Paper BI-10-2015

SAS® Enterprise Guide® System Design

Jennifer First-Kluge and Steven First, Systems Seminar Consultants, Inc.

ABSTRACT

A good system should embody the following characteristics: It is planned, maintainable, flexible, simple, accurate, restartable, reliable, reusable, automated, documented, efficient, modular, and validated. This is true of any system, but how to implement this in SAS Enterprise Guide is a unique endeavor. We will provide a brief overview of these characteristics and then dive deeper into how an Enterprise Guide user should approach developing both ad hoc and production systems.

INTRODUCTION

A tool is only as good as its user. Enterprise Guide is a software product with a vast array of tasks, coding choices, options, and other objects. There is so much potential functionality there that it can be easy to get lost in EG's capabilities. It is easy to be overcome by eagerness and to just dive right in to a project, without planning and putting the proper structure in place. Creating a well designed system in Enterprise Guide can be simple and will pay off over and over again in the long run.

OVERVIEW OF A SYSTEM

A standard system usually follows this type of pattern:

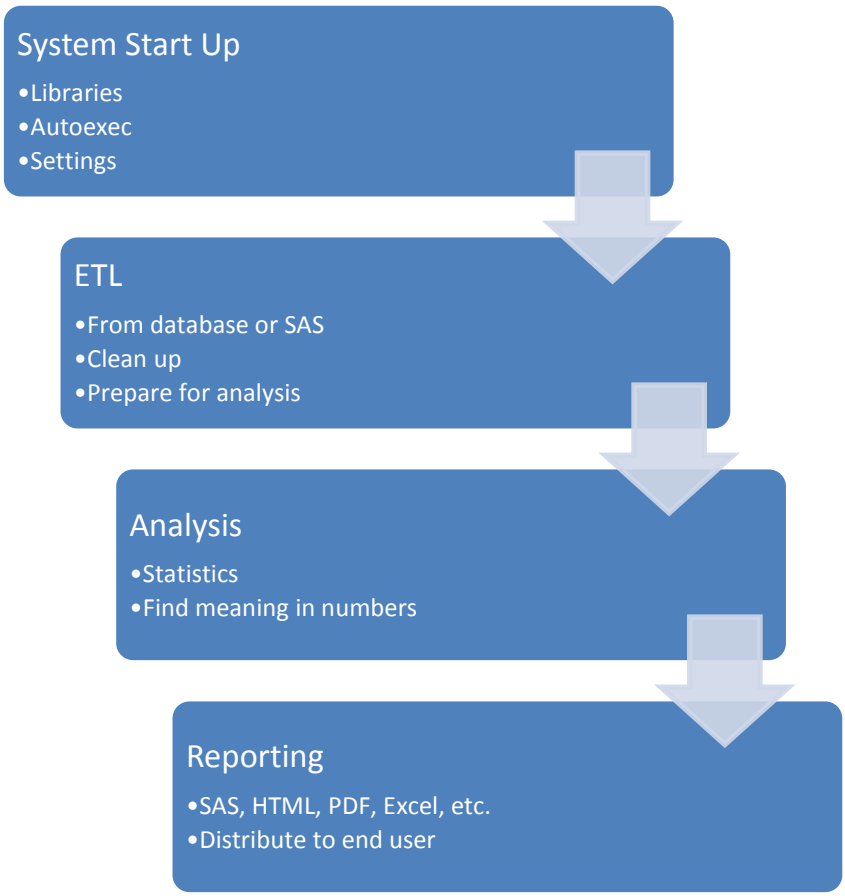


Figure 1. Typical System

A good system should exemplify the following characteristics:

