

Papers 257-31 and 258-31

SAS® Enterprise Guide® Stored Processes,

Part 1: The Information Consumer's View

Part 2: Creation and Deployment

Marje Fecht, Prowerk Consulting LLC, Cape Coral, FL
Peter R. Bennett, SAS International, Marlow, UK

ABSTRACT

SAS®9 and SAS Enterprise Guide 3.0 provide an array of powerful tools for analyzing data and generating the information that organizations demand in order to address their business needs. This two-part tutorial focuses on Stored Processes within SAS Enterprise Guide. Part 1 presents the view of the "end user" or "information consumer." Part 2 focuses on the technical aspects of creating and deploying Stored Processes.

In [Part 1](#), we explore the flexible and responsive reporting environment and demonstrate benefits such as:

- The user has easy access to Stored Processes in Microsoft Excel and Word via the SAS® Add-In for Microsoft Office.
- Because the processing is done by the Stored Process, consistent answers are provided to all users.
- Parameter-driven Stored Processes allow powerful and flexible "drill-down" capabilities.
- Because Stored Processes use the original external data, the user is not limited by Excel row limitations.
- The user can perform ad hoc analyses of the data, using many of the SAS analytical functions while working in Excel or Word.
- Using the SAS® Information Delivery Portal, all users can access and use the same Stored Processes.

In [Part 2](#), we explore the tasks that are required to create and deploy Stored Processes, including:

- Preparing the environment in SAS® Management Console.
- Defining the metadata repository in the SAS Enterprise Guide Administrator.
- Defining and creating Stored Processes.
- Adding parameters (macro variables) to the appropriate Stored Processes for flexible reporting.
- Defining reporting styles to be used.
- Restricting access to data, reports, and Stored Processes. Authorized users can be defined individually or by group.

INTRODUCTION

Enterprise Guide provides a powerful development environment to create and deploy information for the information consumer. This paper focuses on the use of **Enterprise Guide to develop Stored Processes** that the **information consumer accesses in their chosen environment**. This is just one of many approaches to using Enterprise Guide.

STORED PROCESSES – THE VIEW OF THE INFORMATION CONSUMER

Stored Processes provide the information consumer with a powerful, flexible, and responsive reporting vehicle. And since Stored Processes can be accessed using Office products, the user works with tools they are already familiar with, rather than learning a new tool. Some of the possibilities include:

- Access to Stored Processes in Excel / Word via the SAS Add-In for Microsoft Office.
 - Since the processing is done by the Stored Process, consistent and timely information is provided to all users.
 - Parameter Driven Stored Processes allow powerful and flexible "drill down" capabilities
 - Since the Stored Process uses SAS to process the original external data, the user is not limited to usual Excel row limits.
- The user can perform ad hoc analyses against the data, using many of the SAS analytical functions while working in Excel or Word.
- Using the SAS Information Delivery Portal, all users can access and use the same Stored Processes.
- Using Web Report Studio the user can produce reports against a standardized business view of the corporate data as well as using the same Stored Processes mentioned above.
- Access to the information is controlled centrally via the SAS Management Console.

EXAMPLE 1: SUPPLIER LISTS

We all suffer from those folders of “outdated” information that we statically store on our computers. And having someone run a daily update of *common reports* to be stored as static is not really productive. A better solution is live access to the data you need, in a format you can easily access and understand.

Stored Processes, ideally created in Enterprise Guide, enable you to get up to date information when you need it and how you need it. This first example is quite simple, just to get going. In later examples, we will increase the control and complexity of the reporting environment.

The first example Stored Process report produced by Enterprise Guide is a simple Excel listing of suppliers by country. When the Stored Process is created by the programmer in Enterprise Guide, the output format could also be RTF, PDF, and other possibilities.

Before Stored Processes can be run in Microsoft Excel or Word, the **SAS Add-In to Microsoft Office** must be installed on your PC. Once installed, you are ready to begin running Stored Processes.

To view the supplier listing in Excel,

- Start Excel
- Select **SAS** on the Excel menu bar
- Select **Browse SAS Programs**
- Log on to the **SAS Metadata Server**
- Navigate to and select the folder containing the Stored Processes
- Select the first Stored Process to run (**Supplier Listing**)
- Click **Run**
- The listing appears in Excel.

The screenshot shows a Microsoft Excel window titled "Microsoft Excel - SAS Analysis". The active worksheet is "Suppliers Listed by Country". The data is presented in a structured format with sections for different countries. Each section starts with a header row, followed by a row for the country name, and then a table of supplier records. The table columns are Supplier ID, Supplier Name, Supplier Address, City Name, and Postal Code.

Suppliers Listed by Country				
<i>Country=Australia</i>				
Supplier ID	Supplier Name	Supplier Address	City Name	Postal Code
14162	Health-Sports	142 George St	Mt Vernon	2759
11427	Typhoon Clothing	120 Station Street	Rosehill	2118
10692	Toto Outdoor Gear	44 Norway St Kelburn	Strawberry Hills	2010
<i>Country=Belgium</i>				
Supplier ID	Supplier Name	Supplier Address	City Name	Postal Code
5922	Force Sports	Dr Vandepereleln 139	Wijnegem	2110
<i>Country=Canada</i>				
Supplier ID	Supplier Name	Supplier Address	City Name	Postal Code
16733	Fuller Trading Co.	15 boul Wallberg	Dolbeau	
16814	CrystalClear Optics Inc	4697 rue Beaubien E	Montréal	
<i>Country=Germany</i>				
Supplier ID	Supplier Name	Supplier Address	City Name	Postal Code
6246	Schroeder Trim GmBH	Kulmbacher 9		95460
6991	SnowSports International GmBH	Olof-Palme-Straße 13		60439
1037	Meister Pharma & Chemie GmBH	Worringerstr. 1		50668

In its present state, the Stored Process does not enable interaction. However, since the report is displayed in Excel, the user is free to enhance the report if needed without affecting the source data.

