Paper 058-2007

Data on Demand with Enterprise Guide: A Cheap and Easy Approach to Real-time Database Access

Richard F. Pless, Ovation Research Group a division of ICON Clinical

ABSRACT

SAS Enterprise Guide is often promoted as an easy introduction to SAS or as a tool for business analysts who prefer graphic interfaces to writing extensive amounts of code. However, it has some interesting functionality that even the truest die-hard programmer could appreciate such as its ODBC interface.

A frequent task for SAS programmers and DBAs alike is creating SAS datasets from databases. Typically this process involves either an exporting utility (such as DBMS/Copy) or implementing additional SAS Modules such as SAS Access or SAS Share and using their ODBC functionality. However, SAS does have another simple way to gain read-only access to a database, the ODBC functionality in SAS Enterprise Guide.

This paper will describe a process for accessing real-time data from a SQL Server database using SAS Enterprise Guide. Additionally it will provide best practices for data structure and developing SAS Enterprise Guide programs.

INTRODUCTION

SAS Enterprise Guide on a PC helps solve the problem of real-time data access through the use of ODBC drivers and query objects. This paper will walk you through the steps of configuring an Enterprise Guide project that creates permanent SAS datasets from views on an enterprise database. The steps are very straightforward. First, create an ODBC Data Source in your operating system. Second, configure an ODBC data source in your Enterprise Guide application. Third, query your data to create SAS datasets. Fourth, manipulate your data with Code Objects.

The steps illustrated below will show you how to connect to a database housed on a local SQL Server database. The steps will be similar for connecting to other platforms or over networks. Now let's look at each of these steps in more detail.

ESTABLISHING THE ODBC CONNECTION

The first step in accessing a database from an Enterprise Guide project is to make the operating system aware of the data source. So our first task is to properly configure an ODBC connection in our operating system; this connection is also referred to as a data source. An ODBC connection is a link from the operating system to a target database. ODBC stands for Open Data Base Connectivity; it is a programming standard that all modern database systems support.

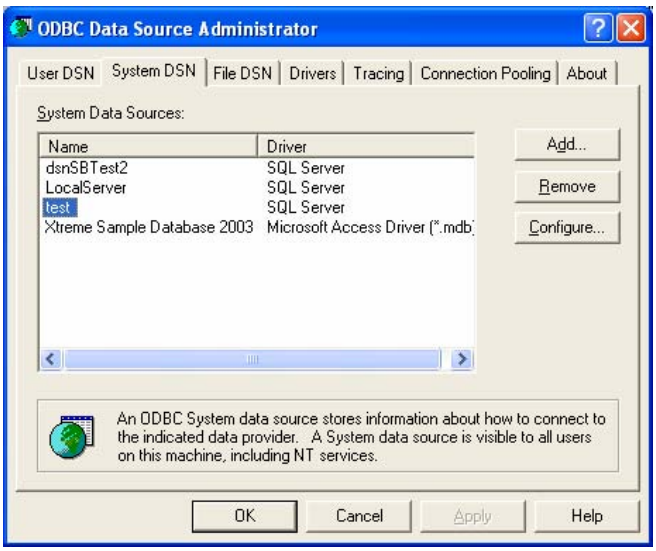

Figure 1. Creating a DSN through the Data Source Administrator.

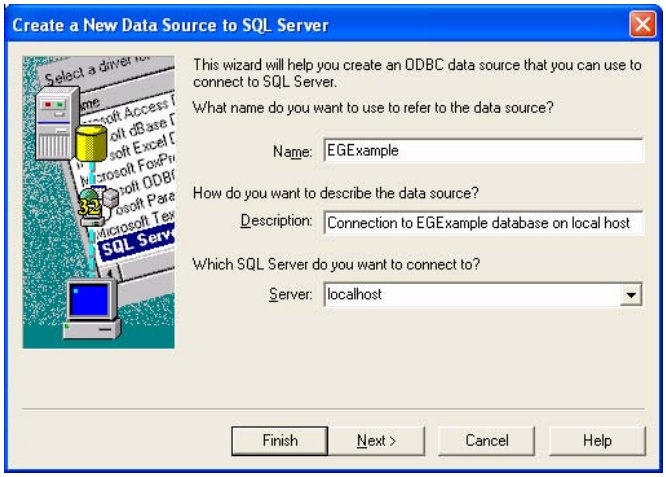

Figure 2. The New Data Source Wizard.

We start by configuring a data source in the operating system. Go to Start, Control Panel, Administrative Tools, and open the ODBC Data Source Administrator. You will see a window similar to the one shown in Figure 1 above. It lists all of your available data sources. Note that there are three types of Data Source Names (DSNs). A User DSN resides on the local machine and cannot be accessed by other users or Windows Services. A File DSN is a file that can be shared with other users who have similarly configured drivers. A System DSN is local to the machine but accessible to all users as well as

Windows services. It cannot be transferred to other machines. In this example we will configure a System DSN. The other DSNs would work for our example as well.

Select the DSN tab and click on the 'Add…' button. You will be asked to select a driver. Select the driver for the appropriate database platform. In our case it will be 'SQL Server'. Click 'Finish'. The New Data Source wizard will begin prompting you for information to establish the connection. The first screen, shown in Figure 2, will ask for a name, description and which server you want to connect to.

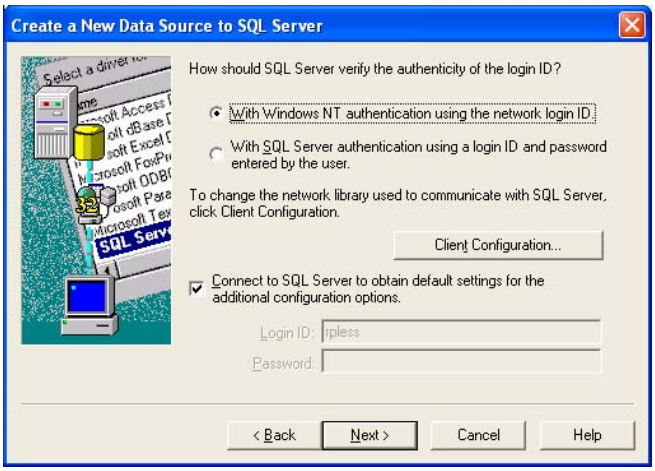

Figure 3. Select the appropriate authentication method for your connection.

