

**An example of ODS, FTP, PROC FORMAT with HTML linked to EXCEL, then the output is
Emailed via PCSAS. PS its automated.
Zeke Torres, 1016inc.com, Chicago, IL**

ABSTRACT

I will review a variety of popular SAS components. The concept is to obtain SAS Data (transport) file from a remote host. Use FTP (not SAS/Connect). Perform a PROC TABULATE or PROC PRINT etc. I get a little fancy with ODS and PROC FORMAT adding HTML to it allowing links to the Excel output which is also generated from this code. Finally I have my PCSAS Session email all of the output to "end users" and then PAGE myself with a message telling me it all worked. All from my PC SAS Session. And its automated using MS Win98 Schedule Task Manager. So when I am having breakfast at home my PC at work is actually working! When I finally do get to the office I have a custom SASLOG I can review since I set up the SAS CONFIG file to be specific for this code.

INTRODUCTION

You will benefit from this example if you have a PC SAS session and need to get 'data' from a host machine daily, weekly etc then run a 'static' report against it. You can use this to automate these tasks giving you the freedom to do other things. This process is automated from the PC side. I will show you how you can use the MS Win98 Schedule Task software to accomplish this. You will see an example of how you can send the "final output" via email using PCSAS. I provide a few examples of how you can send quick messages to a PDA or TEXT capable pager that the report 'FINISHED'.

In this example I employ a PROC FORMAT concept that incorporates HTML and gives the finished product a very good look and feel of "drill down" ability. You'll be able to take this example further and get some very good reports and HTML output easily with PROC FORMAT from now on. The output generated by ODS is not only in HTML but I create a printable PDF page for my end users. I'll go over a concept I started using to make it easy for the user to not only view the ODS HTML but have choices, ie PDF output, for print. I also set this program up to generate Microsoft Excel workbooks and they are linked to the HTML page via the links in the PROC FORMAT I mentioned earlier.

Finally since this is automated and in case of problems you might need to review the SAS LOG, I present an example of how to use the SAS CONFIG file with SAS.EXE upon initiation. You'll learn some basic ways to accomplish this on your PC SAS session and be able to have 'custom' SAS sessions.

INTENDED AUDIENCE

This paper is intended for SAS users who have a technical skill level that begins at the intermediate level thru advanced. The author has made every attempt to keep the examples of code focused around Base SAS. The use of supplemental software or other languages are general and it is assumed that the audience will obtain sufficient information via this abstract to adequately manage changes required by their own scenario(s). While some parts of this code are specific to V8.2 (ODS, EMAIL) the concept and idea can be used with earlier versions of SAS.

ABOUT THE AUTHOR

Ezequiel 'Zeke' Torres is the President of 1016 Incorporated located in Chicago Illinois. With 10 years of SAS experience he is not just a SAS junkie but a "sas-aholic". Some of the clients that have benefited from his addiction to SAS are: Allstate Insurance,

GE, Transamerica Distribution Finance, AEIG, Trustmark Insurance. Zeke also worked as an employee of Household International. Zeke develops end-to-end solutions using SAS for projects and business information needs clients present. This only feeds his addiction to SAS because creative solutions are not a choice, but a requirement.

PC AND HOST INFORMATION

In this example the HOST is an IBM os/390 MVS mainframe. PC is operating on a WIN98 machine. FTP has been used via the DOS or command prompt on the PC. SAS on the PC is version 8.2. SAS on the HOST is not required for this example or your site except if you need to have SAS datasets created there for you to pick up. Email software in this case is MS Exchange. The 'automation' was accomplished using MS Win98 Schedule Task Manager. This software is generally found on standard MS Windows operating systems. The author will present alternatives to this automation but not discuss them in detail. A basic example of how to set up a 'task' to run according to your needs is provided. The author will mention the use of WINZIP in this abstract and presentation. WinZip is not required. The author has made every attempt to employ supplemental software tools that are in his opinion reliable, easily obtainable, and for practical purposes affordable i.e.: Shareware, Freeware or minimal cost.

SECURITY AND PROTOCOL

The password and userid in this example are created specifically for this process. Consult with your site administrator on the security requirements you need to consider when exposing your information in this way. Do not deploy this code or any similar concept in your organization, pc or site without consulting with your manager and or security administrator first. Email in this example was available but not all sites will allow you to automate this feature when you are not present or physically there and or in consideration of what protocols your site might find important. *See WARNING

SPACE AND MEMORY REQUIREMENTS

Consider space requirements on either the HOST or your PC. You might not have enough hard drive space for the data you wish to pick up from the HOST. Sending large datasets can cause problems if your machine is running low on space. Even if you get the data on your PC, the RAM or memory may not be enough to sufficiently process the SAS code or programs you create. If you do not carefully test and calculate what your PC is capable of sustaining this concept can cause your PC to fail or at a minimum operate inadequately. The FTP process can be a security risk depending on the standards established by your site. Email and the use of email via SAS should be used with caution. Remember the person you email may not have enough space for an output file you send. There may also be size restrictions on attachment files via email and or restrictions on the number of files per email.

MESSAGE TO PDA, PAGER MOBILE DEVICE

This paper will document how using PC SAS to email to a PDA or TEXT Message capable mobile device such as a PDA, Pager or Phone can work. This will cover a one-way message to that device. Your site, program, circumstances and results may vary according to your service provider and/or mobile device, which may or may not include such additional considerations such as the software on the device. Finally your service provider and service plan may not be easily compatible to this process and/or the use of 'messages' via mobile devices might require additional features in your plan. Some service providers have specific rules, policies regarding what kind of transmission is allowed on their networks, systems consult your service provider first. *NOTE: This paper will not discuss how to interactively work with

reports between the device and your PC. The author uses the features of PCSAS email to send a message via a TEXT message pager. The author will cover related concepts and tips he has during the actual presentation that may interest various levels of users. You also need to consider security issues when sending any information to a mobile device.

OUTLINE OF THE PROCESS

PURPOSE OF PROGRAM

I created this program for a client who needs the output daily. The data originates on the host. It is created or prepared during nightly batch processing and resides in a SAS Data Library and a copy in SAS Transport format. The user who requested this specified that the output would be sent to multiple people via email and likely distributed to more people. The output should provide "totals" and somehow for users to obtain the detail of those totals as they might need to see it. The site prohibited EMAIL from the mainframe. SAS/Connect was not an option.

INVENTORY OF CODE OR FILES USED

LIST OF CODE FOUND IN THIS EXAMPLE.

- SASV8X.CFG Custom config file.
- X1016.SAS SAS Code
- X1016CONFIG.SAS SAS Code
- X1016FTP.INI FTP Syntax
- XSTART.BAT DOS Batch File
- macroexportemew.sas Macro to EXPORT to EXCEL *
- SASLIB.DAT SAS Transport file
- X1016ZIP.LST Syntax for WINZIP **

* Note:

macroexportemew.sas This code was obtained from: <http://www.sas.com/service/techsup/unotes/SN/006/006695.html>
Modifications were made to suffice the desired results for this code. Changes or modifications may be required for your site, pc, station. 1016 Incorporated did not author this code.