EXAMPLE 2: SALES CHANNEL ANALYSIS

In the first example, the Stored Process fully specified all reporting parameters. Often, Stored Processes are generalized so that the information consumer can identify reporting parameters such as time frame, region, format, etc. The Sales Channel Analysis demonstrates a simple parameter driven report.

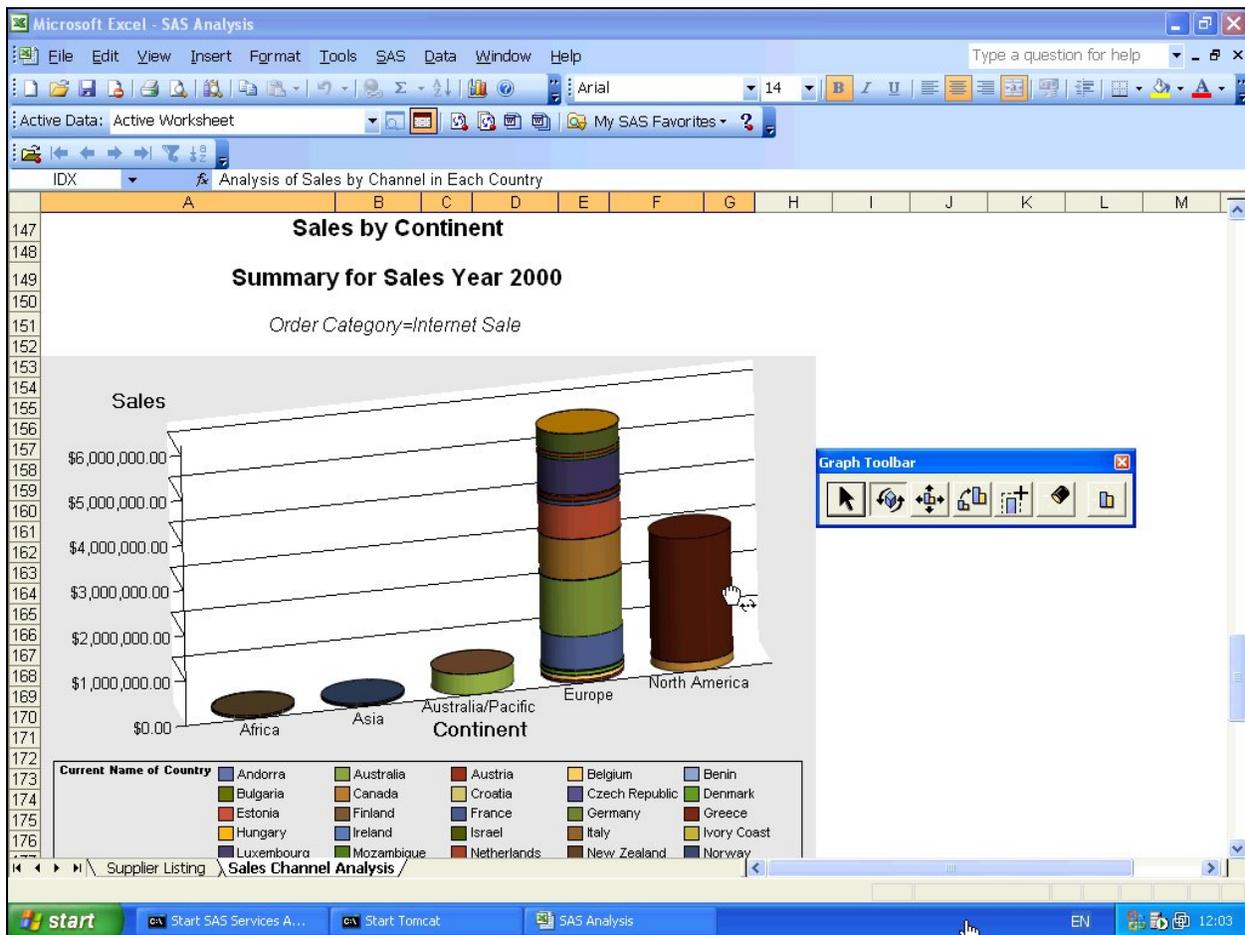
From Excel,

- Select **SAS** on the Excel menu bar
- Select the Stored Process, **Sales Channel Analysis** and **Run**
- In the dialog that appears, choose the year to analyze and **Run**. Notice that each Stored Process has placed its output on a separate Excel sheet
- Scroll to the bar chart and view the pictorial representation of the sales channels
- Right click on the chart and select **Graph Toolbar**. Notice the functionality that is available for altering the appearance of the chart to meet your needs.

The screenshot shows a Microsoft Excel window titled "Microsoft Excel - SAS Analysis". The active worksheet is "Analysis of Sales by Channel in Each Country". The report content is as follows:

Analysis of Sales by Channel in Each Country					
Summary for Sales Year 2000					
Continent Name Africa	Sales Channel				
	Catalog Sale		Internet Sale		
	Amount	%	Amount	%	
Country					
Benin			\$95	100	
Ivory Coast			\$2,244	100	
Morocco	\$1,467	100			
Mozambique	\$465	52.44	\$422	47.56	
Senegal	\$1,323	100			
South Africa	\$9,552	21.62	\$34,633	78.38	
Tunisia	\$484	22.06	\$1,710	77.94	
All	\$13,291	25.37	\$39,103	74.63	

Analysis of Sales by Channel in Each Country					
Summary for Sales Year 2000					
Continent Name Asia	Sales Channel				
	Catalog Sale		Internet Sale		
Country					



While this example did enable user input (the year), all other reporting parameters were pre-defined as part of the Stored Process.