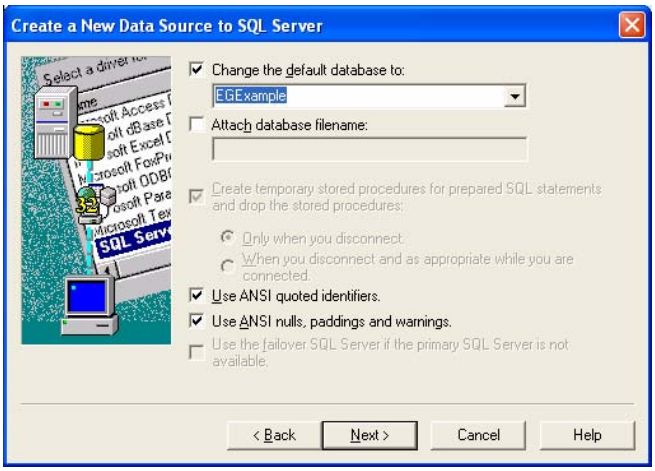

Figure 4. Point to the correct data source.

After answering these questions, you will be prompted for your authentication information as shown in Figure 3. Remember that this screen will differ for different databases. Next, you will be asked for details about the data source, including which database you want to access on the server and connection details. The default connection settings for SQL Server are shown in Figure 4. Note that we selected the EGExample database.

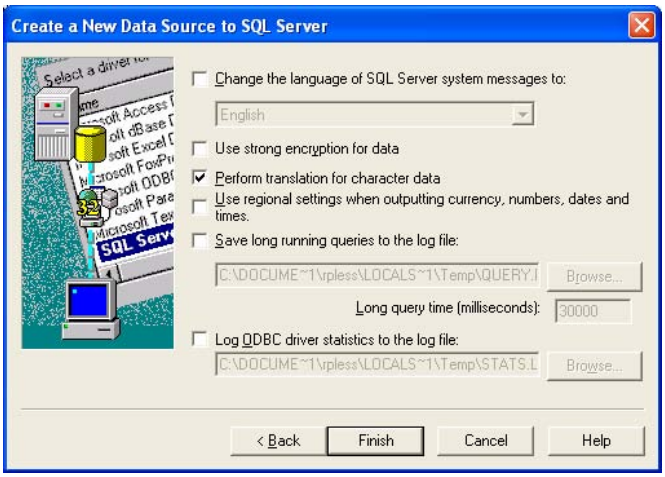

Figure 5. Accept the default database connection settings.

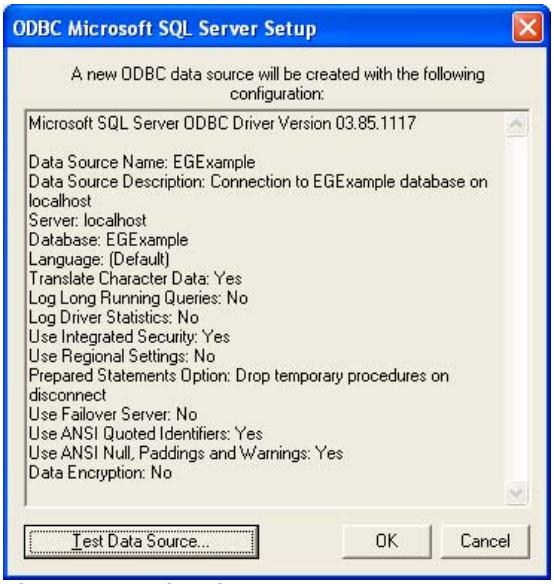

Figure 6. Testing the Data Source.

The final window will prompt will prompt you for additional details about the connection as shown in Figure 5. Once you finish configuring the database, click 'Finish' and you will see a summary of your connections. Click on the Test Data Source button and the OS will try to access the data source. You should see a message window saying that the test was successful. If the test results window indicates that the test was unsuccessful, ensure that you have appropriate access to the database and that your connection parameters are correct. This will probably require the assistance of your Database Administrator.

NAVIGATING THE ENTERPRISE GUIDE INTERFACE

Now that you've configured your data source, you are ready to get started with Enterprise Guide. If you already know Enterprise Guide, then skip to the next section. If you are new to Enterprise Guide, the description below should give you just enough of an introduction to allow you to create data extracts.

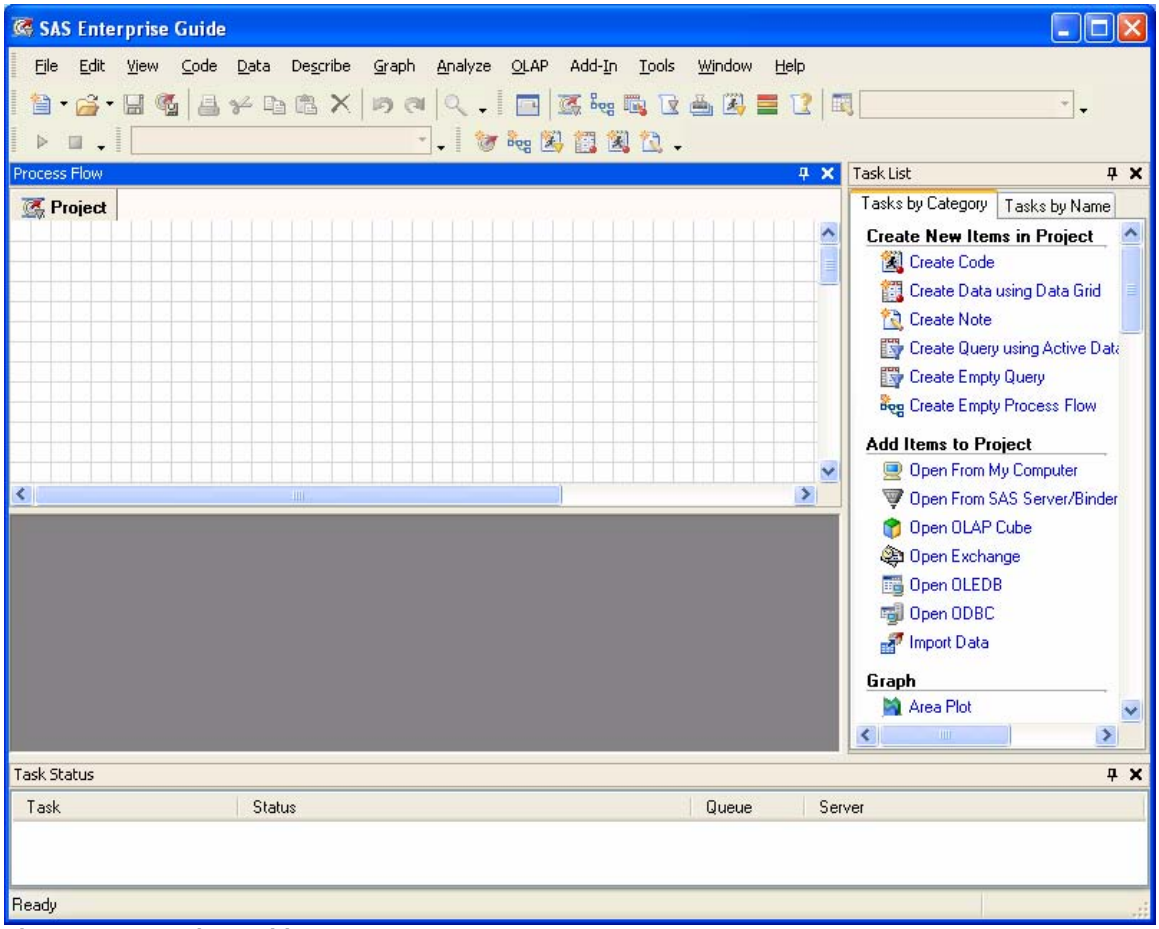

Figure 7. Enterprise Guide Screen Layout.