** Note:

This abstract will not cover this line of code. The author will cover this in actual presentations. WINZIP is required for advanced or additional EMAIL options. This example or code does not require WINZIP. The basic EMAIL features are documented. The code presented here will applicable line of code is commented out.

THE CODE

In this example we have the pc set up with the following file structure: There is a main folder: c:\SUG\

This folder contains 3 folders the entire path for those are:

- c:\SUG\OUT
- c:\SUG\RUNTIME
- c:\SUG\TEMP

In the main folder (c:\SUG\) we have the following files:

- macroexportemew.sas
- saslib.dat
- sasv8x.cfg
- x1016.sas
- x1016config.sas
- x1016ftp.ini
- x1016zip.lst
- xstart.bat
- sug_zt1016b.doc * Word Version of presentation
- sug_zt1016b.txt * Text Version of presentation

XSTART.BAT

```
SAS.EXE
-sysin c:\sug\x1016.sas
-config c:\sug\sasv8x.cfg
```

* end of xstart.bat;

This will actually begin the process of executing the SAS session and code. It also calls the specific SAS CONFIG file we want to use for this program or session. By using this file with an icon or as we will see with the schedule software with windows it allows us to deploy quickly and easily different variations of the PC SAS system we might use. One might execute specific SAS code to generate a report or data. Another might have SAS code and a special CONFIG file which sends output to a specific folder and adjusts memory and SAS lib's and Catalog also. Say a "testing" vs "production" CONFIG file. Where one is for TEST data and the other is actually using "live" data. Yet the code we'll use and work on will obviously be swappable to either session. Let's look at the CONFIG file a little. But lets recap. The SAS.EXE file will start. It will use the CONFIG file we have specified and then initiate the SYSIN file we have defined.

C:\SUG\SASV8X.CFG

```
-compress=yes
-nocenter
-fontsloc "lsasroot\core\resource"
-TRAINLOC ""
-AUTOEXEC "c:\sug\x1016config.sas"
-ALTLOG "c:\sug\runtime\x1016log.log"
-configure 3m
```

* end sasv8x.cfg;

This is a very powerful feature and file once you understand how and what you need to do with it. SAS uses this file to establish the default options your session will have. We are all familiar with OPTIONS COMPRESS=YES NOCENTER; and many others. Well in the CONFIG file we have the ability to initialize the session with all of the options we want. We generally set options at the start of our code or program. The idea of customizing your CONFIG file is the same but at a much earlier stage. Your options are many obviously. This CONFIG file is allows you much more options to change or modify since it happens when SAS starts. The options statement in your code or the SAS system options window will have more options closed or reserved from changes once the session is running.

Basically with this CONFIG file you can modify many more options. Lets look closer at our example again: In XSTART.BAT we are executing SAS.EXE but we are overriding the SYSIN value in the CONFIG FILE (either default config file or our sasv8x.cfg). And we are overriding our CONFIG default for the SAS.EXE file. It would normally go after the first CONFIG file it will see. We are now specifying where the file is we want the session to use.

So why use this OPTIONS, CONFIG part? Well once your session is running your options for changing the default OPTOINS are fewer when using the SAS system options window or OPTIONS statements in your SAS code. There are more OPTIONS you can change at the initialization of the session than once it is started and you start using it.

I will list a few options that might interest you but in the reference section I document where a more detailed list of OPTIONS are found. I hope someone or to myself write a paper on all of the various features the CONFIG file in windows can offer users. I'll

also cover how to add an icon to your SAS session (REGISTER option!). **WARNING:** Make a back up copy of this file before you start to use it. I suggest making 2 copies and making sure you put one on a floppy disk should your 'testing' prove frustrating and you need to revert to the original. These are just a few of many options you can work with and determine if your situation calls for using it differently or basically USING it!

- **ALTLOG** - Send the SAS Log to a specified file on your pc. Remember a valid PC path is required.
- **COMPRESS** - Are you going to compress your 'work' environment SAS Session to try and save space or not. YES or NO?
- **EMAILSYS** - MAPI or VIM well discuss this a little later. The default is MAPI. Let's leave this for now.
- **EMAILID** - Yes your email id. Starting to get the picture here? Well with this little OPTION you can get some pretty big results. We'll get to that part in a minute.
- **EMAILPW** - Same as EMAILID. As with any kind of security protocols at any organization consideration must be given to what were doing. Ultimately it's up to you if you are going to hard code your email id and password on a file on your pc. CONFIG file or not SAS or otherwise it is something you should consider a privacy or security risk.
- **LOG** - The file your SAS log will write to. See ALTLOG.
- **NEWS** - Yes that "NEWS" file that is basically old news! We'll put a valid path and point to some NEWS for god sake or make your own text. Its up to you.
- **PRINT** - Yes you can set the PRINT output to go to a specific *.lst file.
- **REGISTER** - Did you ever want an "ICON" in the menu bar in your PC SAS session? Thank SAS for thinking ahead as their standard operating procedures dictate. I'll cover how to quickly add an icon you can later customize.
- **SASAUTOS** - OK this one is as beneficial to you as all the others I'm presenting. But here you can specify which MACROS you want to have up and running for this session.
- **SORTSIZE** - Yes you can bump up that amount of space you have for sorts. But careful it is important to consider how much your pc can handle. Only bump up a few MB at a time say 1mb then 2mb. The SAS book (see reference) recommends a value of between 2 and 8mb. But you have 128 mb of ram right? Or 512mb? Or a 2 gig hard drive? Sure show off, but remember you have other windows applications running and they need space too. Of course you need those applications running since you're a very productive person and while your pc SAS session is running your working on some other important project.
- **SYSPRINT** - Which printer did you want that output sent to? Yep you guessed it. Put a valid printer name and path and your users in a remote location will be really impressed right!
- **SPLASH** - Ok I must stress that I swear by SAS and all of my friends, family, clients abruptly stop me from getting on the SAS sermon when they see its coming. Sometimes (few) you just do not need to see the SAS banner at the start of the session, right? Say with a program running in 'batch', why have the 'splash screen' your not going to see it.
- **SYSIN** - This points to the code that will be executed by SAS.EXE.

Ok these are but a very few of the options you can switch or change for your session. Please review them in more detail and see which settings are best for you. Again with all of the different combinations and choices along with your PC hardware, LAN, WAN, Email system and other software applications you can

really set up some interesting sessions to make your work or day easier.

But ok you forgot to ask the most important thing, where is the SAS CONFIG file for you to MODIFY in the first place? Well try looking in the folder or directory where the SAS System is installed in. For Windows installations try looking in:

/PROGRAM FILES/SAS Institute/SAS/V8/

However you might have a different SAS version. You may also have a different installation method at your site. Generally the CONFIG file would be found in the same folder as the SAS.EXE file on your machine. Try looking for this via the FIND FILES feature on your operating system. But try to find the SAS.EXE file and then look in that folder for the CONFIG file. The various versions of SAS have different (slightly) naming conventions for the CONFIG files. Someone at your site or pc might have already modified the file or file name so it could already have a different name.