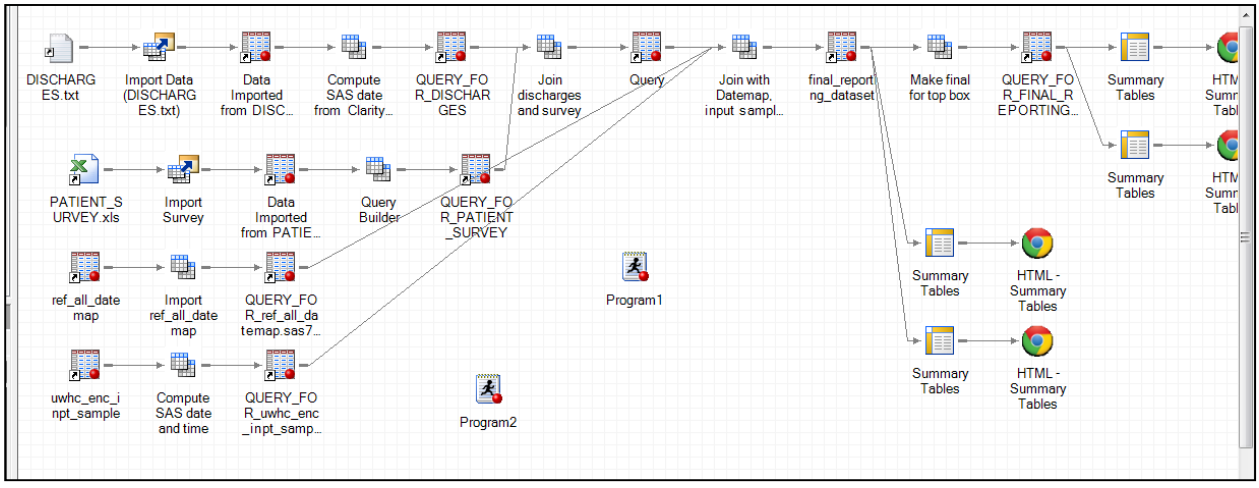
- Planned-user should design system before jumping into coding
- Maintainable-the system should be able to be run and understood by anyone of reasonable intelligence
- Flexible-the system should allow for future updates, changes, and exceptions
- Simple-the system shouldn't overcomplicate things so that they are confusing to other users
- Accurate-the system should be pulling the correct data, performing the correct operations
- Restartable-a large system should be divided into run sections so that if it crashes the entire project doesn't need to be rerun
- Reliable-the user should be able to depend on the system and the results
- Reusable-a system should be developed so that it can be used for similar applications in the future
- Automated-the system should run with as little interactive work as possible
- Documented-the system should be well documented including run time instructions, change logs, and complicated calculations
- Efficient-the system should run in a reasonable amount of time and take advantages of SAS's many ways to speed up systems
- Modular-the system should be developed in distinct pieces so that it is easier to debug, reuse, or restart.
- Validated-the system should have checks and balances (such as record counts, sums, etc.)

SAS Enterprise Guide® is the perfect vehicle to develop a well designed system. It has many built in tools that the user can leverage.

START UP

WORKING WITH PROCESS FLOWS

Once a user begins a project in SAS Enterprise Guide, the work is by default going into a process flow. A process flow is basically a collection of objects including programs, data, output, notes, etc. Once a user starts working in Enterprise Guide it is very easy for things to get messy very quickly. The following process flow isn't too bad yet, but it is approaching the point that lines start crossing and if more is added, it will certainly not be as neat as it could be.



Display 1. Example of a "messy" SAS Enterprise Guide project

Instead of throwing everything into one process flow, resulting in a tangled mess, the user can create multiple process flows. The process flows can function as folders to organize a SAS Enterprise Guide project.

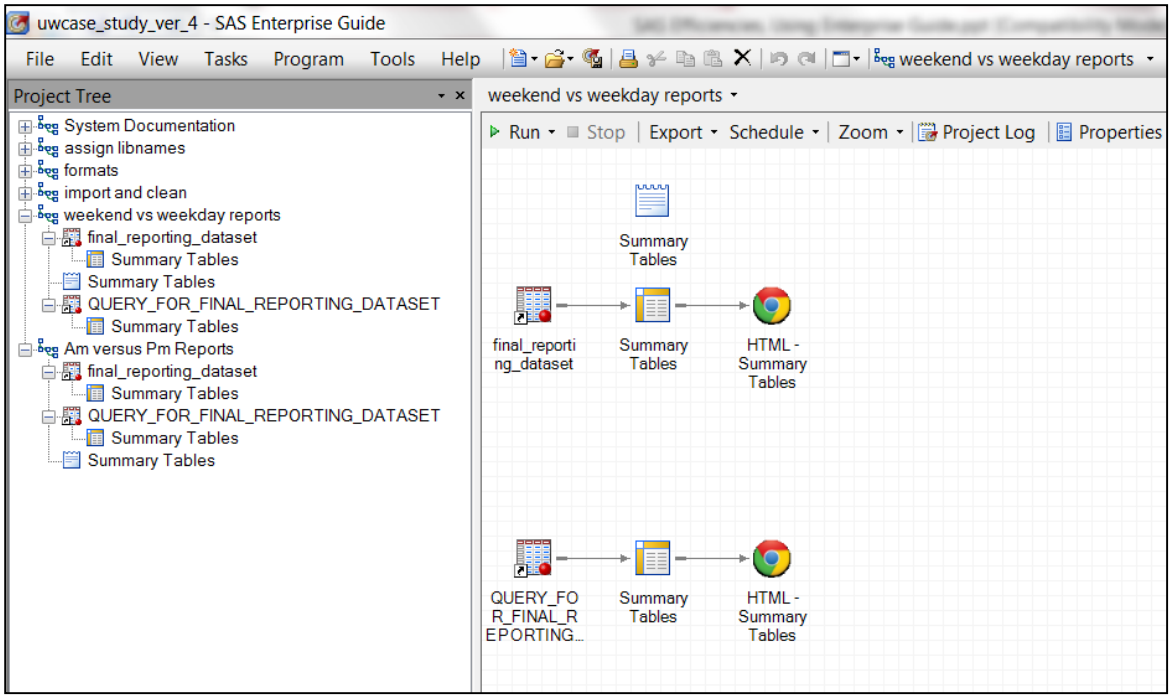
Process flows could be organized according to logical breaks. For example, a user could break up a financial projection system, into 3 Process Flows: Income Projections, Expense Projections, and Production Projections. Each process flow has its own data, tasks, code, output, and notes. It is easy to keep track of all of the moving parts and make changes.

