

Paper 404-2013

## Consistent and Organized Analysis: Moving Beyond Pie-in-the-Sky to Actual Implementation via SAS® Enterprise Guide®

Amber Schmitz, Prime Therapeutics

### ABSTRACT

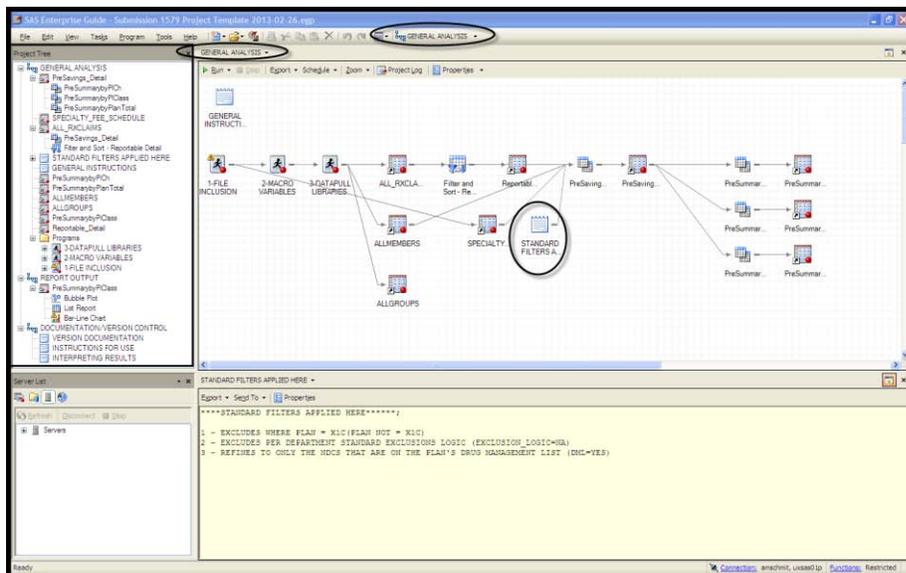
Consistency and timeliness are two goals that every reporting department strives to achieve. SAS® Enterprise Guide® provides tools that support these goals that many analysts overlook. The goal of this paper is to demonstrate the use of SAS Enterprise Guide to construct a master project template that can be used to report standard metrics for various clients. The template is built around SAS Enterprise Guide tools that allow for consistent analysis methods, project organization, and version control documentation. This paper explores: 1) Utilizing program code for efficient data pulls, 2) Using SAS Enterprise Guide tasks for analysis, 3) Exploiting built-in tools such as Notes and Process Flows for template organization and version control, 4) Demonstrating business intelligence via built-in graphs.

### INTRODUCTION

Many companies have transitioned to SAS Enterprise Guide as their preferred SAS interface. While SAS Enterprise Guide may seem like a simplified version of BASE SAS, it is a truly a unique tool that allows for organized analysis and template creation for reporting needs which is especially useful when you have multiple users accessing a master template for report generation. SAS Enterprise Guide maintains the basic functionality of BASE SAS while also offering a broad suite of built-in functionality with drag and drop capabilities. This paper will explore how to work with both program code & SAS Enterprise Guide's built-in functionality to create a project template that produces consistent and timely results. The project template used throughout this paper reflects a template that analyzes pharmacy claims data.

### ORGANIZATION WITHIN SAS ENTERPRISE GUIDE

SAS Enterprise Guide has simplified the complexity of creating an organized analysis project by allowing the user to create multiple Process Flows within a SAS Enterprise Guide project. The incorporation of multiple Process Flows into the layout of the project allows analysts to build easily interpretable analysis projects since the analyst can visually follow the analysis flow. Display 1 below demonstrates a typical SAS Enterprise Guide Session layout. The Project Tree on the left-hand side of the graphic shows three Process Flows housed within the project: General Analysis, Reporting Output, Documentation/Version Control. The titles of each provide a large amount of information as to the contents of each and are essential in building a master project template that can be easily used by multiple team members.



