Top Ten Reasons to Switch to SAS® Enterprise Guide
(Say Goodbye to PC SAS)

Heather Y. Seeno, Highmark Inc., Pittsburgh, PA

ABSTRACT

My love affair with PC SAS began on May 13, 1996, my first day of work as Data Analyst at a new job. After years of learning “just enough” JCL to run mainframe SAS programs, PC SAS was a delight. A Program Editor, Output Window, Log Window, File Explorer and command line all available at a glance! With Version 8 we gained an Enhanced Editor with color-coding and collapsible code! Life was good. %INCLUDE became my standard method of indicating the flow of SAS code throughout a project. Proc EXPORT and DDE bridged the gap to Excel. ODS meant I could share SAS output with non-SAS users. Then along came SAS Enterprise Guide. No need to use %INCLUDE now that we have process flows. DDE won’t work. Why give up PC SAS?

This paper outlines the Top Ten Reasons why SAS Enterprise Guide is the number 1 tool in my SAS toolkit.

CAVEAT

This paper references several techniques which use SAS Enterprise Guide in conjunction with other products from the SAS 9 Enterprise Intelligence Platform including the SAS Add-In for Microsoft Office.

Reason # 10: Projects – A California Closet System for your SAS stuff!!

Modern man owns so much “stuff” that organizing “stuff” is now it’s own industry. There are professional organizers who, for a fee, will take all your “stuff” and separate it into three piles: keep, toss, and give away. The “stuff” that is kept gets arranged in neat Rubbermaid bins, or, better yet, specialized closet systems. Well, SAS Enterprise Guide provides the California Closet for SAS files and they call it Enterprise Guide Projects. Projects hold everything you need, from SAS code, to SAS datasets, external datasets, SAS logs, even documentation text files. It’s all together in one project file, which has a .egp extension. So take that long vacation to Jamaica without fear, because your co-worker will have no problem (mon!) locating the necessary files to do your job in your absence.
FIGURE 1

Reason #9: Process Flows – A picture is worth a thousand words

Did you ever inherit a coworker’s project with multiple SAS programs and multiple SAS datasets? What is the first thing you did as the new analyst? You scanned the program code, identified the datasets, and essentially figured out the order of what came first and what followed. You found the logs and output files, and put each piece of the puzzle in its place until a complete picture emerged. The Enterprise Guide Process Flow does exactly that, by default, saving us lots of time.

FIGURE 2

When code generates data, a link, indicated by an arrow, is automatically added to the Process Flow. You can also add your own links, by selecting an existing object, right-clicking, and choosing “Link to…”
You can use a process flow to run the entire project or to run only specific tasks. My reliance on the %INCLUDE macro has dropped dramatically now that I no longer need it to run programs sequentially.

Reason # 8: Notes

In our projects, each process flow begins with a note. Standard documentation in the note is the name of the Process Flow, date and initials of developer. This is a great place to explain the purpose of the flow, the inputs, outputs, etc.

FIGURE 3

Reason # 7: Point and Click Wizards

The introduction of point and click wizards is a bitter sweet development for us old-timer SAS programmers. For years we've commanded exorbitant salaries (ha ha) solely because we knew such intricacies as:

⇒ Proc Freq uses a Tables statement to list variables while Proc Print users a Var statement!
⇒ Proc SQL requires comma delimiters between variables while BASE SAS procs hate commas!
⇒ If you've been writing a lot of SQL and then forget you're in BASE SAS and put commas in your variable list you'll see the following errors in your log:

```sas
proc freq data=internet.final4;
tables client_num, client_name;
```

```
22
__________
202
```

ERROR 22-322: Syntax error, expecting one of the following: a name, ;, (, *, -, /, :, _ALL_, _CHARACTER_, _CHAR_, _NUMERIC_.

ERROR 202-322: The option or parameter is not recognized and will be ignored.