Enterprise Guide's graphical user interface consists of a number of windows and is highly customizable. This is great for development but can make demonstrations challenging. For these examples, I will focus on three windows; Process Flow, Task Status, and Task List. You can add windows by going to the View menu and selecting the window that you would like to add. You can move or remove windows as you would in any other Windows application, by highlighting the window header and dragging it or by clicking on the 'X' in the upper right, respectively. My preferred layout is shown in Figure 7 above with the Process Flow on the upper left, the Task List on the right and the Task Status on the bottom. Note the gray area in the middle of the Enterprise Guide Window. This is my workspace. This is where I will edit the code inside of Code Objects or review log files.

Enterprise Guide files are known as projects. Each project consists of a number objects. Projects are constructed by adding tasks to the Process Flow window. Objects can be connections to data sources, standard SAS data step operations like sort or transpose, statistical analyses like regressions or ANOVAs, base SAS procedures, Graphics or customized SAS code. For any object other than a Note or a custom code object, a wizard will start once the object has been added to a project. The wizard will prompt you for of all of the parameters that you need to successfully execute the object.

To add an object to a project, simply click on the desired object in the Task List, and drag the object onto the Process Flow window. To modify objects simply double click on them and they will open for editing. When a project executes, it executes objects from left to right and top to bottom. To change the order of execution, you simply move the object from one position in the Process Flow window to another. To execute a project, right click on the Project tab and select 'Run Project'.

For a more thorough introduction to Enterprise Guide, the interested reader should use the SAS Enterprise Guide tutorial located in the Help Menu.

CONFIGURING DATA SOURCES IN ENTERPRISE GUIDE

As described earlier, to add a data source you simply drag and drop the 'Open ODBC Data Source' onto the Process Flow window.

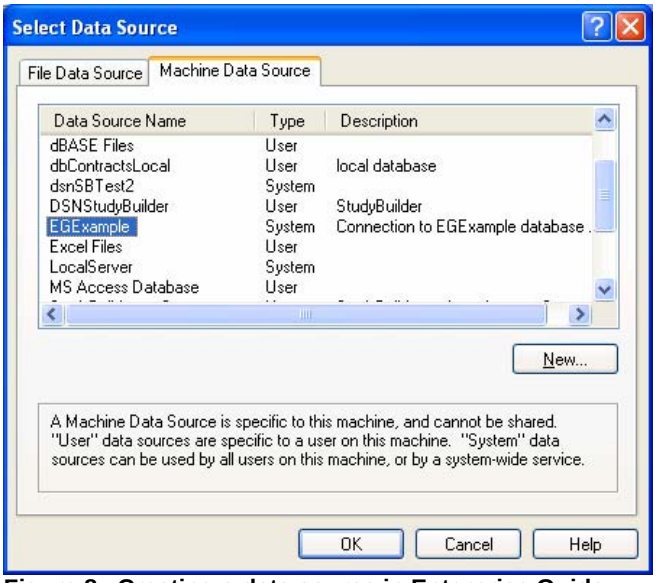

Figure 8. Creating a data source in Enterprise Guide.

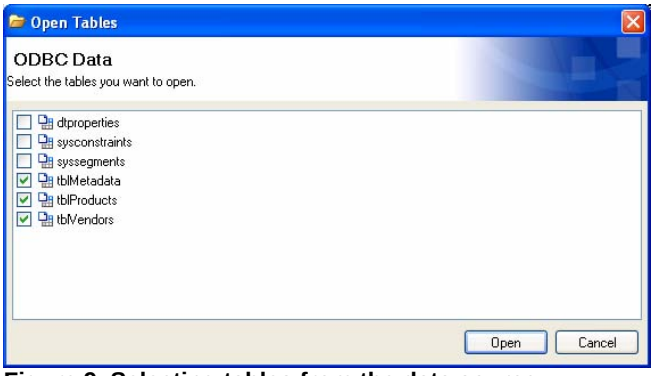

Figure 9. Selecting tables from the data source.

After you have dropped the ODBC Data Source object onto the Process Flow, you will be prompted to select a data source and then tables within that data source as shown in Figures 8 and 9 above. Note that you will be able to select any table or view that you have read access to in the database. In Figure 9 above the example database has three data tables; tblMetadata, tblProducts and tblVendors. The other three tables are system tables.

After clicking the 'Open' button, each table (or view) will be represented by a Data Source object in the Process Flow window as shown in Figure 10 below. Each Data Source object will occupy its own line and they will all be in one column on the left side of the Process Flow window.

There are two very important issues that should be remembered when configuring Data Source Objects. First, varchar data can lead to problems with SAS datasets. The ODBC connection will base the length of the character field in SAS on the longest varchar value in the table. when it is first used. For example, suppose that you have a field called fname and the longest name in that field is "Bill". When you first create that Data Source object in SAS EG, the length of the character field will be 4. If you later enter new data and one of the values for fname is "William" the length of the field in the Data Object will remain 4. Whenever possible use hardcoded char(xx) datatypes in your database to ensure that you prevent data loss.

Second, SAS has difficulty with some datatypes such as Timestamps. If you are trying to access Timestamp data, the field will result in nulls in SAS. In some cases the entire table may be unavailable if a single datatype is Timestamp. This can also affect views that have been built off of tables with those datatypes. If you are having problems accessing data in views or tables, check the datatypes in the tables. You may need to create new tables without those datatypes in order to access the data.