C:\SUGX1016CONFIG.SAS

```
options compress=yes noxwait nonotes;
x 'ftp -v -s:c:\sug\x1016ftp.ini trfs';

options notes;
%include "c:\sug\macroexporternew.sas";
filename input1 'c:\sug\saslib.dat';

proc cimport infile=input1 lib=work;
run;
```

* end x1016onfig.sas;

In our example we are setting options to suite the purpose of the code were running. Here I have specifically done this outside of our custom SAS CONFIG file (sasv8x.cfg). I have chosen to consider it part of the CONFIG file by way of the SYSIN statement.

Once again what has happened in our example so far. We started XSTART.BAT. That starts the SAS Session, uses a custom config file (sasv8x.cfg). That config file executes this code (x1016config.sas). The config file also picks up the filename for the SYSIN option from the XSTART.BAT file. It will tell the SAS.EXE (session) to run that after the CONFIG file is done processing. Well we just told the CONFIG file to process our code to set some preliminary options and execute a set of FTP statements via the X command prompt in SAS itself. This means that once the FTP is done our data will be ready for the SAS code we are pointing to in the SYSIN statement. Hey we just eliminated a manual process! By the time our PC SAS Session is ready for us to use we've picked up the data we need to work with!

C:\SUGX1016FTP.INI

```
useridxx
userpass
binary
get 'hlq.sasdata.port' c:\sug\saslib.dat
bve
```

* end x1016ftp.ini;

This is a simple FTP script. You'll need to enter your userid, password for the host. I specified binary since I am picking up a SAS transport file from the mainframe. Again use caution with userid and password left on your pc. Please note that the file we are 'getting' is already prepared as a SAS Transport file. It is generated by the batch process in this environment or example. If you needed to pick up a flat file or some other file you would need to use almost the same code but maybe not the BINARY

command. Say it was a flat file. So you pick it up get it to your PC and then use INFILE statements etc.

Now lets get to the SAS Session. So we can get started with the code. We've initiated the SAS System on our PC, established a custom CONFIG file setting, picked up data, and now were going to execute the code we wanted to run.

Ok we just covered the initialization and execution of the code we want to run. We still did not cover the automation. Lets cover that before we go into my example of SAS code.

AUTOMATE WITH WINDOWS SCHEDULE

I have to make this it's own section. In my situation I needed the ability to execute my SAS program on a routine basis. I found my windows software had a Schedule Task Manager Program. After working with it a little I found it to be sufficient for programs to be initiated and manageable. I have since had a few associates mention other scheduling software or 'batch' software. I would suggest that you try what you have and see how you like it. I'll go over a few ways to set up a job and let you decide. In the reference section I list some other software names and web links lets also remember that the SAS Institute does have its own set of solutions.

BEFORE WE START

Configure the Task Manager (JOB). Ok there are two approaches I'll document on how to set up the Job or Task via the wizard. I will highlight how I prefer to do it in the applicable steps that are different. In either way you are going to need to have a few things ready. 1st the path to the SAS.EXE file on your computer. You'll need to know the exact directory path. Here is an example which might be exactly the same or similar on your pc:

`C:\Program Files\SAS Institute\SAS\SV8`

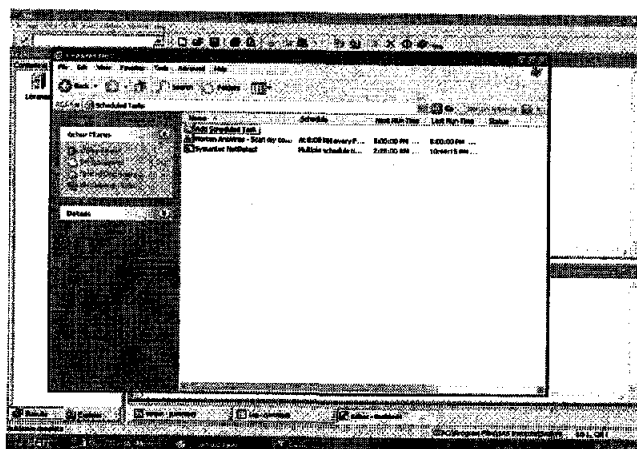
In some cases your windows system may require it be typed like this next example:

`C:\PROGRA~1\SASINS~1\SAS\SV8`

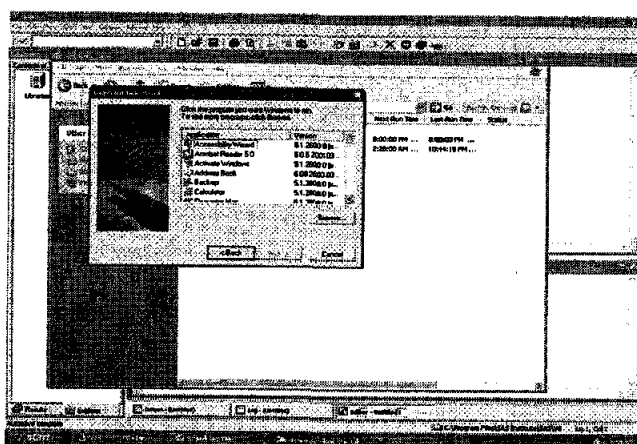
Both of these 'path' examples are essentially the same. However the windows system and/or the application requires the 'short' name version of the path.

Make sure you have the exact path! We want to make sure we are on the right track. Now lets find the Scheduled Tasks application on your machine. It takes a little looking for via the START button from Windows. I've found it in the Accessories group many times. Another SAS user told me he found it in the Administrator tools. In XP I found it in the Control Panel Items. The name is the same Schedule Tasks. Once you find it lets go over these screens. It's pretty easy. The screen prints presented here are from Windows XP.

Click on: Add Scheduled Tasks



Now you will see a "Wizard" which will help you thru the process. XP and Windows 98 are slightly different. But the concept is the same. You are presented with this next screen in both versions.

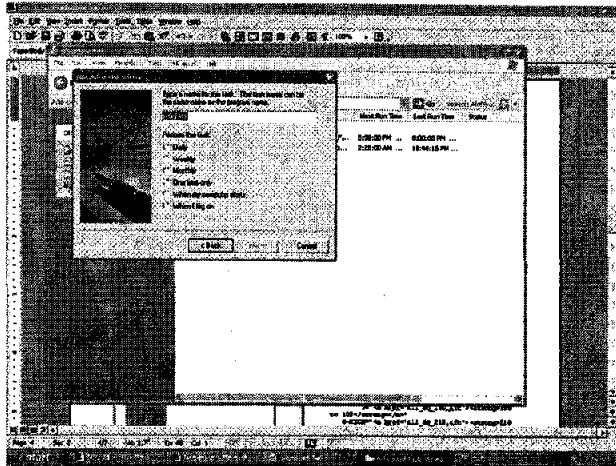


You will be presented with a choice of applications on your pc that the wizard will allow you to "schedule". Depending on how SAS was installed on your PC you will even find the SAS application. Lets pick the SAS Application choice.