EXAMPLE 3: AD HOC REPORT GENERATOR

Stored Processes can let the user "design" a customized report via a simple parameter menu. In the past, this would be accomplished by passing the requirements to the IT department and waiting for the report to be defined and produced. Probably this would have involved several iterations until the desired results were obtained.

From Excel,

- Select **SAS** on the Excel menu bar
- Select the Stored Process **Ad Hoc Report Generator** and **Run**
- In the following dialog, select the parameters to define your report. Notice that all parameters in the example are required. However, Stored Processes can be created with a mix of required and optional parameters.

Ad Hoc Report Generator

General

Enter year to report: 2002 *

Value to use for page break: NONE *

What value do you want to summarize?: Order Value *

How do you want to summarize the data?: Total *

Row Dimension: Country *

Do you want a row total?: YES *

Column Dimension: Customer Gender *

Do you want a column total?: YES *

Supply a report title: My Ad Hoc Report *

(* denotes required parameter)

sas

Run Cancel

- Select **Run**.

Again, note that each Stored Process creates a separate sheet in your Excel Workbook, so you have all of the results available.

The screenshot shows a Microsoft Excel window titled "Microsoft Excel - SAS Analysis". The active worksheet is "Sales by Customer Gender". The pivot table is titled "Sales by Customer Gender" and is based on data for the year 2000. The pivot table is structured as follows:

Current Name of Country	Customer Gender		All
	Female	Male	
	Order Value	Order Value	Order Value
	Sum	Sum	Sum
Andorra	6798.7	5278.5	12077.2
Australia	1712086.48	1840875.06	3552961.53
Austria	41204.63	57983.7	99188.33
Belgium	629124.64	916676.36	1545801
Benin	.	95.2	95.2
Bulgaria	382.9	548	930.9
Canada	144851.65	188547.35	333399
Croatia	2315.7	91	2406.7
Czech Republic	14887.1	.	14887.1
Denmark	593203.71	560595.33	1153799.04
Estonia	240	451	691
Finland	28922.7	18449.92	47372.62
France	3169740.56	3687762.43	6857502.98
Germany	4544320.69	5187528.56	9731849.25
Greece	.	7135.6	7135.6
Hungary	966.9	4262.3	5229.2

The taskbar at the bottom shows the Start button and several open applications: "Start SAS Services A...", "Start Tomcat", and "SAS Analysis". The system tray on the right shows the language set to "EN" and the time as "12:05".

If you prefer to produce your reports in **WORD** instead of Excel, the steps are identical to those outlined for Excel. Note that after producing your reports in WORD, you may need to adjust the pagination by manually inserting page breaks to meet your needs.

Analysis of Sales by Channel in Each Country
Summary for Sales Year 2002

Continent Name Africa	Sales Channel			
	Catalog Sale		Internet Sale	
	Amount	%	Amount	%
Country				
Benin	\$608	48.97	\$633	51.03
Ivory Coast	\$267	42.73	\$358	57.27
Morocco	\$1,297	92.16	\$110	7.835
Mozambique	\$189	100		
Senegal	\$727	100		
South Africa	\$32,369	43.15	\$42,651	56.85
Tunisia			\$138	100
All	\$35,459	44.60	\$43,800	55.39

Page: 1

EXAMPLE 4: ACCESSING SAS DATA IN MICROSOFT OFFICE

In addition to being able to run Stored Processes, the SAS Add-In to Microsoft Office allows the Excel user to work with SAS data files. The real bonus of this feature is that you can access data **in excess of the Excel maximum row limit!** In this example, we will explore a SAS data table with 951,669 rows.

- On the Excel menu bar, select **SAS** → **Open SAS Data Source**
- Login to the SAS Metadata Server
- Select **Servers** → *the server to use* → **SAS data library** → **SAS table** → **Open**
- Use the **SAS Data Analysis** toolbar to explore the data



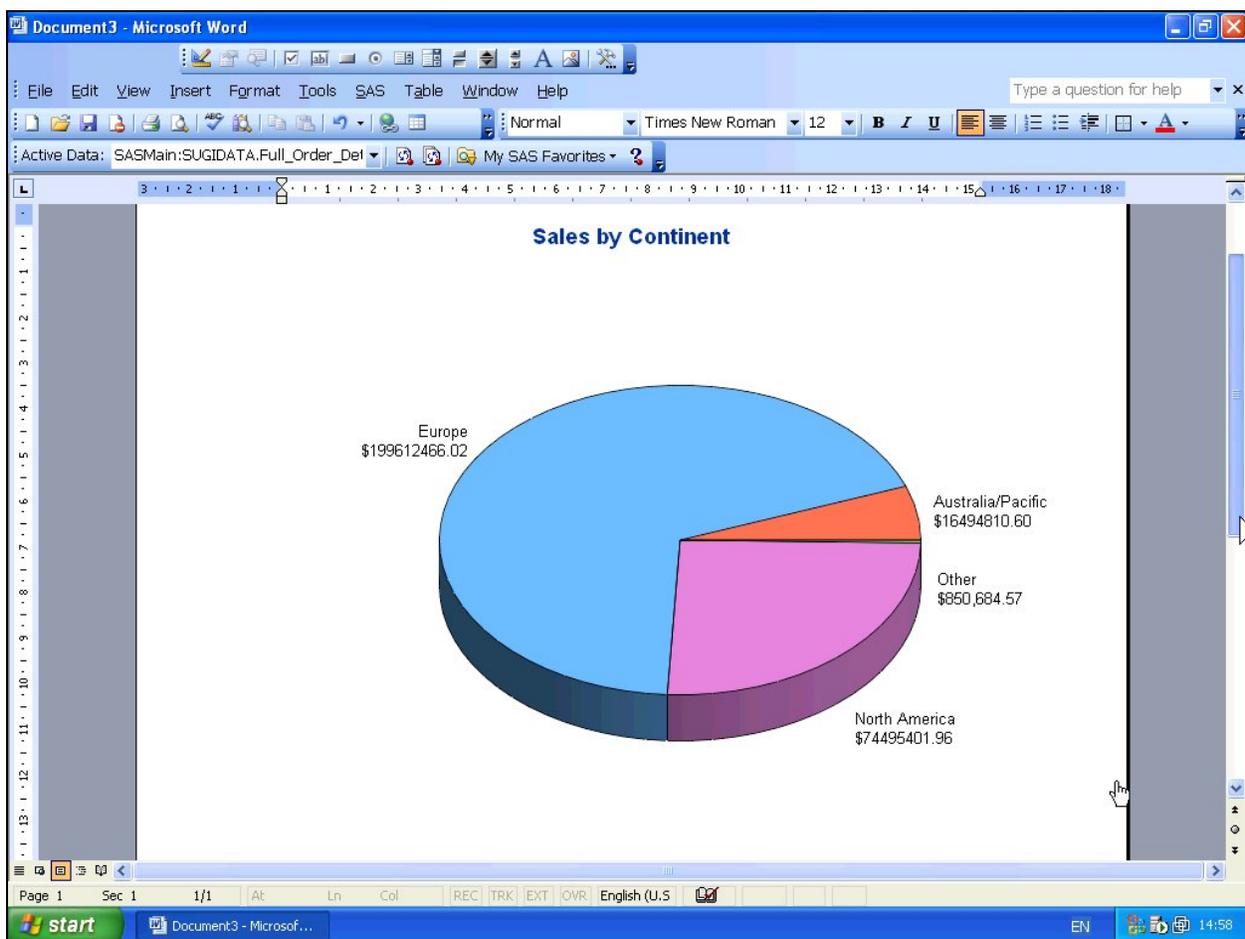
- Initially, just the first 5000 rows are displayed. To view more rows, click the first right arrow.
- If you want to move to the end of the table, click the second right arrow, which will show you that there are 951,669 rows in total.

