

Paper 169-2008

Starting Out with SAS® Enterprise Guide®

Pete Bennett, I.T. Training Services Limited, United Kingdom

ABSTRACT

SAS® Enterprise Guide® is an excellent entry point to the SAS® System for users who prefer a point-and-click environment over working with the complexities of program code. In this paper, we look at some of the fundamental concepts and terminology (for example, projects, tasks, and property sheets), and then we concentrate on how to access, manage, and analyze data using SAS Enterprise Guide. Finally we look at ways in which we can share the results and processes we develop.

INTRODUCTION

This paper can only provide a brief introduction to SAS Enterprise Guide as the scope of the product is vast. The intent, then, is to give the audience a taste of what can be done and a feel for how to build projects within the product. Hopefully people will leave wanting to start exploring the software on their own.

The bulk of the paper will be in the form of demonstrations showing how to

- access different data files stored locally and via a SAS® Metadata Repository
- produce tabular and graphic reports
- join multiple tables
- share static reports
- create SAS® Stored Processes to enable users to generate their own reports

These demonstrations will be described in this document with more detail being presented on the day. The author will provide a PDF of the slides which can be downloaded from his website:

<http://www.ittrainingservices.com/papers.htm>.

FUNDAMENTAL CONCEPTS AND TERMINOLOGY

Before we start looking at the demonstrations, it is important that we understand some of the terminology and concepts involved in SAS Enterprise Guide. This section attempts to highlight and explain some of the key terms.

- **Task.**
A task is a specific analysis, activity, or report that you may run against your data (for example, list data, sort, summary table, bar chart, statistical analyses, etc.)
- **Results.**
Output from a task or process flow in the form of an HTML document, an RTF file, a PDF file, a SAS report, or a text file.
- **Process Flow.**
An interactive graphical representation of a set of linked tasks. You can think of them as a series of SAS programs made up of data files, program steps and output.
- **Project.**
A project is a "container" for a group of process flows. These process flows will usually be related in some way although this is not an absolute requirement. The information about the structure of the project is stored in a project file (.egp)