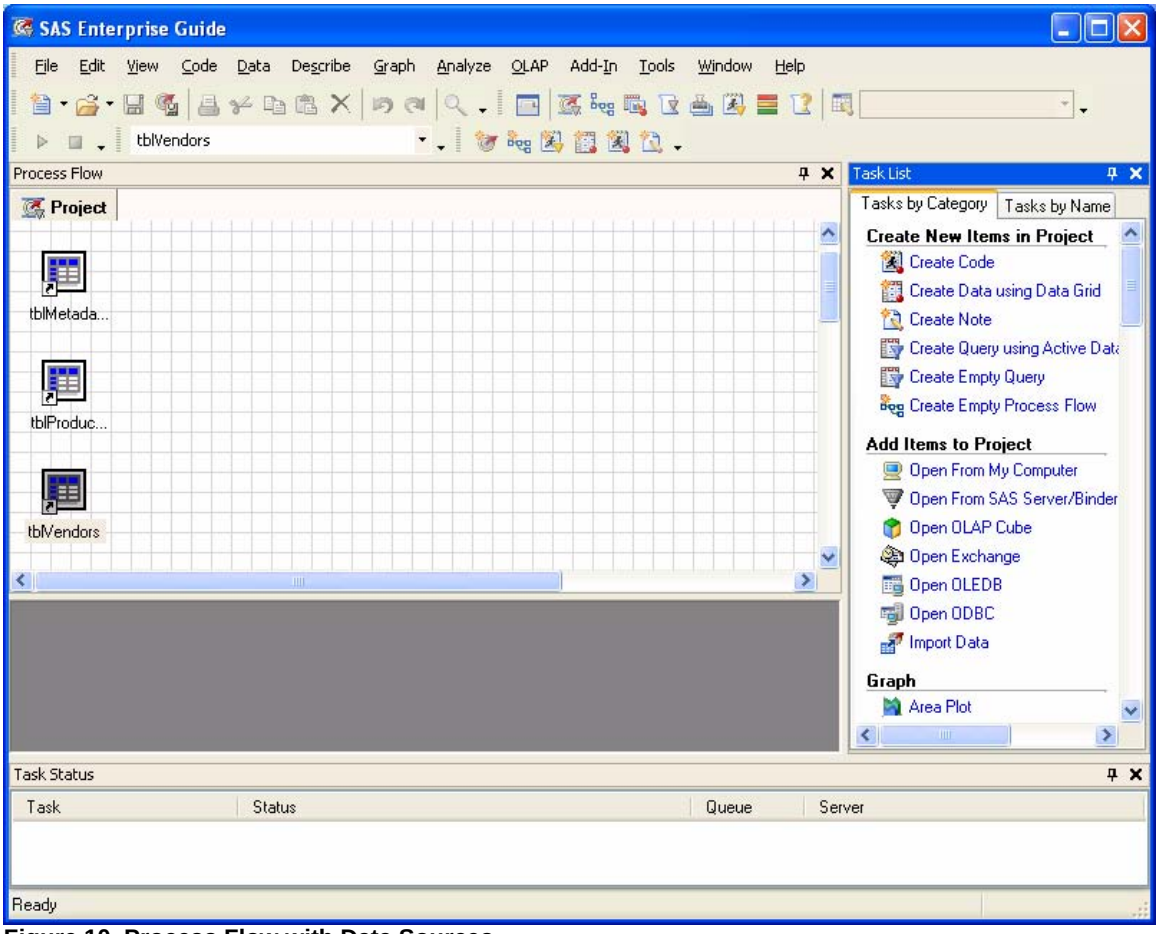

Figure 10. Process Flow with Data Sources.

QUERYING YOUR ENTERPRISE GUIDE DATA SOURCES

Now that you have access to your data, you're ready to begin manipulating it. You can manipulate data in Enterprise Guide by creating Code objects (as we'll see below) or by creating Query objects. Code objects allow you to execute SAS code against datasets. Query objects allow you to query ODBC data sources or SAS datasets.

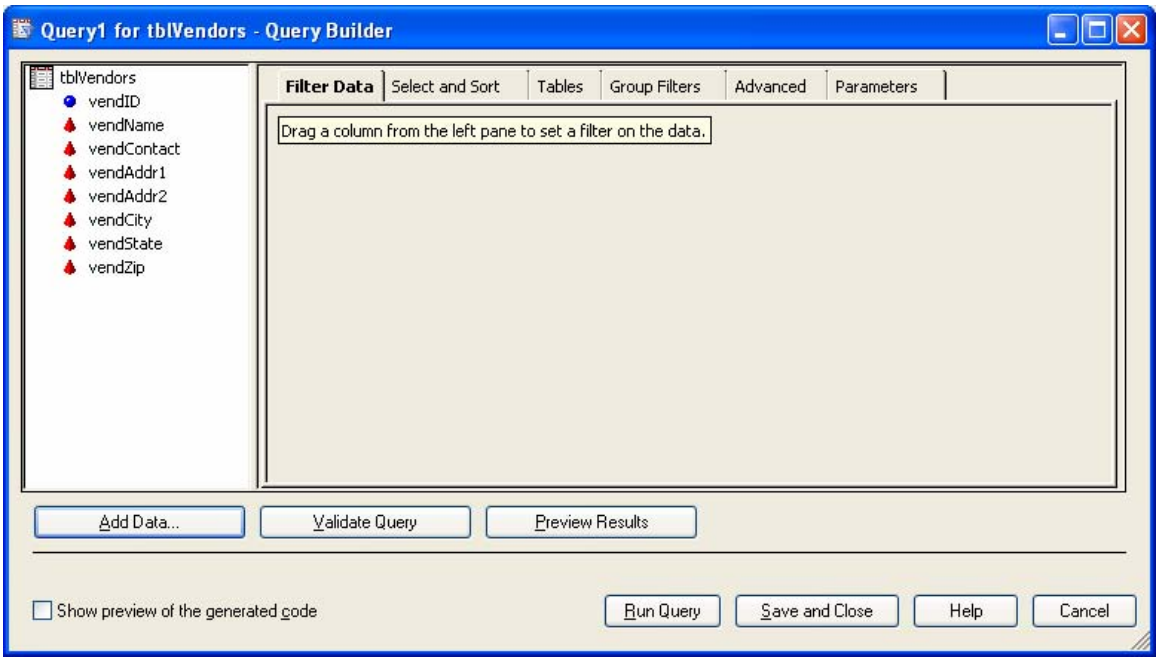

Figure 11. Creating a Query.

You can start a query in one of three ways. If you right click on a data source, and select Create Query, the Query object will open for editing as in Figure 11. Note that the data source is pre-populated with the table name and the columns to the far left. A second way to start a query is to highlight a data source and select 'Create Query using Active Data' from the Task window. Again, the active data sources and their component columns will be listed to the left. Finally, you can drag a 'Create Empty Query' from the Task list to the Process Flow. A new query will open without any data sources listed.

The Query window is divided into three parts: a data source listing to the left of the window, action buttons on the bottom of the window and a series of tabs to the right side of the window that allow you to modify the query. The data listing will include every active table in the Query as well as each of its component variables. Numeric variables are identified with a blue ball while character variables are identified with a red cone.