The **SAS Data Analysis** toolbar enables you to subset and manipulate the data.

- To subset, select the filter tool (funnel shape)
- Using the dialog, select a subset :
country = Netherlands.
This will produce a view of 62,533 rows, but will in now way impact the original data.
- To sort the data, select the sort tool and specify the sort columns in the dialog.

In addition to viewing the data, you have the ability to analyze the data in MS WORD or EXCEL. For variety, switch to WORD.

- Select **SAS** from the WORD menu bar
- Select **Browse SAS Programs**
- Expand the **SAS Tasks** node and select **Graph**
- Select **Pie Chart**
- The **Pie Chart property** sheet will open.
Note that the property sheet is the same as used in Enterprise Guide.
- On the property sheet, select **Simple Pie**
- Select **Task Roles** and drag the variables to the appropriate roles
 - **Column to Chart**
 - **Sum of**
- Provide title and footnote text by selecting **Titles**
- **Run.**

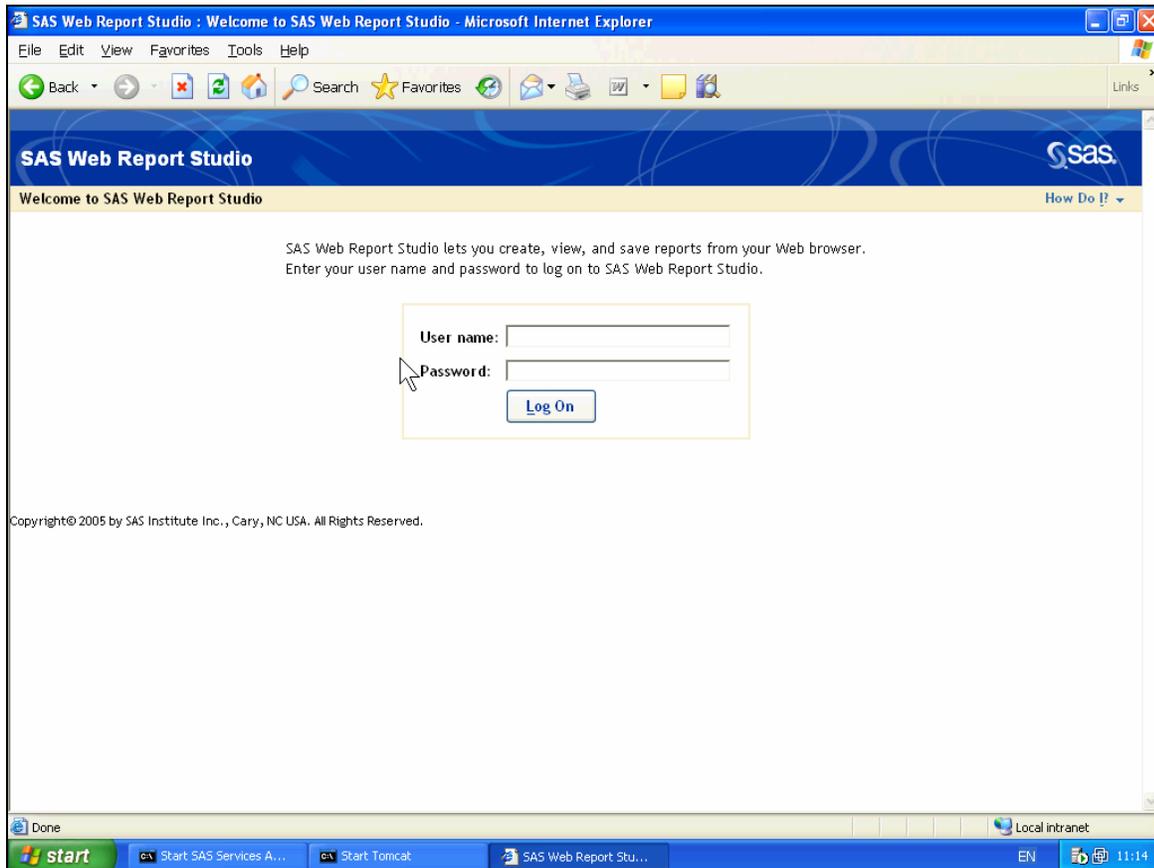


EXAMPLE 5: RUNNING STORED PROCESSES USING SAS WEB REPORT STUDIO AND SAS INFORMATION DELIVERY PORTAL

Stored Processes are available to run from many environments, aside from Microsoft Office. If you have SAS Web Report Studio, or the SAS Information Delivery Portal, these environments can be customized to your preferences and can be used as your information delivery channel.