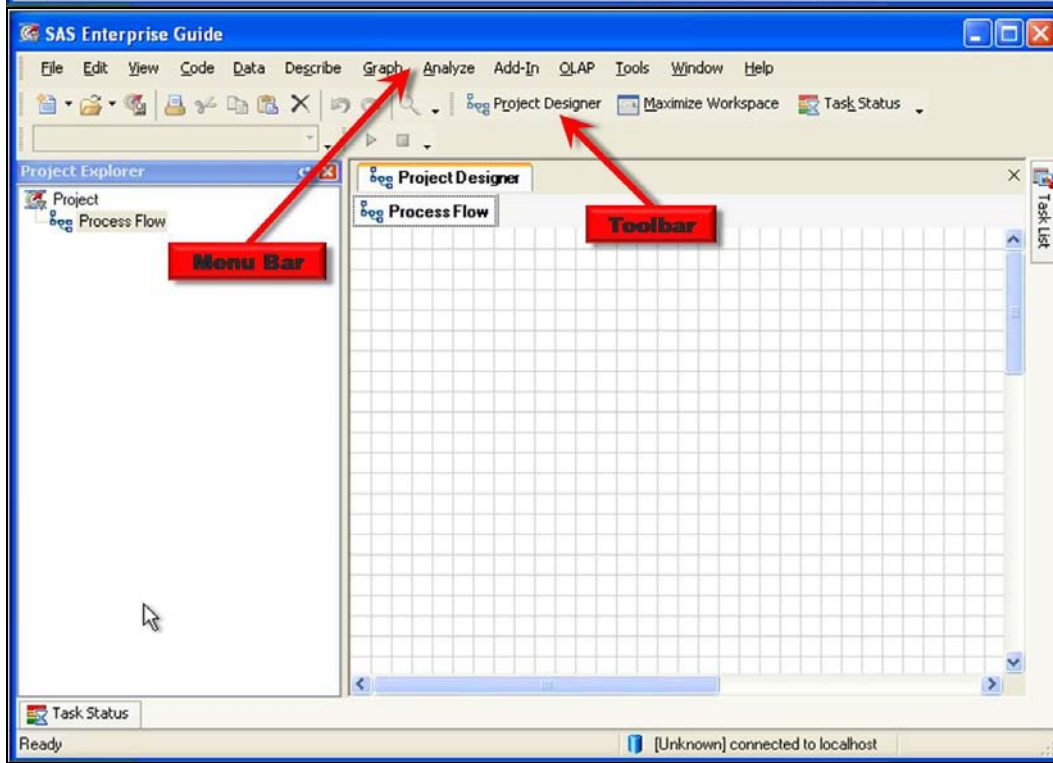
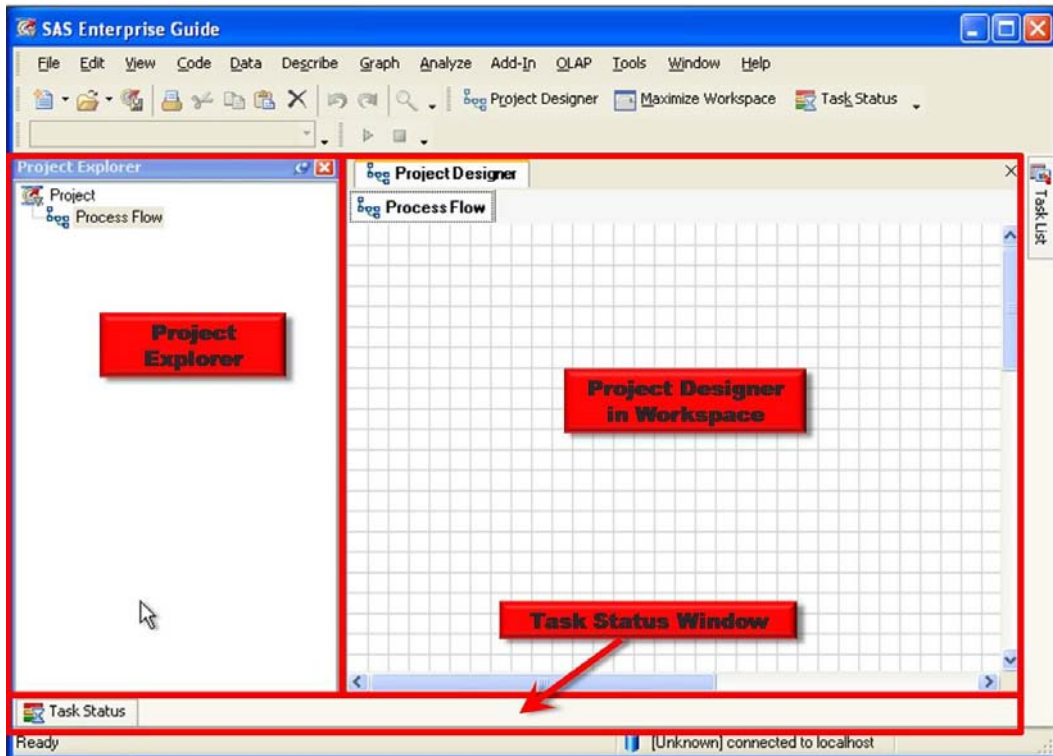
THE SAS ENTERPRISE GUIDE INTERFACE

On starting SAS Enterprise Guide, you will be presented with the windowing environment provided as part of the software. This section explains the features of that environment.

Assuming that you have not customized the layout, you will see two main parts to the window. On the left is the Project Explorer which displays a map of the project in a tree structure. On the right is the Workspace showing the Project Designer open on the Process Flow display. It is here that you will see the project flows as they are constructed.

At the bottom of the window is the Task Status tab. Selecting this opens a window showing what part of the process flow is running.

