Domo Arigato SAS®: Turn Your PC Into A BASE SAS® Robot

Michael McCarthy PE, Austin Energy, Austin TX USA

ABSTRACT

Data can take a long and winding path before graduating to become a SAS data set. Using the native OS can frequently be an easier way to navigate this journey. This paper is written for BASE SAS® programmers who want to control their PC from the DOS command prompt using the "CALL SYSTEM ()" command. A method to command DOS from a SAS script is discussed, and a step-by-step example demonstrating file download from a web server and subsequent file operations from a BASE SAS® macro is presented.

INTRODUCTION

The CALL SYSTEM command can send virtually any instruction to the command line. While this could be familiar DOS file commands such as zip, unzip, move, or copy, the palette is not limited to only these. A SAS program can use this command to open and run any program on the host computer, passing parameters as needed. In other words, if you can find the executable file on your OS, then SAS can run it using the parameters you choose. Often, this can be less complicated than trying to perform a similar process from the SAS environment alone.

This flexibility is particularly useful when accessing a web server via https. Note that https sites typically use the SSL (Secure Socket Layer) protocol to encrypt the transaction between your computer and the server, and this protocol requires passing login and certificate credentials. Navigating a file server connection can be much less complicated from a browser, such as Chrome, than the native SAS environment. In particular, Chrome was specifically designed to seamlessly manage the public encryption key, whereas SAS was not. But this is not really a problem since SAS can easily send any number of file requests from the browser search window via the command prompt, thereby turning the browser into a SAS controlled robot.

METHODOLOGY

Since the punctuation can be tricky, it's best to leverage the CALL SYSTEM command by defining each instruction as a string. Storing the instructions as strings helps define the proper punctuation, enables us to utilize a variety of SAS string commands, and makes a repeated web server query very convenient with macro variables. For example, to open a Chrome browser on the Google search page, we would define the string OPENIT='start chrome "www.google.com"', and then use the command CALL SYSTEM (OPENIT) to send this to the command prompt.

Instructions can get even more specific when we open a program from its location on the OS drive. For example, we can open Chrome from DOS and point it at a file on the web server as follows. In this case the file name is “2017.01.01”:

```
data _null_;
call system('"C:\Program Files (x86)\Google\Chrome\Application\Chrome.exe"
https://WEBSERVER/FILENAME?NAME='||"2017.01.01");
run;
```

As long as we’ve already logged onto the https site through Chrome, we will be able to download the file. If we replace `"2017.01.01"` with a macro variable, then the script above will download any file we select from the web server. Furthermore, once downloaded SAS can tell the OS to ZIP, UNZIP or MOVE the files as needed.

BASE SAS®

The BASE SAS® code presented in the Appendix shows how a macro called “SASDOS”, can be used to repeatedly download a zip archive from an https web server. To run this code, first open Chrome and login to establish the https connection to the remote web server. In BASE SAS®, the “CALL EXECUTE” command (step 10) will send the date string &GETDATE to SASDOS (steps 1-8), which will define DOS instructions as strings (steps 3-6), and then send each to DOS using the “CALL SYSTEM” command. The script utilizes the command “options noxwait xsync” (step 2) to close the xwindows session when done. Since the downloaded files may be large, it’s handy to use the SAS command “wait = sleep(60,1)” (step 7) to pause the script while the download completes. Once completed, the downloaded file can be moved to the archive folder with the command MOVEIT (step 5). Then the command CLOSEIT (step 4) will close all of the open Chrome sessions.

CONCLUSION

The BASE SAS® code presented in this paper demonstrates the use of the “CALL SYSTEM” command to send instructions to a DOS command prompt, controlling programs and performing file operations according to a SAS script. As shown, a SAS macro can repeatedly open a Chrome browser and access a secure web server to download and process files. This strategy makes for a very economical combination of the native OS, installed programs, and SAS since each is doing what it does best. Domo Arigato SAS for turning my PC into a robot.

PERMISSION TO PUBLISH

The authors of the paper/presentation have prepared these works in the scope of their employment with AUSTIN ENERGY and the copyrights to these works are held by AUSTIN ENERGY.