A project could also be divided into functional breaks based on a system level and how the process will be run, such as the following flows:

1. Overall System Documentation
2. Startup Processes, including Assigning Libraries and Setting Options
3. Data Import, Data Cleanup, Data Preparation
4. Preliminary and Exploratory Analysis
5. Final Analysis
6. Graphing and Reporting
7. Production, Scheduling, and Distribution

The main point is that a project should be broken down somehow into multiple process flows that are easier to work with and understand and easily viewed and printed. Here is an example of a more clearly organized project showing the last two process flows expanded.

Any of the process flows on the left can be expanded in the project tree to show the corresponding graphical flow on the right.



Display 2. A well organized SAS Enterprise Guide project

LIBRARIES

Native SAS data files are created within EG and SAS and provide the most efficient way to pass data to other parts of the SAS system. The decision of where to store this data can be complex as there are many choices of where to store SAS data, but if done correctly it will make the user's lives much simpler and should not cause performance problems.

Traditionally SAS files are broken into groups as follows:

1. WORK is a temporary location that exists while SAS jobs are running. This is a good place to store small and quickly created files. Using WORK for long running queries becomes inconvenient when data does not change much but is needed many times.
2. SASUSER can be an individual area for SAS files that remain after the SAS session ends. Files remain there until someone or some SAS process deletes them. This is a good location for one user's personal files that are inconvenient to recreate.
3. SASUSER could also be an area that all users share. While this provides a space for intermediate or longer term storage and does allow one user to share data with another, users could however overlay or delete data from other users and if the area fills it affects all users.
4. Other SAS libraries can be defined that point to a directory accessible to the SAS server. This can provide a medium or long term place to store data and is a good place to store more project oriented data that might be shared between users and stored for a long time. Advantages of using permanent library pointing to a directory is that naming conventions are usually already in place and users are comfortable with directories.
5. Another place to store SAS files is as a table within a RDBMS such as SQL server. Disadvantages to this method are that permissions need to be granted to the RDBMS and names need to be unique, and files are stored as RDBMS tables, so it takes time to convert from SAS to and from the RDBMS format. Advantages are that DBA's can have more control and these files look like any other SAS file, and other non SAS systems can access the RDBMS tables. This is also very useful if a SAS file needs to be joined to a RDBMS table. The SAS file can be loaded to the RDBMS, and the join of the now RDBMS tables can occur in the RDBMS and it will generally run much quicker than if joining a SAS file to a RDBMS file.

A user can create their own permanent library to store SAS files by selecting "Assign Project Library" from the Tools menu. Permanent data can be stored here and reused elsewhere in Enterprise Guide. The ability to create and store permanent files can however often be limited in a SAS environment by the SAS administrator and is not always an option. The library can point to a specific folder or server location and can also be referenced to bring in input files. Within this library location, it is a good idea to set up directories for input, output, and other items.

DOCUMENTATION

Documentation is a critical component of any good system, but it is all too often overlooked. It is important to document what is being done, when, by who, and why (for coding and point and click tools). What you are doing now may be obvious but down the road or when it is passed on to someone else, it should still make sense.

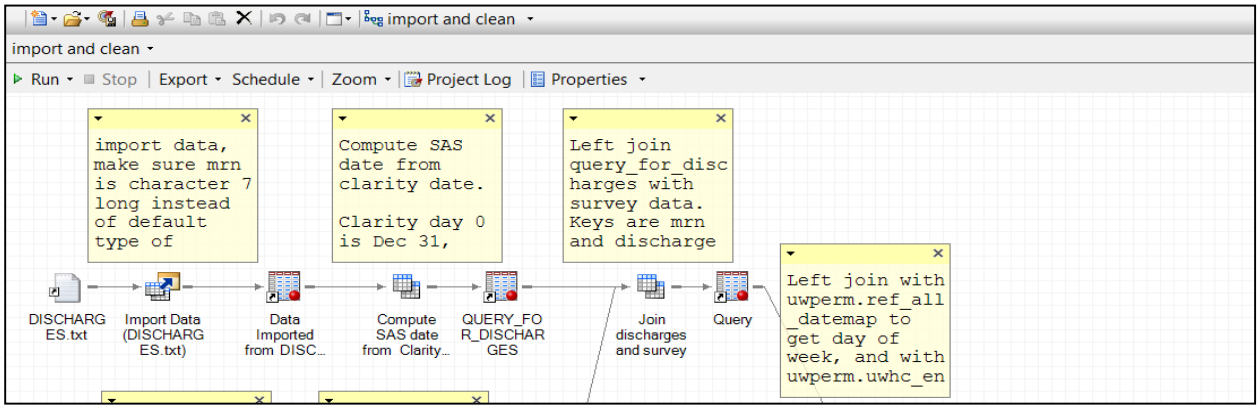
A built in way to document in SAS Enterprise Guide is by using 'Notes'. An Enterprise Guide note is a text document where a user can type in any important details. It will appear as an object in the project. It can be linked to a data set, task, code module, output, or log or it can be independent.

```
3/14/2013
Created by Jennifer First-Kluge, SSC, jfirstkluge@sys-seminar.com

Change Log:
8/1/2014-Jennifer added in new pie types
12/14/2014-Jennifer updated with 2015 price information

Run Instructions:
-Receive updated pieshop.xls file from Tom Miron
-Import .xls file to create SAS dataset
-Double check column lengths and format when import is complete
-This data will feed our corporate production estimates
```