To the right of the Workspace is the Task List tab. Opening this presents the user with a list of tasks that can be brought into the project.
At the top of the screen you will see the menu bar and toolbar familiar to users of Windows applications. The following images show these features.



ACCESSING, MANAGING AND ANALYZING YOUR DATA

In this section we will be using SAS Enterprise Guide 4.1 to

- access data of different types and from different locations.
- perform simple and complex tasks.
- join multiple tables.
- generate text and graphic reports.

The data we will be using is from a fictitious travel company which operates a number of hotels in different locations in Europe and the United States.

Information is maintained on target and actual occupancy rates for each hotel, the revenue generated and the name of the manager responsible for each country operation.

This data is not intended to be realistic; it has simply been randomly generated to give some data to work with.

DEMONSTRATION 1 – STARTING A PROJECT AND ACCESSING SAS DATA SETS

In this demonstration we will

- start a new project.
- access a SAS data set on the local computer.
- produce a listing of that data.
- save the project.

To see screen shots of the process, download the PDF of the slides mentioned in the introduction. This will be the case for all demonstrations.

1. Start SAS Enterprise Guide and select **New Project**.
2. Select **File ► Open ► Data**.
3. Select the **Local Computer** icon.
4. Select the data set to be used (**targets.sas7bdat**).
5. A data set is opened in a data grid which can be scrolled to examine the data. Close the grid by clicking its close button .
6. The data set is now represented by an icon on the **Process Flow**.
7. Click the **Task List** tab to expand it and select **List Data** under the **Describe** category.
8. The **List Data** wizard is opened in the **Task Roles** section.
Drag **Location**, **month**, and **target** from the list of available variables to the **List variables** role.
Drag **Country** to the **Page by** role. Notice that it will also be added to the **Group Analysis By** role.
9. Select **Options** in the left panel and deselect **Print the row number** check box.
10. Select **Titles** in the left panel, select **Report Titles**, deselect **Use default text** and change the default title to something more appropriate. Repeat the process to delete the default footnote.
11. Optionally select **Preview code** to inspect the SAS code that has been generated. Close the code window after viewing.
12. Select **Run** to run the task and view the results. Close the results tab after viewing the report.
13. Select **File ► Save Project As...** and follow the prompts to define the location and name of the saved project.
(**Note:** the project should be saved at regular intervals to prevent data loss, however this paper will include no further reminders).

DEMONSTRATION 2 – USING NON-SAS DATA

In this demonstration we will add a Microsoft Access™ database to the project.

1. Return to the **Process Flow** and select **File ► Open ► Data**.
2. Select the **Local Computer** icon.
3. Navigate to the project data folder and select the Microsoft Access database (**revenue.mdb**).
4. Select the second icon to open the database as a SAS data set.
(**Note:** using the other icon allows you to read the database directly rather than importing it. However, formats and attributes for displaying the data will be chosen automatically and may be inappropriate or cause problems later. For example, character columns tend to be given a default length of 255 bytes).
5. Select the columns in turn and if necessary change the format by clicking **Display format** and then the ellipsis ... to open a window to select the required format.
6. After selecting the required formats, click **Run**. Close the data grid after viewing the data.

DEMONSTRATION 3 – JOINING TABLES AND COMPUTING NEW COLUMNS

Now we will join the two tables in our project and use the data to compute a new column to show the difference between the target and actual occupancy rates.

