Paper TU02

Delivering Multi-Sheet Excel Reports from a Parameterized Stored Process

Richard DeVenezia, Independent Consultant
Harry Droogendyk, Stratia Consulting Inc.

ABSTRACT

The advantage of using parameterized stored procedures is the ability to make functionality available to a wide audience through any number of interfaces. When the flexibility of stored processes is combined with the options provided by ODS, the possibilities are many. This tutorial will demonstrate the importance and usefulness of SAS®'s BI architecture by walking through the various steps required to develop and deploy a stored process that creates publishable Excel output. The paper concludes with some examples showing different ways to invoke the stored process.

INTRODUCTION

This paper will illustrate how Enterprise Guide 4.1 (EG) can be used to create process flows that generate simple listing reports, tabular output and pie charts. Within EG, the flows can be incrementally developed using a *run, review, and modify* approach. Once the process flows are deemed ready for deployment, they are turned into a Stored Process using an Enterprise Guide wizard. The wizard facilitates defining and registering the flow code in a metadata repository. Once in a repository, a stored process can be invoked a number of ways. We will show examples using EG, the Excel Add-in and SAS Stored Process browser interface. Generating multi-sheet Excel output will require some minor code tweaks in the stored process .sas file.

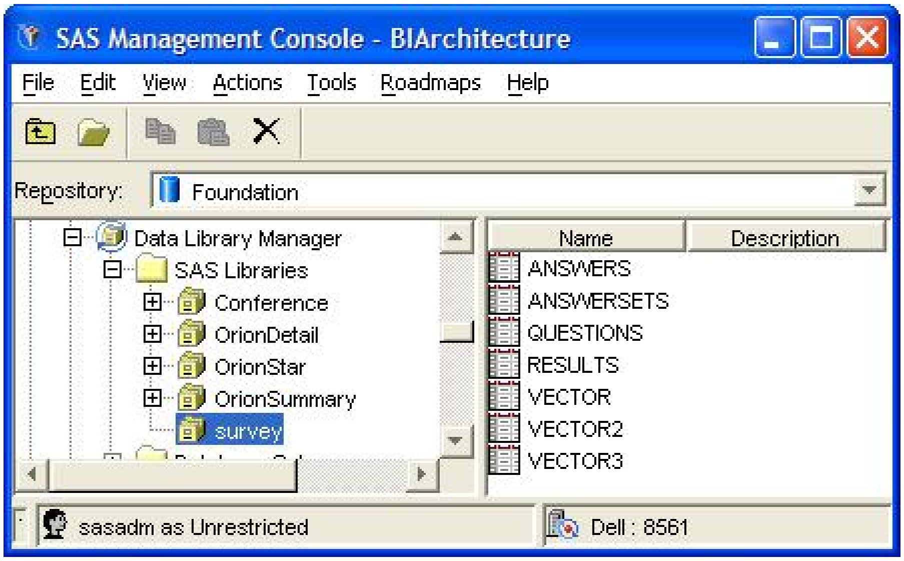
The survey data used in this paper is created from the code shown in Appendix A. For the sake of brevity, questions that elicit a single response are used. One question requires a simple yes/no answer, the remaining questions have seven answer choices, of which one must be chosen. Demographic information includes the gender, income level, country and city of the responder. Answers to the survey questions were randomly determined, as was the responder demographic data.

The structure of the data is as could be found in a real world situation. The code in Appendix A also reshapes the data for improved utilization in DATA Steps, SQL joins and SAS Procedures

Explaining the various components of the SAS Enterprise BI suite and their interrelation is beyond the scope of this paper. Please see Greg Nelson's *excellent* SGF 2007 paper: http://www2.sas.com/proceedings/forum2007/207-2007.pdf.

CREATING A SAS PROCESS WITH ENTERPRISE GUIDE

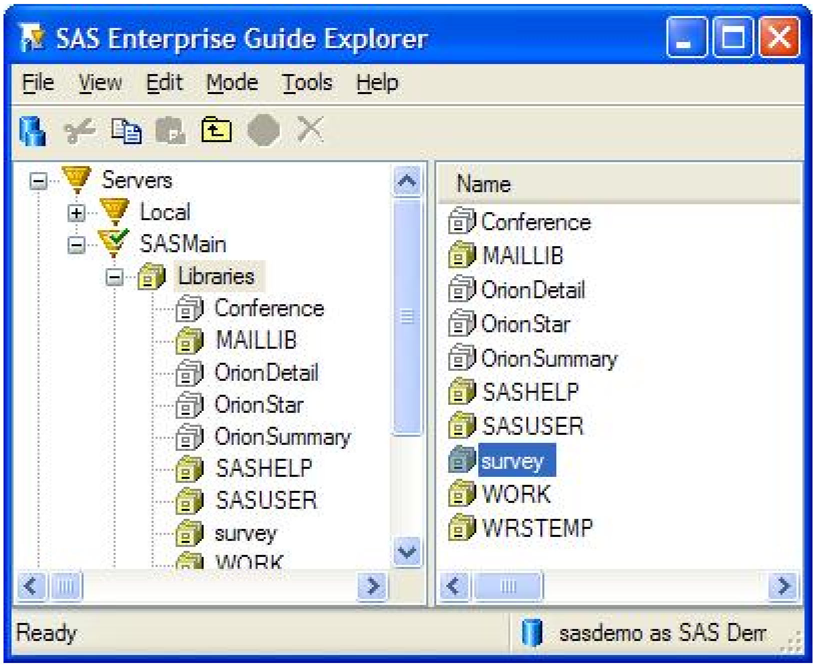
After running the program in Appendix A, the SAS Management Console was used to define the library, its tables and required authorizations to the metadata repository. Datasets in this library are referenced as the source data for the various reports.



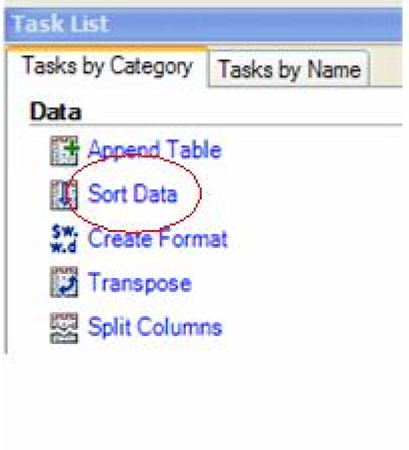
Repository: Foundation

Name	Description
ANSWERS	
ANSWERSETS	
QUESTIONS	
RESULTS	
VECTOR	
VECTOR2	
VECTOR3	

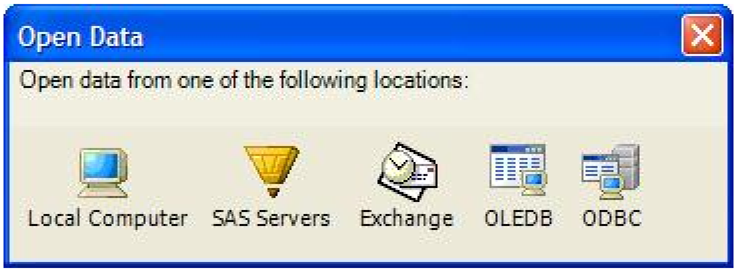
In SAS Enterprise Guide, click Tools, SAS Enterprise Guide Explorer to verify EG can see the new library:



Click File, New, Process Flow to begin defining the new process flow. Once the blank Process Flow appears, begin the development of the list report by dragging Sort Data from the Task List toolbar to the grid.

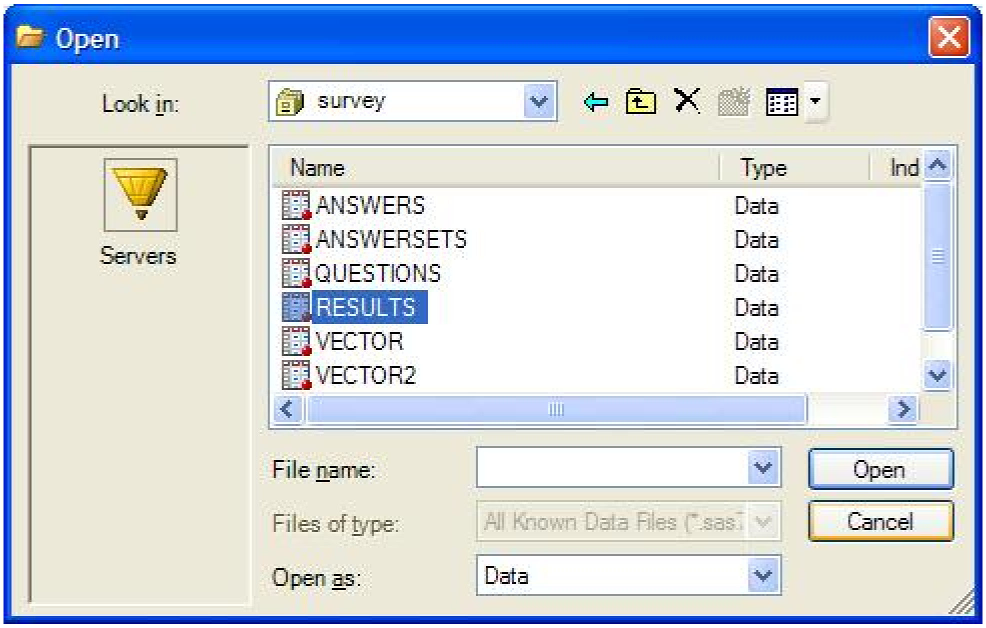


Before the Sort wizard is initiated, you must specify the location of the source data set.



Open data from one of the following locations:

Click SAS Servers, SASMAIN, Libraries, SURVEY and select the RESULTS dataset.



Look in: survey

Name	Type
ANSWERS	Data
ANSWERSETS	Data
QUESTIONS	Data
RESULTS	Data
VECTOR	Data
VECTOR2	Data

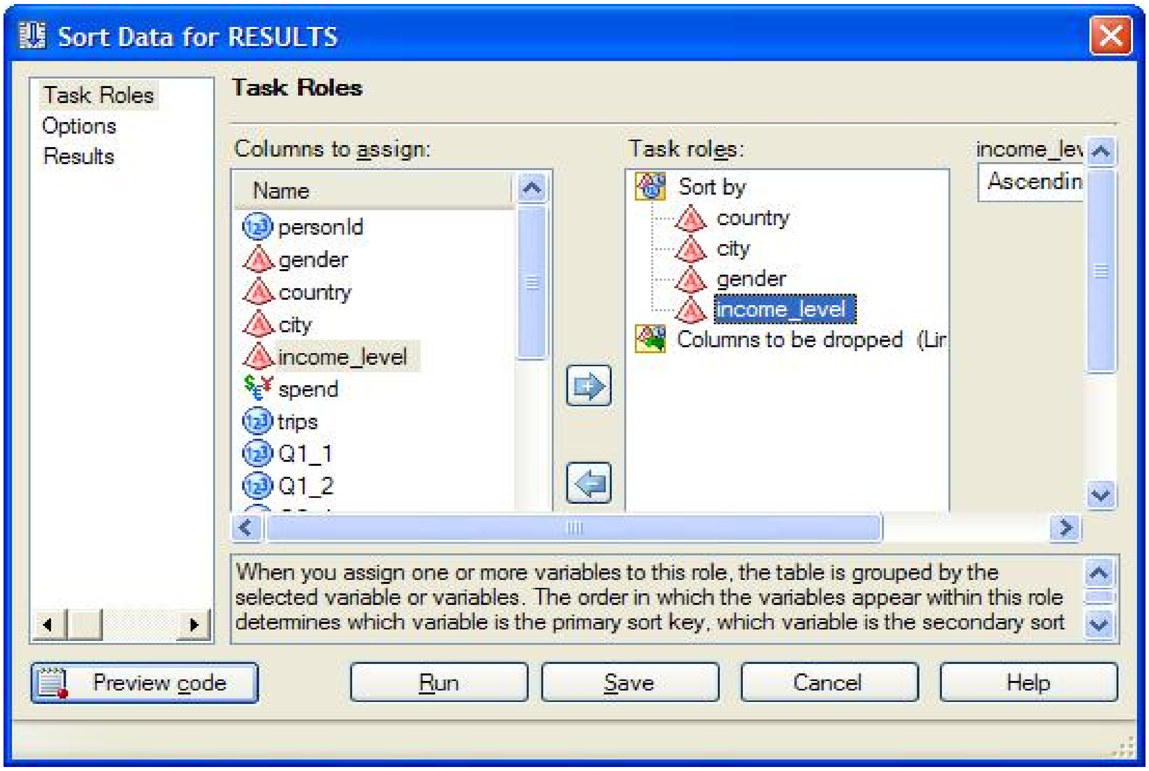
File name:

Files of type: All Known Data Files (*.sas:

Open as: Data

Click Open.

When the Sort Data wizard appears, drag variables from the **Columns to assign** box to the desired **Task roles** box. Arrows beneath the Task roles fields may be used to change the order of the sort variables. Click Run when complete.



Columns to assign:

Name
- personId
- gender
- country
- city
- income_level
- spend
- trips
- Q1_1
- Q1_2

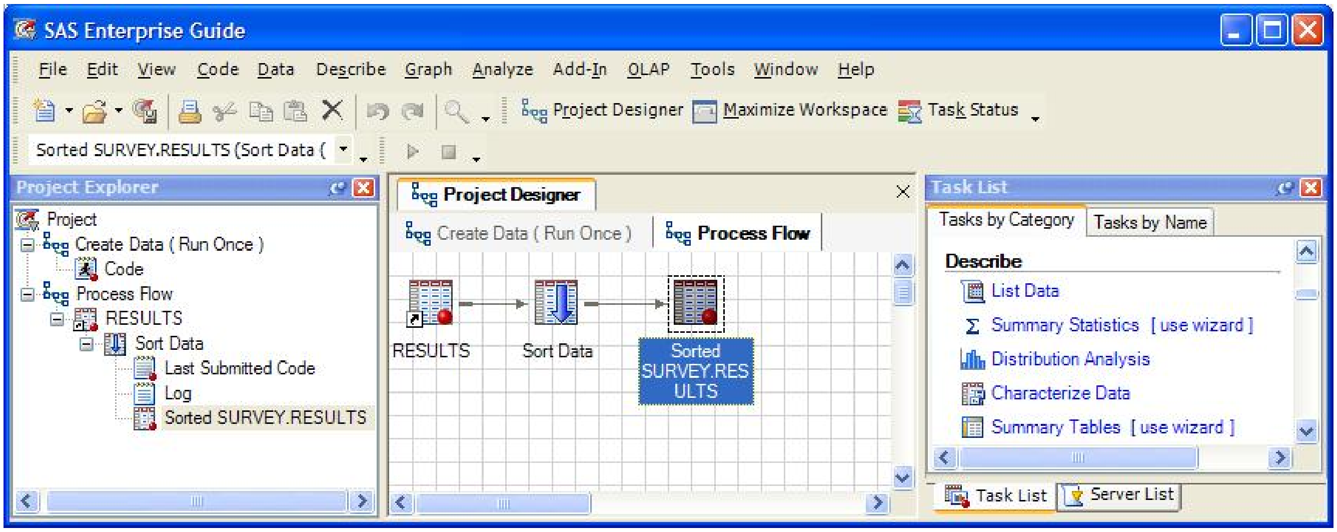
Task roles:

- Sort by
  - country
  - city
  - gender
  - income_level
- Columns to be dropped (Lir