Display 3. An Enterprise Guide note

It is very useful to create a Note linked to each task used in the Project. It should contain the what, who, when, and why of the task. It is quick to do, and it will save time in the long run. It is also useful to have notes to document overall pieces of your Project. They should explain the purpose of multiple tasks working together, including dependencies, and the overall purpose of the process.

Expanded Notes are a new feature in EG 6.1. They can be used to annotate and better document tasks and projects.



Display 4. Annotated Notes in SAS Enterprise Guide

Another piece of good documentation is good naming conventions. Good, consistent naming conventions are self documenting. Enterprise Guide will use default names like 'WORK.QUERY_FOR_DAILY1' and 'Calculation'. The user should specify meaningful names, assign variable labels, and use good default formats for numeric fields. Good naming applies to data, variables, output, tasks, programs, and process flows and good labels and formats will make the displayed data more attractive.

The overall project should be documented, and a great way to do this is to dedicate the first process flow to overall documentation. It could include system documentation, maintenance logs, run time instructions, and wrap up instructions.

Of course, code should be documented as well. It is helpful to use a header box and change log at the top of each program. It is easy to comment as one develops code (or even before), but don't wait until after. Especially focus comments on difficult code, such as complicated calculations. Remember, especially in Enterprise Guide, a non-programmer may need to read and decipher this code.

AUTO ARRANGE

When working in a process flow, right clicking allows the user to uncheck "Auto Arrange" so that the objects in the process flow can be rearranged. The user can eliminate crossed lines and make the

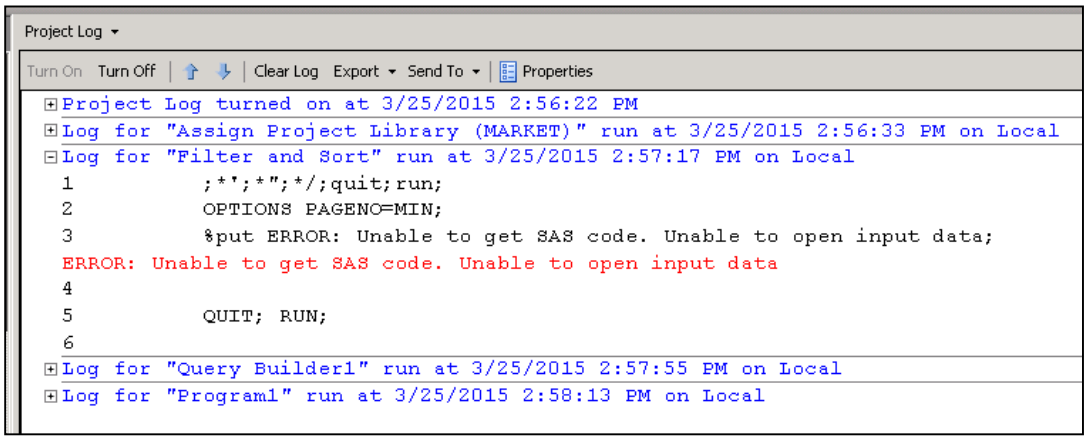
appearance better and less confusing. If the lines in the flow are crossing or not arranged well, uncheck and move the objects around.

PROJECT LOG

Project logs contain all of the SAS logs that have been generated for a project. This gives an excellent place to view all runs of the project which can help with error checking and provide a later audit to answer questions concerning what filtering was used, what joining occurred etc.

The project log defaults to being off. It can be turned on and all activity will be logged, or it can be cleared after testing and a clean log generated. Note that to include the autoexec log in the project log, the project would have to be exited and reentered.

Below is an example of a project log. Each item could be expanded to show details. In the example we have expanded to see the details of an error.



Display 5. Example of a Project Log

AUTOEXEC

If a process flow named “Autoexec” is present at the beginning of a project, the flow will automatically run when the project is opened. This is an excellent place to create a SAS program to establish SAS libnames and SAS options so that anyone running the project will get consistent data and results. It does ask the user if he or she wants to run the autoexec or that message can be suppressed. Autoexec process flows run only at project open and not when the run project menu is used.