1. Right click on the imported revenue data and select **Filter and Query** from the popup menu. The **Query Builder** window will appear.
2. Click **Add Tables** in the **Query Builder** window.
3. On the **Open Data** menu, select **Project** as the location for the additional table.
4. On the **Add from Project** menu, select the data set to add (**targets**).
5. In the **Query Builder**, double click the required columns to add them to the output data set. In the demonstration, we add all the columns from the **revenue** table and **target** from the **targets** table. (**Note:** instead of double-clicking, the columns can be dragged).
6. Click **Join** in the **Query Builder**.
7. In the **Tables and Joins** window, you will see a link has been established between the two tables on one of the variables (**month**). We also need links for **Country** and **Location**. Select one of these in the left table and drag a link to the same variable in the right table. Repeat for the other variable. (**Note:** All joins are inner joins (matching rows only) by default. This is what we want, but if needed we could change the join specification by right clicking the join line and selecting **Modify Join**).
8. Close the **Tables and Joins** window.
9. Click **Computed Columns** in the **Query Builder**.
10. In the **Computed Columns** window, click **New** and select **Build Expression**.
11. In the **Advanced Expression Builder** window, double click the **Occupancy** column, click the **minus** button, and double click the **target** column. The expression should read **REVENUE.Occupancy – TARGETS.target**
Click **OK**.
12. In the **Computed Columns** window, select the new column and click **Rename**.
13. Enter a new name (**Difference**) and click **Close**.
14. In the **Query Builder** window, select the new column and click the **Properties** icon from the right of the panel.
15. In the **Properties** window, click **Change...** to the right of **Format:**
16. In the **Formats** window, select **Numeric** and **PERCENTNw.d**.
17. Click **OK** to close the **Formats** window.
18. Click **OK** to close the **Properties** window.
19. In the **Query Builder** window, click **Run** and view the resulting table.
20. Close the data grid.

DEMONSTRATION 4 – ADD A BAR CHART TO THE PROJECT

Having joined the two tables, we will now produce a bar chart to show how each hotel has performed against its target.

1. Select the joined table from the last query, click the **Task List** tab, and select **Bar Chart (use wizard)** from the **Graph** category.
2. In the **Bar Chart** wizard, select the type of chart to produce (**Vertical Colored Bars**).
3. Select **Task Roles**.
Drag **month** to **Column to chart**, **Location** to **Group charts by**, and **Difference** to **Sum of**.
4. Select **Titles** and amend the default title and footnote as in Demonstration 1.
5. Click **Run**.
6. Scroll through the results and then close the **Bar Chart** tab to return to the **Process View**.

DEMONSTRATION 5 – ACCESSING DATA FROM A SERVER

In the previous demonstrations, the data was located on the local computer. In a modern corporate environment, it is likely that data will be retrieved via a server. This demonstration shows how to

- select a SAS metadata repository and server
 - access SAS data sets from the server
1. First start a new project (**File ► New ► Project**) from the menu bar.
 2. Select **Tools ► Options** from the menu bar.
 3. In the **Options** window, select **Repository and Server** under **Administration**.
Select the required metadata repository (**ITConfig Metadata Repository**).
 4. Select the required server (**SASMain**) and click **OK**.
 5. Select **File ► Open ► Data...**
 6. On the **Open Data** menu, select **SAS Servers**.
 7. In the **SAS Servers** window, select the required server (**SASMain**) and click **Open**.
 8. When prompted, enter your userid and password to log in to the server.
 9. In the **Open from SAS Servers** window, double click **Libraries**.
 10. Select the library to open (**SAS Global Forum Server Files**).
 11. Select the data set to open (**TARGETS**) and click **Open**.
 12. Continue to build the project as in the previous demonstrations, using SAS data sets from the server.
(**Note:** when accessing the **REVENUE** data, you will not need to import it as it is a SAS data set and not a Microsoft Access™ database).

SHARING YOUR RESULTS AND PROCESSES

It is rare that you will be the only person who needs to see the reports you generate. Typically you will need to share them with someone else. This may be by distributing a static copy on paper or via the web or you may want to provide the means for others to produce the reports themselves. This section looks at some of the ways which SAS Enterprise Guide provides for both these approaches.

DEMONSTRATION 6 – SHARING STATIC REPORTS

This demonstration explores the different formats in which static reports (text and graphic) can be created for distribution to other parties. We will just use the Bar Chart we created earlier as an example, but the techniques used could be applied to any report.