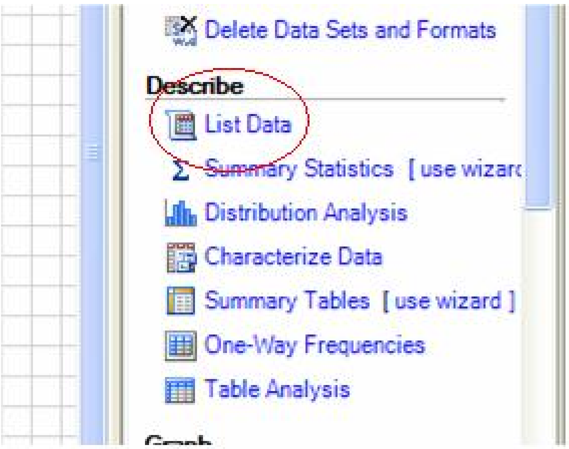
income_lev: Ascendin

When you assign one or more variables to this role, the table is grouped by the selected variable or variables. The order in which the variables appear within this role determines which variable is the primary sort key, which variable is the secondary sort

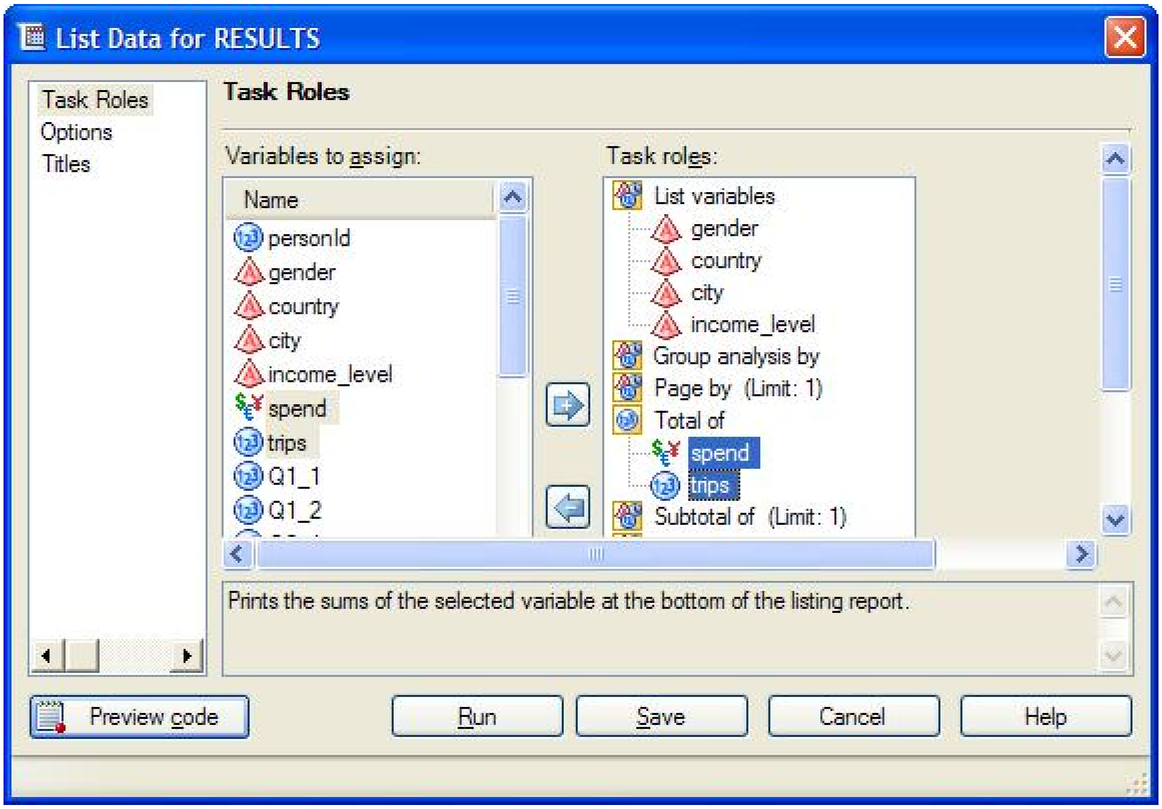
When the SORT completes, a temporary dataset is created and displayed in the process flow.



Drag **List Data** from the Task List toolbar (or Describe, List Data from the menu bar) onto the grid. EG will rightly assume the temporary sorted dataset is to be the source for this report.



When the List wizard appears, drag the columns from **Variables to assign** to the **Task roles** categories. Click Run.



Task Roles

Variables to assign:

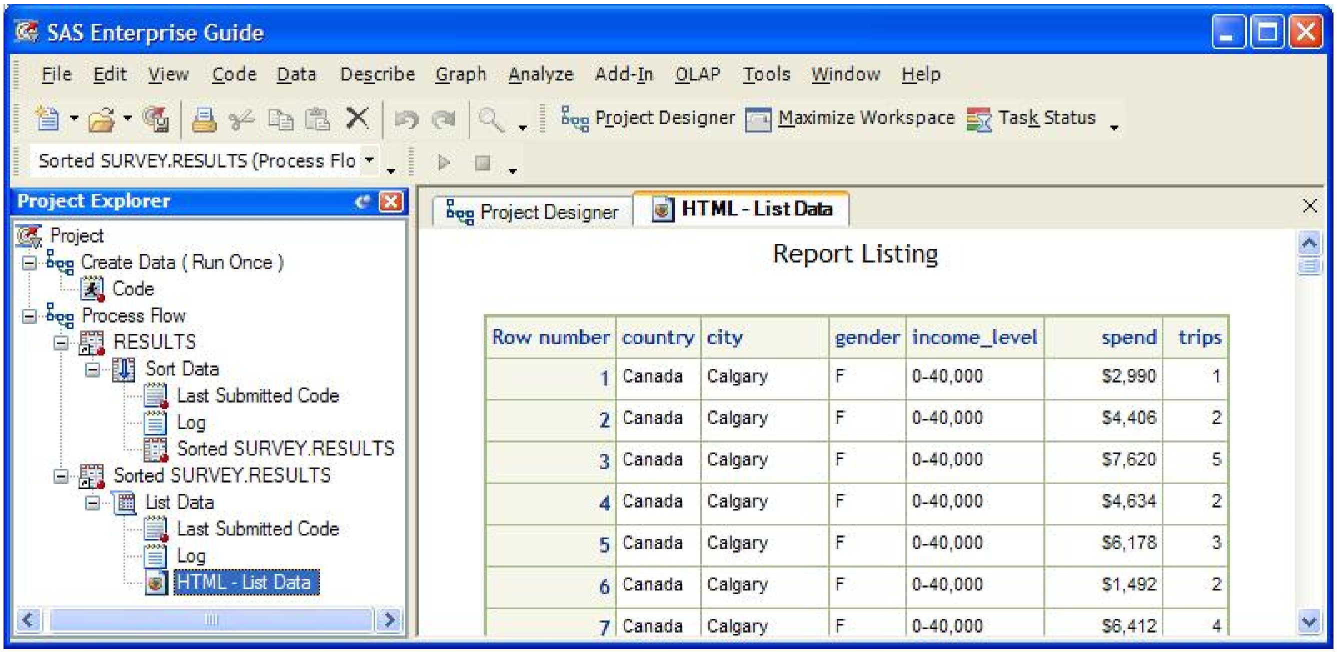
- Name
- personId
- gender
- country
- city
- income_level
- spend
- trips
- Q1_1
- Q1_2

Task roles:

- List variables
  - gender
  - country
  - city
  - income_level
- Group analysis by
- Page by (Limit: 1)
- Total of
  - spend
  - trips
- Subtotal of (Limit: 1)

Prints the sums of the selected variable at the bottom of the listing report.

When the list process completes, an HTML report (using the EG default style) is generated and displayed.

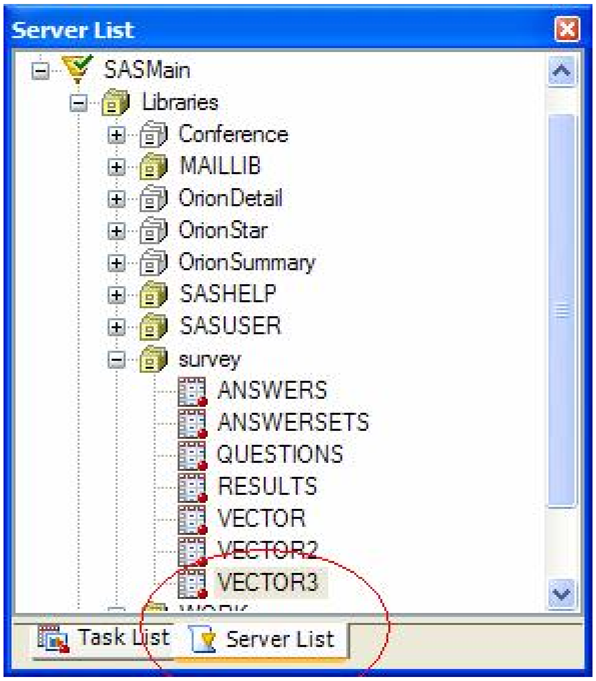


- Project
  - Create Data (Run Once)
    - Code
  - Process Flow
    - RESULTS
      - Sort Data
        - Last Submitted Code
        - Log
        - Sorted SURVEY.RESULTS
    - Sorted SURVEY.RESULTS
      - List Data
        - Last Submitted Code
        - Log
        - HTML - List Data

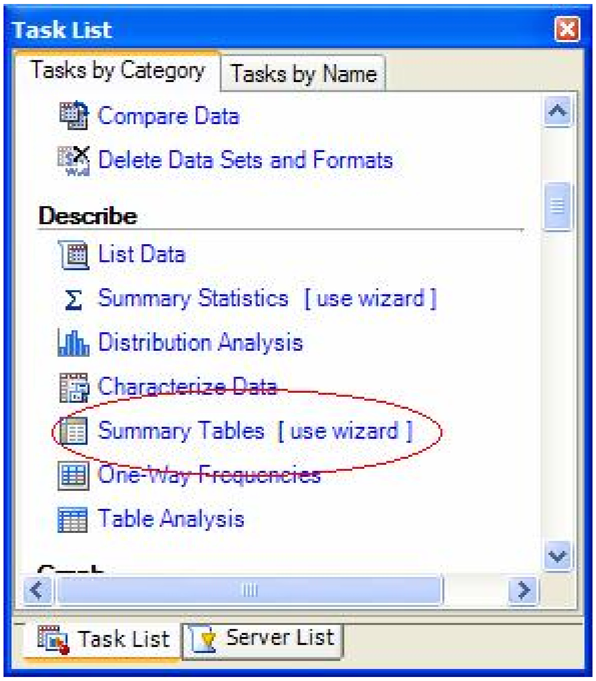
Report Listing

Row number	country	city	gender	income_level	spend	trips
1	Canada	Calgary	F	0-40,000	$2,990	1
2	Canada	Calgary	F	0-40,000	$4,406	2
3	Canada	Calgary	F	0-40,000	$7,620	5
4	Canada	Calgary	F	0-40,000	$4,634	2
5	Canada	Calgary	F	0-40,000	$6,178	3
6	Canada	Calgary	F	0-40,000	$1,492	2
7	Canada	Calgary	F	0-40,000	$6,412	4

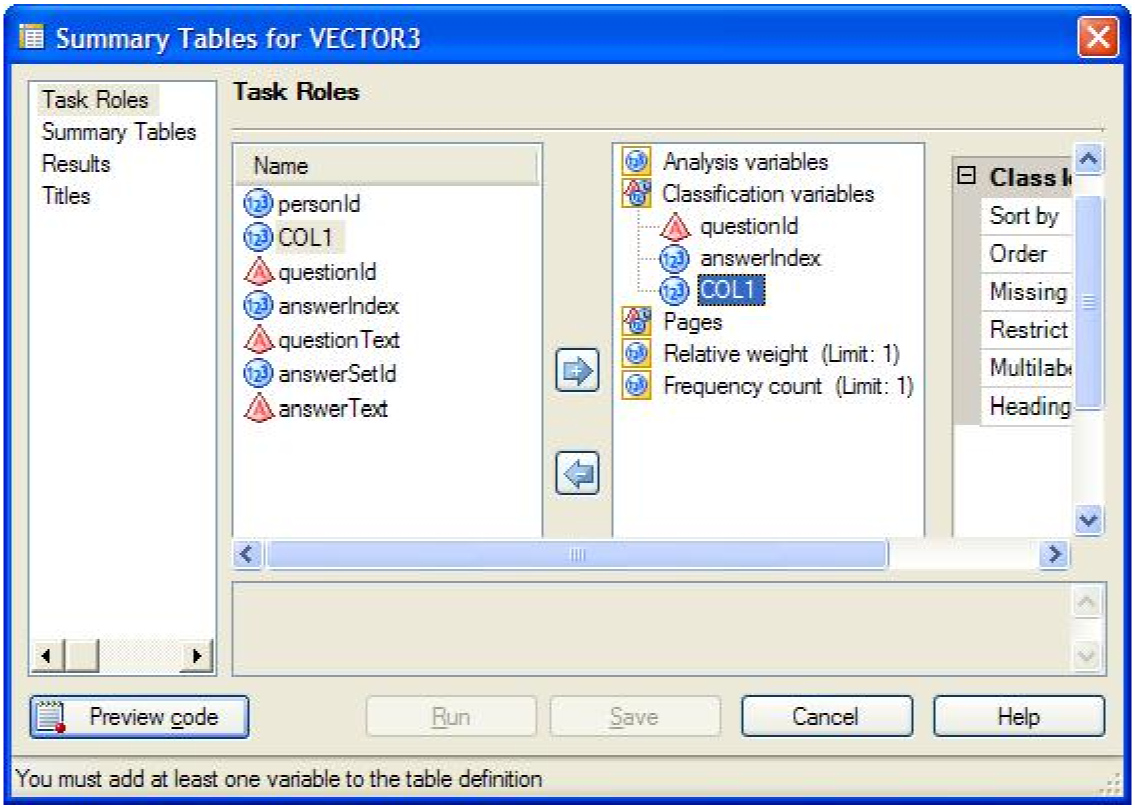
The Tabulate report will be created from a different dataset than the simple list report just generated. However, EG will always assume the source table of subsequent process flows to be the last active dataset. To force a new dataset to be used, click the **Server List** tab on **the Task** List toolbar, navigate to the library and dataset, and drag the dataset to the EG desktop. When the contents of the dataset are displayed, close the table display and a shortcut to the dataset will remain.



To create the Tabulate report, drag the **Summary Tables** icon to the grid.



The Summary Tables report will be connected to the VECTOR3 shortcut created in the prior step and the TABULATE wizard will initiate. Drag and drop fields to the **Task Roles** locations as required.