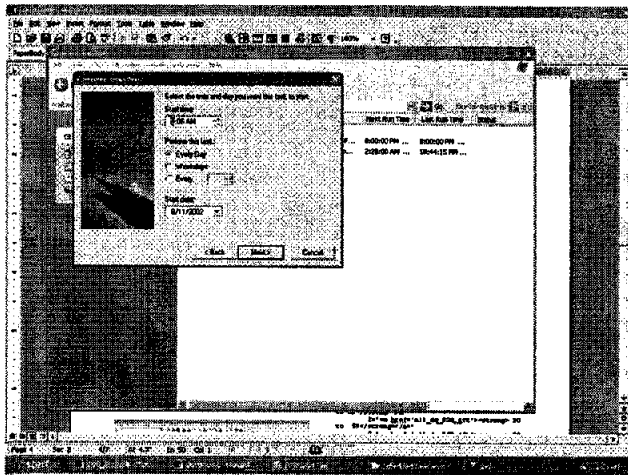
Here is where I like to do things slightly different. I use the Browse button to find the "folder" which has my XSTART.BAT file. I make sure the rest of the process this wizard might over look is managed via that BAT file. Benefits with this way are that you can modify the BAT file much quicker and easier than say going back to the SCHEDULED TASK process were covering now.

By using the applications the wizard presents you please pay attention to which SAS application you select. Sometimes you'll need to carefully read all of the details presented in this box to make sure you are selecting the correct SAS - Application. You may have choices that appear at first hand to be the "SAS Application" but it could be the 'SAS System Viewer' or another SAS component. If you do not find 'SAS' or cannot tell which application is SAS keep reading we'll make sure its resolved. Let's click Next and move on.

Now we get a box that gives us a choice about when this task will run. It is up to you. You are also able to modify the 'title' of the task.



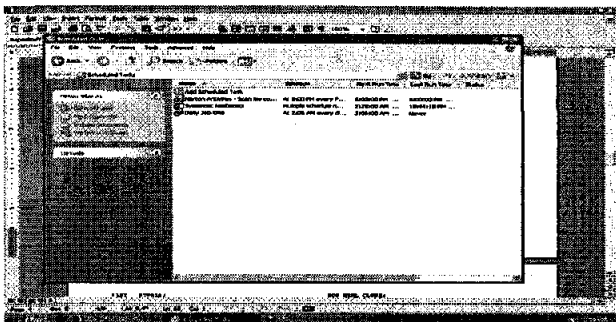
Lets change the title to "daily job One" and set it to run daily.



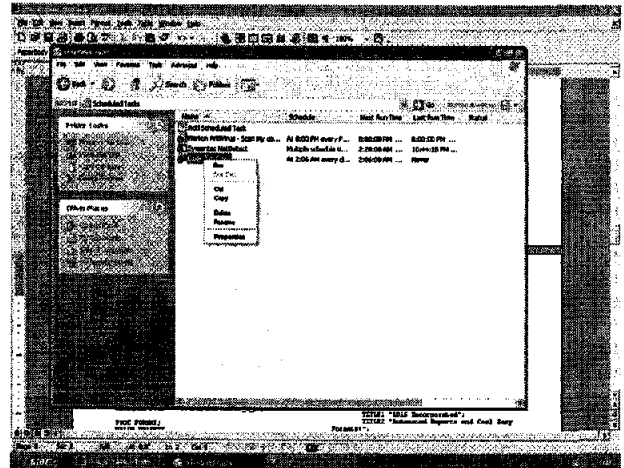
The next choice you have is over what time and other details pertinent to the 'daily' choice. I will not cover all the other choices with weekly, monthly, one time. But you'll find there are many variations that are easily set and established which will allow you to get the right one for you.

With Windows XP you are going to have one more choice. I'll mention it but not cover it since not all users have XP. The next screen allows you to enter the user name, password, id etc required by the operating system. When this job runs it will be as if this 'user' is at the control.

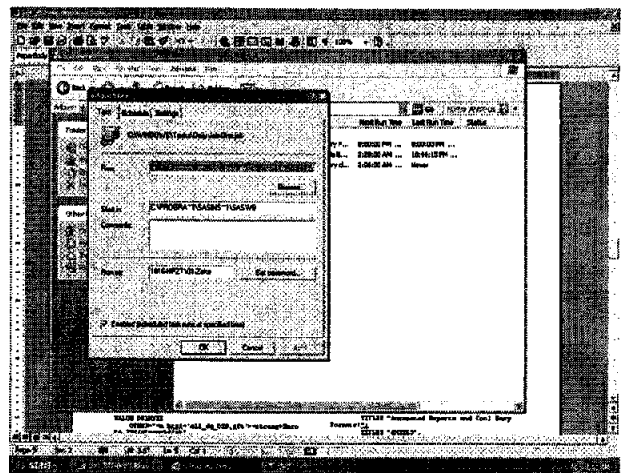
Windows 98 and XP again: Once were done with final selections we can wrap up with the wizard and see the finished 'job'. We are not finished we still want to make some minor adjustments.



Lets RIGHT click once on the finished job.



Select Properties (Left Click)



In both Windows 98 and XP we'll see a similar screen. There is a RUN, START IN properties box.

If you used the method I prefer the RUN box will have our BAT file.

If you did not this next part applies to you.

We need to make sure that the path in the RUN box is set to our SAS.EXE file. Your wizard might have entered the incorrect information for our goal of automating this and using the specific CONFIG file we set up. Or maybe you selected the wrong application earlier with the wizard. Remember, because of how long the path information can be for your PC set up, NOT all of the path information is visible in this box. You will need to carefully click in this box and use the arrow keys to scroll thru and validate it. Most important is that the wizard generally defaults the value, which includes the SAS CONFIG file. In this example we want to use OUR own not the SAS default one.

The value in the START IN box should be the path to the folder where the SAS.EXE file is.

You are done. So long as your pc has the correct date and time you should have your 'job' run according to the specifications you've just selected.

Some things that I noticed are more windows related and specific to my operating system. I found other users did not necessarily experience the same problem and/or their pc configuration was a factor in how much or little they experienced these items. After a TASK executes the residual window from the Schedule Task application clutters up the desktop. I did not get bothered enough to dig into windows with/out SAS to close these windows. They actually helped me know that the Scheduled item was initiated and SAS did start up. Other users complained that the PC left for a few days straight resulted in a cluttered desktop. Until it is closed it will not run again.

You can set up jobs to run weekly, monthly, hourly the sky is the limit. Please remember that as the jobs start to run they will begin to use your computers resources. So setting up jobs to run at the same time might result in slower performance of your computer.

LETS REVIEW

I want to outline what we've learned so far in a step process that will allow you to make sure you set up the same kinds of files and process on your pc. These 5 steps are important in getting your PC SAS to do the repetitive parts "automatically".

XSTART. BAT

This is the syntax which will start the process. It should be executing the SAS. EXE file on your PC. It should have a SYSIN statement pointing to the SAS code you want executed and if you need it should have a "CONFIG" statement pointing to the configuration file you set up for this code to run under.