AD HOC VS PRODUCTION

The same rigor and discipline should be applied to temporary ad hoc systems that is applied to production systems. More often than not, ad hoc processes are used beyond their initial intent. They are reused or replicated, and quite often they actually turn into production processes. Putting in that small time investment up front will pay off in the long run.

ETL

Enterprise Guide requires data to be a nice rectangle. EG can use SAS Data Files or RDBMS Tables (Oracle, SQL, DB2, etc.). Text or Excel files can be imported into SAS to create SAS Data Files. The way that SAS Enterprise Guide handles data is very useful to the user, but there are a few tricks that are very helpful to know.

JOINING SAS AND DATABASE FILES

SAS tables that are stored in an RDBMS look and act very much like native SAS files to the user. It is transparent to the user to just click on a SAS file and join with another SAS data set that is actually a RDBMS table, and the join works but very slowly. Whenever SAS can, it passes filtering and joining to the RDBMS which is usually the most efficient place to do it. However, if one table exists in SAS and the

other is in the RDBMS, what can happen, beneath the scenes, is a full table scan occurs and the resulting table is downloaded to SAS for the join to occur and probably discard most rows.

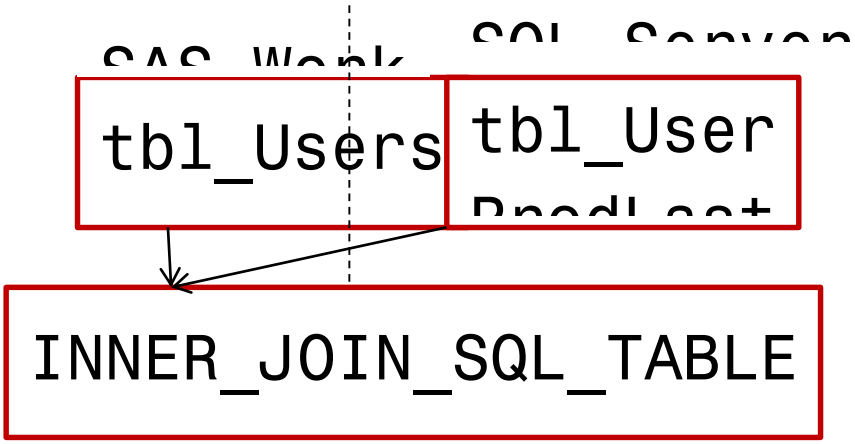


Figure 2. Joining a SAS Table with a RDBMS Table

There are a few ways to address this issue:

1. If there is an equivalent table in the RDBMS to use instead of the native SAS file, the join occurs in the RDBMS and query time will be greatly reduced. The user may have to upload a SAS file to the RDBMS before the join. This requires write access to the RDBMS.
2. The DBMS table can be filtered and downloaded to SAS before the joins take place. This may not help unless extensive filtering can occur.
3. Coding a PROC SQL that utilizes the DBKEY option to download matching rows. This works best when very few rows match against a very large RDBMS file.
4. The PROC SQL pass through facility can be used to work exclusively in the RDBMS system and only download the created SAS table. This gets SAS out of the way and again would require all tables to exist in the RDBMS. This is an option available in the SAS Enterprise Guide query builder.

5. A new SAS proc called FEDSQL is available in the newest release of SAS which has stellar performance in a “Federated” environment and will probably be worked into Enterprise Guide into the future.

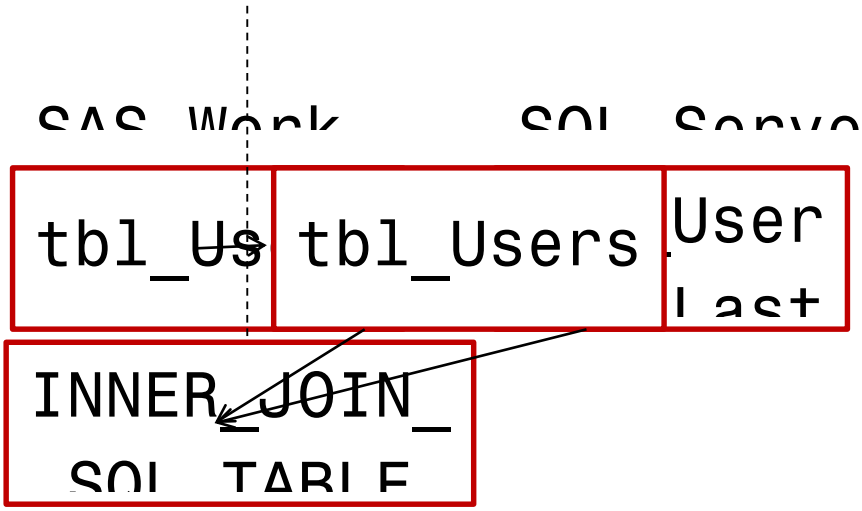


Figure 3. A better way to join SAS with a RDBMS file

SAS FUNCTIONS ON RDBMS FILES

Another transparency that multiplies run time is using a SAS function that doesn't exist in the RDBMS. Run times can be much higher if the RDBMS doesn't filter the data but leaves it for SAS to filter. A very common example is using the DATEPART part function in a filter (Where clause).