Task Roles

Name
personId
COL1
questionId
answerIndex
questionText
answerSetId
answerText

Analysis variables

Classification variables

- questionId
- answerIndex
- COL1

Pages

Relative weight (Limit: 1)

Frequency count (Limit: 1)

Class l

Sort by

Order

Missing

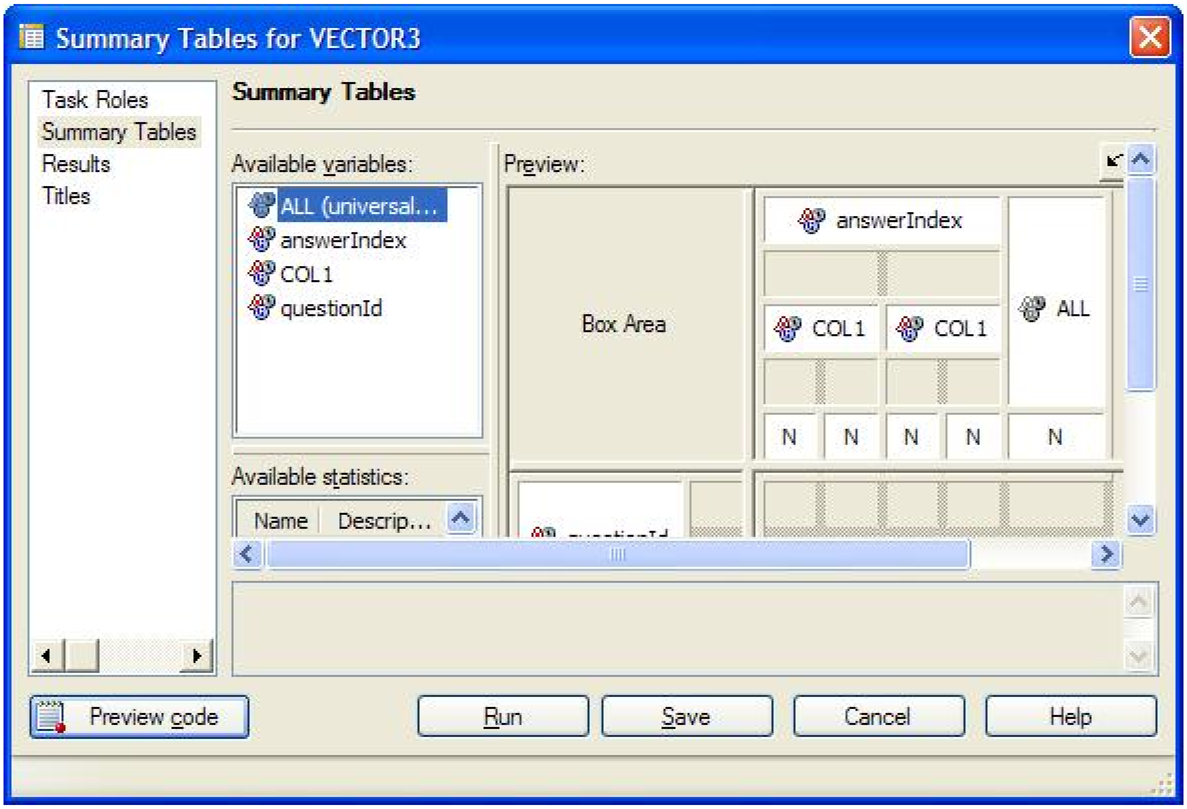
Restrict

Multilab

Heading

You must add at least one variable to the table definition

Click **Summary Tables** and specify fields as below. When complete, click **Preview Code** to insert a WHERE clause to limit the rows brought into the TABULATE.



Summary Tables

Available variables:

- ALL (universal...
- answerIndex
- COL1
- questionId

Available statistics:

Name	Descrip...

Preview:

Box Area

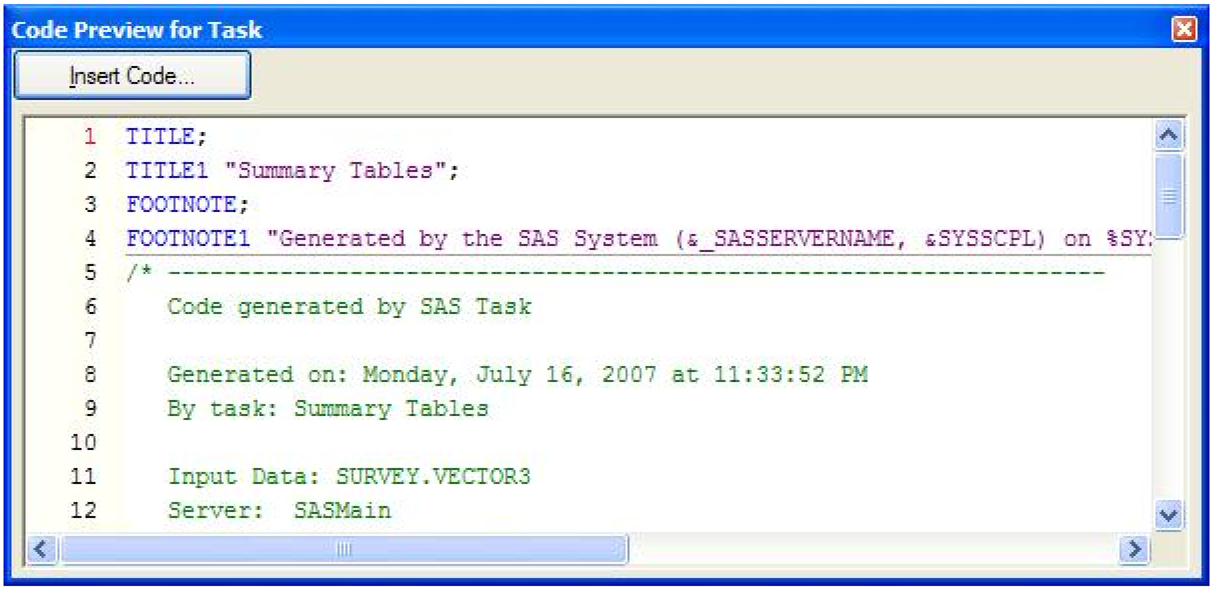
answerIndex

COL1 COL1 ALL

N N N N N

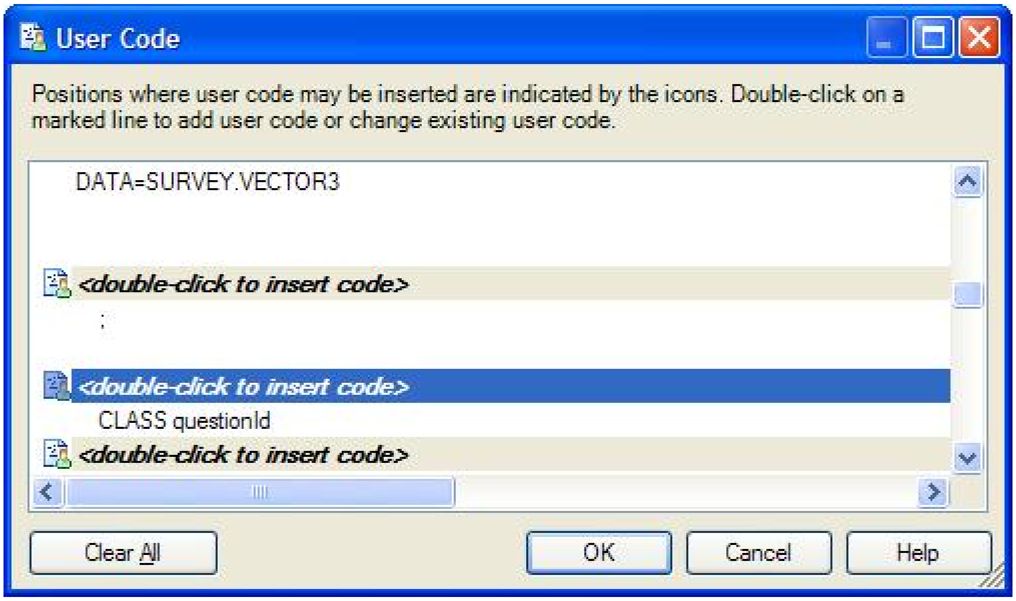
questionId

When the code window is display, click **Insert Code**:



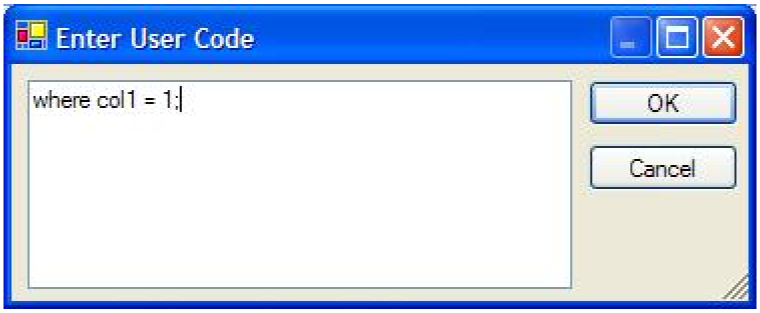
```
TITLE;
TITLE1 "Summary Tables";
FOOTNOTE;
FOOTNOTE1 "Generated by the SAS System (&_SASSERVERNAME, &SYSSCPL) on %SY
/* -------------------------------------------------------------------
   Code generated by SAS Task

   Generated on: Monday, July 16, 2007 at 11:33:52 PM
   By task: Summary Tables

   Input Data: SURVEY.VECTOR3
   Server:  SASMain
```

Double click the *<double click to insert code>* after the DATA= specification and enter the code snippet in the dialog box that opens and click **OK**. The code will be inserted, click OK again and close the Code Preview window.



Positions where user code may be inserted are indicated by the icons. Double-click on a marked line to add user code or change existing user code.

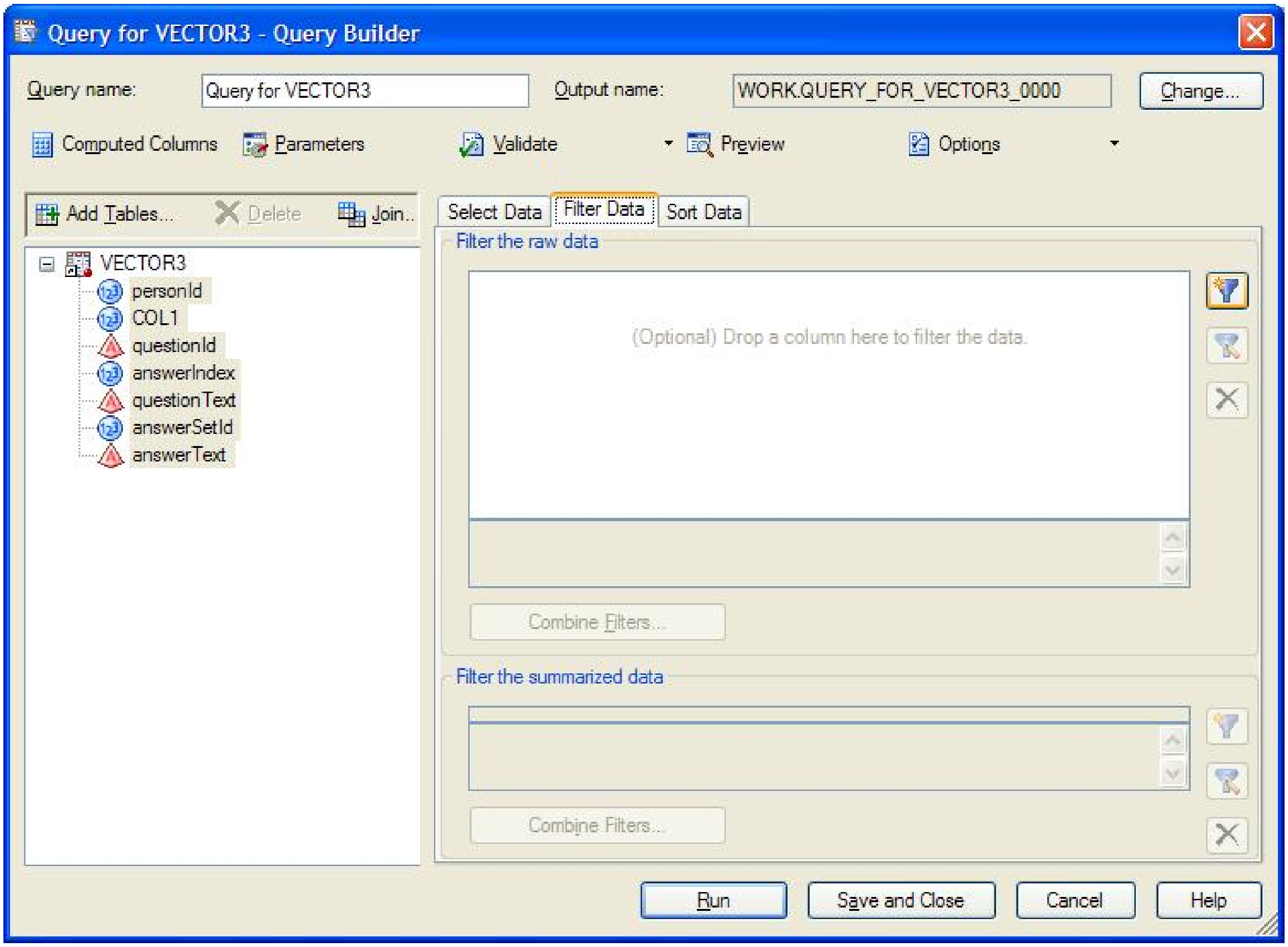
```
DATA=SURVEY.VECTOR3


<double-click to insert code>
  ;

<double-click to insert code>
  CLASS questionId
<double-click to insert code>
```



```
where col1 = 1;
```

Click **Run** in the Summary Tables wizard to generate the TABULATE output.

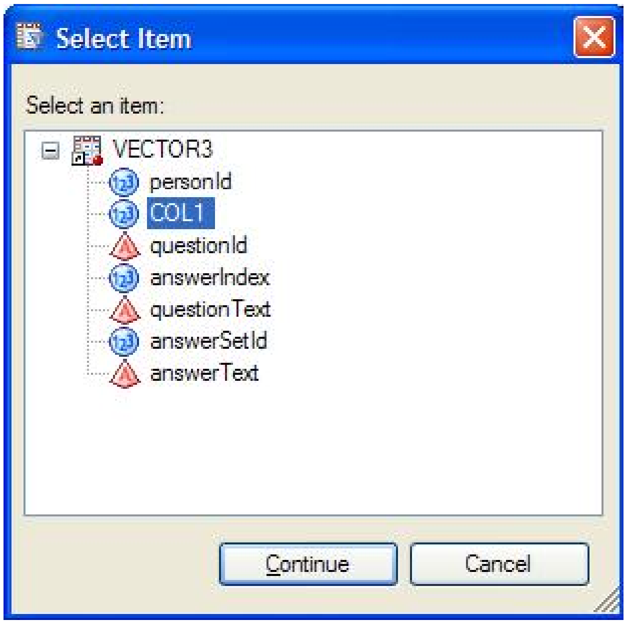
The next step illustrates a different method of subsetting, required for the pie chart creation. After clicking the VECTOR3 dataset to ensure it's active, add a data filter to the process. Click **Data** in the menu bar and select **Filter and Query** and use the wizard to define the filtering criteria.

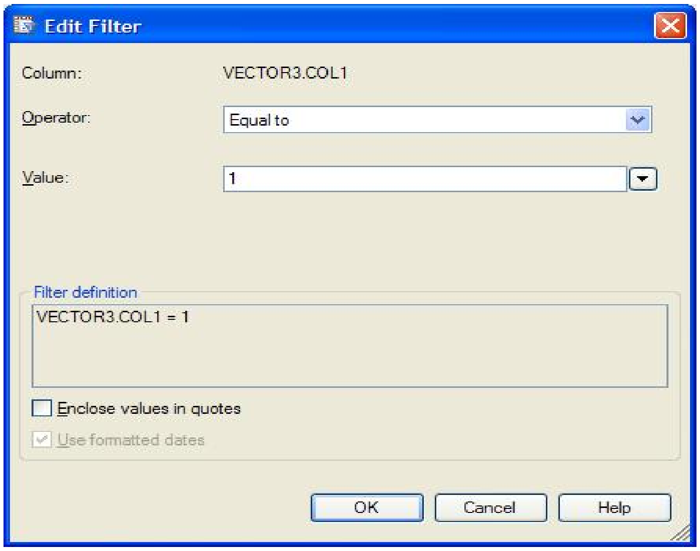


Query name: Query for VECTOR3

Output name: WORK.QUERY_FOR_VECTOR3_0000

Drag the required fields into the **Select Data** area and click on **Filter Data** and click the new filter icon and select **New Filter**. When the field list is displayed, select COL1 and click **Continue** to specify the filter condition (COL1 = 1).