The buttons at the bottom of the screen are self explanatory. You can add data with the 'Add Data' button. You can validate a query (not generating results) with the 'Validate Query' button. You can preview query output with 'Preview Query' button. You can run the query without closing the window with the 'Run Query' button.

When you click on the 'Add Data' button you will be prompted for where to look for other data sources (e.g., 'Project', 'Exchange', 'SAS Servers', etc.). Once you select a location, you will be prompted for a data source. Once you select the data source, it will appear in the data listing at the left of your window below your current data set. Note that when you add a data source, Enterprise Guide will attempt to link the new table to existing tables. You will receive a warning if it can't make a link. It will attempt to link tables on similarly named columns. If it's faced with numerous choices, it will make a guess. Be careful to check the links created in the Tables tab before proceeding with the query (we'll discuss the Tables tab shortly).

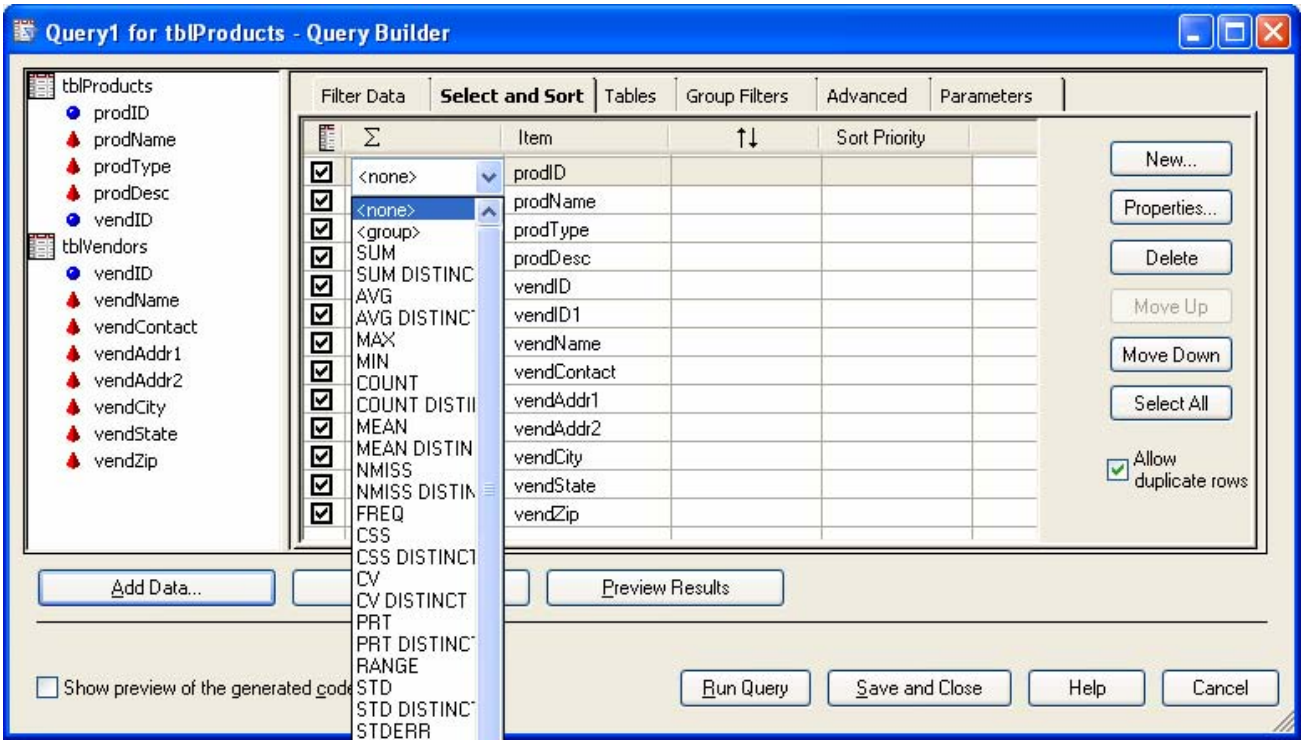

Figure 12. Selecting and sorting variables in a query.

Now let's take a closer look at the various tabs that allow you to edit your query. Rather than starting with the Filter tab, let's start with the Select and Sort tab as shown in Figure 12 above. As the name implies the Select and Sort tab allows you to select variables to include in the query. By default, all variables are included in the query; no summary functions or sorts are applied. You can delete fields by unchecking the select box to the left of the variable name. You can add a variable by dragging it from the data list to the left of the window or by typing a name into the Item column.

You can apply a sort by clicking on the box next to the variable name in the third column. You will be prompted to select Ascending or Descending order. If you apply more than one sort, you can order the sort priority by using the box in the fourth column. You can group fields or apply aggregate functions by selecting them from the drop down list in the first column next to the corresponding variable name.

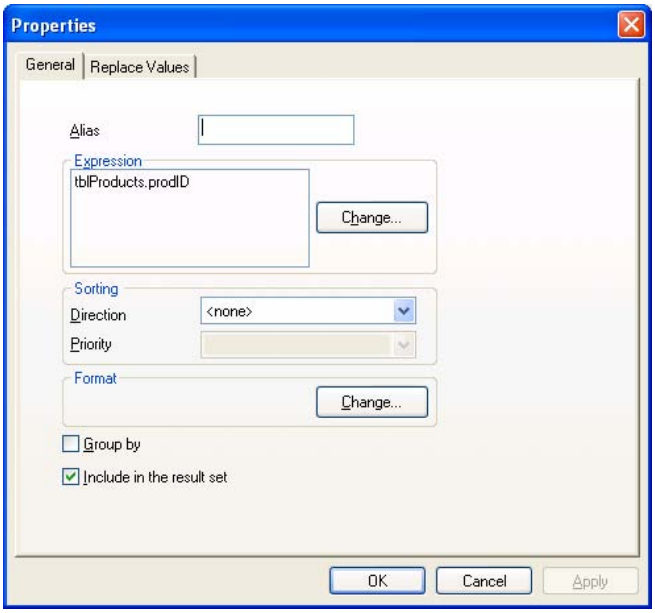

Figure 13. General field attributes.