Display 1. Multiple Process Flows Example

**TIP:** Moving between Process Flows is incredibly simple. You can switch between Process Flows via any of the following methods:

- Click on a Process Flow from the Project Tree on the Left
- Select a Process Flow from the Process Flow drop-down list
- Select another Process Flow from the general drop-down list

Another organization tool that every SAS Enterprise Guide user needs to be aware of is Notes. Notes are useful when the need arises to call out specific analysis criteria within the project flow. For example, the note in Display 1 directly specifies the analysis step where standard filters are applied in the template along with identifying exactly what those restrictions are.

**TIP:** Notes can either be stand-alone notes (e.g. GENERAL INSTRUCTIONS) or they can be linked to an analysis step (e.g. STANDARD FILTERS). To link a note, use either of the following methods:

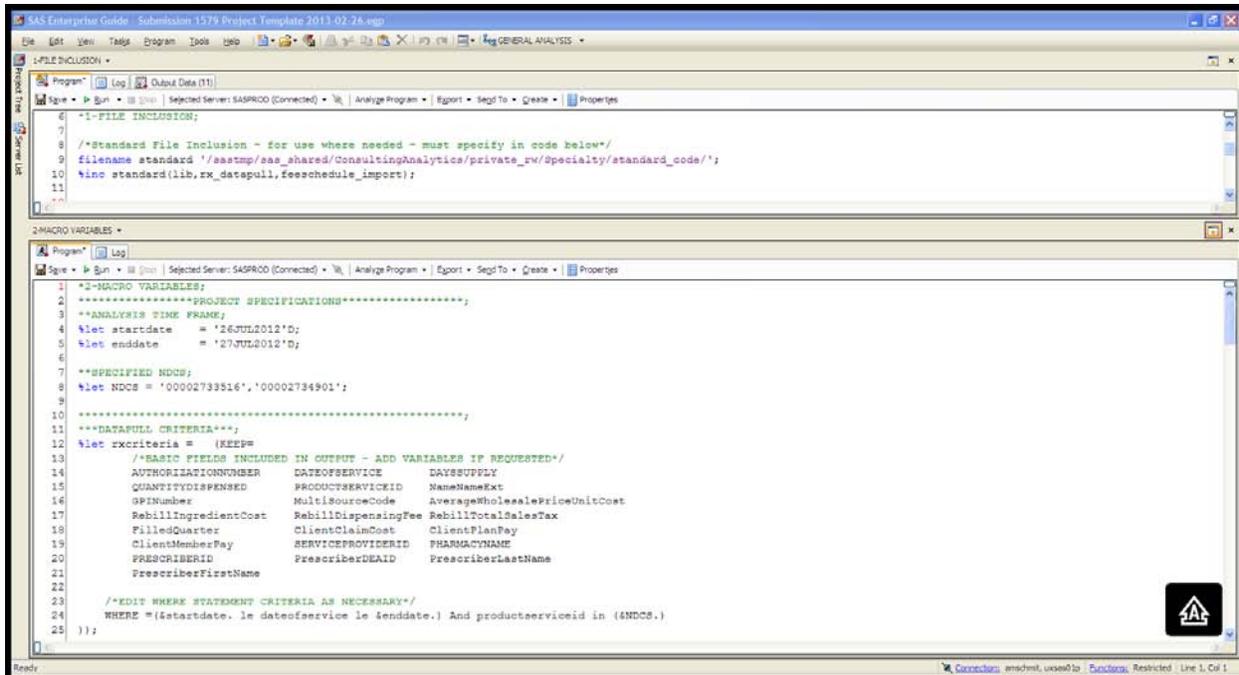
- Right-click on the note and select the 'Link to' option & appropriate item
- Hover on the edge of the note to get crosshairs. Click & drag the crosshairs to the appropriate item

A final organizational tool that SAS Enterprise Guide supplies is the 'Ordered List' task. This tool allows a user to define a specific order of operation for the selected components of the list. While not used in the project template outlined in this paper, I find that the 'Ordered List' function is most useful when a project template requires multiple external reference files to be imported for use.

## CREATING PROCESS FLOW 1: GENERAL ANALYSIS

The General Analysis Process Flow of our template will be used to house the basic analysis pieces of this project. We will complete our data extract via code blocks that utilize macro variables, and then we will employ the built-in data set creation options that SAS Enterprise Guide supplies in order to maintain only the reportable population, join to client specific detail, and create variables that meet standard reporting request specifications. The ultimate goal of this paper is to build a master project template that is robust enough to allow the user to input report specifications on the front-end via macro variables and then have no interaction with the process until the data is ready for the analyst to build the final deliverable report. Between user interactions, the template itself will create and incorporate a standard set of filters and reporting variables for the analyst to use in the final reporting package.