Column: VECTOR3.COL1

Operator: Equal to

Value: 1

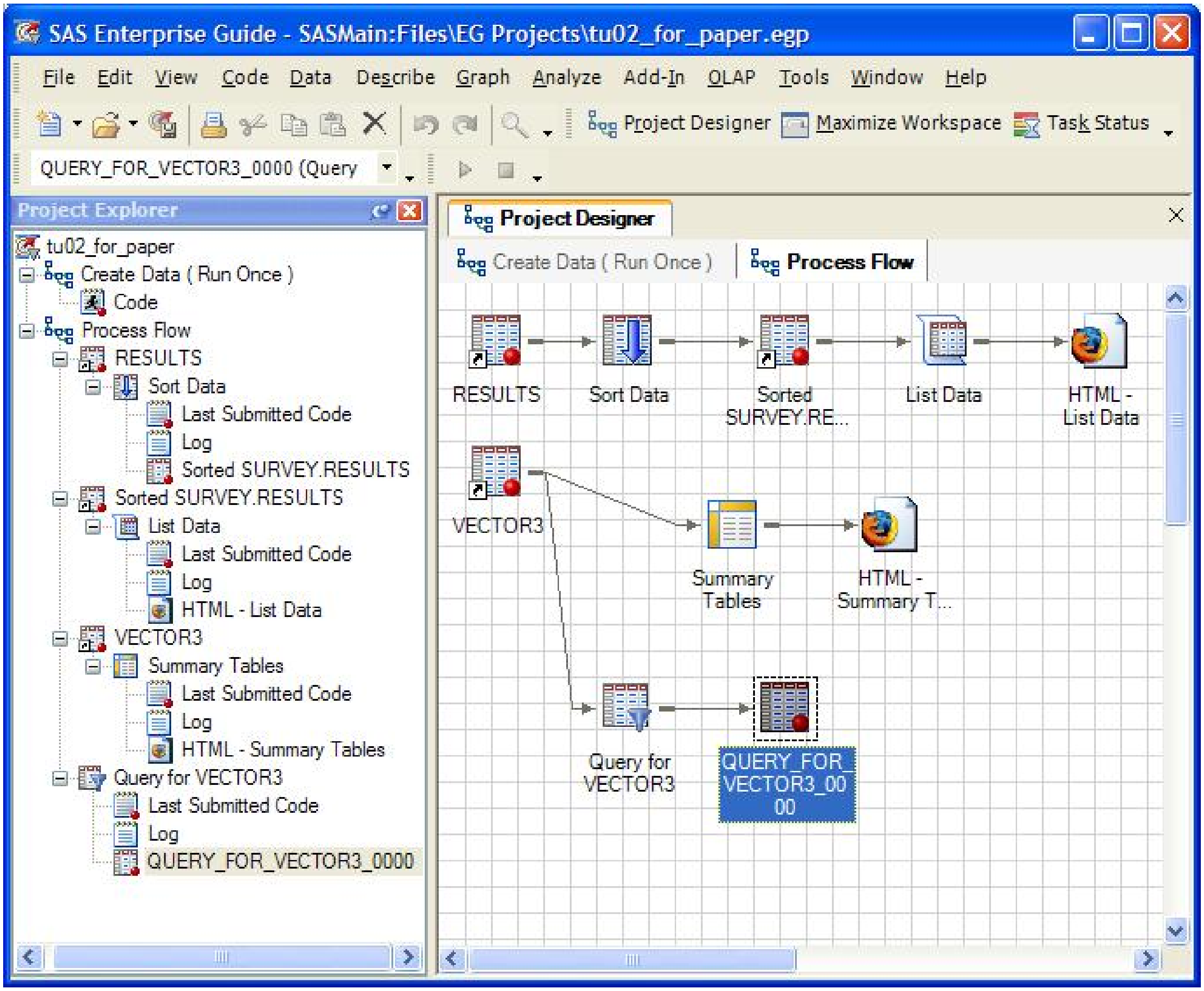
VECTOR3.COL1 = 1

☐ Enclose values in quotes

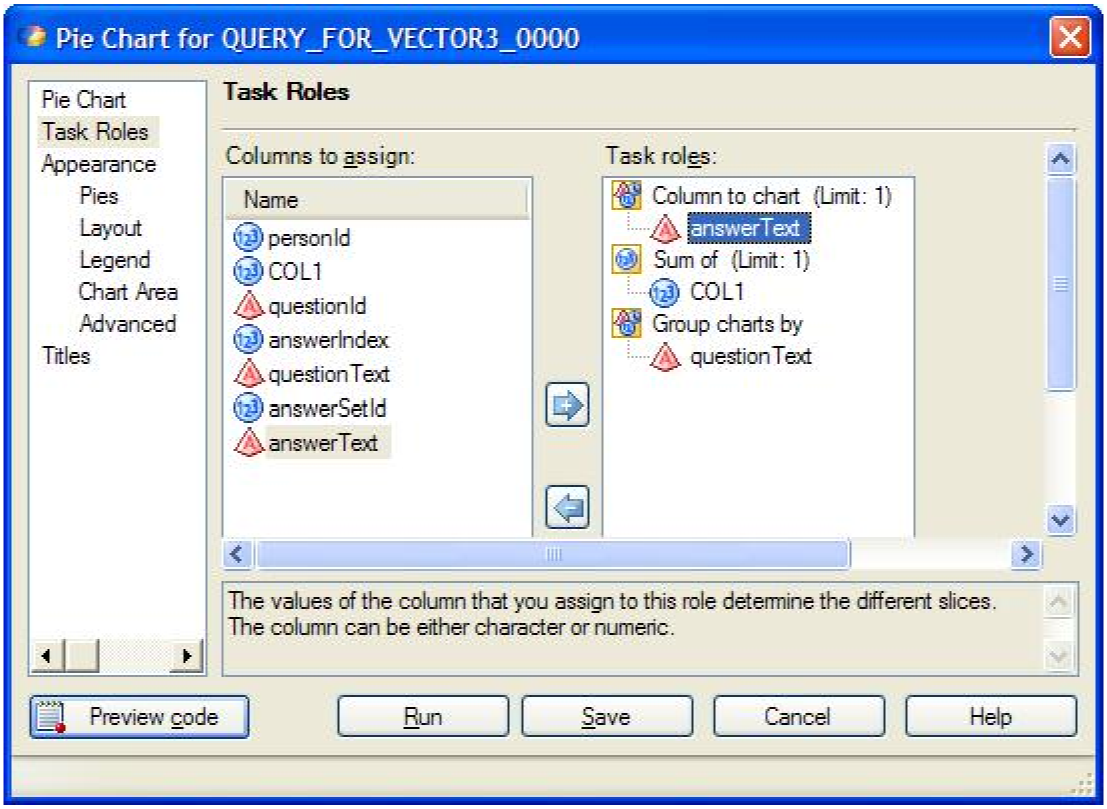
☑ Use formatted dates

Click the **Sort Data table** and specify that the data be sorted by QuestionText and click **Run**.

The process flow is beginning to look a little more interesting!



The third report to be defined is a pie chart showing the distribution of the answers to the seven survey questions. Drag Pie Chart from the **Task List** to the palette. When the wizard begins, drag/drop fields as required and click **Run**.



Task Roles

Columns to assign:

Name
personId
COL1
questionId
answerIndex
questionText
answerSetId
answerText

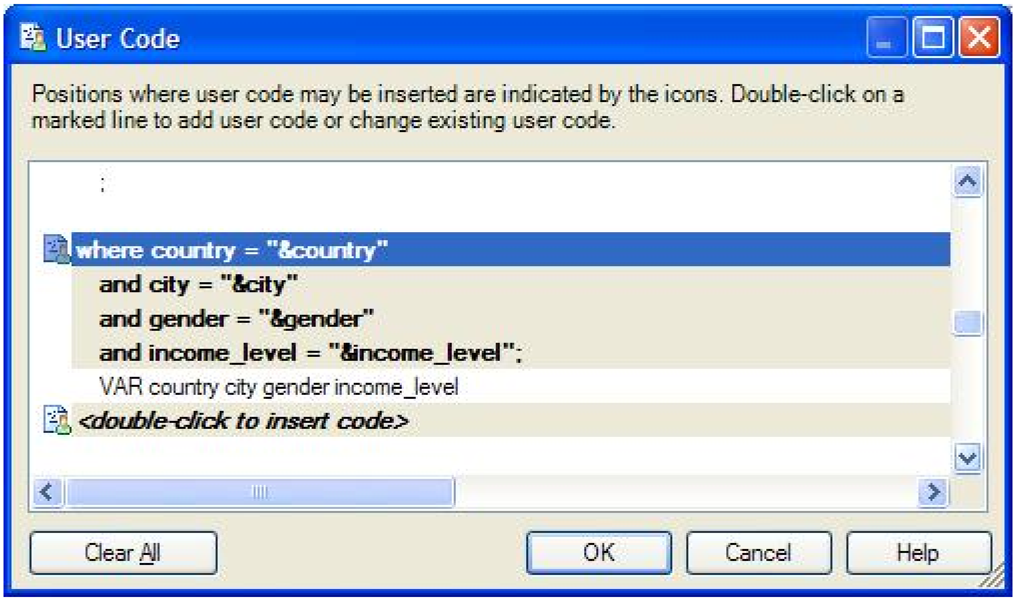
Task roles:

- Column to chart (Limit: 1)
  - answerText
- Sum of (Limit: 1)
  - COL1
- Group charts by
  - questionText

The values of the column that you assign to this role determine the different slices. The column can be either character or numeric.

CREATING A STORED PROCESS

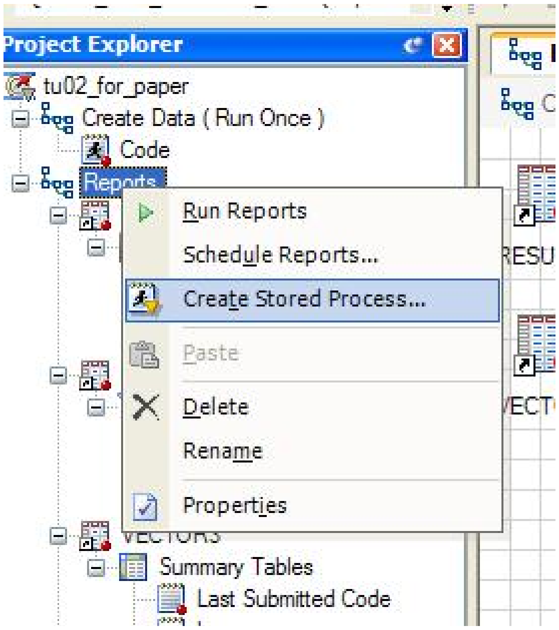
Now that the process creating the different reports has been defined, it's time to turn it into a Stored Process. This isn't an onerous task, again a wizard will help us. Before we begin, we want to add some parameters to the listing report we created initially. This will allow the report to be run for selected responders in the RESULTS dataset. Begin by right-clicking on the List Data icon and opening the customization wizard. Click Preview Code and then Insert Code in the preview window. Scroll down to the PROC PRINT and add a where clause after the DATA= specification. Stored process parameters are really just SAS macro variables that the stored process wizard detects and prompts us to specify criteria/values for each. Click OK and exit the code preview window, click Save.



Positions where user code may be inserted are indicated by the icons. Double-click on a marked line to add user code or change existing user code.

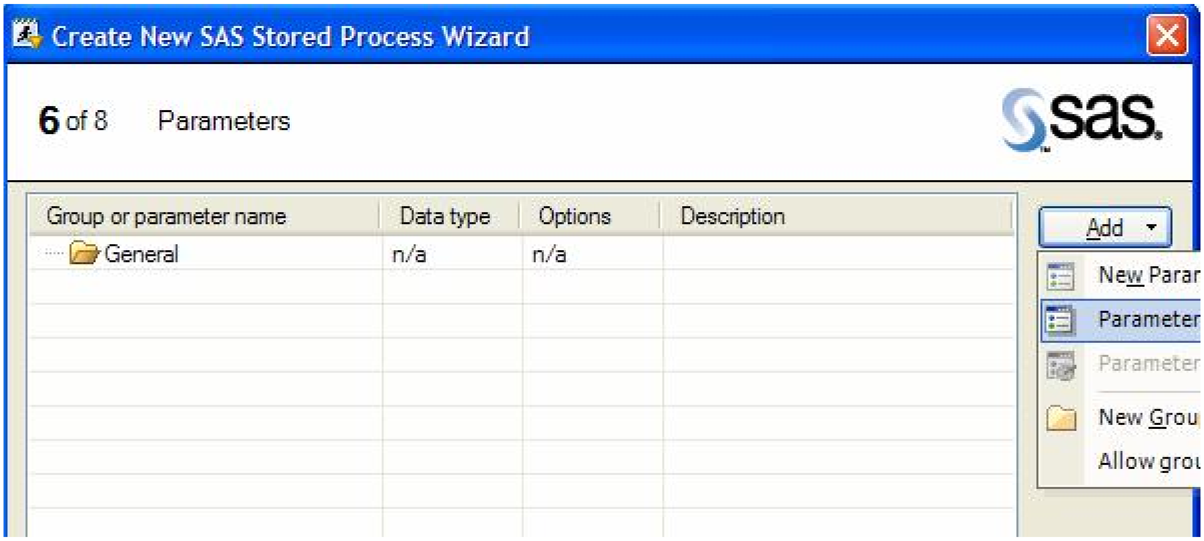
```
    ;

where country = "&country"
  and city = "&city"
  and gender = "&gender"
  and income_level = "&income_level";
  VAR country city gender income_level
<double-click to insert code>
```

To initiate the creation of a stored process for the entire process flow, right click on the process name in the Project Explorer and select Create Stored Process. The eight step wizard prompts for necessary information, on occasion, asking for confirmation.