Now in one fell swoop my knowledge of syntax lost its value. The brilliant folks at SAS bottled up all that syntax and sold it as GUI! Now if I want a frequency on client_num and client_name I simply select One-Way Frequencies from the Describe menu, and drag the analysis variables over from the Variables to assign screen into the Task roles screen.
FIGURE 4

But take a look at the code EG generated - its BEAUTIFUL!!! The code is fully-documented down to the generation date, time (in minutes and seconds!)

/* -------------------------------------------------------------------
Code generated by SAS Task
Generated on: Wednesday, May 07, 2008 at 5:22:41 PM
By task: One-Way Frequencies
Input Data: INTRNET.FINAL4
Server: SASMain
------------------------------------------------------------------- */

PROC SQL;
%_SASTASK_DROPDS(WORK.SORT);
QUIT;

/* Data set INTRNET.FINAL does not need to be sorted. */
PROC SQL;
    CREATE VIEW WORK.SORT
    AS SELECT CLIENT_NUM, CLIENT_NAME FROM INTRNET.FINAL;
QUIT;
TITLE;
TITLE1 "One-Way Frequencies";
TITLE2 "Results";
FOOTNOTE;
FOOTNOTE1 "Generated by the SAS System (&_SASSERVERNAME, &SYSSCPL) on %SYSFUNC(DATE(), EURDFDE9.) at %SYSFUNC(TIME(), TIMEAMPM8.)";
PROC FREQ DATA=WORK.SORT
    ORDER=INTERNAL
    ;
    TABLES CLIENT_NUM /  SCORES=TABLE;
    TABLES CLIENT_NAME /  SCORES=TABLE;
RUN;
RUN;
/* -------------------------------------------------------------------
End of task code.
------------------------------------------------------------------- */
RUN; QUIT;
PROC SQL;
%_SASTASK_DROPDS(WORK.SORT);
QUIT;
TITLE; FOOTNOTE;

Now where this really comes in handy is for those infrequently used pros. For instance, I haven't had an opportunity to pull a random sample in years. Suddenly I wanted to do just that. I could write some code from scratch, dig up an old macro that I used years ago, or select Random Sample from the Data menu in Enterprise Guide. As every good programmer, I took the quickest route and selected Random Sample from the Data menu! A quick preview of the code behind Random Sample shows a proc that is brand new to me: Proc SurveySelect!! Who knew? This is a great way to learn SAS code, too.

PROC SURVEYSELECT DATA=WORK.SORTTempTableSorted
    OUT=WORK.RANDRanDomSampleFINAL(LABEL="Random sample of INTRNET.FINAL")
    METHOD=SRS
    N=2
    ;
    STRATA CLIENT_NUM;
    ID CLIENT_NUM CLIENT_NAME;
RUN;

By the way, if you like how I've highlighted my comments in Buttercup Yellow, navigate to Tools/Options/SAS Programs/Editor Options, where you can switch colors around to your heart's delight.

Reason #6: Multiple Servers

SAS Enterprise Guide can access several different types of servers when used as part of the SAS Enterprise Intelligence Platform.
The SAS Workspace Server is your nuts and bolts basic server, which functions similar to PC SAS. From within EG it is possible to submit code to one or more workspace servers, or even to your PC SAS server, identified as “Local.” At Highmark the primary Workspace Server available for me to use is called “SASMain.” But I also have access to SASFore and SASApp1. SASFore is a Workspace Server running on a second UNIX box, which also has SAS High Performance Forecasting and SAS Enterprise Miner products. SASApp1 is a Stored Process Server. From within the comfort of SAS Enterprise Guide I can connect to any one of these servers, depending on the task at hand. I can barely wait until our admin group grants me access to an OLAP server! Of course, all these choices create complexity…. Which server is set as my default? Where am I operating now? Luckily I can check the bottom of my screen, where EG will inform me of my logon and metadata identity as well as which server I am using, identified by the IP address and port number. Since I do a lot of switching back and forth between environments, I really appreciate the reminder.