The first portion of the template is built to allow the user to input the request specifications via macro variables. A data extract loop is then created to identify the analysis population for the request. This is all done via program code that utilizes macro variables to build a dynamic data inclusion statement for the analysis. The top pane, or Process Flow pane, in Display 1 demonstrates the process flow used for the data extraction. The local 1-FILE INCLUSION program has an INCLUDE statement that brings in the standard stored program files and reference data sets while 2-MACRO VARIABLES defines the request specific macro variables. These macro variables are then incorporated into the WHERE clause of a KEEP statement which itself is a macro variable. The KEEP statement also contains a base set of reporting variables that is editable to allow for flexibility should more data fields be required for a specific report. The rx\_datapull.sas program builds the RX\_EXTRACT macro by incorporating the defined KEEP statement and specified macro variables. All of these restrictions are applied to the SET portion of the DATA step in order to process only the data that meets the report specifications. This is a great way to improve the efficiency of data pulls from the source data set! The 3-DATAPULL LIBRARIES program then allows the user to execute the RX\_EXTRACT macro for all applicable client libraries with the final data compiled into one data set.



```

1  *1-FILE INCLUSION;
2
3  /*Standard File Inclusion - for use where needed - must specify in code below*/
4  filename standard '/sasmp/sas_shared/consultingAnalytics/private_rw/specialty/standard_code/';
5  %inc standard(lib,rx_datapull,feeschedule_import);
6
7
8
9
10
11
12
13
14
15
16
17
18
19
20
21
22
23
24
25

```

```

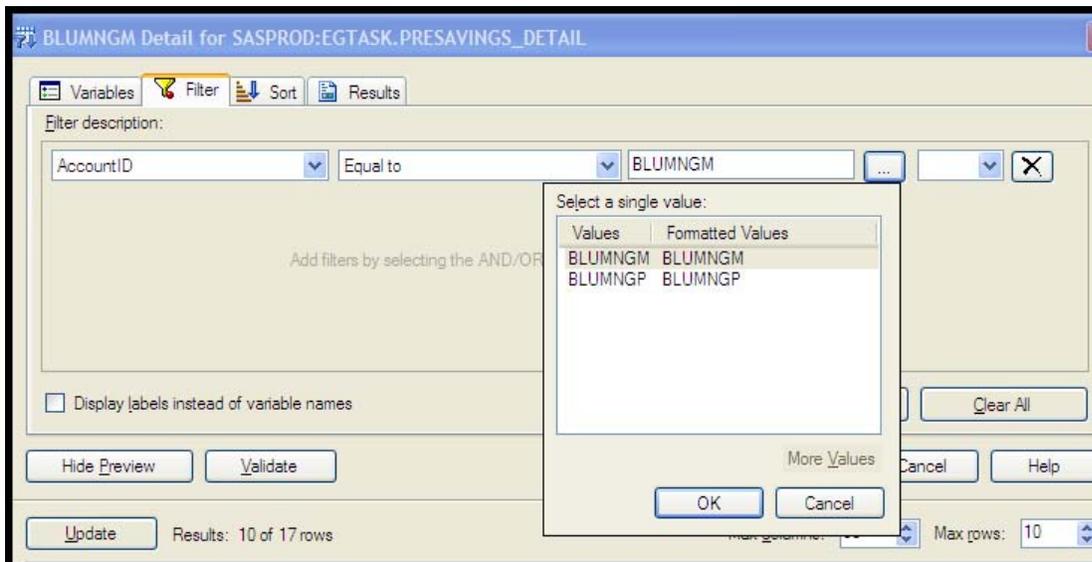
1  *2-MACRO VARIABLES;
2  *****PROJECT SPECIFICATIONS*****
3  **ANALYSIS TIME FRAME;
4  %let startdate = '26JUL2012'D;
5  %let enddate = '27JUL2012'D;
6
7  **SPECIFIED NDCS;
8  %let NDCS = '00002733516','00002734901';
9
10 *****
11 ***DATAFULL CRITERIA***;
12 %let rxcriteria = (KEEP=
13 /*BASIC FIELDS INCLUDED IN OUTPUT - ADD VARIABLES IF REQUESTED*/
14 AUTHORIZATIONNUMBER DATEOFSERVICE DAYSSUPPLY
15 QUANTITYDISPENSED PRODUCTSERVICEID NameNameExt
16 GPINumber MultiSourceCode AverageWholesalePriceUnitCost
17 RebillIngredientCost RebillDispensingFee RebillTotalSalesTax
18 FilledQuarter ClientClaimCost ClientPlanPay
19 ClientMemberPay SERVICEPROVIDERID PHARMACYNAME
20 PRESCRIBERID PrescriberDEAID PrescriberLastName
21 PrescriberFirstName
22
23 /*EDIT WHERE STATEMENT CRITERIA AS NECESSARY*/
24 WHERE =(startdate .le dateofservice .le enddate.) And productserviceid in (&NDCS.)
25 );