Steps:

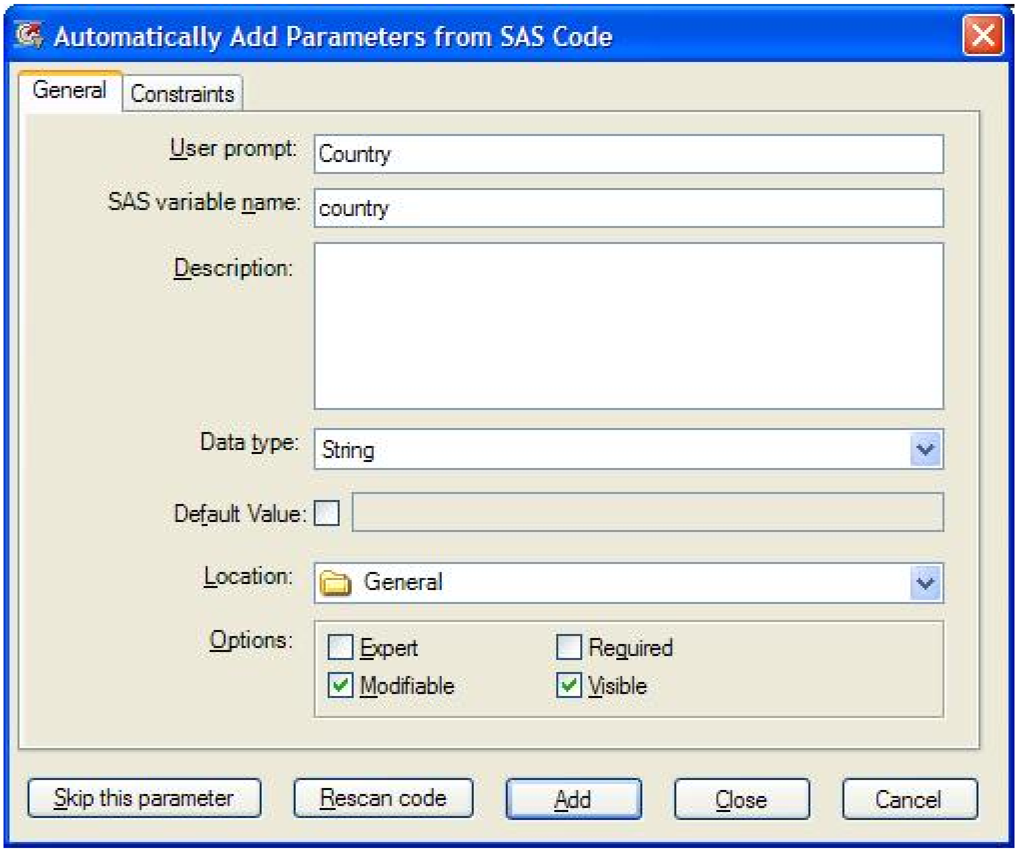
1. Enter SESUG 2007 Survey Reporting as the stored process name, click **Next**.
2. Verify the SAS code, click **Next**.
3. Select the metadata location for the stored process, click **Next**.
4. Verify entries, click **Next**.
5. Specify the inclusion of any non “built-in” libraries for inclusion in the stored process, click **Next**.
6. Specify stored process parameters



Group or parameter name	Data type	Options	Description
General	n/a	n/a	

Click Add and select Parameters from SAS Code. When the wizard begins, click **Skip Parameter** for all macro variables detected that are NOT the parameters you defined, eg. SASSERVERNAME

When a user-defined parameter is encountered, alter the options as necessary. To populate a dropdown with valid choices, click the **Constraints** tab.



Automatically Add Parameters from SAS Code

User prompt: Country

SAS variable name: country

Description:

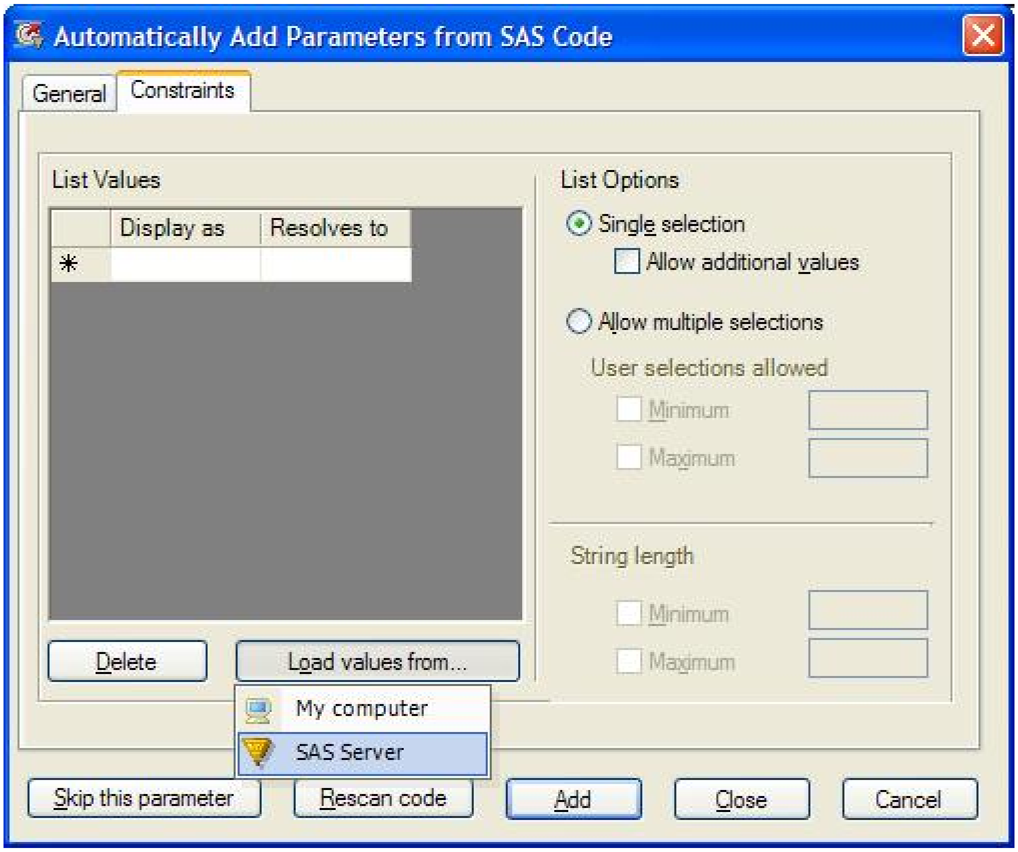
Data type: String

Default Value: □

Location: General

Options: □ Expert □ Required ☑ Modifiable ☑ Visible

Click **Load values from** and select SAS Server. Navigate to the RESULTS dataset (ie. source for List Data stream).



Automatically Add Parameters from SAS Code

List Values

	Display as	Resolves to
*		

List Options

● Single selection

□ Allow additional values

○ Allow multiple selections

User selections allowed

□ Minimum

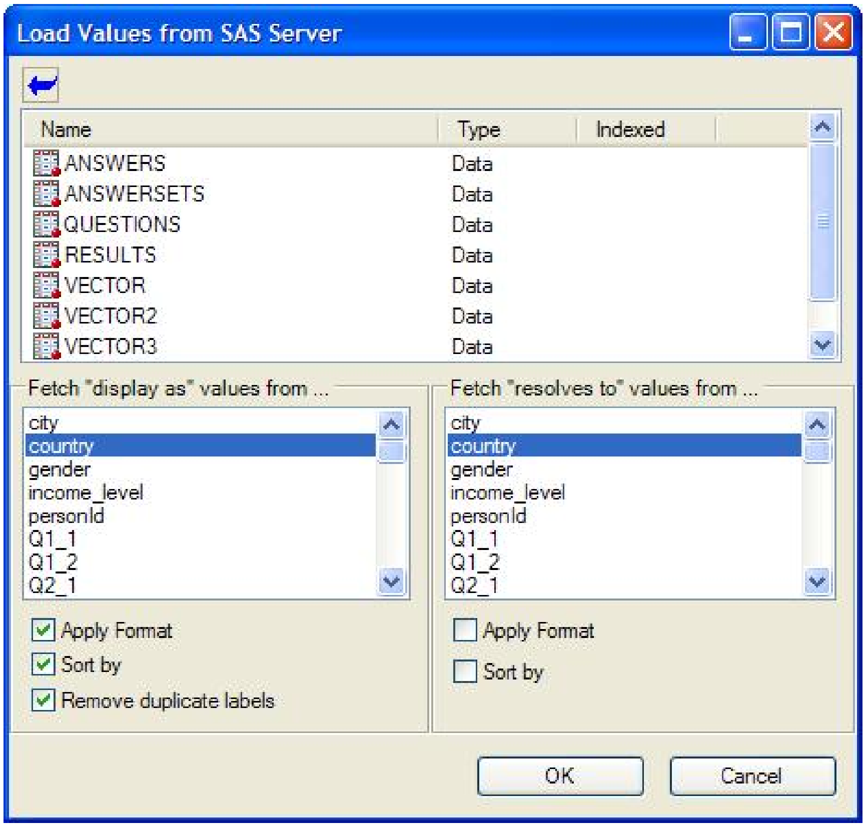
□ Maximum

String length

□ Minimum

□ Maximum

Select the appropriate field, applying options as required.



Name	Type	Indexed
ANSWERS	Data	
ANSWERSETS	Data	
QUESTIONS	Data	
RESULTS	Data	
VECTOR	Data	
VECTOR2	Data	
VECTOR3	Data	

Fetch "display as" values from ...

city
country
gender
income_level
personId
Q1_1
Q1_2
Q2_1

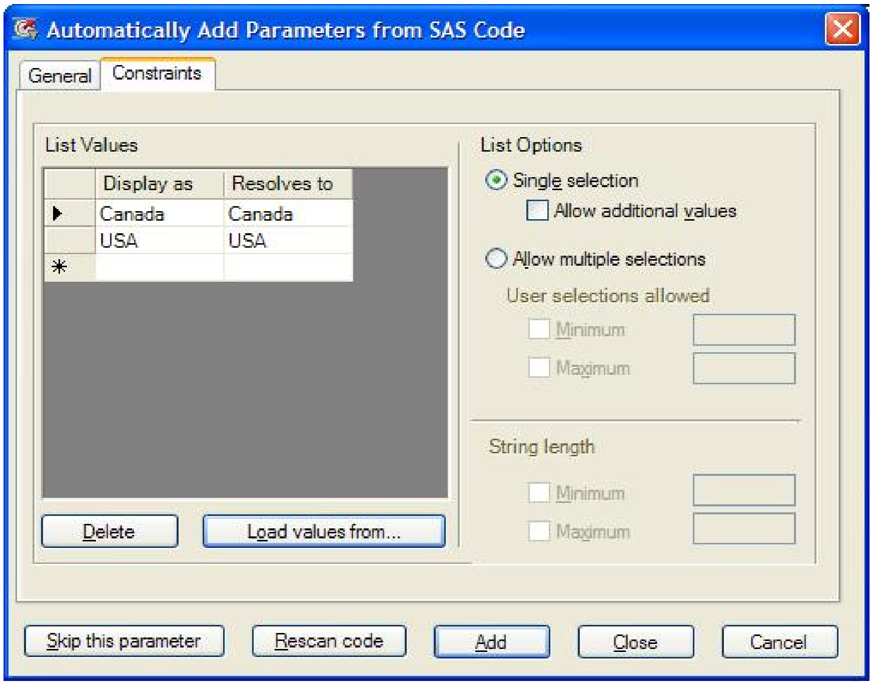
☑ Apply Format
☑ Sort by
☑ Remove duplicate labels

Fetch "resolves to" values from ...

city
country
gender
income_level
personId
Q1_1
Q1_2
Q2_1

☐ Apply Format
☐ Sort by

Click **OK** and the current values of country will be displayed. Click **Add**. Continue for all parameters in the stored process.



List Values

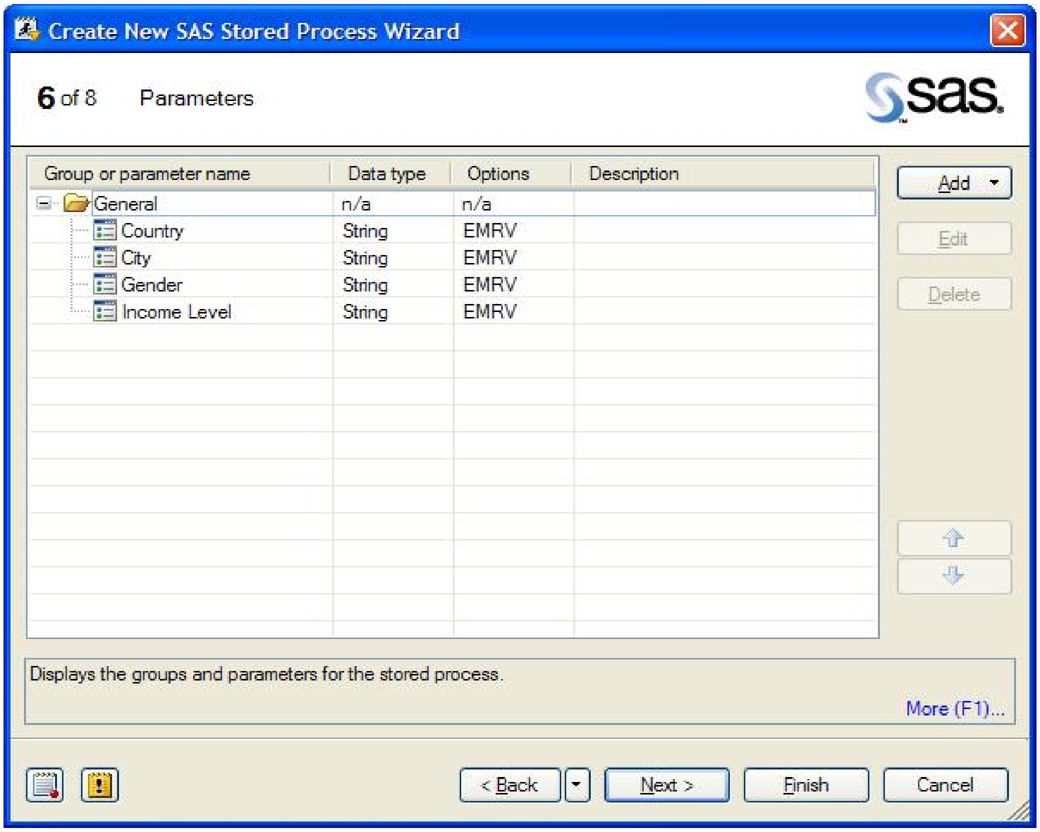
	Display as	Resolves to
▶	Canada	Canada
	USA	USA
*		

List Options

● Single selection
☐ Allow additional values
○ Allow multiple selections
User selections allowed
☐ Minimum
☐ Maximum

String length
☐ Minimum
☐ Maximum

Upon completion, all stored process parameters are displayed, click **Next**.