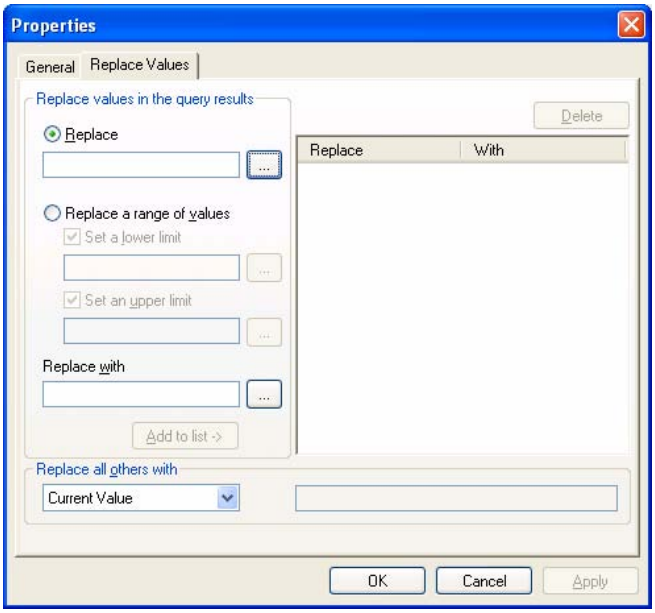

Figure 14. Replacing field attributes.

You can change other field attributes by highlighting a field and selecting the Properties button to the right. A Properties window will open as shown in Figure 13 above. Here you can rename the field by applying an alias, you can create a SQL expression in the Expression box, or you can change sort orders or formats. You can Group your results by that field or exclude them from the result set. The second tab in the Properties window is the Replace Values tab. In this tab you can substitute raw data values for others that you specify. You can also set upper or lower bounds of the values. For the purposes of this example we won't change any of the properties.

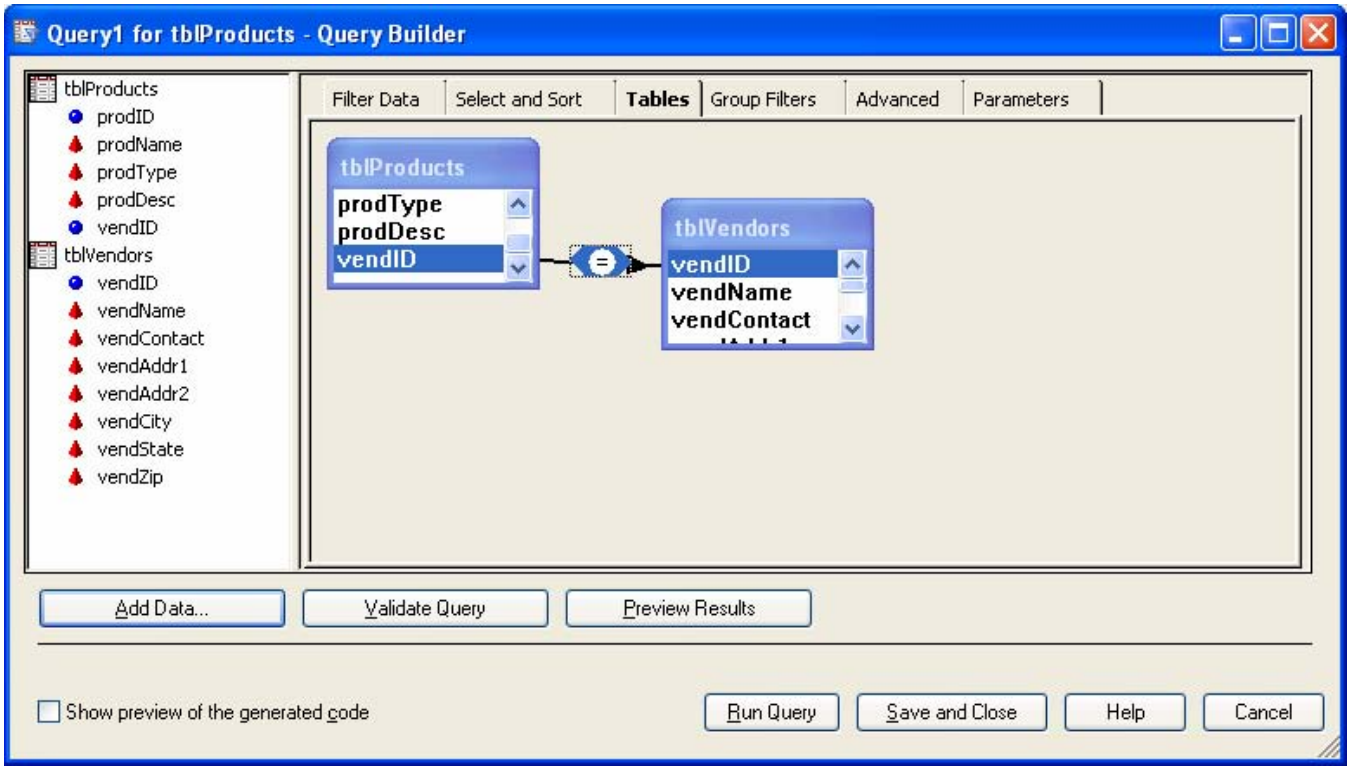

Figure 15. Joining Tables in the Table Tab.

The Tables tab allows you to manage table joins. Two tables can be joined manually by dragging and dropping the foreign key field from one table to the matching key on the other table. A line will appear in connecting the two tables with a blue hexagon in the middle as shown in Figure 15 above.

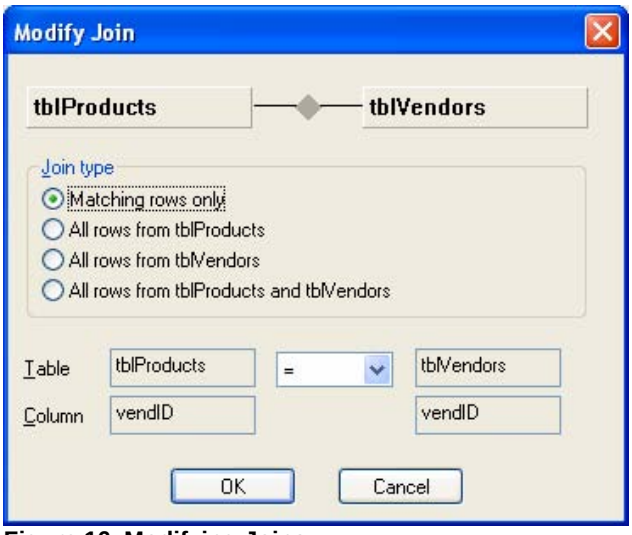

Figure 16. Modifying Joins.

Right clicking on the blue hexagon will allow you to modify or delete the join. The Join Modification window is shown in Figure 16 above. Note that you can't change the tables or columns in the Join, only the join type and the criteria between two fields.