```

## Display 2. Macro Variables Example

**TIP:** Programs within a SAS Enterprise Guide project can either be saved only as a piece of the .egp project file (local) or they can be saved as a stand-alone .sas program file that can be referenced in multiple projects (stored). In order to save the program as a stand-alone program, select the 'Save Program As' option from the drop down menu and select the appropriate storage location and program name.

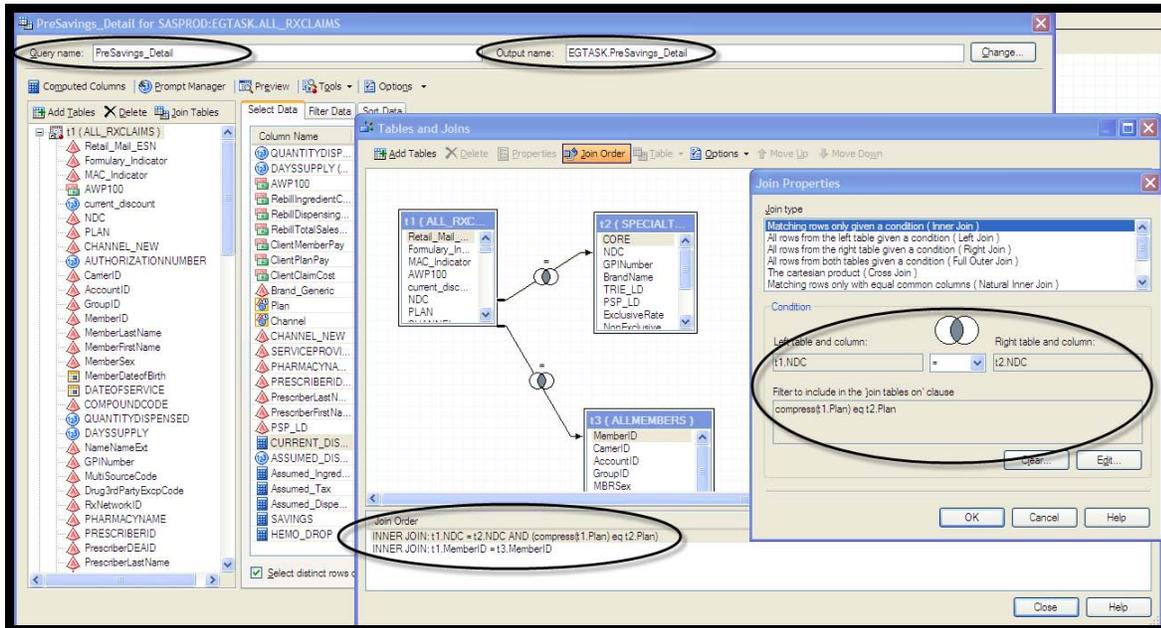
Once the data extract is complete, the data refinement continues via a standard analysis flow which contains all of the data cleansing and standard calculation steps. The most basic method of creating a new data set is via the Filter & Sort task. This task is only available when there is no need to merge tables or create calculated variables. Display 3 demonstrates how to apply a filter to output only the data rows that meet the specified AccountID value which is also the step in the template that maintains only the reportable population.