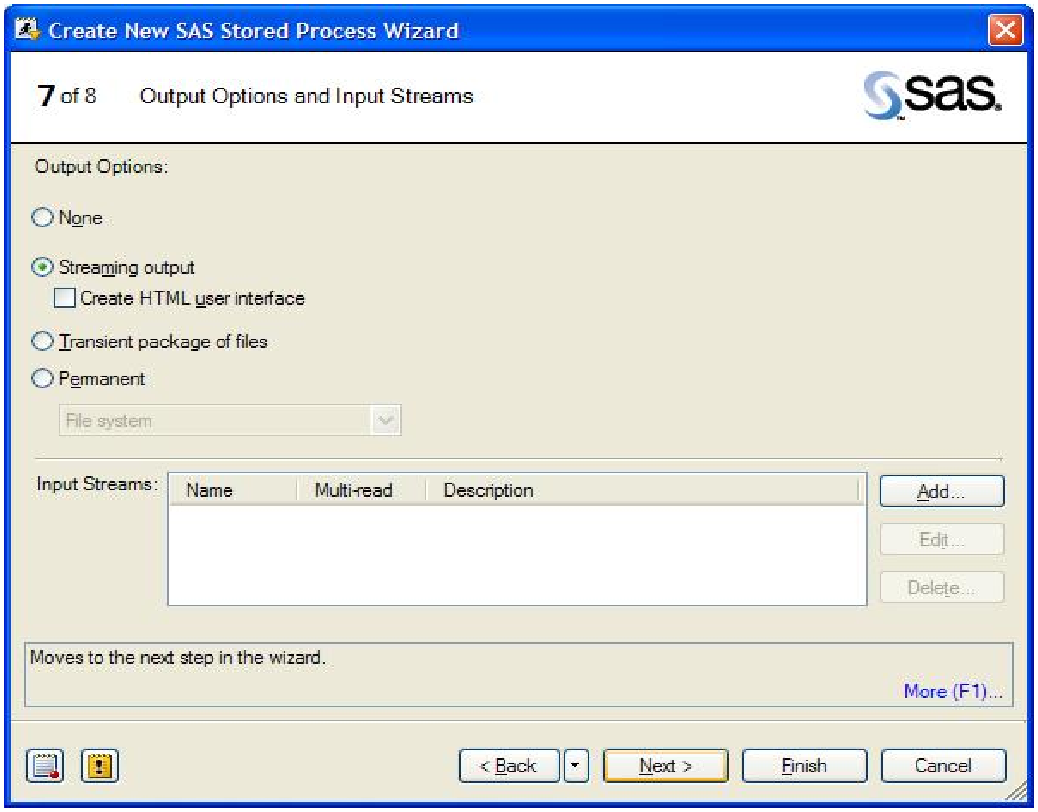


6 of 8 Parameters

Group or parameter name	Data type	Options	Description
General	n/a	n/a	
Country	String	EMRV	
City	String	EMRV	
Gender	String	EMRV	
Income Level	String	EMRV	

Displays the groups and parameters for the stored process.

In step 7 specify **Streaming Output**, click **Next**. Click **Finish** on step 8.



7 of 8 Output Options and Input Streams

Output Options:

○ None

● Streaming output

□ Create HTML user interface

○ Transient package of files

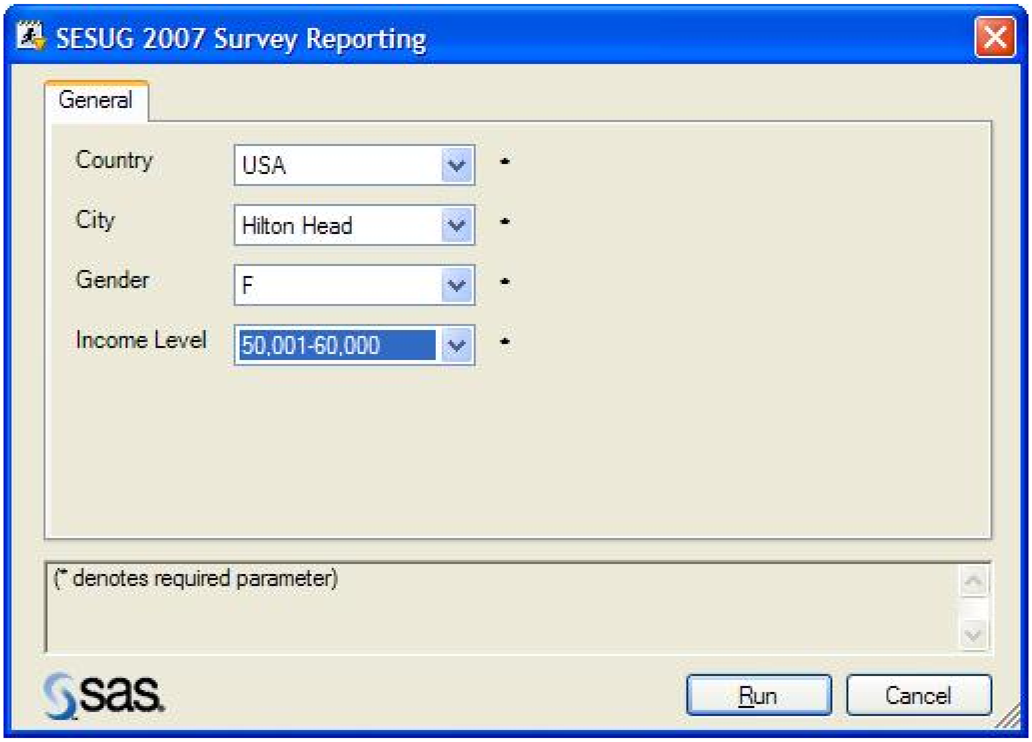
○ Permanent

Input Streams:

Name	Multi-read	Description

Moves to the next step in the wizard.

When the Stored Process wizard completes, EG registers the stored process in the metadata and places a new Stored Process icon on the process desktop . Right click on the new icon and select **Run SESUG 2007 Survey Reporting**. The parameter selection dialog box will be displayed. Make selections from each dropdown and click **Run**.



SESUG 2007 Survey Reporting

General

Country: USA *

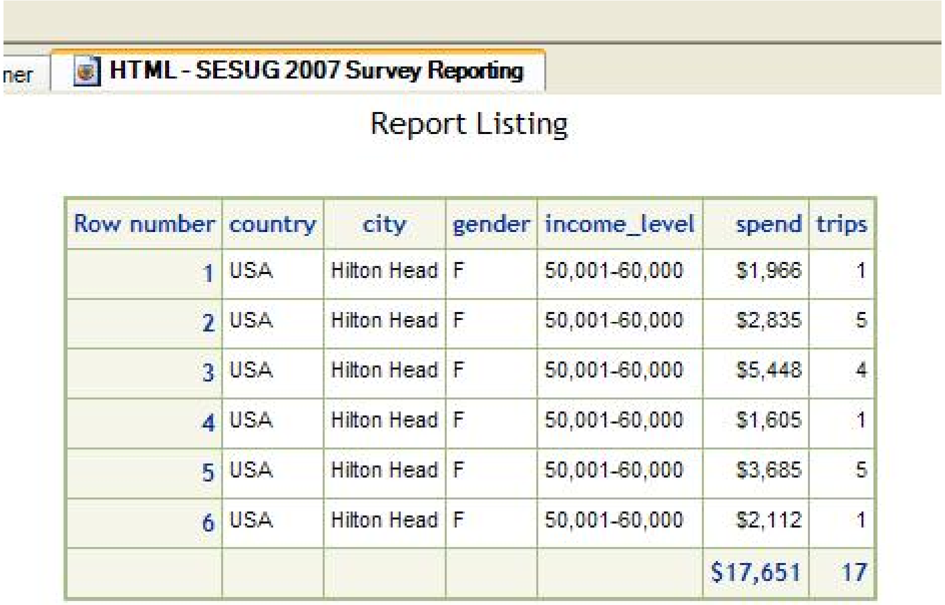
City: Hilton Head *

Gender: F *

Income Level: 50,001-60,000 *

(* denotes required parameter)

Results are returned to EG (only the first PRINT output is shown here).



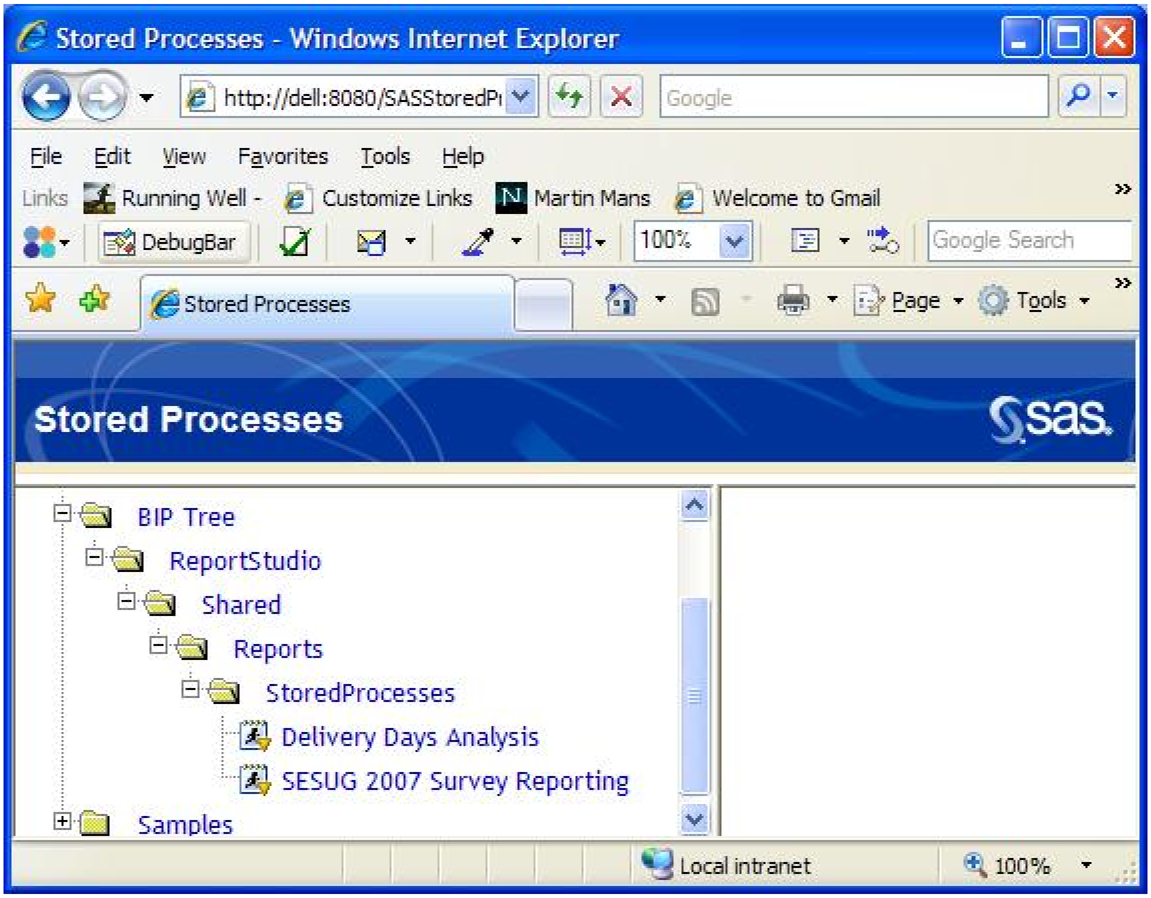
Report Listing

Row number	country	city	gender	income_level	spend	trips
1	USA	Hilton Head	F	50,001-60,000	$1,966	1
2	USA	Hilton Head	F	50,001-60,000	$2,835	5
3	USA	Hilton Head	F	50,001-60,000	$5,448	4
4	USA	Hilton Head	F	50,001-60,000	$1,605	1
5	USA	Hilton Head	F	50,001-60,000	$3,685	5
6	USA	Hilton Head	F	50,001-60,000	$2,112	1
					$17,651	17

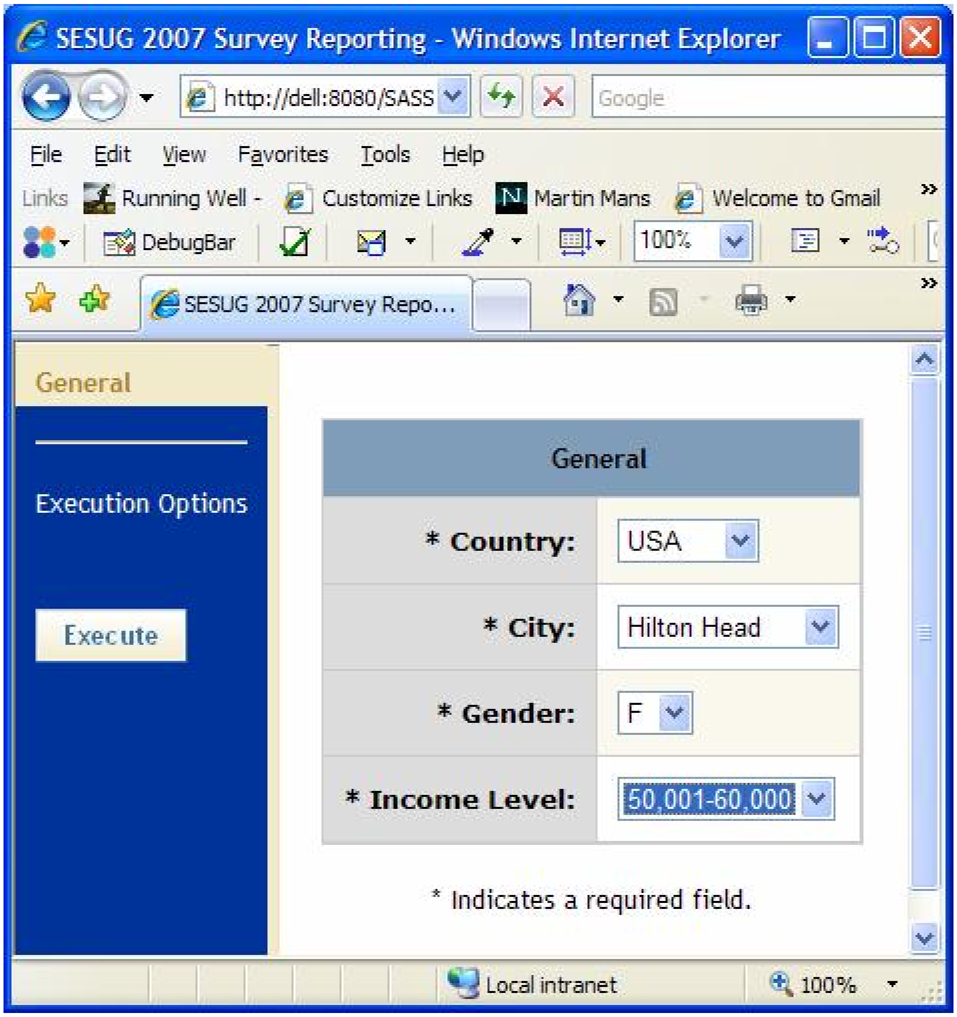
EXECUTING A STORED PROCESS OUTSIDE ENTERPRISE GUIDE

We've already seen how to execute the stored process in EG. The next method covered in this paper involves the browser. The URL for the Stored Process page will be http://*your_server_here*:8080/SASStoredProcess/do?action=index.

Navigate to the location where the Stored Process was stored and click the SESUG 2007 Survey Reporting link. The Stored Process properties are displayed for verification, click **Execute**.


Stored Processes - Windows Internet Explorer
http://dell:8080/SASStoredPr
Google
File Edit View Favorites Tools Help
Links Running Well - Customize Links Martin Mans Welcome to Gmail
DebugBar
100%
Google Search
Stored Processes
Page Tools
Stored Processes
sas
BIP Tree
ReportStudio
Shared
Reports
StoredProcesses
Delivery Days Analysis
SESUG 2007 Survey Reporting
Samples
Local intranet
100%

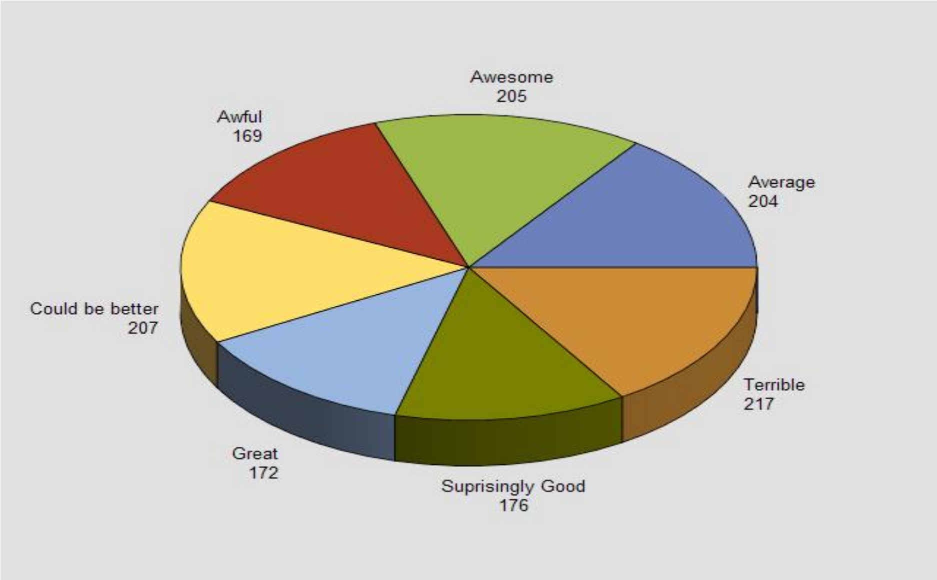
When the parameter prompts are displayed, make your selections and click **Execute**. The same report seen in EG will appear in a new browser window. One of the pie chart segments is displayed below.