**FIGURE 5**
Reason #5: No need for separate FTP software!

File Transfer Protocol, or FTP, is a method for transferring (downloading or uploading) files. Before the advent of Enterprise Guide at Highmark, moving SAS code from a local or network drive to the UNIX SAS server required the use of FTP software. With EG, this process is simplified. Simply open the code from the Local Computer and SAVE AS to the SAS Server. Voila!

**FIGURE 6**

![Open Code From](image1.png)

**FIGURE 7**

![Save As](image2.png)
Reason #4: The Task Status Window

Have you ever had an analysis that pulled back oodles of data in a gazillion steps and ran for hours and hours? You go to lunch, participate in a 2:00 meeting, and when you return to your desk, it’s STILL running? By now you’re wondering if you’re going to be able to leave work in time for 5:00 yoga or if you should plan on hitting the gym tomorrow. The EG Task Status Window is a great tool that will help you track how far along you are in the process. The Task Status Window lists each task and its current status: Running, Waiting or Completed. The status report indicates the server name on which the task will run. This is also where you can stop tasks from running, with the choice of remaining connected to the server or severing all connections.

**FIGURE 8**

<table>
<thead>
<tr>
<th>Task Status</th>
<th>Status</th>
<th>Queue</th>
<th>Server</th>
</tr>
</thead>
<tbody>
<tr>
<td>SP00003_Underwriting_Control_Audit</td>
<td>Running..</td>
<td></td>
<td>SASApp3-Logical Stored Process Server</td>
</tr>
</tbody>
</table>

Reason #3: Parameters (Soon to be known as Prompts)

A few years back I inherited maintenance of an existing report. This report could be generated for various customers and various time periods. The client number and begin and end dates were all the user needed to change prior to running the report. The user was opening the Enhanced Editor in PC SAS, editing these few lines below, and submitting the code to the UNIX SAS server.

```sas
%let num=CLH_1.CL_N in ( '768910','256882') ;    /* Client Numbers */
%let styr='01jul2005';                         /* Year to date start date */
%let endyr='30jun2006';                         /* Year to date end date */
%let stqtr='01apr2006';                         /* Quarter start date */
```

This particular user had no knowledge of SAS; however. When she accidentally forgot a single quote or a semi-colon, I raced upstairs to interpret the log and fix the error. Obviously this was not an efficient use of my time. Then I discovered I had access to SAS/IntrNet! After working with SAS Admin to build an IntrNet service, I was able to migrate this report to the web. But I still needed some sort of front-end for the user to input client numbers and dates. Having limited knowledge of web programming, I opted to use DATA step PUT statements to write out HTML which would build an input form as illustrated below:

```sas
data _null_;  
file _webout;  
*.-Get the service name, url, and pgmlib to use in broker callback--;  
url=symget('_url');  
service=symget('_service');  
put '<html>';  
put '<form action="' url +(-1) '" method=post>';  
put '<input type="hidden" name="_service" value="' service +(-1) '"';  
put '<br>';  
put '<table border="0" align="center" width="595" height="100" cellspacing="9" face="Tahoma">';
```
This worked fine, but took some time to program.

With SAS Enterprise Guide, creating an input form is simply a matter of declaring macro variables in the project Parameter Manager....

**FIGURE 9**

...and assigning them to code in the properties tab of a program....
so that when users submit the code for execution an input form is automatically generated....
which assigns values to each of the macro variables in the current SAS session, enabling their use in SAS code like this:

```sas
data _null_;

p1 = resolve('&Client_no');
if missing(p1) = 0 then where_cls1 = " AND cl_n = " || p1 || " ";

p2 = resolve('&Group_no');
if missing(p2) = 0 then where_cls2 = " AND gp_n = " || p2 || " ";

p3 = resolve('&District_code');
if missing(p3) = 0 then where_cls3 = " AND sd_n = " || p3 || " ";

p4 = resolve('&Region');
if missing(p4) = 0 then where_cls4 = " AND coarg_c = " || p4 || " ";

p5 = resolve('&Contract_ID');
if missing(p5) = 0 then where_cls5 = " AND substr(ch_agr_id,1,9) = " || p5 || " ";

p6 = resolve('&Member_Id');
if missing(p6) = 0 then where_cls6 = " AND member_nbr = " || p6 || " ";

where_clause = where_cls1 || where_cls2 || where_cls4 || where_cls5 || where_cls6 ;
call symput('where_clause',where_clause);
run;
```
%%put &where_clause;