Because the DATEPART function is not honored by some DBMS's, the unfiltered rows must be downloaded, where SAS does the filtering. Usually a different way of specifying the filter can do the equivalent.

Examples:

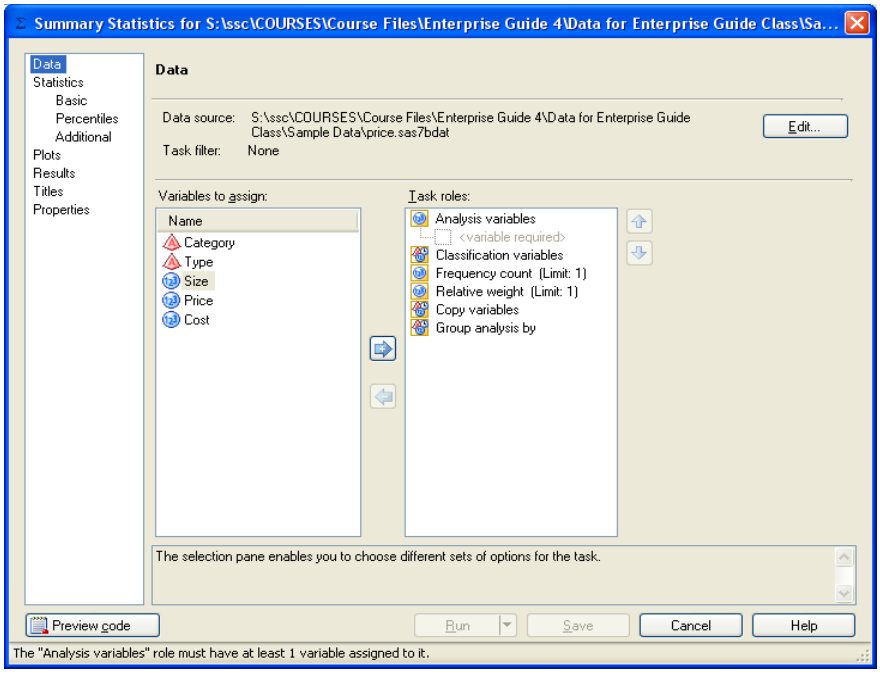
```
WHERE datepart(LastUsedDate) >= '01JAN2014'd;       /* SAS filters later, is slower */

WHERE t1.LastUsedDate >= '01JAN2014:00:00:00'dt; /* RDBMS can filter, is faster */
```

There are options that can display the SQL passed to the RDBMS which provides some cryptic but useful messages. The key is to see whether the WHERE clauses or ON clauses for joining are part of the RDBMS query or not. The SAS documentation for each RDBMS documents which functions are passed to the RDBMS and the SASTRACE option displays the RDBMS SQL.

DATA ANALYSIS AND SUMMARIZATION

Once a user has all of the data in place, it is time for data analysis and summarization. Of course this will vary greatly depending on the nature of the application. The user may calculate simple statistics like percentages or may perform more complex statistical analysis like regression analysis or multivariate correlations. They also may summarize their data, such as by month, department, employee, customer, or product.

The Summary Statistics Task (PROC SUMMMARY) is an excellent task to use for this. It is a very easy point and click way to calculate basic statistics such as mean, standard deviation, minimum, maximum, and more. Variables can also be "classified" to roll them up into groups (such as category or type). Here is an example of the interface:



Display 6. The Summary Statistics Task

The task will generate a Summary Statistics Table as output. Here is an example:

Summary Statistics
Pie Prices

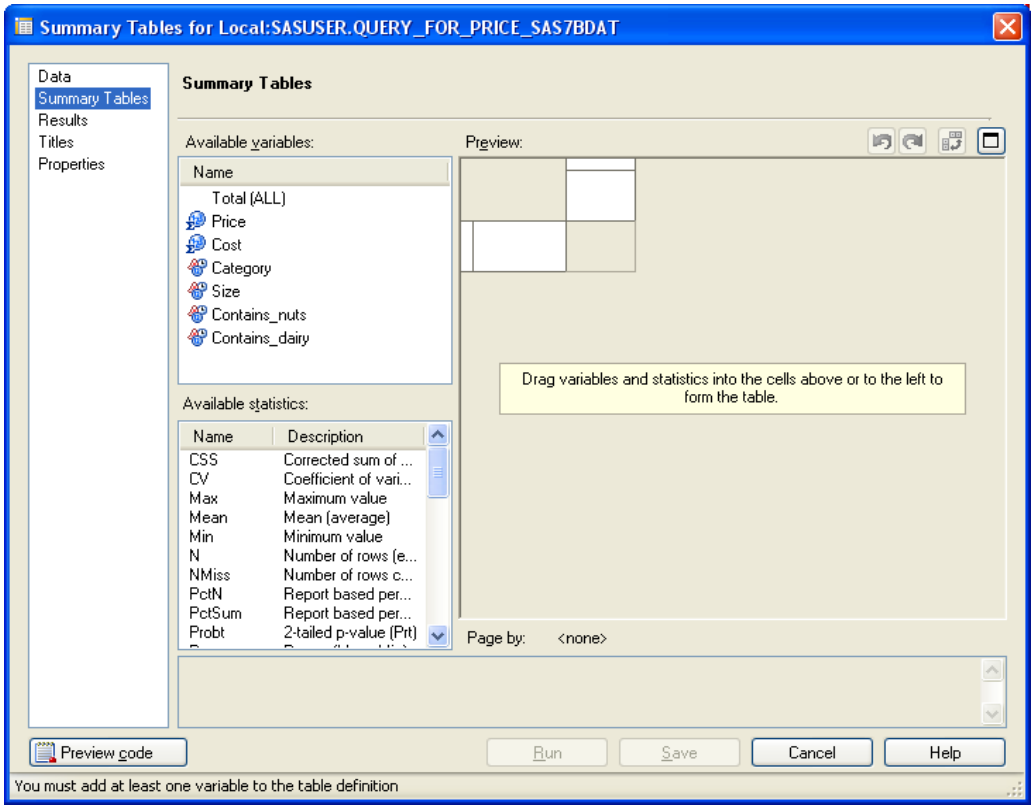
The MEANS Procedure

Category	Size	N Obs	Variable	Mean	Std Dev	Minimum	Maximum	N
Classification Variables						Statistics		
			Cost	4.33	0.83	3.20	5.25	5
	10	5	Price	10.99	1.58	8.99	12.99	5
			Cost	5.61	0.85	4.75	6.85	5
Fruit Pies	8	7	Analysis Variables					
			Price	4.99	0.66	2.60	4.55	7
	10	7	Price	8.85	1.35	6.99	10.99	7
			Cost	4.43	0.71	3.70	5.65	7

Generated for Pie Palace at 8:29 AM

Display 7. Summary Statistics Output

Another excellent task for displaying summarized data is the Summary Tables Task (PROC TABULATE). This also uses analysis and classification variables to roll up data. It calculates selected statistics about a data set and displays the statistics in a highly customizable table. This can be customized in a Report Preview Area. So the user can see the rows and columns, variable roles, and how things are going to be rolled up and calculated before the report is produced. Here is a view of the Preview Area:



Display 8. Preview Area in Summary Tables Task

Here is an example of the output that the Summary Tables Task can produce:

Pie Price and Cost Summary Report

Cost and Price Statistics		Category														All			
		Chocolate Pies						Fruit Pies								The ALL Variable			
		Size (in inches) — Classification Variables						Size (in inches)											
		8				10				8				10					
		Price		Cost		Price		Cost		Price		Cost		Price		Cost		Price		Cost	
		Min	Max	Min	Max	Min	Max	Min	Max	Min	Max	Min	Max	Min	Max	Min	Max	Min	Max	Min	Max
Contains Dairy	Contains Nuts	Analysis Variables — Statistics																			
Classification Variables										$4.99	$6.99	$2.60	$3.70	$6.99	$8.99	$3.70	$4.50	$4.99	$8.99	$2.60	$4.50
Yes	No	$6.99	$10.99	$3.20	$5.25	$8.99	$12.99	$4.75	$6.85	$7.99	$8.99	$4.20	$4.55	$9.99	$10.99	$5.10	$5.65	$6.99	$12.99	$3.20	$6.85
	Yes	$9.99	$9.99	$5.00	$5.00	$11.99	$11.99	$6.00	$6.00									$9.99	$11.99	$5.00	$6.00

Generated for Pie Palace at 4:36 PM

Display 9. Summary Tables Output

There are other methods of summarization and analysis, ranging from basic to complex. SQL (or Enterprise Guide's Query Builder) is another excellent tool for this.

Regardless of the method for summarization, it is a great idea to include some tie out reports and probably put them in a separate process flow if convenient. The most useful reports are usually simple statistics like:

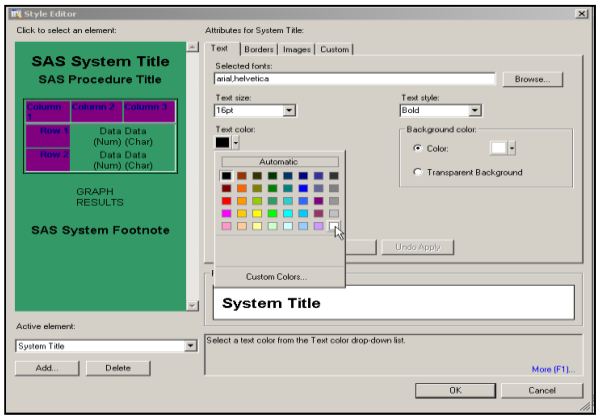
1. Frequency reports (row counts)
2. Univariate Statistics like sums, means, minimums, maximums etc.
3. Percentages

These types of reports will provide a quick snapshot that will flag problems in the data or process.