SESUG 2007 Survey Reporting - Windows Internet Explorer
http://dell:8080/SASS
Google
File Edit View Favorites Tools Help
Links Running Well - Customize Links Martin Mans Welcome to Gmail
DebugBar
100%
SESUG 2007 Survey Repo...
General
Execution Options
Execute
Local intranet
100%

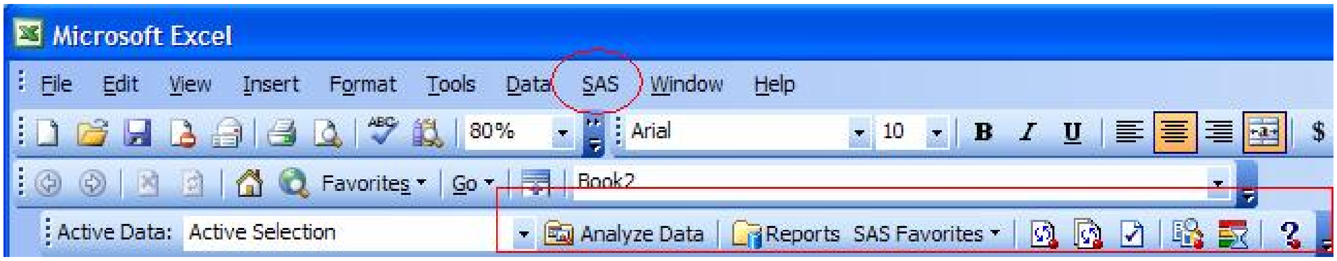
General

* Country:	USA
* City:	Hilton Head
* Gender:	F
* Income Level:	50,001-60,000

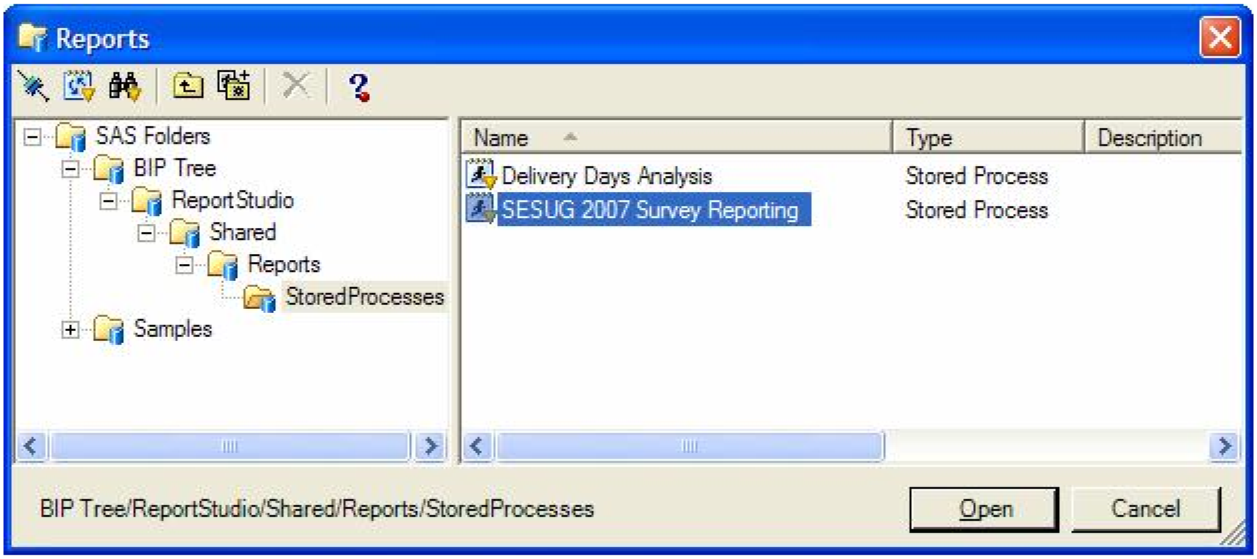
* Indicates a required field.



Stored Processes can also be executed from Excel if the Microsoft Add-On is installed. Note the SAS menu bar item and the SAS toolbars installed by the MS Add-On.



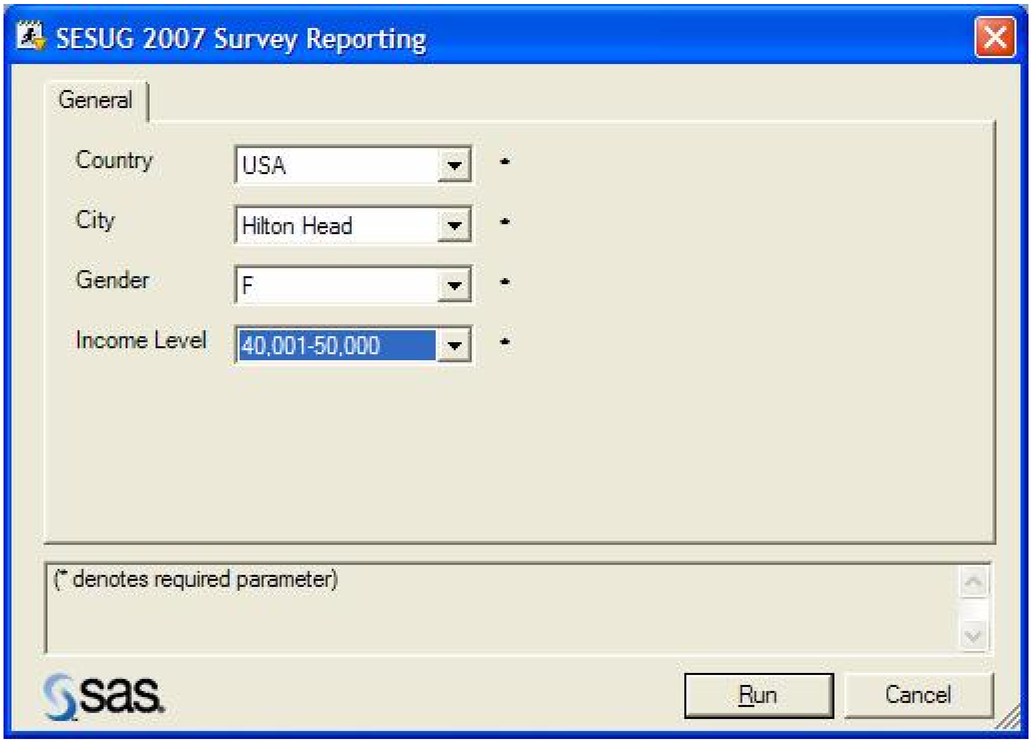
Click **Reports** in the SAS toolbar and select the SESUG 2007 Survey Reporting Stored Process.



Name	Type	Description
Delivery Days Analysis	Stored Process	
SESUG 2007 Survey Reporting	Stored Process	

BIP Tree/ReportStudio/Shared/Reports/StoredProcesses

When **Open** is clicked, as you might expect, the parameter prompt appears. Make the required selections and click Run.



Country USA *

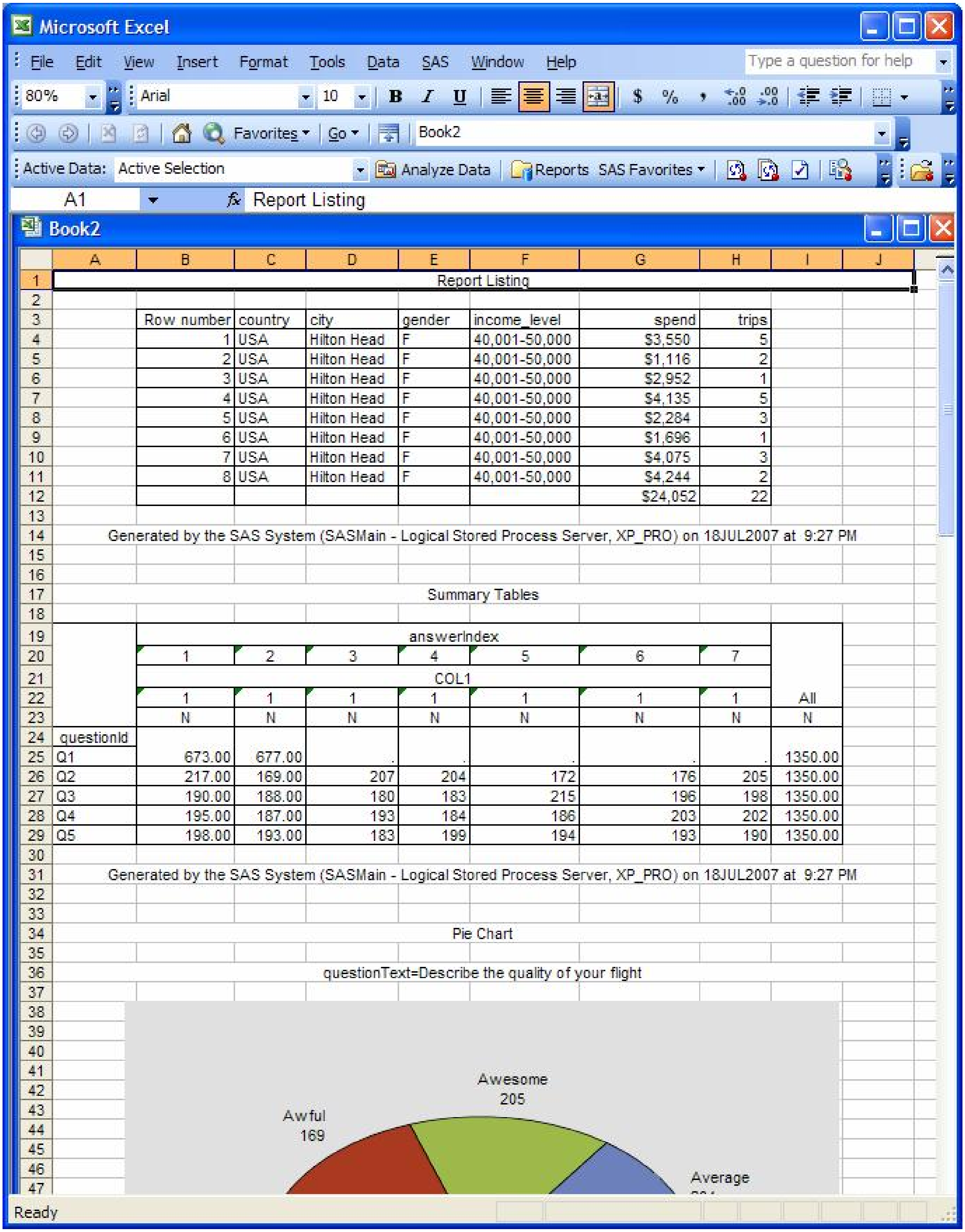
City Hilton Head *

Gender F *

Income Level 40,001-50,000 *

(* denotes required parameter)

The report results are returned to Excel, though all reports are in the same worksheet.



Report Listing

Row number	country	city	gender	income_level	spend	trips
1	USA	Hilton Head	F	40,001-50,000	$3,550	5
2	USA	Hilton Head	F	40,001-50,000	$1,116	2
3	USA	Hilton Head	F	40,001-50,000	$2,952	1
4	USA	Hilton Head	F	40,001-50,000	$4,135	5
5	USA	Hilton Head	F	40,001-50,000	$2,284	3
6	USA	Hilton Head	F	40,001-50,000	$1,696	1
7	USA	Hilton Head	F	40,001-50,000	$4,075	3
8	USA	Hilton Head	F	40,001-50,000	$4,244	2
					$24,052	22

Generated by the SAS System (SASMain - Logical Stored Process Server, XP_PRO) on 18JUL2007 at 9:27 PM

Summary Tables

	answerIndex							
	1	2	3	4	5	6	7	
	COL1							
	1	1	1	1	1	1	1	All
	N	N	N	N	N	N	N	N
questionId								
Q1	673.00	677.00	.	.	.	.	.	1350.00
Q2	217.00	169.00	207	204	172	176	205	1350.00
Q3	190.00	188.00	180	183	215	196	198	1350.00
Q4	195.00	187.00	193	184	186	203	202	1350.00
Q5	198.00	193.00	183	199	194	193	190	1350.00

Generated by the SAS System (SASMain - Logical Stored Process Server, XP_PRO) on 18JUL2007 at 9:27 PM

Pie Chart

questionText=Describe the quality of your flight

So far, we've seen reports delivered in EG, a browser window or in a single Excel worksheet. Doesn't the title of the paper say something about multi-sheet Excel ?! As you might know, the ODS ExcelXP tagsets *will* create multi-sheet Excel output when used in conjunction with Excel XP or Excel 2003. Unfortunately, it is NOT possible to invoke a stored process in Excel and *stream* ExcelXP tagset output directly to the application. And, EG doesn't deal very well with the XML output generated by the tagset either. The desired multi-sheet output can be delivered to a browser by doing the following work-around. Modify the Stored Process code so it contains additional directives and specifies the correct content-type.