SASV8X.CFG

Is the SAS Configuration file we modified to fit our purpose. We turned off the SPLASH screen. Bumped up the SORT memory. Set up which LOG file things will go to. Things we do not want to see in our code but want our code to have.

X1016CONFIG.SAS

This is code that our SASv8x.cfg file executes for us. It is SAS code which we might change once in a while but it will basically stay the same over time. The code that will be executed in the following steps might change often as we perform normal maintenance or revisions.

X1016FTP. INI

These are the user variables that identify our program to the HOST. This is nothing more than a script or FTP instructions that SAS will execute to get the DATA or information the program will later use.

SCHEDULE JOB (AUTOMATE)

We want to make sure we set up a JOB via the MS Schedule Task Manager. The point is to have this job run when we are not there. Or run when we are away from our desk and still need that job to run so when we come back things are ready for us to go to work.

So if we have been able to set up those files/tasks on our PC now we can look at the code we are going to execute automatically every morning before we get to work.

DAILY SAS JOB C:\SUG\X1016.SAS

PURPOSE OF THE JOB

Every morning managers need to see what the Delinquency level is for a specific portfolio. The HOST system has a batch process which finishes daily at 1 am. Part of that batch process sets up a copy of the system file in a SAS DATA format and finally it copies that SAS DATA into a SAS Transport file which we will pick up via the FTP process I've outlined earlier.

Our code will then deliver the data to our SAS DATA Library and

continue to produce tabulate output with ODS. This ODS output will be in two types of output HTML and PDF. Our HTML is for 'online' viewing and sharing via the 'browser' method. We will also use the ODS PDF method to give the user the ability to print the same information or share it in hard copy format.

In this example the managers requested the ability to see the detail of what the TOTALS represent. In this code we use PROC FORMAT in the ODS statement and add HTML HREF tags to it. These tags will point to EXCEL files that have the detail. So via the ODS HTML output our user will see the report totals. They can click on the FORMAT Category and go to excel files which have the detail. Or they can use the PDF version to print and share the output.

We also have code at the very end to take all of these output files and bundle them into a zip file that we will email. This email can have multiple attachments so we do not need to ZIP all of the files. But since we are sending 'detail' the email can potentially be too big to get delivered. We also do not want our user to get so many files that it would get confusing.

Finally we send an email text message to our cell phone when it's done.

Lets look at the code.

FILE REFERENCES AND PROC FORMAT WITH HTML.

```
%LET FT=XLS;
%LET FOLD=SUG;
```

The &FT variable will tell my program what kind of file extension I will send data to in this program. It is required in the PROC FORMAT example we are about to see.

```
PROC FORMAT;
VALUE DELKYIIa
OTHER="<a href='all_dq_a029.&ft'>
      <strong>Zero to 29</strong></a>"
2="<a href='all_dq_a030.&ft'>
   <strong>30 to 59</strong></a>"
3="<a href='all_dq_a060.&ft'>
   <strong>60 to 89</strong></a>"
4="<a href='all_dq_a090.&ft'>
   <strong>90 to 119</strong></a>"
5="<a href='all_dq_a120.&ft'>
   <strong>120 to 149</strong></a>"
6="<a href='all_dq_a150.&ft'>
   <strong>150 to 179</strong></a>"
7="<a href='all_dq_a180.&ft'>
   <strong>180 to 109</strong></a>"
8-HIGH="<a href='all_dq_a210.&ft'>
        <strong>210 - Plus</strong></a>"
;
RUN;
```

The PROC FORMAT has HTML syntax we have set up for each of the values the FORMAT will encounter. This HTML will allow the user to see a report that gives them the ability to hover their mouse cursor over the 'FORMAT' value and if they click take them to this 'href' file. We have the ability to point them to another Web Page, or as in this case the detail of that FORMAT value. If we look closer we see that with this example we are using the Macro Variable "&FT" to let us use this code for multiple kinds of output. Say PDF, EXCEL (XLS), TEXT, you name it. However in the code that I am presenting there are 3 kinds of output. One for each part of the portfolio. So we need to set up distinct FORMAT outputs so that they point to the correct file(s).

```

PROC FORMAT;
VALUE DELKYIIB
OTHER="<a href='all_dq_b029.&ft'>
      <strong>Zero to 29</strong></a>"
2="<a href='all_dq_b030.&ft'>
   <strong>30 to 59</strong></a>"
3="<a href='all_dq_b060.&ft'>
   <strong>60 to 89</strong></a>"
4="<a href='all_dq_b090.&ft'>
   <strong>90 to 119</strong></a>"
5="<a href='all_dq_b120.&ft'>
   <strong>120 to 149</strong></a>"
6="<a href='all_dq_b150.&ft'>
   <strong>150 to 179</strong></a>"
7="<a href='all_dq_b180.&ft'>
   <strong>180 to 109</strong></a>"
8-HIGH="<a href='all_dq_b210.&ft'>
        <strong>210 - Plus</strong></a>"
;
RUN;

```

```

PROC FORMAT;
VALUE DELKYIIC
OTHER="<a href='all_dq_c029.&ft'>
      <strong>Zero to 29</strong></a>"
2="<a href='all_dq_c030.&ft'>
   <strong>30 to 59</strong></a>"
3="<a href='all_dq_c060.&ft'>
   <strong>60 to 89</strong></a>"
4="<a href='all_dq_c090.&ft'>
   <strong>90 to 119</strong></a>"
5="<a href='all_dq_c120.&ft'>
   <strong>120 to 149</strong></a>"
6="<a href='all_dq_c150.&ft'>
   <strong>150 to 179</strong></a>"
7="<a href='all_dq_c180.&ft'>
   <strong>180 to 109</strong></a>"
8-HIGH="<a href='all_dq_c210.&ft'>
        <strong>210 - Plus</strong></a>"
;
RUN;

```

We have just set up three distinct FORMAT statements. Each of which will provide 3 distinct types of HREF file names (a, b, c). They will be used in our ODS statement and in the specific step they will resolve to the values the code needs at that time. We could reduce this example to 1 FORMAT statement and use additional 'macro variables' to assist in the process of differentiating the output file HREF statements. But you would need to initiate the PROC FORMAT statement at key parts of the program.

The PROC FORMAT statement and the MACRO variable used here make this code flexible for adjusting output to go to various file types, Excel, PDF. However since it uses a MACRO variable the process does become STEP dependant or JOB specific. In a larger job with many more repetitive steps or outputs I would modify this concept to work in that fashion. With only 3 kinds of output I wanted to keep the statements and macro variables easy to manage as I made revisions for the users.

We need one more FORMAT to present the information in our PDF output. This final FORMAT will not have HTML or HREF. It's the FORMAT we're generally familiar with.

```

PROC FORMAT;
VALUE DELKYXI
OTHER="Zero to 29"
2="30 to 59"
3="60 to 89"
4="90 to 119"
5="120 to 149"
6="150 to 179"
7="180 to 109"
8-HIGH="210 - Plus"
;
RUN;

```

This is essentially the same FORMAT we used in the earlier 3 examples with the only difference being the HTML is not used. In our PDF output we want to see these labels and not the HTML.