Therefore, AUSTIN ENERGY hereby grants to SCSUG Inc. a non-exclusive right in the copyright of the work to the SCSUG Inc. to publish the work in the Publication, in all media, effective if and when the work is accepted for publication by SCSUG Inc.

This the 7th day of September, 2017.

ACKNOWLEDGEMENTS

Special thanks to John Trowbridge, PE (DABI Engineering Supervisor, Austin Energy), Liz Jambor, DABI Manager, Austin Energy), and Debbie Kimberly (VP Customer Energy Solutions, Austin Energy).

REFERENCES

Working Outside the SAS® Box: Calling Other Applications from Within SAS
James A. Zeitler, Harvard Business School, Boston, MA
http://www.lexjansen.com/nesug/nesug06/cc/cc01.pdf

CONTACT INFORMATION

I value your comments. You can contact the author at:

Michael McCarthy, PE
Austin Energy, Customer Energy Solutions, Austin TX USA
michael.mccarthy@austinenergy.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.

APPENDIX – SASDOS MACRO

```
/*************************************************************************************/
/* DATA SASDOS SCRIPT                                                                */
/* WRITTEN BY: MMCCARTHY, PE                                                         */
/* DESCRIPTION: USE THE CALL SYSTEM () FUNCTION TO DOWNLOAD                          */
/* AND ARCHIVE WEB SERVER FROM A CHROME BROWSER                                      */
/*************************************************************************************/
/*STEP*/
/*1 */   %macro SASDOS(GETDATE); /*PROCESS WEB SERVER FILE BY DATE*/
/*2 */   options noxwait xsync; /*CLOSE THE XWINDOWS SESSIONS WHEN DONE*/
          data _null_;
/* LIST OF COMMAND STRINGS*/
/*3 */   OPENIT = 'start chrome "www.google.com"'; /*OPEN CHROME EXAMPLE-NOT CALLED*/
/*CLOSE CHROME – CLOSES ALL CHROME SESSIONS*/
/*4 */   CLOSEIT = 'TASKKILL /PID Chrome.exe';
/*MOVE FILE – ALL ZIPPED FILES MOVE TO ARCHIVE*/
/*5 */   MOVEIT = 'move /Y C:\Users\USERNAME HERE\Downloads\*.zip C:\YOUR DIRECTORY HERE';
/*GET FILE FROM WEB SERVER*/
/*6 */   GETIT = '"C:\Program Files (x86)\Google\Chrome\Application\Chrome.exe"
https://WEBSERVER/FILENAME?NAME='||"2017.01.01");
/*SEQUENCE OF COMMANDS-CALL ANY DEFINED COMMANDS ABOVE*/
          call system(GETIT); /*CALL THE WEB SERVICE AND DOWNLOAD A FILE*/
/*7 */   wait = sleep(60,1);  /*SAS COMMAND TO WAIT 60 SEC FOR DOWNLOAD TO COMPLETE*/
          call system(MOVEIT); /*MOVE THE FILE TO ANOTHER DIRECTORY*/
          call system(CLOSEIT);/*CLOSE GOOGLE*/
          run;
/*8 */ %mend SASDOS;

/*9 */   Data _null_;
          SDATE = mdy(9,1,2017); EDATE = mdy(9,1,2017); /*RANGE OF VALUES*/
          do DATE = SDATE to EDATE; /*REPEATEDLY CALL MACRO WITH DATES*/
          DateStamp = cats(Year(Date),'.',Month(Date),'.',Day(Date));
/*ADJUST DATE FORMAT TO MEET WEB SERVICE CALL FORMAT STRING*/
          if Day(Date) LT 10 then DateStamp = cats(Year(Date),'.',Month(Date),'.0',Day(Date));
          if Month(Date) LT 10 then DateStamp = cats(Year(Date),'.0',Month(Date),'.',Day(Date));
          if Month(Date) LT 10 and Day(Date) LT 10 then DateStamp =
          cats(Year(Date),'.0',Month(Date),'.0',Day(Date));
/*CALL MACRO SASDOS-FILENAMES ARE ACCESSED BY DATE IN THIS EXAMPLE*/
/*10 */ Call EXECUTE ('%SASDOS(GETDATE='||DateStamp||')');
          end;
          run;
```