## Display 3. Filter Example

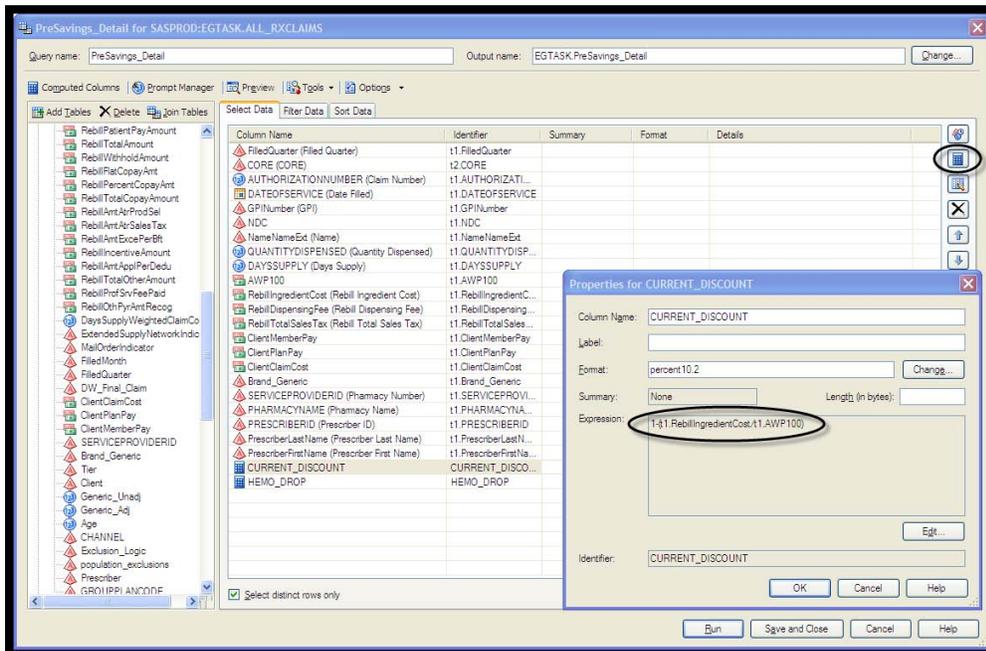
Once the data has been filtered, the Query Builder task is used to construct a more complex dataset. This interface allows us to create a new data set by 1) selecting the necessary variables, 2) joining data tables, 3) calculating

variables. The basis of Query Builder is its drag & drop functionality for selecting the variables in the output data set while also allowing complex joining of data sets. Display 4 demonstrates two different joins with varying complexity. The first is a simple inner join of ALL\_RXCLAIMS (t1) to ALLMEMBEMBERS (t3) on MemberID. The second is an inner join of ALL\_RXCLAIMS (t1) to SPECIALTY\_FEE\_SCHEDULE (t2) on NDC and Plan where the t1.Plan variable is specified via the compress function. In order to utilize the compress function (or any other function), the Join Properties menu must be used from the direct inner join on NDC. These two joins link the client specific information with the claims data.



Display 4. Complex Join Example

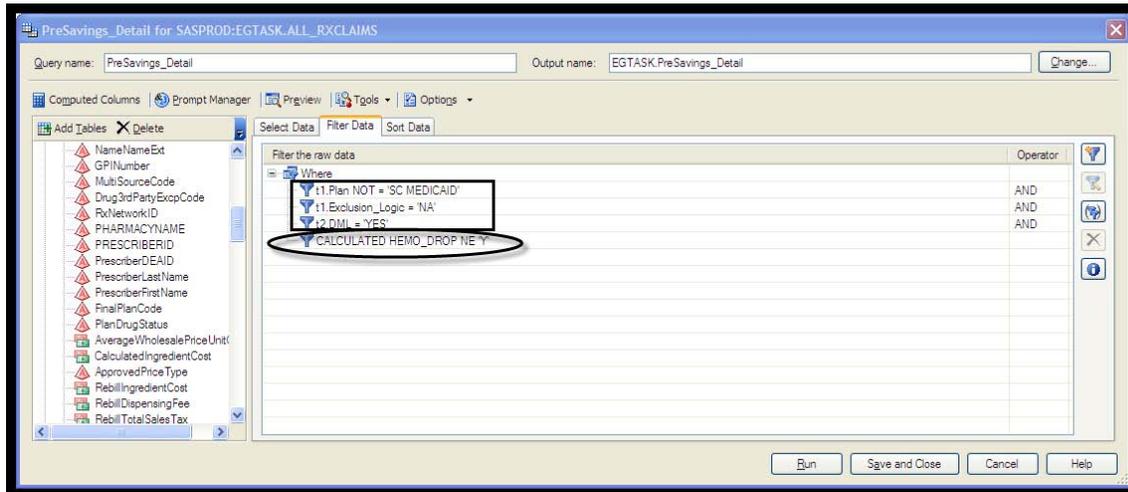
**TIP:** When creating new data sets in SAS Enterprise Guide, be sure to name the Query and the Output data set with a logical name; otherwise, identifying the correct data set when building the project template can become quite complex.