OPTIONS NOTES;

```

DATA
ACCT1
ACCT2
ACCT3;
SET ACCT;
IF PRODUCT_TYPE IN ('202','201') THEN OUTPUT ACCT1;
IF PRODUCT_TYPE GE 203 AND
PRODUCT_TYPE LE 301 THEN OUTPUT ACCT2;
IF PRODUCT_TYPE GE 302 THEN OUTPUT ACCT3;
RUN;

```

This is my SAS code example. But here you would have your own code. Notice we have turned on the NOTES option.

ON TO ODS HTML

```

ODS LISTING CLOSE;
ODS HTML BODY='XBODY'
STYLE=SASWEB
TRANTAB=ASCII
FRAME='STARTPAGE.HTML'
CONTENTS='DAILYYY.HTML'
FILE='DAILYX.HTML'
;
OPTIONS MISSING=0 NOCENTER PAGENO=1
LINE SIZE=255;

```

Here we are simply specifying the output file names but note that we are not assigning a 'drive' or path. We want to make sure it's ambiguous so that no matter where these pages are viewed so long as they are in the same folder they will function as a unit. In this way we can distribute it among users and have little or no problems caused by different folder names or drive letter assignments. The other options shown here are my own preference.

```

PROC TABULATE DATA= WORK.ACCT1 MISSING;
TITLE1 "1016INC.COM SAS EXAMPLE";
TITLE2 "OUTPUT ON PORTFOLIO - A -";
TITLE3 "PRODUCT TYPE REPORT";
FOOTNOTE "<A HREF='DAILY_A_AP.PDF'><STRONG>PDF
-PRINTABLE- VERSION!</STRONG></A>";
KEYLABEL SUM ='TOTAL';
KEYLABEL N   ='# OF ACCTS';
KEYLABEL ALL ='TOTAL';
CLASS PYMPSTDU /STYLE={FONT_SIZE=2
PROTECTSPECIALCHARS=OFF};
FORMAT PYMPSTDU DELKYIIA.;
VAR BALANCE /STYLE={FONT_SIZE=1 };
LABEL BALANCE  = 'ACCOUNT BALANCE';
LABEL PYMPSTDU = 'PAST DUE RANGE';
TABLE (PYMPSTDU ALL),
      N*(F=COMMA8.0)
      BALANCE *(SUM*F=DOLLAR15.2)
      /RTS=75 INDENT=3;
KEYWORD ALL /
STYLE={FONT_SIZE=2 FOREGROUND=BLACK};
RUN;

```

Ok the TABULATE procedure. Our TITLE statements need no explanation. They FOOTNOT however does. We are using HREF or HTML to make our FOOTNOTE stand out to the user. The user will be directed to the PDF version of this output that we will create in a few minutes via a click-able link on the same output we are generating. Our user will now have a quick and easy way to obtain the same results in our HTML tabulate but via PDF for printing.

Now take note of the CLASS statement for the variable PYMPSTDU. We are adding the syntax "PROTECTSPECIALCHARS= OFF". This will enable the

use of HTML in the FORMAT we will invoke. It will allow the HTML syntax to be moved without being PROTECTED by the quotes which otherwise would simply show HTML code not the HTML results.

The FORMAT statement for PYPSTDU is referencing the code we set up earlier. Remember this FORMAT has HTML in it. Not only that but the HTML references FILES. That means that each category the FORMAT will create will be a click-able link. We are letting SAS generate a pseudo-drill-down report or web page. By using PROC FORMAT and HTML in this fashion you have just enabled the reporting capabilities of your existing reports to be further enhanced easily. Now you can connect the TABULATE or PRINT or other ODS output with FORMAT to something like an EXCEL sheet with the detail behind these totals. The next few TABULATES are the same but lets remember they are set up for different Data Sets and output.

```
PROC TABULATE DATA= WORK.ACCT2 MISSING;
  TITLE1 "1016INC.COM SAS EXAMPLE";
  TITLE2 "OUTPUT ON PORTFOLIO - B -";
  TITLE3 "PRODUCT TYPE REPORT";
  FOOTNOTE "<A HREF='DAILY A BP.PDF'><STRONG>PDF -
PRINTABLE- VERSION!</STRONG></A>";
  KEYLABEL SUM ='TOTAL';
  KEYLABEL N   ='# OF ACCTS';
  KEYLABEL ALL ='TOTAL';
  CLASS PYPSTDU /STYLE=[FONT_SIZE=2
PROTECTSPECIALCHARS=OFF];
  FORMAT PYPSTDU DELKYIIB.;
  VAR BALANCE /STYLE=[FONT_SIZE=1 ];
  LABEL BALANCE = 'ACCOUNT BALANCE';
  LABEL PYPSTDU = 'PAST DUE RANGE';
  TABLE (PYPSTDU ALL),
    N*(F=COMMA8.0)
    BALANCE *(SUM*F=DOLLAR15.2)
    /RTS=75 INDENT=3;
  KEYWORD ALL /
    STYLE=[FONT_SIZE=2 FOREGROUND=BLACK];
RUN;
```

```
PROC TABULATE DATA= WORK.ACCT3 MISSING;
  TITLE1 "1016INC.COM SAS EXAMPLE";
  TITLE2 "OUTPUT ON PORTFOLIO - C -";
  TITLE3 "PRODUCT TYPE REPORT";
  FOOTNOTE "<A HREF='DAILY A CP.PDF'><STRONG>PDF -
PRINTABLE- VERSION!</STRONG></A>";
  KEYLABEL SUM ='TOTAL';
  KEYLABEL N   ='# OF ACCTS';
  KEYLABEL ALL ='TOTAL';
  CLASS PYPSTDU /STYLE=[FONT_SIZE=2
PROTECTSPECIALCHARS=OFF];
  FORMAT PYPSTDU DELKYIIC.;
  VAR BALANCE /STYLE=[FONT_SIZE=1 ];
  LABEL BALANCE = 'ACCOUNT BALANCE';
  LABEL PYPSTDU = 'PAST DUE RANGE';
  TABLE (PYPSTDU ALL),
    N*(F=COMMA8.0)
    BALANCE *(SUM*F=DOLLAR15.2)
    /RTS=75 INDENT=3;
  KEYWORD ALL /
    STYLE=[FONT_SIZE=2 FOREGROUND=BLACK];
RUN;
ODS HTML CLOSE;
ODS LISTING;
```