To produce your report in SAS Web Report Studio,

- Open SAS Web Report Studio and login



- Select **Open Report**
- Navigate to the **Stored Processes** Folder
- Select the Stored Process to run
- View the results, by selecting **Reports → Open**.
- Repeat these steps for other Stored Processes.

The **SAS Information Delivery Portal** gives you the ability to organize information according to your needs and preferences. Before accessing the Stored Processes, add a custom portal page.

- Start the SAS Information Delivery Portal and log in
- Select **Options → Add (under Pages)**
- Add a name and description for the page
- Select **Add**, and **Done**.

Now add a Portlet (Container) for the Stored Process links.

- Select **Options → Portlet** (under Current Page)
- Add a name and description for the portlet
- Select **Add**, and **Done**.

Now you are ready to add the links.

- Select the **Edit** icon on the portlet



- Click **Add Items**
- Use the **Search** tab to locate and select the SAS Stored Processes
- When finished selecting, Select **Add**, and **Done**.

You can reorder the items by using the arrows. When finished, select **OK**. You are now ready to run the Stored Processes.

SAS Information Delivery Portal - Microsoft Internet Explorer

File Edit View Favorites Tools Help

Back Forward Stop Home Search Favorites Refresh Print Mail Stop Taskbar Links

SAS Portal Home SUGI Options Search Log Off Help

SUGI Stored Processes

Click here to refresh collection.

- Supplier Listing**
List of Suppliers Names and Addresses grouped by Country
Foundation/BIP Tree/ReportStudio/Shared/Reports/StoredProcesses
- Sales Channel Analysis**
Analyze sales channels by year, continent and country
Foundation/BIP Tree/ReportStudio/Shared/Reports/StoredProcesses
- Ad Hoc Report Generator**
Generate ad hoc reports on Order Details data
Foundation/BIP Tree/ReportStudio/Shared/Reports/StoredProcesses

Local intranet

start Start SAS Services A... Start Tomcat SAS Information Deliv... EN 11:26

EXAMPLE 6: USING AN HTML APPLICATION IN YOUR BROWSER

SAS Stored Processes are not restricted to SAS client applications. In this example, an HTML application has been created to select and run the Stored Processes. Frames are used to surface a selection menu which is the parameter interface for the selected report.

For this example, when you start your browser and open the application, you would see a selection menu, containing the available Stored Processes. Select the Stored Process you want to run, and supply the parameters as shown in the below application. Notice that the available Stored Processes are in the upper left of the window. When you select **Run**, your report displays in the frame on the right.

Sales by Gender
Using data for 1999

Continent Name Africa

	Customer Gender		All
	Female	Male	
	Order Value	Order Value	Order Value
	Sum	Sum	Sum
Current Name of Country			
Benin	.	10.90	10.90
Ivory Coast	98.50	36.60	135.10
Morocco	244.70	746.80	991.50
Mozambique	.	1831.50	1831.50
Senegal	83.60	2512.80	2596.40
South Africa	26376.56	48089.48	74466.04
All	26803.36	53228.08	80031.44

PART 1 – SUMMARY

In Part 1, we have focused on the powerful reporting and analytical environment available to the information consumer, as a result of Enterprise Guide Stored Processes. Not only is information available on demand, but up-to-date data is as well, for those times that further analytics and exploration is required.

In part 2, we will focus on how to create what the information consumer needs!

ENTERPRISE GUIDE

STORED PROCESSES

Enterprise Guide 3.0 provides a powerful environment and toolset for end-to-end reporting. Since the information consumer may not use Enterprise Guide for their reporting environment, Enterprise Guide also provides the flexibility to create Stored Processes that enable access to the data and reporting in an alternate delivery channel.

Stored Processes can be designed to create reports in a variety of formats depending on requirements

- Static reports
- Parameter driven reports
- Drillable reports.

Reports produced using Stored Processes, can be designed for delivery

- as standard HTML (**Static only**)
- via Microsoft Office products (Word, Excel) using the SAS Add-In for Microsoft Office
- via the SAS Information Delivery Portal
- via SAS Web Report Studio.

PREPARING THE ENVIRONMENT

Before development or reporting or Stored Processes can begin, there are tasks to complete including the setup of the operating system environment and the SAS Metadata environment. Although this may seem to add complexity to the process, typically most of the effort will only be expended once for a set of related projects as they are likely to use the same resources and environment. Benefits of this approach include control of access to the corporate data and computing resources.

Some key concepts we will reference in our discussion include:

- **METADATA SERVER**
The SAS[®] Metadata Server stores information about servers, users, and Stored Processes to provide to client applications. The information is stored in Metadata Repositories.
- **STORED PROCESS**
A Stored Process is simply a set of SAS code that is available for execution by client applications.
- **STORED PROCESS SERVER**
A Stored Process Server executes Stored Processes for client applications.

The **Administrator** will need to carry out the following tasks when setting up the **operating system environment**.