With the code above the user can enter one or more parameter values and the NULL dataset builds the corresponding WHERE clause for the query. The RESOLVE function returns the value of the macro variables. The MISSING function returns a value of “1” if true and “0” is false. If a macro variable has no value, that piece of the where clause is blank. Each of the individual where clauses are concatenated together to form one statement. The where clause statement is then stored in a new macro variable named “where_clause” for use in a data extract.

Reason #2: Calendar Object

For years our customers have used calendars to select dates in various types of applications. Finally SAS programmers can offer the same functionality to our users. Simply indicate that a parameter is a DATE type, and automatically the input will be via the calendar object.

FIGURE 12

CAUTION: The Version 9.1 calendar date object automatically returns a macro variable whose value is a date constant in the SAS DATE9. format, like this: ‘01Jan2008’d. To query Teradata®, I needed to convert the value of the macro variable “Begindt” to SAS yymmdd10. format, like this: ‘2008-01-01’. Version 9.2 will bring the ability to modify the date format as needed. Meanwhile, use the code below. The PUT function returns the date as a character value in the desired date format. This value is then stored in a character variable named “Tempvar.” The CALL SYMPUT routine saves the value of Tempvar into a new macro variable named “Begin.”

data _null_;  
tempvar = put(&begindt,yymmdd10.);  
call symput('begin',tempvar);
\begin{verbatim}
run;
\end{verbatim}

NOTE: \texttt{begindt = '03Jul2008'd}, \texttt{begin = 2008-07-03}

**Reason #1: Stored Processes**

A Stored Process is a vehicle that allows non-SAS programmers to run SAS programs. A SAS stored process is just a SAS program stored in a special place called a source code repository. Information about the stored process—such as which server to run the stored process on, what parameters the stored process has, what the name of the .sas program is—is registered in the metadata repository.3

Stored Processes enable users to run SAS programs

- In SAS Enterprise Guide
- In SAS web clients (SAS Web Report Studio, SAS Information Delivery Portal)
- In Microsoft® Office products (Word, Excel™, PowerPoint) via the SAS Add-In for Microsoft Office

To run Stored Processes from the web, users do not need any client-side software other than a browser, such as Internet Explorer. The SAS product, Add In for Microsoft Office, must be loaded on the client side for users to run Stored Processes from within Excel, Word or PowerPoint. The Add In creates a SAS menu item on the Office toolbar. Any Stored Process that is registered with the metadata server can be run from this menu.

Getting back to that report I inherited, my customer now gets the data she needs without leaving the comfort of Excel. And thanks to the inclusion of parameters in my stored process, my customer can customize her request as needed. She can run her own report on the first of each month by simply changing the dates and submitting. My schedule is no longer tied up by operational business unit deadlines.
CONCLUSIONS

SAS Enterprise Guide takes SAS programming to a whole new level. The power of SAS is harnessed in a flexible, highly functional user interface. Say goodbye to PC SAS – you won’t regret it!

ACKNOWLEDGEMENT

Portions of this paper were suggested by my colleague, Harsha Kotian. I'd like to express my gratitude for her assistance and support.

REFERENCES


CONTACT INFORMATION

Heather Y. Seeno
Highmark, Inc.
1701 Center Street
SP3W
Camp Hill, PA
Work Phone: 717.302.2361
FAX: 717.302.3502
e-mail heather.seeno@highmark.com

SAS and all other SAS Institute Inc. product or service names are registered trademarks of SAS Institute Inc. in the USA and other countries. ® indicates USA registration.