The output for this part of the program should result in 3 html files. Which have links in what is the FORMAT categories. These links should point to the EXCEL and from the FOOTNOTES the PDF files. Lets create the PDF files and then the EXCEL.

```
ODS LISTING CLOSE;
ODS PDF FILE='DAILY_A_AP.PDF' STYLE=SASWEB
NOTOC;
OPTIONS MISSING=0 NOCENTER PAGENO=1;

PROC TABULATE DATA= WORK.ACCT1 MISSING;
  TITLE1 "1016INC.COM SAS EXAMPLE";
  TITLE2 "OUTPUT ON PORTFOLIO - A -";
  TITLE3 "PRODUCT TYPE REPORT";
  FOOTNOTE " ";
  KEYLABEL SUM ='TOTAL' ;
  KEYLABEL N   ='# OF ACCTS';
  KEYLABEL ALL ='TOTAL' ;
  CLASS PYPSTDU /STYLE=[FONT_SIZE=2 ];
  FORMAT PYPSTDU DELKYXI.;
  VAR BALANCE /STYLE=[FONT_SIZE=1 ];
  LABEL BALANCE = 'ACCOUNT BALANCE';
  LABEL PYPSTDU = 'PAST DUE RANGE';
  TABLE (PYPSTDU ALL),
    N*(F=COMMA8.0)
    BALANCE *(SUM*F=DOLLAR15.2)
    /RTS=75 INDENT=3;
  KEYWORD ALL /
    STYLE=[FONT_SIZE=2 FOREGROUND=BLACK];

RUN;
ODS PDF CLOSE;
ODS LISTING;
```

In this ODS PDF example we generate only one TABULATE. One for each of the portfolio segments we are reviewing. Notice how the FOOTNOTE statement is intentionally blank. We want to remove the values that would have had a residual HTML syntax from our last procedure. The CLASS statement for PYPSTDU has no syntax for the special character settings. And the FORMAT statement is for our usual FORMAT statement that has no HTML. Remember this is PDF. Our user(s) will not have HTML on that format. The TABULATE is essentially the same. We'll see 3 individual ODS PDF blocks for each of our examples.

```
ODS LISTING CLOSE;
ODS PDF FILE='DAILY_A_BP.PDF' STYLE=SASWEB
NOTOC;
OPTIONS MISSING=0 NOCENTER PAGENO=1;

PROC TABULATE DATA= WORK.ACCT1 MISSING;
  TITLE1 "1016INC.COM SAS EXAMPLE";
  TITLE2 "OUTPUT ON PORTFOLIO - B -";
  TITLE3 "PRODUCT TYPE REPORT";
  FOOTNOTE " ";
  KEYLABEL SUM ='TOTAL' ;
  KEYLABEL N   ='# OF ACCTS';
  KEYLABEL ALL ='TOTAL' ;
  CLASS PYPSTDU /STYLE=[FONT_SIZE=2 ];
  FORMAT PYPSTDU DELKYXI.;
  VAR BALANCE /STYLE=[FONT_SIZE=1 ];
  LABEL BALANCE = 'ACCOUNT BALANCE';
  LABEL PYPSTDU = 'PAST DUE RANGE';
  TABLE (PYPSTDU ALL),
    N*(F=COMMA8.0)
    BALANCE *(SUM*F=DOLLAR15.2)
    /RTS=75 INDENT=3;
  KEYWORD ALL /
    STYLE=[FONT_SIZE=2 FOREGROUND=BLACK];

RUN;
ODS PDF CLOSE;
ODS LISTING;
```

```

ODS LISTING CLOSE;
ODS PDF FILE='DAILY_A_CP.PDF' STYLE=SASWEB
NOTOC;
OPTIONS MISSING=0 NOCENTER PAGENO=1;

PROC TABULATE DATA= WORK.ACCT1 MISSING;
  TITLE1 "1016INC.COM SAS EXAMPLE";
  TITLE2 "OUTPUT ON PORTFOLIO - C -";
  TITLE3 "PRODUCT TYPE REPORT";
  FOOTNOTE " ";
  KEYLABEL SUM ='TOTAL' ;
  KEYLABEL N   ='# OF ACCTS';
  KEYLABEL ALL ='TOTAL' ;
  CLASS PYPSTDU /STYLE={FONT_SIZE=2 };
  FORMAT PYPSTDU DELKYXI.;
  VAR BALANCE /STYLE={FONT_SIZE=1 };
  LABEL BALANCE = 'ACCOUNT BALANCE';
  LABEL PYPSTDU = 'PAST DUE RANGE';
  TABLE (PYPSTDU ALL),
    N*(F=COMMA8.0)
    BALANCE *(SUM*F=DOLLAR15.2)
  /RTS=75 INDENT=3;
  KEYWORD ALL /
  STYLE={FONT_SIZE=2 FOREGROUND=BLACK};

RUN;
ODS PDF CLOSE;
ODS LISTING;

```

MS EXCEL FOR THE HTML

There are different approaches to getting SAS Data to EXCEL. I must stress that for this example I needed to have separate EXCEL workbooks for each of the categories. The users would only be interested in specific workbooks according to their department etc. This example also uses a specific SAS Dataset, which has the variables, labels, and formats properly set up for facilitation of output. Your results on your own data or site would need to take this into consideration when you begin to get different results in EXCEL after performing this same kind of export. The macro I used was adjusted to work for this code. You're welcome to use the macro I've mentioned but please configure it to work for your scenario.

Prepare for the 1st Excel output.

```

DATA REVW029A REVW030A REVW060A REVW090A
REVW120A REVW150A REVW180A REVW210A;
SET WORK.ACCT1;
FORMAT PYPSTDU DELKYXI.;
IF PYPSTDU LT 2 THEN OUTPUT REVW029A;
IF PYPSTDU EQ 2 THEN OUTPUT REVW030A;
IF PYPSTDU EQ 3 THEN OUTPUT REVW060A;
IF PYPSTDU EQ 4 THEN OUTPUT REVW090A;
IF PYPSTDU EQ 5 THEN OUTPUT REVW120A;
IF PYPSTDU EQ 6 THEN OUTPUT REVW150A;
IF PYPSTDU EQ 7 THEN OUTPUT REVW180A;
IF PYPSTDU GE 8 THEN OUTPUT REVW210A;
RUN;

```

Since my example calls for separate tabs for each of the portfolios according to the FORMAT Categories we need to output EXCEL workbooks for each and so we want to have separate SAS Datasets. We'll do the same for each portfolio. Notice the DATASET names.

```

DATA REVW029B REVW030B REVW060B REVW090B
REVW120B REVW150B REVW180B REVW210B;
SET WORK.ACCT2;
FORMAT PYPSTDU DELKYXI.;
IF PYPSTDU LT 2 THEN OUTPUT REVW029B;
IF PYPSTDU EQ 2 THEN OUTPUT REVW030B;

```

```

DATA REVW029C REVW030C REVW060C REVW090C
REVW120C REVW150C REVW180C REVW210C;
SET WORK.ACCT3;
FORMAT PYPSTDU DELKYXI.;
IF PYPSTDU LT 2 THEN OUTPUT REVW029C;
IF PYPSTDU EQ 2 THEN OUTPUT REVW030C;

```

Now we have distinct SAS Datasets for each of the portfolios. One additional item is that on each we have used the FORMAT that does not use HTML.

```

OPTIONS MPRINT MACROGEN NOSYMBOLGEN;
OPTIONS NOXWAIT NOXSYNC;

X "DEL C:\%FOLD\ALL_D*.&FT";