- Define user accounts and groups
- Define physical locations for
 - Enterprise Guide projects
 - Stored Process code
 - HTML applications
- Grant permissions, as appropriate
- Copy development HTML code into the production location, when ready
- Start the **SAS Services Application**
 - **Start → Programs → SAS → *profilename* → Start SAS Services Application**
- Start **Tomcat** (or alternative).
 - **Start → Programs → SAS → *profilename* → Start Tomcat.**

Using the SAS Management Console, the **administrator** will define **the metadata**:

- Define the **Metadata Repository**
- Add libraries into the **Data Library Manager**
- Import tables into the **Data Library Manager**
- Create a location for the Stored Process metadata, in the **Stored Process Manager**
- If using **SAS Web Report Studio**, copy the metadata for completed Stored Processes from the development folder into the Web Report Studio Shared folder in the Stored Process Manager.

DEVELOPMENT EFFORTS

As with any technology solution, development of the reports and Stored Processes benefits from careful planning, and proper preparation of the EG environment.

Requirements gathering

It is easy to send a report to someone, when requested, but deploying information to an entire organization benefits from up-front planning, including

- Report requirements – determine requirements such as
 - Source data
 - Segmentation or summary levels
 - Granularity, for investigative purposes
 - Further tools required to explore the data
 - Deployment requirements – which delivery methods are to be used? (HTML, Web Report Studio, SAS Information Delivery Portal, Excel, Word, etc.).
- Development requirements
 - Source data format, location, and access
 - Metadata
 - Execution location.
- Security Requirements
 - Who requires access to top-level reports?
 - Which users are permitted access to more granular data, for further exploration?

GETTING STARTED WITH ENTERPRISE GUIDE 3.0

Our focus will be on defining and creating the reports, Stored Processes, and applications that the Information Consumer will use. Note that the example reports are displayed in Part 1.

PROGRAMMER ROLE: REPORTING VIA EG

The Programmer's role will be similar to their role in preparing standard reporting. However, a portion of the efforts will focus on the extensibility of the reporting. For example, the Stored Processes will enable the user to produce further analyses and reports, beyond those originally provided. As the programmer, you will

- Perform any necessary data manipulation (e.g. Joins)
- Generate reports & Graphs in Enterprise Guide
- Create Stored Process to be used by the end user
- Define reporting styles to be used.

EXAMPLE 1: CREATING SUPPLIER LISTS

In this first example, we produce a simple listing of supplier names and addresses by country, using data stored on the BI server. When the user accesses the report, they will always enjoy the most current data (as opposed to a static report created in some batch process). The goal is to create a Stored Process that produces the report whenever it is run. The steps for the report production are provided here, and the Stored Process is created in a later step.

Creating the report:

- Start Enterprise Guide and open a new project
- Select the Repository and Server (**Tools → Options → Administrator**)
- Add the data to the project
 - Select **Open from SAS Server/Binder**
 - Review the data and close the viewer
- Create the report
 - Open the properties sheet (**Describe → List Data**)
 - Drop the variables to the desired roles
 - Select appropriate options, titles, footnotes, etc
 - Run the task and view the results in Enterprise Guide and in the browser
- Save the project.

The Supplier report has been created. Now we need to create a Stored Process for access by the user.

Creating the Stored Process:

- In the **Process Flow** window, right click on the List Data node
- Select **Create Stored Process** and assign a name, description and key words.
- Step through the wizard to assign
 - Metadata location
 - Execution environment
 - Library assignment
 - Parameters (we will use default for this report)
 - Output options
 - HTML-based user interface.

To test usage of the Stored Process and the HTML interface which was generated,

- Use Windows Explorer to navigate to the folder where the new interface was created
- Double click on the HTML file
- Select **Run**
- Login when prompted and view the results.
Since the Stored Process always processes the most current data, your results are always up to date.

EXAMPLE 2 – SUMMARY SALES REPORT AND BAR CHART

The users are accustomed to receiving a summary sales report comparing countries and channels. Typically, the report would be made available on a periodic basis. It is desired to make the information available when needed, with the most current data available. Further it would be nice to publish the report complete with a graphical representation of the sales figures.

In this example, we will produce the summary report, and a supporting graph. In the next example, we will create both together as a Stored Process.

Creating the report:

- Add the data to the project
 - Select **Open from SAS Server / Binder**
 - Navigate to the data to be used
 - View the date when loaded, for confirmation
 - Close the data viewer.
- Generate the report
 - Select **Describe → Summary Tables** to open the properties sheet
 - Drag the appropriate variables to the desired roles
 - **Analysis Variables** role
 - **Page** by role
 - **Classification variables** row
 - Define the layout for the summary table
 - Select **Summary Tables**
 - Drag the variables to the position they will appear in the summary table
 - Select the statistics to display
 - Specify the formats for the data cells
 - Customize the column and row headings
 - In turn, click on each columns and row and select **Heading Properties**
 - Enter an appropriate header. NOTE: a blank header suppresses the header from appearing in the table
 - Right click in the **Box Area** and select **Box Area Properties**
 - Select **Show Page Heading**.
 - Customize the report title and footnote
 - **Run** and view the report.

Enhance the Summary report with a bar chart showing sales by continent, with division on country within continent. Each channel will be displayed in a separate bar chart.

The data are the same as the previous example.

To produce the chart,

- Select the order date used in the previous example
- Double click the **Bar Chart** task and select **Stacked Vertical Bar**
- Assign columns to the appropriate task role, including **Column to Chart**, **Stack**, **Sum of**, and **Group Charts by**
- Select the **Layout** property sheet and change the **Shape** to **Cylinder**
- Select the **Horizontal Axis** property sheet and supply an appropriate label
- Select the **Vertical Axis** property sheet and supply an appropriate label
- Select the **Titles** property sheet and change the text
- RUN and view the chart.

EXAMPLE 3 – CREATING A PARAMETER DRIVEN STORED PROCESS

Once the users work with your Stored Processes for a while, you will find that they request extensibility and customization. Using the Summary Report and Bar Chart from the previous example, we will combine both into a single Stored Process, and provide the user with the option to specify parameters.