Display 5. Calculated Variable Example

Another useful aspect of Query Builder is its capability to create calculated variables. This project template is setup to create a standard set of calculated variables that can be used in the reporting output. To create a calculated variable, select the blue calculator in the Query Builder box and specify the calculation or case statement as needed. As Display 6 demonstrates, the calculated variable can then be referenced in other calculations or filters in the same Query Builder box by specifying CALCULATED 'variable-name' in the new variable's calculation or filter.

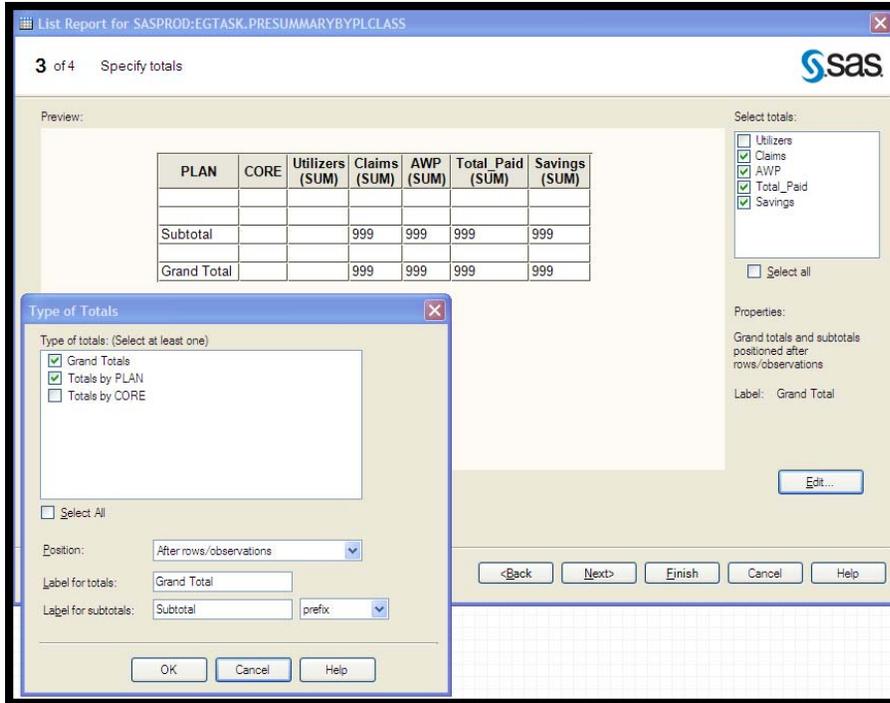
**TIP:** Using the reference to the CALCULATED variable is extremely useful should the actual calculation's formula change in future revisions of the project template. By using the CALCULATED 'variable-name' reference, the current calculation is always referenced instead of a hard-coded formula.