```

Now lets prepare for our new output. This macro uses the DDE method. You need to remember that without a NOXWAIT your session will sit there and wait for whatever response the prompts called by X or other routines will wait for. I use an 'X' command to clean out old excel files from the folder we are working in.

```

data setup; dsname1='work.REVW029A';
dsname2=scan(dsname1,2);
%exloop(c:\%fold\all_dq_a029.xls,yes); run;
data setup; dsname1='work.REVW030A';
dsname2=scan(dsname1,2);
%exloop(c:\%fold\all_dq_a030.xls,yes); run;
data setup; dsname1='work.REVW060A';
dsname2=scan(dsname1,2);
%exloop(c:\%fold\all_dq_a060.xls,yes); run;

```

```

data setup; dsname1='work.REVW090A';
dsname2=scan(dsname1,2);
%exloop(c:\%fold\all_dq_a090.xls,yes); run;
data setup; dsname1='work.REVW120A';
dsname2=scan(dsname1,2);
%exloop(c:\%fold\all_dq_a120.xls,yes); run;
data setup; dsname1='work.REVW150A';
dsname2=scan(dsname1,2);
%exloop(c:\%fold\all_dq_a150.xls,yes); run;
data setup; dsname1='work.REVW180A';
dsname2=scan(dsname1,2);
%exloop(c:\%fold\all_dq_a180.xls,yes); run;
data setup; dsname1='work.REVW210A';
dsname2=scan(dsname1,2);
%exloop(c:\%fold\all_dq_a210.xls,yes); run;

```

This set of macro steps will export the first segment or portfolio portion of our data to EXCEL. Please note that the remainder of the code for this step in this specific part of the code is not presented in this abstract, as it is repetitive. However the working example does have the entire code. We are basically DONE. Lets Email it out!

EMAIL THE OUTPUT

```

FILENAME MAIL EMAIL "sasuser@1016inc.com"
SUBJECT="Report - Automatic ";

DATA _NULL_;
FILE MAIL;
PUT "You can put what you want";
PUT "to say in the body of the";
PUT "message, email here and ";
PUT "its going to work!";
PUT " ";
PUT "http://www.1016inc.com";
PUT "!EM_CC!info@1016inc.com";
PUT "!EM_ATTACH!(\"c:\sug\report.zip\")";

RUN;

```

*** end x1016.sas;**

Here is where I use the SAS email features to facilitate distribution of the output. I do take the files and zip them with WINZIP. This makes it easier to send via email for size and facilitates the distribution of so many files. However if you have a single page output or simply wish to notify someone or yourself that the new report is finished and ready on an Intranet or Internet web site then this would also suffice.

The users I was designing this for were located in different locations and many did not have the same authority to browse a single web server site or something of that nature.

The WINZIP method sufficed for my scenario because it was widely available among many users. It had easy syntax and instructions that I incorporated into my code. In this next example you can see how I executed WINZIP via the X command.

```
x 'c:\progra~1\winzip\wzzip
c:\ta_sherm\sherman_daily_report.zip
@c:\ta_sherm\temp\outzip.lst';
```

What I've done is used the WINZIP 'list' feature to specify which files will be added/updated to this zip file.

And you can even send a message to your mobile phone. The key to this is your service provider needs to have the appropriate email configuration and your phone service plan should provide you with some 'text'-messaging messaging feature. For example I simply send an email message to my EMAIL id provided by my mobile phone service company.

7735551212@mobilephone.com

What happens is the email message will be routed to that company and forwarded to my phone. As I am driving in to work in the morning my phone has already prompted me via this message as to the completion of the task I've assigned.

CONCLUSION

Certainly the techniques I used are a variety of basic thru intermediate level concepts. The goal was to successfully distribute the information to the users and employ some of the newer features PC SAS has to offer. Combined with some 3rd party software like EXCEL, Schedule Task Manager and EMAIL and if you desire WINZIP you end up getting a very flexible concept for producing functional reports that users will find very easy to navigate and work with.

ADDITIONAL CODE OR DATA USED

C:\SUGMACROEXPORTERNEW.SAS

macroexporternew.sas was obtained from:
<http://www.sas.com/service/techsup/unotes/SN/006/006695.html>
Modifications were made to suffice the desired results for this code. Changes or modifications may be required for your site, pc, station. This code is available with sufficient documentation and as such this abstract will not cover it.

*** end macroexporternew.sas;**

SAS DATASET WORK.ACCT (SASLIB.DAT)

This is a SAS dataset on the host that we pick up as a SAS Transport file. However this can be almost any file you can transfer via FTP to your PC and then read with PC SAS. Say a flat file. Our example works because on the host we have already set up the data for this transport in batch. You can also have a dataset or file available for pick up and still use this method.

ACKNOWLEDGMENTS

The author would like to thank the following individuals for their invaluable assistance in the preparation, and technical review of this paper.

Dana Laskowski
Transamerica
Hoffman Estates IL

Sally Ernst
SAS Consultant
Kansas City KS

Steve First
System Seminar Consultants
Madison WI

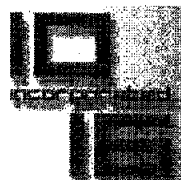
I would especially like to thank my newborn baby daughter and son for their wonderful and above all cute support. With the many sleepless nights they indirectly gave me it made completing this paper in record time possible, I think? I would also like to thank my very patient wife who kept waking me up as I typed or thought I was at night. And for her endless reminders to get a few hours of sleep before it was time to go to work. Or to throw out the garbage.

FINAL CONSIDERATIONS

Please note that the code supplied in this paper is designed only to illustrate the concepts being discussed and may need to be modified to work in other applications and/or environments. The author of this paper does not support modified code.

AUTHOR'S CONTACT INFORMATION

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



Ezequiel 'Zeke' Torres
1016Inc.com
Chicago, Illinois 60630
Work Phone: (773)467-8492
Fax: (773)467-8492
Email: info@1016inc.com
Web: www.1016inc.com

WARNING

This code and the concept and/or your ability to execute it does not indicate you are complying with the security protocols or standards your site, office, place of work have established. The author makes no warranty, assurances implied or otherwise that this method is in anyway 'secure'

or complies with your site standards or protocols. The code, information and concept is presented 'as is'. The use and/or manipulation of USER ID's, Passwords and similar confidential user information can be considered ILLEGAL and against policy or standards at your location and in some cases by public agencies, communications companies, telephone, internet service providers and/or law enforcement agencies. You are responsible for taking that into consideration before applying this model in any way.

Use the Information, concept presented here at your own risk.

REFERENCES

BOOKS

SAS Companion for the Microsoft Windows Environment,
Version 8

Cary, NC SAS Institute Inc., 1999. pp.562

SAS Publishing ISBN – 1-58025-524-8

SAS Institute Inc.

CONFIG related pages cited here:

P.10. Files used by the SAS system

P.396.Changing SAS System Option Settings

P.398 Summary of System Options for Windows

The Complete Guide to the SAS Output Delivery System,
Version 8

Cary, NC SAS Institute Inc., 1999. pp 310.

SAS Publishing ISBN – 1-58025-484-5

WEB SITES REFERENCED

<http://www.sas.com>