First, you will create a **Code item** to store the **LIBNAME**. In previous examples, this was not required since when a Stored Process is created from a task in the project, a **LIBNAME** statement is automatically created. But when creating a Stored Process from scratch, the **LIBNAME** statement is not automatically assigned. Time and effort is saved if the statement is copied from an existing Stored Process.

- Open the **Supplier Listing** Stored Process
- Select **Preview code**
- Locate the **LIBNAME** statement and copy it to the Windows clipboard
- Cancel out of the Stored Process window
- Select **File → New → Code**
- Paste the **LIBNAME** statement into the code window
- If required, modify the **LIBNAME** statement (for example, check that the semicolon is included at the end of the statement)
- Save the **Code item** to the local computer
- Close the **SAS Code Editor** window and the Stored Process.

Next, create the basic Stored Process.

- Select **File → New → Stored Process**
- Assign a **Name**, **Description** and **Keywords** for the Stored Process
- Click **Next**
- Click **Insert SAS Code**
- Select **Project**
- Select the **LIBNAME** entry
- Click **OK**
- Move the cursor to a new line after the code
- Repeat for the **Summary Table** and **Bar Chart** entries
- Continue creating the Stored Process as in the earlier example
- Test by producing the reports.

Now that the Stored Process is working, we are ready to add parameters. Our focus will be on subsetting the data for a particular sales year and then including a subtitle to show the selected year.

- Open the code for editing
 - Right click the Stored Process node and select **Open**
 - Select **Preview code**
 - Expand the code window to a useful size
 - Select **Edit mode**.
- Locate the first **TITLE** statement and add a second title line including a parameter for the sales year. **Remember to use double quotes so that the parameter can be resolved.**

```
Title2 "Summary for Sales Year &year";
```

- Locate the **PROC TABULATE** step and add a **WHERE** statement
- Locate the **PROC SORT** step in the Bar Chart section of the code and
 - add **WHERE** statement

```
Where year(order_date) = &year;
```

- add **Order_Date** to the list of kept variables
- Locate the **TITLE** statement for the Bar Chart and add a second title line to show the sales year
- Close the **SAS Code Editor**
- Now we need to define the parameters to the Stored Process:
 - Select the **Parameters** property sheet
 - Click **Add**
 - Select **Parameters from SAS Code**
 - Skip the **dsname** parameter
 - Enter a prompt for the **year** parameter
 - Check the **Required** box and supply a default value
 - Select the **Constraints** tab
 - Select **List of values** from the pull down menu
 - Enter the years **1998** to **2002** into the **List Values**
 - Click **Add** to go to the next parameter
 - Skip the remaining parameters
 - Close the **Add Parameters** dialog
 - Select **Save and Run**
 - Select a year to display and click **Run**
 - View the results.

EXAMPLE 4 – CONVERTING EXISTING SAS MACRO CODE TO A STORED PROCESS

In the past, to meet the demand to run ad hoc reports for the user community, we created a complex SAS macro program which contained a number of **%let** statements used to specify the content and layout of the report. By converting it to a parameter-driven Stored Process, we can make the program available to the users so that they can run it, thus freeing up valuable development time for our programmers, and providing a quicker turnaround for the users.

What did the OLD way look like? Remember this required the programmer or somebody to input values into the program prior to running. We all know how easy it is to *accidentally type over a semicolon* thus wreaking havoc.

The OLD way

```
%let row=Country_Name;
%let col=Order_Category;
%let analvar=Order_Value;
%let statistic=max;
%let rowtotal=YES;
%let coltotal=YES;
%let title=My Table;
%let page=NONE;
%let year=2002;

%macro adhoc;
  %if &coltotal=YES %then %do;
    %let coltotal=ALL;
  %end;
  %else %do;
    %let coltotal= ;
  %end;
  %if &rowtotal=YES %then %do;
    %let rowtotal=(ALL * &analvar * &statistic);
  %end;
  %else %do;
    %let rowtotal= ;
  %end;
  %if &page = NONE %then %do;
```

```

        %let page= ;
        %let pagedim= ;
    %end;
    %else %do;
        %let pagedim=&page,;
        %let page=&page;
    %end;
proc    tabulate  data=sugidata.full_order_details;
        where      Order_Year=&year;
        class      &row  &col &page;
        var        &analvar;
        table      &pagedim
                    &row &coltotal,
                    (&col * &analvar * &statistic) &rowtotal;

title1 "&title";
title2 "Using data for &year";
run;

%mend adhoc;
%adhoc;

```

The NEW way

Ad Hoc Report Generator

General

Enter year to report: 2002 *

Value to use for page break: NONE *

What value do you want to summarize?: Order Value *

How do you want to summarize the data?: Total *

Row Dimension: Country *

Do you want a row total?: YES *

Column Dimension: Customer Gender *

Do you want a column total?: YES *

Supply a report title: My Ad Hoc Report *

(* denotes required parameter)

sas

Run Cancel

Note: The appearance of the interface will vary depending on the viewing environment (browser, MS Office, Web Report Studio, Information Delivery Portal)

The original macro code could be converted manually by

- o Adding Stored Process statements (*ProcessBody, %global, %stpbegin, %stpend)
- o Removing %let statements
- o Registering the Stored Process and defining the parameters in the Stored Process Manager in the SAS Management Console.

As an alternative, Enterprise Guide provides a **one-stop environment for doing all the above** while having the additional benefit of reducing the risk of creating errors by automating some of the tasks (adding Stored Process statements, identifying parameters from the code, registering the Stored Process).

To load an existing SAS program into a new Stored Process:

- o Select **File** ⇒ **New** ⇒ **Stored Process**
- o Enter **Name**, **Description** and **Keywords**
- o Click **Insert SAS code** ⇒ **Project** ⇒ **LIBNAME.sas**
- o Position the cursor after the **LIBNAME** statement
- o Click **Insert SAS code** ⇒ **Local Computer**
- o Navigate to your SAS program and select it
- o Maximize the **SAS Code Editor**.