**Display 6. Calculated Variable Reference Example**

## CREATING PROCESS FLOW 2: REPORTING OUTPUT

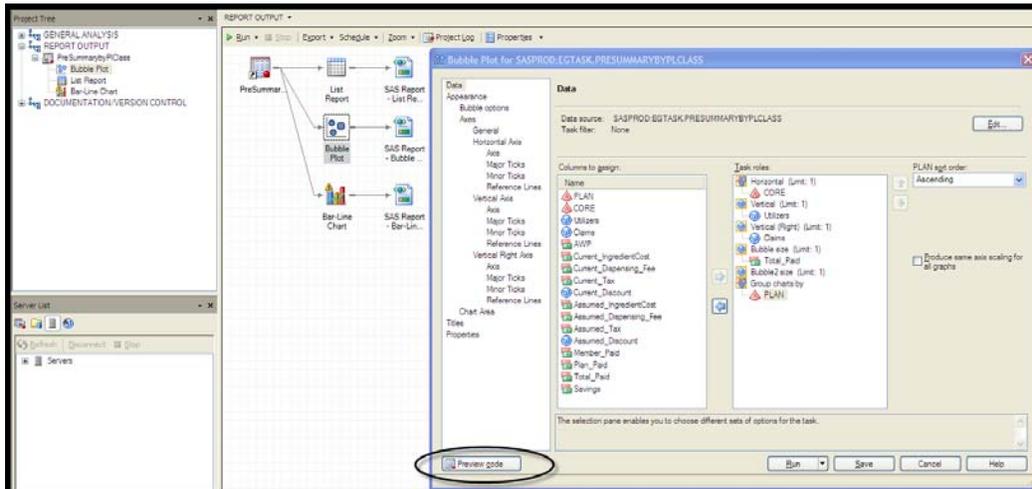
Once the General Analysis process flow is complete, all of the standard data cleansing and reporting calculations have been incorporated for the analyst. The Reporting Output process flow can be built to meet business needs for report content. The master template is structured to create a high-level list report summary and then include graphical breakouts of pertinent data components. Since each of the items are built with SAS Enterprise Guide tools, they are all easily edited or copied to allow for request specific slicing & dicing of the data. The incorporation of the list report along with the graphical representations of the data should appeal to all business requestors since the numerical data is provided in an easily referenced format while the graphical data provides the visual story. The analyst can use the template's reporting output components to build the final report using the Report functionality of SAS Enterprise Guide. The Report creation wizard allows the user to drag and drop each component of the report (listing reports, graphs, etc.) into a coordinated order prior to generating the final report.



Display 7. List Report Wizard Example

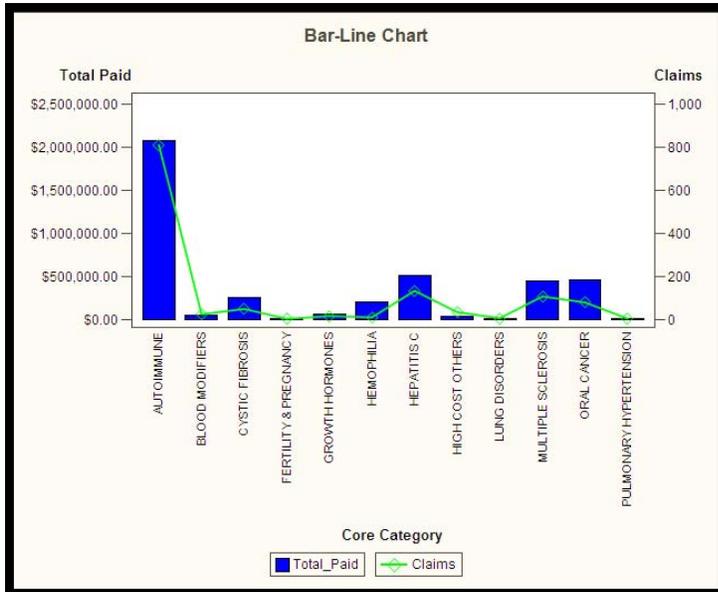
The project template utilizes the List Report Wizard to construct the basic tabular output for the report based on the standard calculated variables. As shown in Display, the List Report Wizard allows you to designate the data source, filter as needed, order the report columns, format output, and add in subtotals as appropriate. The simple interface of the wizard allows the analyst to build the numerical portion of the final report prior to drilling-down into the data layer-by-layer to uncover themes.

After the list report is generated, graphical output can be constructed to give a visual representation of each layer of the data. The drag and drop interface of SAS Enterprise Guide's Graph suite allows analysts to quickly build various types of graphs that best demonstrate the data. Every aspect of the output can be modified such as chart grouping, colors, axis options, and titles. In this template, a standard Bubble Plot is added to the project to demonstrate the Total Paid values along with Claims and Utilizers in each core category. As highlighted in Display 8, the analyst can select the 'Preview Code' button in order to review the code that will generate the graph. Within the 'Preview Code' button there is also an option the 'Insert Code' if specific display options are needed for the graph output.