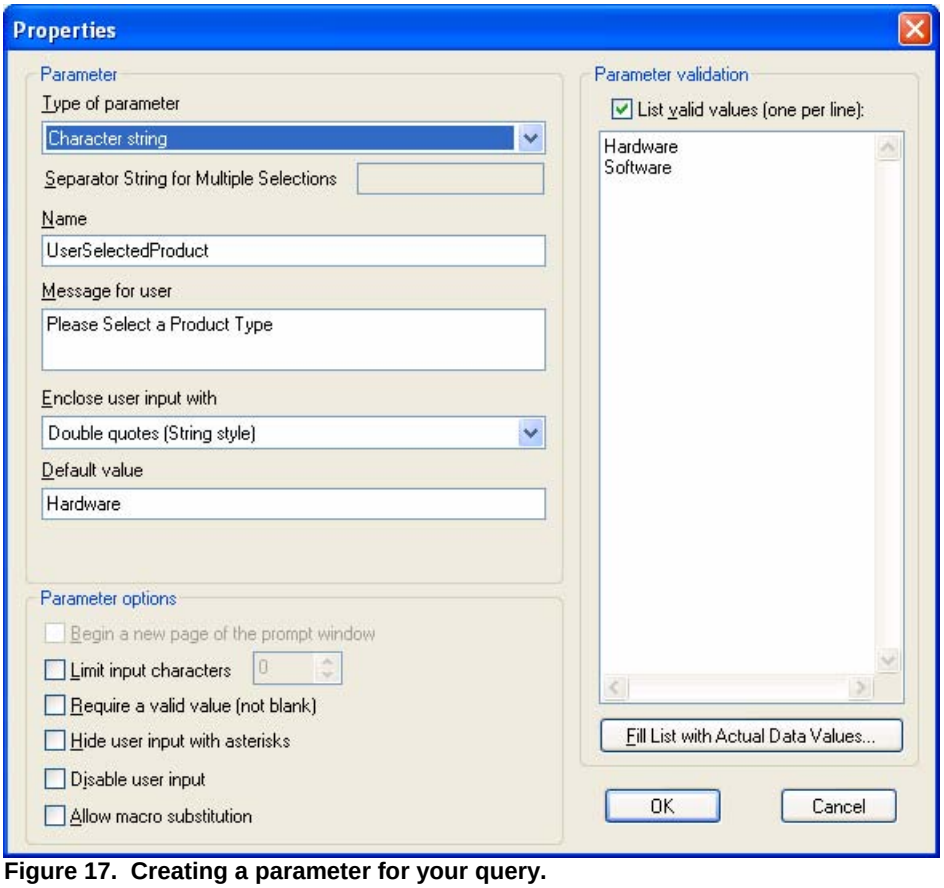

Figure 17. Creating a parameter for your query.

You can also create parameters that prompt users for inputs from the Parameters tab. These parameters can then be used in filters. Creating a parameter is simple, just click on the New Button and you will be prompted for parameter properties as shown in Figure 17. Name is the name of the parameter and it follows normal SAS naming conventions. Message for User is what is displayed in the prompt window. You can limit the user to a number of predefined responses by listing valid values in the 'Parameter validation' section. Usually you'll want to fill this list with values from an existing table. In this example, I only wanted values that were in the ProdType field of the product table. Clicking on the 'Fill List with Actual Data Values…' button prompts you for the source of allowable values. As you can see in Figure 17, the list has been populated with the two allowable values in my ProdType field.

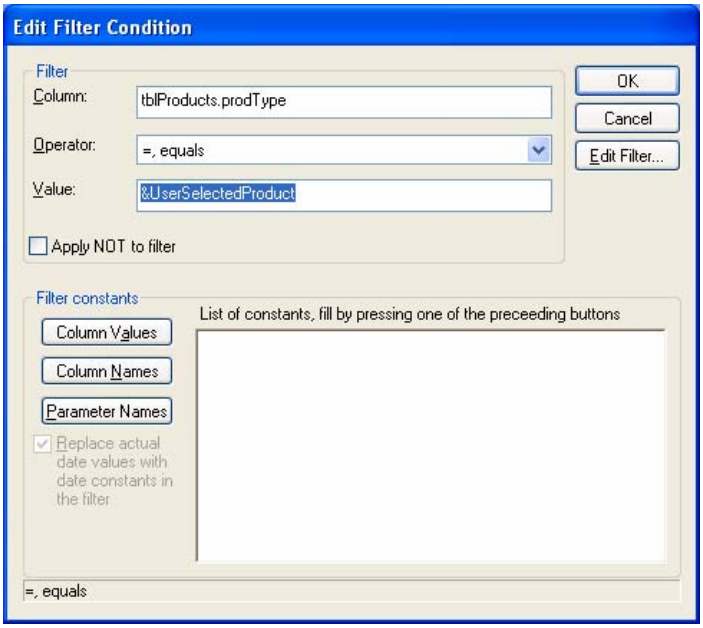

Figure 18. Creating a Filter.

Now that I have created my parameter, I can use it in a filter by going to the Filter tab. To apply a filter, drag and drop a variable from the field list onto the Filter tab. An 'Edit Filter Condition' window will appear as shown in Figure 18. Here you can edit the filter. The name of the variable is in the Column field. The default operator is '='. You can enter a value into the Value field or you can select it from a list of Column Values, Column Names or Parameter Names. To generate the list, click on the corresponding button and the list will appear in the 'List of constants' window. Double click on the desired value and it will move to the Value field. Click OK to save the filter.