REPORTING AND DELIVERING DATA AND REPORTS

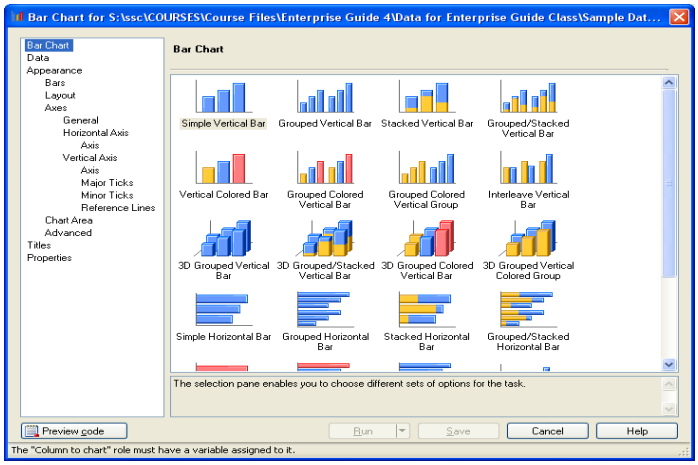
Reports can be produced in a variety of formats, including a SAS report, HTML, PDF, Excel, and others. These reports can be saved to a specific location or the distribution can often be automated, depending on licensing. There is an option to "do the export as a step in the project" and then when the project is run, the report would be exported into Excel. There is also another option to email as a step in a project. There are more moving parts here than just SAS though, so these capabilities may depend on your platform, email server, and other software and environment considerations.

The format of reports can also be somewhat automated by taking advantage of CSS (Cascading Style Sheets). Standard styles can be used or a user can develop their own. For example, a user could develop a style that utilizes company colors and includes their company logo on the top of every report. Here is an example of the Style Sheet Editor.



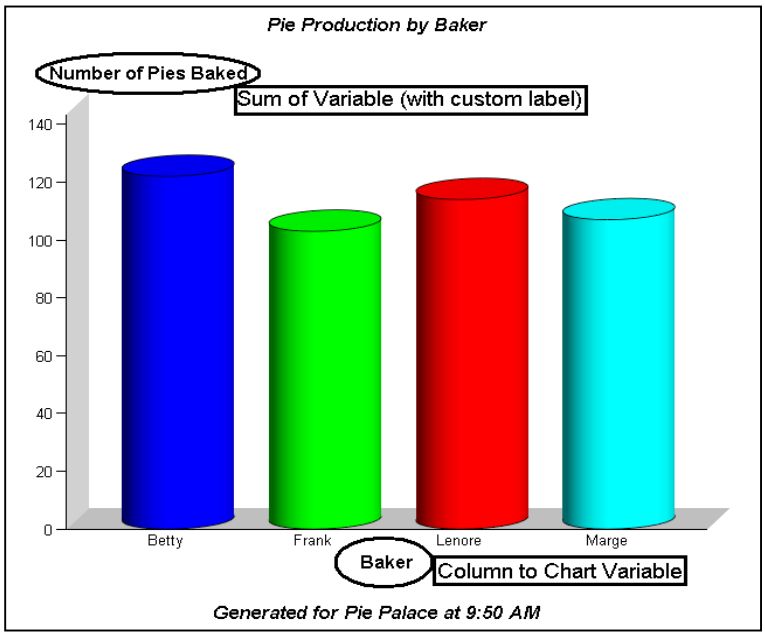
Display 10. Style Sheets in Enterprise Guide

Enterprise Guide can also be used to create graphs. It can be done without having to code cumbersome SAS/GRAPH code. The interfaces are very easy to use and intuitive, such as the bar chart example below:



Display 11: Bar Chart Task in SAS Enterprise Guide

The graph output is attractive and easy to update. Some types of graphs may actually be interactive so that the user can change small things like color, chart type, etc. without waiting for the data to rerun.



Display 12. Example of a Graph in SAS Enterprise Guide

Production runs and scheduling are something that should be discussed with a user's SAS administration. SAS Enterprise Guide use Windows Scheduler to do scheduling. This can work fine for an individual PC, given that the user has administrator permission on the machine. However in a more complex environment, there are many things that must be considered in putting systems into scheduled production.

CONCLUSION

Enterprise Guide is an amazing tool for data analytics and reporting. With a few best practices, users can develop systems that are easier to use, more efficient, and more maintainable.

RECOMMENDED READING

- *The Missing Semicolon Blog® – www.sys-seminar.com*
- *The SAS® Dummy Blog®*

CONTACT INFORMATION

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

Jennifer First-Kluge
Systems Seminar Consultants, Inc.
608-278-9964
train@sys-seminar.com
www.sys-seminar.com