next option XSYNC causes the operating environment command to run synchronously with your SAS session. That is, control is not returned to the SAS System until the command has completed.

In step #3 the batch file is deleted using the Filename PIPE statement. Considering the fact that the batch file contains a sensitive data like your login ID and Password, this step is mandatory one.

#### EXAMPLE #2: Accesing a Secure FTP server via psftp

```

*** 1. Creating a text file with all commands;
data _null_;
file "C:\Data\UploadReports.txt"
noprint notitles;
* uploading reports and data for Client1;
put "cd /chroot/home/client1/Reports";
put "del *.*";
put "lcd C:\Reports\StatPack";
put "mput *.*";
put "cd /chroot/home/client1/Data";
put "del *.zip";
put "lcd C:\Zipfiles";
put "mput *.zip";
* uploading reports and data for client2;
put "cd /chroot/client2/Reports";
put "del *.*";
put "lcd C:\Reports\StatPack";
put "mput *.*";
run;

*** 2. Executing the commands above;
options xwait xsync; * synchronizing the downloading process and SAS steps;
x "C:\sshtools\psftp.exe -P 2251 -b C:\Data\ UploadReports.txt
-l &sftpID -pw &sftpPW courier.state.gov";

```

```

*** 3. Deleting the text file;
Filename delit PIPE "del C:\Data\Uploads\UploadReports.txt";
Data _null_; Infile delit; Run;

```

#### EXPLANATION NOTES:

The steps here are very similar to the steps in the Example#1. The Step 1 we are creating a text file, containing all commands the software need to execute. In this case we are uploading the final reports to a secure server. In case of several clients receiving different set of reports, that example could be very helpful.

In Step 2, SAS is triggering the PSFTP.exe file via X command. The PSFTP software should be installed in advance. It is free application for secure file transfer, available on Internet. The X command contains the path to the PSFTP.exe and the name of the server to connect to. The following options are needed to complete the task:

- P: specify a port number
- l: specify the user name to log in – in this case the log in is in a macro variable
- pw: specify a password – stored in a macro variable, also
- b: display batch commands as they are run – in this case they are stored in the text file, created in Step 1

The macro variables, containing the user ID and the password can be created within the AUTOEXEC.sas to avoid hard coding them in the program.

Step 3 deletes the text file with the batch commands to clear the space.

#### EXAMPLE #3: Accessing WS server:

```

*** transferring the data;
options xwait xsync; * synchronizing downloading and SAS steps;
X "c:\progra~1\coreftp\coreftp.exe -site CAPS -d /*.txt -p D:\data\capss";

```

#### EXPLANATION NOTES:

This is the shortest example of how to access a WS server, using Core FTP Light free client software for windows. In this case Core FTP has to be installed and a session, containing all prerequisites to connect to the server (user ID, password, port number), has to be established and saved using a particular name. In our example that session is named CAPS. The X command is triggering the coreftp.exe and the following options are in use:

- site: site session/profile name as stored in the Core FTP - CAPS
- d: remote directory to download files from – in this case we are downloading all test files - /\*.txt
- p: destination directory - D:\data\capss

If you need to upload data or reports, then use the following example:

```

*** transferring the data;
options xwait xsync; * synchronizing downloading and SAS steps;
X "c:\progra~1\coreftp\coreftp.exe -site mysite -u /upload/data.txt -p
/remote/store";

```

Other available options are:

- ZIP: zipping the file(s)
- AUTONAME: naming the files after zipping
- output and -log parameter are for information about transfers
- s: silent mode – the application window doesn't open
- o: overwrite if file exists
- oa: overwrite \*all\* if transferring a folder or directory of files.
- on: overwrite files that are newer than existing files.
- og: overwrite destination file if source file size is greater.
- ol: overwrite destination file if source file size is less (smaller).
- os: skip file if exists
- r: resume if file exists
- ra: resume \*all\* files that exist.

Here is an example of zipping the files before uploading them:

```

X "c:\progra~1\coreftp\coreftp.exe -O -ZIP -AUTONAME -u c:\IBM -site mysite
-p /public_html/store/ -output c:\temp\output.log -log d:\temp\log.log -s";

```

-output and -log parameter are for information about transfers. They are optional but listed for informational purposes.

**AUTOMATING THE ENTIRE PROCESS** is a preferred way to go when dealing with repetitive tasks. In this case the entire downloading and uploading programs can be saved and called from within another DRIVING program. Here is an example of the triggering code in the DRIVING program:

EXAMPLE#4: Downloading Driving program:

```
*****;
*** STEP ONE ***;
*** DOWNLOADING MONTHLY EXTRACTS ***;
*****;
**** MAKE SURE THAT THE ID AND PASSWORD ARE SUBMITTED FIRST ***;
%include "C:\Programs\getData.sas";
*** MONITOR THE DOWNLOADING THROUGH THE DOS PROMPT WINDOW POPPING UP ***;
*** CHECK THE FILE SIZE ***;
```

#### EXPLANATION NOTES:

The first step in each data project is to get the data from the customer. Once the SAS program dealing with that process is ready and tested for proper work, it can be saved and called from within another DRIVING SAS program. All the programmer needs to do is to select and submit the part of the code, triggering the data downloading process. Now the entire focus is on the quality and all the tedious typing and coding is removed. The best part is that the room for coding errors is practically zero.

EXAMPLE#5: Uploading Driving program:

```
*****;
*** STEP FIVE ***;
*** UPLOADING THE REPORTS AND THE SAS TABLES INTO SFTP Server ***;
*****;
%include "C:\Programs\UploadingReports.sas";
*** CHECK ALL NEW FILES ARE UPLOADED IN THEIR RELEVANT FOLDERS ***;
```

#### EXPLANATION NOTES:

Once we are done with the data manipulations and reports are ready to upload to different customers, the whole task can be triggered again from within the DRIVING program. Select and submit that part of the code invoking the uploading program, dealing with reports and data distribution and focus on quality control. No mishaps, no errors, no boring job to do. Enjoy the power of automation and focus on quality control.

## CONCLUSION

This paper offers one of the many possible solutions on how to remove the repetitive downloading and uploading tasks by using the power of SAS. The above examples are giving hints on how to deal with different servers. In addition examples are given on how the whole process can be automated. It can be very practical and time saving to implement those examples in the daily routine.

## REFERENCES

SAS Institute Inc. 2002-2008 "SAS OnlineDoc® 9.1.3" Available at: <http://support.sas.com/onlinedoc/913>  
 Gary Sargent 2006 "Core FTP LE Client Instructions"  
 Available at: <http://www.ic.nhs.uk/webfiles/Services/Datasets/MHMDS/MHBScoreftp.doc>  
 Simon Tatham 2001 "PuTTY Manual"  
 Available at: <http://the.earth.li/~sgtatham/putty/0.52/htmldoc/Chapter6.html>

## ACKNOWLEDGMENTS

Special heartfelt thanks to Jim Moore, IT Technician at BCB-ORS for his helpful hints and guidance.

## CONTACT INFORMATION

Name:	Teo Gamishev, M.S. in Statistics
Enterprise:	SC Budget and Control Board, Office of Research and Statistics
Address:	1919 Blanding Street
City, State ZIP:	Columbia, SC 29201
Work Phone:	803-898-9945
Fax:	803-898-9972
E-mail:	<a href="mailto:gamishev@hotmail.com">gamishev@hotmail.com</a>
Web:	<a href="http://sasarticles.blogspot.com/">http://sasarticles.blogspot.com/</a>

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.