1. Right click the **Bar Chart** icon in the **Process Flow** window and select **Properties**.
2. In the **Properties for Bar Chart** window, select **Results** in the left panel.
Select **Override the preferences set in Tools -> Options**.
Select the check boxes for the output types you want to generate. In the demonstration we select **HTML**, **PDF**, and **RTF**.
(**Note:** you can also choose the styles for each type of output if required as well as the format for graphic output. We will leave them unchanged).
Click **OK**.
3. Right click the **Bar Chart** icon and select **Run Bar Chart**.
4. Close the results and you will see icons for each output type. Select each in turn to inspect the output appearance.
5. To make an output object available for others to view, right click the required output icon and select **Export ► Export – xxx Bar Chart** where xxx is the output type (HTML, PDF, or RTF).
6. Choose a location to store the exported file.

Once the exported file has been stored, others can navigate to it and view it. Alternatively, you can add a link to the report on you intranet or email it to the appropriate people.

DEMONSTRATION 7 – SHARING PROCESSES USING SAS STORED PROCESSES

In this demonstration we will look at how we can enable users to run their own reports. We could achieve this by copying the project file to a server location and providing access to the relevant users.

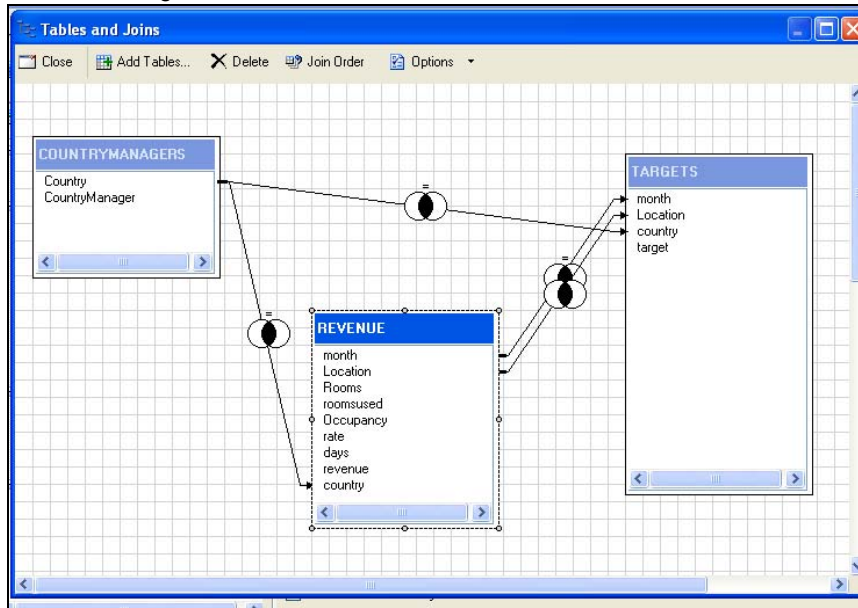
A better solution however is to create a Stored Process which enables users to run processes

- 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

One of the advantages of stored processes is that we can add parameters to prompt users for information so that

they get a report tailored to their specific needs. In the demonstration we will create a stored process to create a summary table for a specific country.

1. In our project, use techniques already demonstrated to add a third table (**CountryManagers**) and join it to the two existing tables as below



2. Once the joined table has been created, select the **Task List** task and choose **Summary Table** from the **Describe** category.
3. In the **Summary Tables** window, select **Task Roles**, drag all the character variables to the **Classification variables** role, and all the numeric variables to the **Analysis variables** role.
4. Select **Summary Tables** and drag variables to define the table as below

5. Right click on each **Mean** statistic header and select **Heading Properties**. Delete the label and click **OK** in the **Heading Properties** dialog.
6. Select **Titles** and amend the titles and footnotes as required.
7. Click **Run**. Close the **Results** tab.

Now we can start creating the Stored Process.