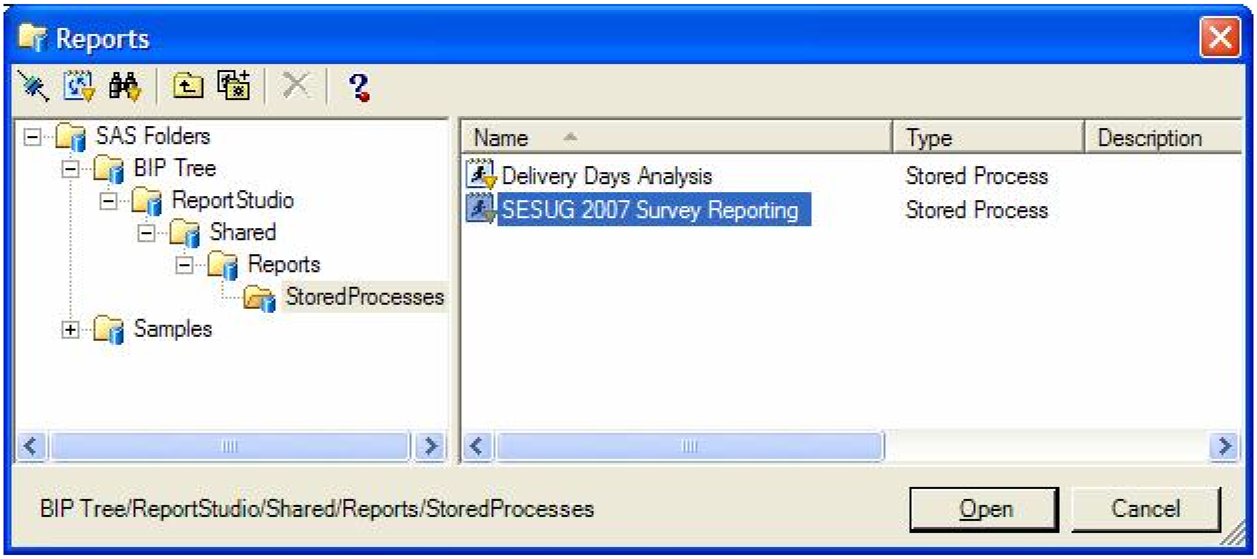
Navigate to the .sas file created by the Stored Process wizard and open it with a text editor. Add the following to the very top of the file, before the *** Begin EG generated code (do not edit this line);** comment.

```
%global _odsdest _odsoptions ;
```

Before the **%STPBEGIN** statement, special macro variables must be assigned indicating the ODS destination (and options thereof). In our case we want specify that the Stored Process is to use the ExcelXP tagset rather than the default ODS HTML destination. Additional ODS destination options may be specified in the _odsoptions variable. Invoke the appsrv_header function to let the browser know that the output we are creating is to be directed to Excel rather than to the browser window proper.

```
%let _odsdest     = tagsets.excelXP;
%let _odsoptions =;
%let rv           = %sysfunc(appsrv_header(Content-type,application/vnd.ms-excel));   * stream output to Excel ;
```

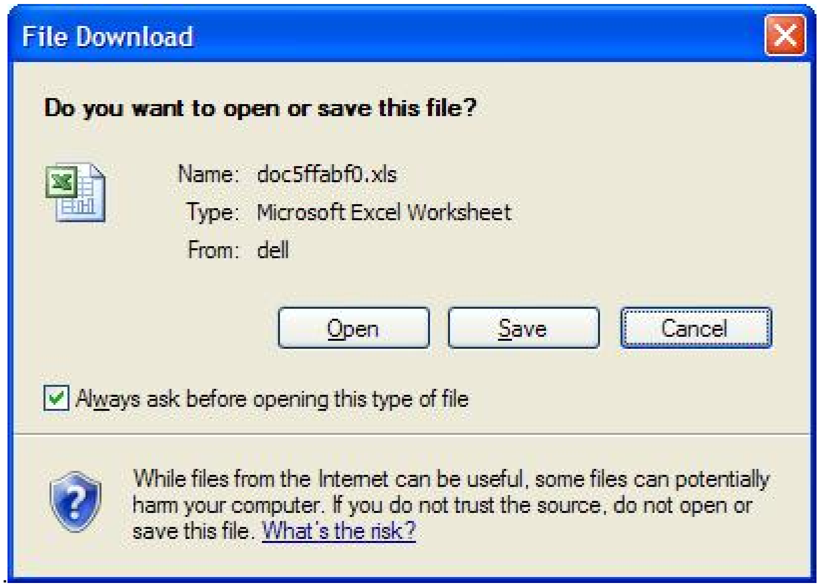
Save the file and direct your browser to the same Stored Process URL used on page 17, navigate to the Stored Process location in the metadata, select the report, click Execute, make the parameter selections and click Execute again.



Name	Type	Description
Delivery Days Analysis	Stored Process	
SESUG 2007 Survey Reporting	Stored Process	

BIP Tree/ReportStudio/Shared/Reports/StoredProcesses

Since the browser has been told to expect Excel output, it displays a prompt looking for direction in dealing with the file. We could save the file locally, or click Open to display the report in Excel.



File Download

Do you want to open or save this file?

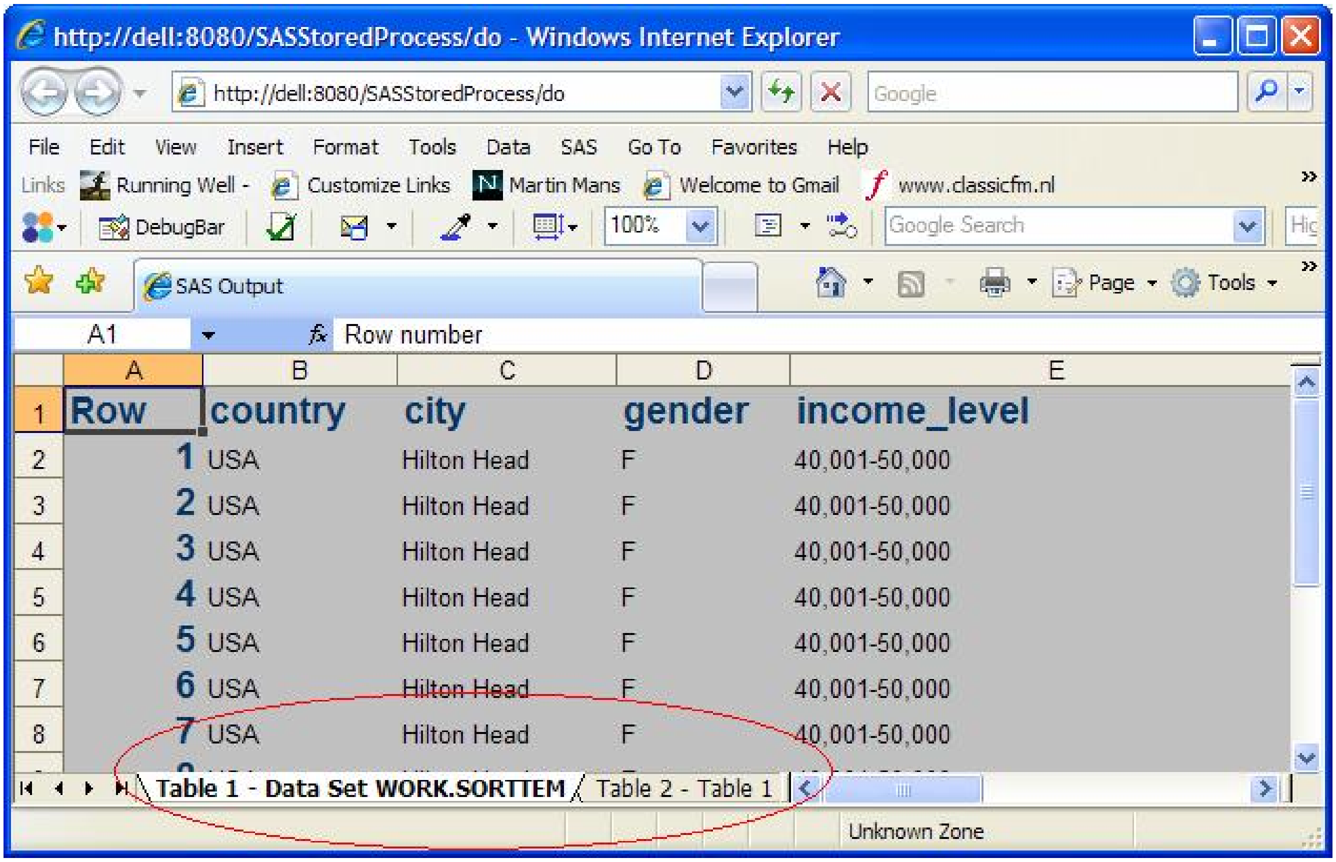
Name: doc5ffabf0.xls
Type: Microsoft Excel Worksheet
From: dell

Open Save Cancel

☑ Always ask before opening this type of file

While files from the Internet can be useful, some files can potentially harm your computer. If you do not trust the source, do not open or save this file. What's the risk?

Since Internet Explorer has the ability to invoke Excel directly in the browser window, a separate Excel application window is not opened. Notice that the workbook contains two worksheets, one for the listing report, another for the tabulate output. Yes, there's only two worksheets created. Unfortunately at this point in time the Microsoft Spreadsheet XML specification used by tagsets.ExcelXP does not support images, i.e. the pie charts, so only the text based reports are returned.



http://dell:8080/SASStoredProcess/do - Windows Internet Explorer

http://dell:8080/SASStoredProcess/do
Google
File Edit View Insert Format Tools Data SAS Go To Favorites Help
Links Running Well - Customize Links Martin Mans Welcome to Gmail www.classicfm.nl
DebugBar
100%
Google Search
SAS Output
Page Tools

A1 Row number

	A	B	C	D	E
1	Row	country	city	gender	income_level
2	1	USA	Hilton Head	F	40,001-50,000
3	2	USA	Hilton Head	F	40,001-50,000
4	3	USA	Hilton Head	F	40,001-50,000
5	4	USA	Hilton Head	F	40,001-50,000
6	5	USA	Hilton Head	F	40,001-50,000
7	6	USA	Hilton Head	F	40,001-50,000
8	7	USA	Hilton Head	F	40,001-50,000

Table 1 - Data Set WORK.SORTTEM / Table 2 - Table 1

Unknown Zone

CONCLUSION

SAS Enterprise Guide is an effective means to create process flows and subsequently convert the flows into a Stored Process. As we've seen, by default, Stored Processes do not create multi-sheet output. It is necessary to manually edit the .sas file associated with the Stored Process to define the appropriate values for the macro variables utilized by the Stored Process to create its ODS output using the ExcelXP tagsets. Unfortunately, the current implementation does not allow streamed XML output to be returned directly to Excel. Instead, one must set the MIME type to force the browser to invoke Excel to deal with the XML data and multi-sheet Excel reports can be generated. Alternatively, the Stored Process output options could be set to "none" and the appropriate tagsets.ExcelXP options set to create an output file which could be opened in Excel.

REFERENCES

DelGobbo, V. 2006. "Creating AND Importing Multi-Sheet Excel Workbooks the Easy Way with SAS® ".
Proceedings of the Thirty-First Annual SAS Users Group International Conference, 31. CD-ROM. Paper 115.
Available http://www2.sas.com/proceedings/sugi31/115-31.pdf.

SAS Institute Inc. "ODS MARKUP Resources". Available http://support.sas.com/rnd/base/topics/odsmarkup/.

ACKNOWLEDGEMENTS

We owe a debt of thanks to **Rupinder Dhillon** for her invaluable assistance in helping us navigate the EG / Stored Process jungle. **Vince DelGobbo**, as usual, was a generous fount of knowledge, even graciously answering dumb questions that a more careful read of his paper would have answered. Thank you to both of you.

CONTACT INFORMATION

Richard A DeVenezia
9949 East Steuben Road
Remsen, NY 13438
www.devenezia.com

Harry Droogendyk
Stratia Consulting Inc.
PO Box 145,
Lynden, ON L0R 1T0
conf@stratia.ca

APPENDIX A

```
libname survey meta library = "survey" repname = "Foundation" metaout=data;

data survey.answerSets;
infile cards dlm='|' truncover;
input answerSetId description $100.;
cards;
1 | Single answer on a scale of 1 to 7
2 | Single answer Yes or No
3 | Single answer on a scale of 1 to 7, flowery
run;

data survey.answers;
infile cards dlm='|';
length answerSetId index 8 answerText $40;
input answerSetId index answerText;
cards;
1 | 1 | Terrible
1 | 2 | Awful
1 | 3 | Could be better
1 | 4 | Average
1 | 5 | Great
1 | 6 | Suprisingly Good
1 | 7 | Awesome
2 | 1 | No
2 | 2 | Yes
3 | 1 | Rotten
3 | 2 | Stunk
3 | 3 | Putrid
3 | 4 | Bleh
```

```
3 | 5 | Worthy
3 | 6 | Fragrant
3 | 7 | Swoony
run;

data survey.questions;
infile cards dlm='|';
length questionId $4 answerSetId 8 questionText $50;
input questionId answerSetId questionText;
cards;
Q1 | 2 | Did you plan your trip using a Travel Agent
Q2 | 1 | Describe the quality of your flight
Q3 | 1 | Rate your car rental service
Q4 | 1 | Rate the demeanor of the receptionist
Q5 | 1 | Please rate the room
Q6 | 3 | My reaction to the survey is
run;

data survey.results(label     = 'Results delivered from external customer'
                                     drop    = cntry: cty: gendr: incme:
                                     );
  length personId             8
                gender               $1
                country              $12
                city
                income_level  $20
                spend                8
                trips                8
                ;

  format spend        dollar20.0
                trips        comma7.
                ;

  array gendr(2)      $1      ('M','F');
  array incme(6)      $20 ('0-40,000','40,001-50,000','50,001-60,000','60,001-70,000','70,001-
80,000','90,000-');
  array cntry(2)      $12     ('Canada','USA');
  array cty(2,10)     $20     ('Toronto','Lynden','Hamilton','Halifax','Montreal','Vancouver','North
Bay','Calgary','Edmonton','Winnipeg'
                                             'Hilton Head','New York','Dallas','San
Francisco','Erie','Raleigh','Chicago','Tacoma','Fort Myers','Bangor'
                                             );

  array A1 Q1_1-Q1_2;
  array A2 Q2_1-Q2_7;
  array A3 Q3_1-Q3_7;
  array A4 Q4_1-Q4_7;
  array A5 Q5_1-Q5_7;
  array A6 Q6_1-Q6_7;

  keep personId -- Q6_7;

  retain seed 200711;

  do i = 1 to 2;
    A1[i] = 0;
  end;

  do i = 1 to 7;
    A2[i] = 0;
    A3[i] = 0;
    A4[i] = 0;
    A5[i] = 0;
    A6[i] = 0;
  end;

  do personId = 1 to 1350;

    i1 = 1 + 2*ranuni(seed);
    i2 = 1 + 7*ranuni(seed);
    i3 = 1 + 7*ranuni(seed);
    i4 = 1 + 7*ranuni(seed);
    i5 = 1 + 7*ranuni(seed);
    i6 = 1 + 7*ranuni(seed);

    A1 [ i1 ] = 1;
    A2 [ i2 ] = 1;
    A3 [ i3 ] = 1;
    A4 [ i4 ] = 1;
    A5 [ i5 ] = 1;
    A6 [ i6 ] = 1;

       _cntry                 = ceil(ranuni(1)*2);
       country                = cntry(_cntry);
       city                   = cty(_cntry,ceil(ranuni(1)*10));
       gender                 = gendr(ceil(ranuni(2)*dim(gendr)));
       income_level   = incme(ceil(ranuni(3)*dim(incme)));
       trips                  = ceil(ranuni(4)*5);
       spend                  = trips*ceil(ranuni(5)*2000)+1000;

    output;

    A1 [ i1 ] = 0;
    A2 [ i2 ] = 0;
    A3 [ i3 ] = 0;
    A4 [ i4 ] = 0;
    A5 [ i5 ] = 0;
    A6 [ i6 ] = 0;
  end;

  stop;
run;

proc freq data = survey.results;
       tables gender trips income_level city*country /nocum nopercent norow nocol;
       tables spend*trips  /nocum nopercent norow nocol;
run;

proc transpose data=Survey.Results out=survey.vector;
  by personId;
  var Q1_1 -- Q5_7;
run;

data survey.vector2;
  set survey.vector;
  questionId = scan(_name_,1,"_");
  answerIndex = input(scan(_name_,2,"_"),best12.);
  drop _name_;
run;

proc sql;
  create table survey.vector3 as
  select vector2.*, questions.questiontext, questions.answerSetId, answers.answertext
  from survey.vector2 as vector2
        , survey.questions as questions
        , survey.answers     as answers
  where vector2.questionId = questions.questionId
    and questions.answersetid = answers.answersetid
       and answers.index = vector2.answerindex
  ;
quit;
```