Once the program is loaded, you will need to remove the %let statements since these will be replaced by Stored Process parameters. You should also review the code to make sure that any **LIBREFs** match the **LIBNAME** statement. After coding changes are complete, continue creating the Stored Process.

- o Resize the **SAS Code Editor**
- o Continue through the wizard until you reach the **Parameters** step
- o Click **Add** ⇒ **Parameters from SAS code**
- o Supply a **User prompt**
- o Supply a **Default Value**
- o Check the **Required** box
- o Select the **Constraints** tab, and set the constraints for the parameters
 - o From the pull down menu, select **List of values**
 - o Type in the values that the user can choose
 - o Select **Single selection**
 - o Click **Add** to go to the next parameter
 - o Continue supplying properties for each parameter
 - o When all parameters have been added, reply **OK** to the message
- o Use the up / down arrows to re-sequence the parameters into the order they will be displayed when the Stored Process is run
- o When complete, click **Next**
- o Step through the remainder of the Stored Process wizard as in previous examples and when complete, run the stored process.

Did you get errors?? When creating a Stored Process from existing code, it is not uncommon to get one or two errors on the first run. These are usually just syntax or logic errors in the code, due to the changes you have made.

To identify and fix any problems

- o Right click on the Stored Process icon and open the log to identify the problem
- o Double click the Stored Process icon, open the code editor and fix the problem
- o Then close and rerun!

Now, no more requests to run these legacy reports. The users can run the reports themselves!

EXAMPLE 6 – CREATING AN APPLICATION

You may want to collect all of the Stored Processes in one container and enable access through a browser. This example will focus on creating an HTML application with frames to display menus and results. The final result enables parameter driven reporting with a choice of reports.

Continent Name Africa	Customer Gender		All
	Female	Male	
	Order Value	Order Value	Order Value
			Sum
Current Name of Country			
Benin		10.90	10.90
Ivory Coast	98.50	36.60	135.10
Morocco	244.70	746.80	991.50
Mozambique		1831.50	1831.50
Senegal	83.60	2512.80	2596.40
South Africa	26376.56	48089.48	74466.04
All	26803.36	53228.08	80031.44

Step 1: – Create an HTML page to run the application

- o Use any HTML editor to create an HTML page (**FrameApp.html**) with 3 frames arranged as above
- o Name the frames **mainmenu**, **appmenu** and **results**

Step 2: – Create the main menu

- o Use any HTML editor to create an HTML page (**MainMenu.html**) to display the applications to be run
- o When an application is chosen, the menu for that application is to be displayed in the **appmenu** frame

Code for MainMenu.html

```
<html>
<head>
<TITLE>MainMenu</TITLE>
</HEAD>
<BODY>
  <a href="Supplier_Listing_frame.html"
    TARGET="appmenu">Supplier Listing</a><br>
  <a href="Sales_Channel_Analysis_frame.html"
    TARGET="appmenu">Sales Channel Analysis</a><br>
  <a href="Ad_Hoc_Report_Generator_frame.html"
    TARGET="appmenu">Ad Hoc Report Generator</a>
</BODY>
</HTML>
```

Step 3: – Create the default pages for the appmenu and results frames

- o Use any HTML editor to create two blank HTML pages (**AppMenu.html** and **Results.html**)

Code for AppMenu.html

```
<html>
<head>
<TITLE>Application Menu</TITLE>
</HEAD>
<BODY>

</BODY>
</HTML>
```

Code for Results.html

```
<html>
<head>
<TITLE>Application Menu</TITLE>
</HEAD>
<BODY>

</BODY>
</HTML>
```

Step 4: – Create application HTML files

- o Create copies of the HTML interfaces produced earlier when the Stored Processes were built in Enterprise Guide. *Note: This isn't essential as we could use the original interfaces, but this way we can*
 - o Make any customisations we might want (for example, remove the option to display the SAS log)
 - o Have the original code as a backup in case we make any mistakes
 - o Still use the original interface in other applications such as the SAS Information Delivery Portal
- o In each interface, locate the **FORM** tag and add a **TARGET** parameter to display the results in the **results** frame.

Amended FORM tag

```
<form method="post"
action="http://localhost:8080/SASStoredProcess/do"
onsubmit="return false;" target="results">
```

Step 6: – Test the application

- o Test the application by opening the FrameApp.html page in your browser. *Note:* the SAS Services Application and Tomcat (or alternative) must be running on the application server.
- o Click an application from the list in the upper left
- o Choose the options, from the parameters listed
- o Click **Run**.

Your HTML application is ready for “Prime Time”. Enjoy!

SOFTWARE REQUIREMENTS

The examples and features discussed in this paper used the SAS software as supplied in the SAS Enterprise BI Server Technology Solution.

CONCLUSION

Enterprise Guide 3.0 provides a powerful environment and toolset for end-to-end reporting. As demonstrated in this paper, efforts expended during development and setup enable flexible reporting and analytics to meet your organization's business needs.

RECOMMENDED READING

To gain a more complete understanding of Enterprise Guide 3.0, the authors recommend the following publications and web links.

- Getting Started with SAS Enterprise Guide – free learning tutorial available from http://www.sas.com/apps/elearning/elearning_details.jsp?pubcode=59271
- Other (chargeable) e-learning on Enterprise Guide from http://www.sas.com/apps/elearning/elearning_courses.jsp?cat=SAS+for+Non-Programmers
- Further information on the SAS® Enterprise BI Server: <http://www.sas.com/technologies/bi/entbiserver/index.html>

CONTACT INFORMATION

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

Marje Fecht

Prowerk Consulting LLC

Email: marje.fecht@prowerk.com

Web: www.prowerk.com

Peter Bennett

SAS

Email: peter.bennett@suk.sas.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.