8. Select **File ► New ► Code**.
9. Right click the first query in the last process flow (where we joined three tables) and select **Open Last Submitted Code**.
10. Select and copy the displayed code. Open the **Code** tab and past the copied code.
11. Return to the **Project Designer** tab, right click the **Summary Tables** task icon and select **Open Last Submitted Code**.
12. Copy this code and past it into the **Code** entry after the previous code.
13. Close all tabs apart for the **Project Designer**.
14. Test the Code object by right clicking it and selecting **Run Code on SASMain**.

To convert this code to a Stored Process we need to add a LIBNAME statement. This isn't automatically generated when we create a multi step Stored Process so we will create a dummy single step Stored Process and "steal" it.

15. Right click on the List Data task icon and select **Create Stored Process**.
16. In the **Create New Stored Process** wizard, click **Next** till you get to the fifth screen.
17. Select the **Use custom LIBNAME statement** checkbox.
Copy the code to the clipboard.
Click **Cancel**.
18. Right click the **Code** icon in the **Process Designer** window. Select **Rename** and supply a new name.
19. Right click the renamed **Code** icon in the **Process Designer** window. Select **Create Stored Process...**
20. In the **Create New Stored Process Wizard** enter a name, description, and keywords. Click **Next**.
21. Paste the LIBNAME statement from the Windows clipboard. Add a semicolon at the end of the statement.
Remove the userid and password parameters from the statement (this will prompt users to supply credentials at run time).
22. Scroll down to the **PROC TABULATE** step in the code. Add a where statement as:
where Country="&country";
(**Note:** &country must be in double quotes)
23. Click **Next**.
24. Specify location to store the stored process code (check with your SAS Administrator). Click **Next** until you get to screen 5.
25. Click **Add** and select **Parameters from SAS Code**.
26. Click **Skip this parameter** until the **country** parameter is displayed.
27. Supply a value for the **User prompt** and select the **Required** check box.
28. Select the **Constraints** tab.
29. Enter the list of valid values. Click **Close, Yes, and OK**. Click **Next**.
30. Select **Streaming output**. Click **Next**.
31. Click **Finish**.
32. Right click the new **Stored Process** icon and select **Run Create Summary Table For Selected Country** where **Create Summary Table For Selected Country** is the name you gave the stored process.
When prompted, supply the country to report and view the results.

DEMONSTRATION 8 – ACCESSING A STORED PROCESS IN MICROSOFT EXCEL

Now we've created a stored process we will access it via Microsoft Excel. To do this, the SAS Add-In for Microsoft Office must be installed and access granted to the user. Consult your SAS Administrator.

1. Start Microsoft Excel.
2. On the Excel menu bar select **SAS**. Select **Reports**.
3. Enter **Logon Credentials** if prompted.
4. Navigate to the location of the stored process, select it and click **Open**.
5. When prompted, supply the country for the report and click **Run**.
6. Complete other prompts as requested to specify the Excel sheet for the report.
7. View the report in the Excel worksheet.

The screenshot shows a Microsoft Excel spreadsheet titled 'Book1'. The active cell is F7, containing the value 344579.17. The spreadsheet displays a table with the following data:

Monthly Average					
Country	Country Manager	Location	Target Occupancy	Actual Occupancy	Revenue
		Nice	90%	84%	221846.67
		Paris	88%	81%	550168.75
France	Pierre Grimaud	St Tropez	89%	87%	344579.17

CONCLUSION

Hopefully this paper has demonstrated that SAS Enterprise Guide provides a flexible and powerful tool for generating code to analyze your data and present the results in a variety of ways. Of course, we've only seen the tip of the iceberg, but you should now have enough information to start exploring the software on your own.

RECOMMENDED READING

The Little SAS Book for Enterprise Guide 4.1 by Susan J Slaughter and Lora Delwiche available from SAS Publications

CONTACT INFORMATION

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

Pete Bennett
 I.T. Training Services Limited
 3 Fynamore Gardens
 Calne
 Wiltshire
 United Kingdom
 Work Phone: (44) 1249 476121
 E-mail: prb@ittrainingservices.com
 Web: www.ittrainingservices.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.