Display 8. Bubble Plot Interface Example

The final piece of this templated process flow is a Bar-Line Chart. This graph displays the distribution of data across Core categories in the population. The bar demonstrates the Total Paid values while the line builds in the number of claims in each of the Core categories. Further layers of the data can be analyzed by changing the data components in the graph interface. As demonstrated in Output 1 below, every aspect of the graph is editable including the data, titles, colors, and axes in the Bar-Line Chart graph interface.

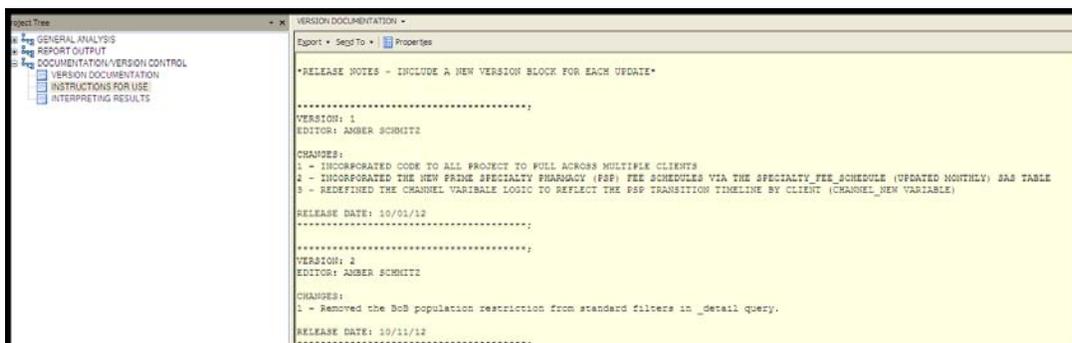


Output 1. Bar-Line Chart Example

Once the analyst builds each piece of necessary reporting output, the Report functionality of SAS Enterprise Guide can be used to construct the final report. The Report option allows the user to drag and drop each component of the final report (listing reports, graphs, etc.) into a cohesive order that tells the story of the data prior to publishing the final report.

### CREATING PROCESS FLOW 3: DOCUMENTATION/VERSION CONTROL

The most crucial component of a master template is documentation of usage and version change control. SAS Enterprise Guide allows for this to be easily implemented. This project template contains a Documentation/Version Control Process Flow that utilizes Notes to incorporate the pertinent information. Documentation of changes to the template is maintained in a consistent manner in the Version Documentation Note. The detail of the release notes are structured in a way that many programmers already incorporate into program code, but now the change documentation applies to the entire project template.



Display 9. Documentation Example

**TIP:** Two other beneficial pieces of information that can be maintained in the Documentation/Version Control process flow are Instructions for Use and Interpreting Results. Each of these topics can be contained within its own Note in the Process Flow. By incorporating both of these items, the need to maintain this information in another document is removed along with the information always being readily accessible whenever an analyst is using the template.

## CONCLUSION

SAS Enterprise Guide is used widely throughout the business world. Becoming more familiar with built-in tasks and organizational tools can greatly benefit many reporting analysts. This paper has explored building a standardized project template that utilizes program code and SAS Enterprise Guide's built-in capabilities to create an organized master project template. Building structured master analysis templates allows for consistent and timely reporting within a team environment while also maintaining flexibility to meet the organization's reporting needs.

## ACKNOWLEDGMENTS

I would like to thank my family, coworkers, and boss for challenging me to share my knowledge.

## RECOMMENDED READING

- *The Little SAS® Book*
- *The Little SAS® Book for Enterprise Guide®*

## CONTACT INFORMATION

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

Name: Amber Schmitz  
Enterprise: Prime Therapeutics  
Address: 5600 West 84<sup>th</sup> Street  
City, State ZIP: Bloomington, MN 55437  
E-mail: [ASchmitz@PrimeTherapeutics.com](mailto:ASchmitz@PrimeTherapeutics.com) or [Akroeger@gmail.com](mailto:Akroeger@gmail.com)

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

Other brand and product names are trademarks of their respective companies.