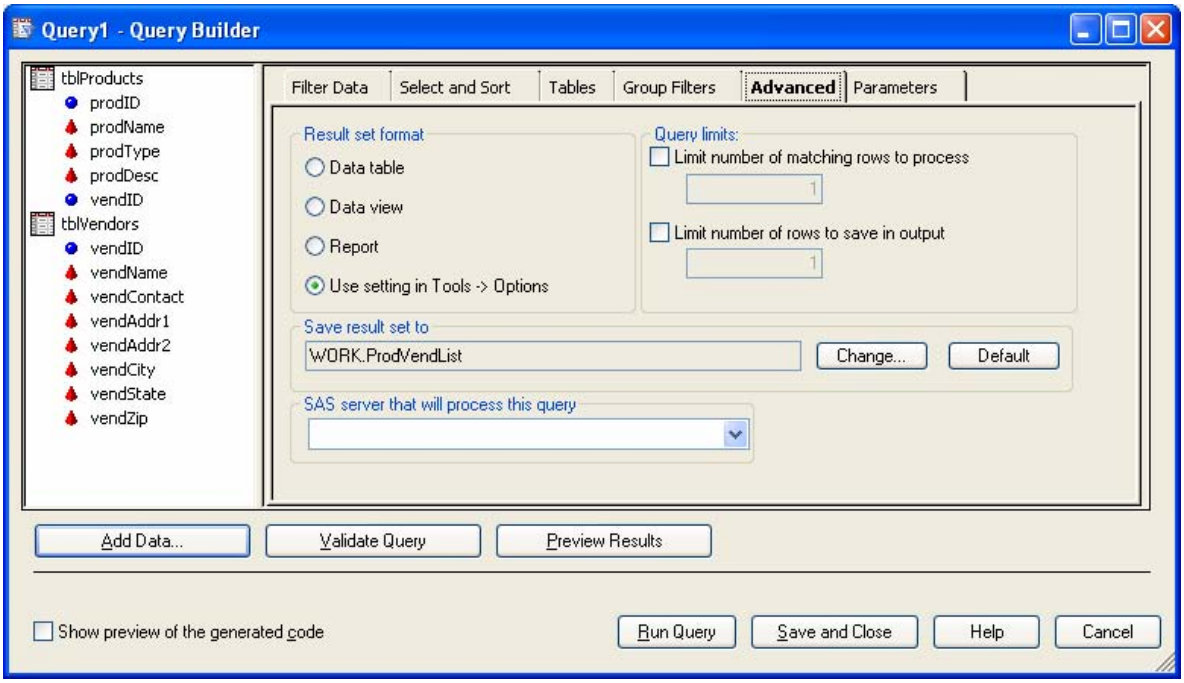

Figure 19. Configuring Output Destination in the Advanced Tab.

The Advanced tab depicted in Figure 19 allows you to modify your output. You can change the dataset names as well as the default library by clicking on the change button. Alternatively, you can send your output directly to a report or a view. You can also control the number of rows processed or output using the Query limits in the upper left.

Figure 20. Prompt for Parameter value.

You're now ready to save and run your project. Click the Save and close button to save your query. Save your project by going to the File Menu and selecting Save Project. Run your project by right clicking on your Process flow tab and selecting Run <Project Name>. Since we created a filter that requires a parameter, you will be prompted for a Product Type.

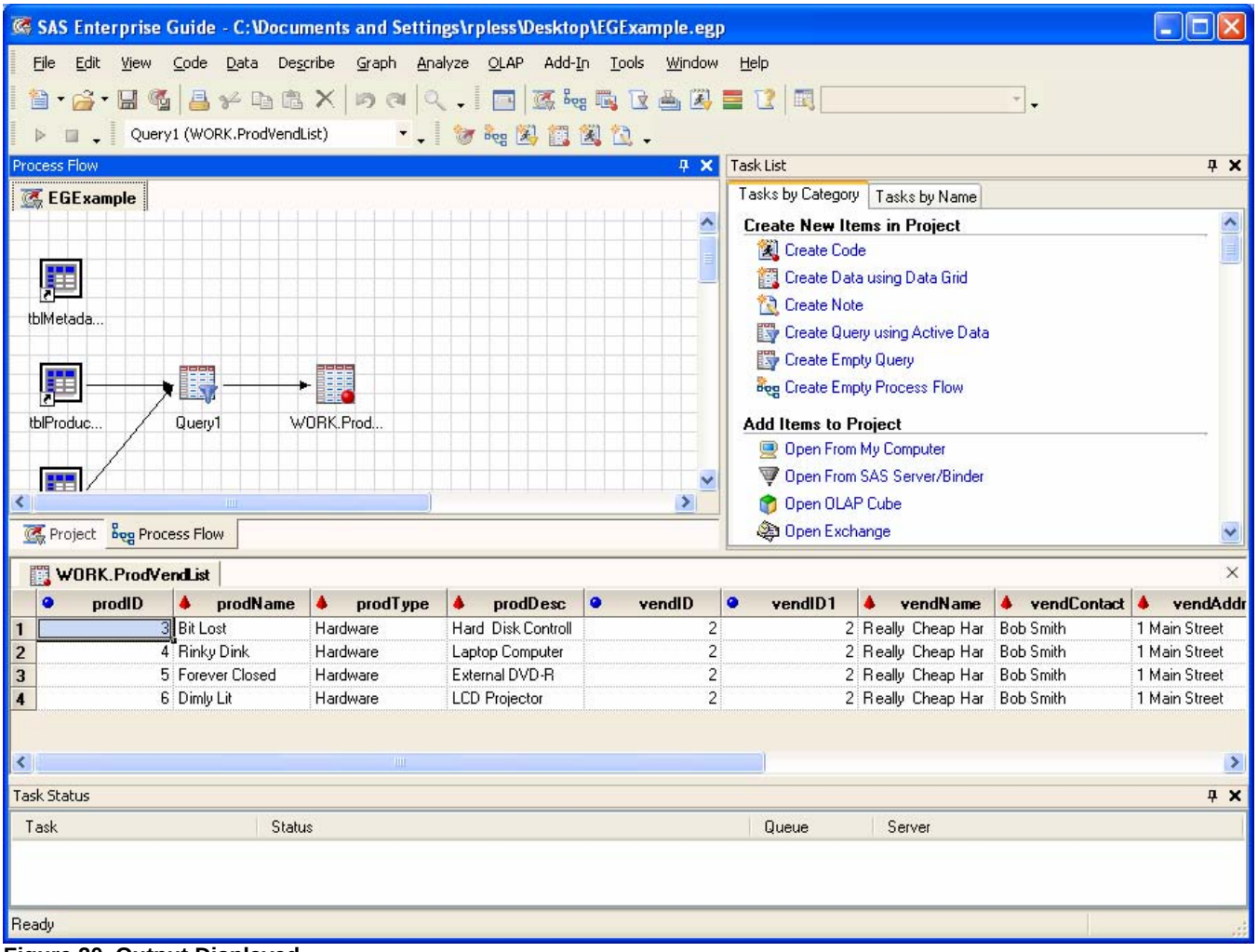

Figure 20. Output Displayed.

After you run your project, note that a temporary SAS dataset is created to the right of your query. If you double click on the SAS dataset you will see the contents in the work area in the center of your screen.

IMPROVING ON THE BASIC DATA ACCESS DESIGN

Thus far we have only created a temporary SAS dataset without labels. Saving the query output to permanent SAS datasets can be done easily with a Code object that defines the library where the permanent SAS datasets will reside and code that copies the temporary datasets (e.g., DATA Steps, PROC COPY, etc.). Another approach is to kill two birds with one stone and add labels as the datasets are being saved. Since the automation is key to so many data management functions, below is an automated approach to applying labels to SAS datasets.

There are a few things to note about the add labels code below. First, it requires another dataset(tblMetadata) that contains a list of SAS temporary dataset names(TABLE), field names(VARIABLE), and labels (LABEL). Second, it assumes that any

dataset that you apply the macro to has corresponding records in tblMetadata. For instance, if you have a tblProduct dataset, it assumes that there is a record in tblMetadata for each variable in tblProduct and that each record has a value in the LABEL field.

The code itself will generate a DATA STEP for each permanent data set that you want to create using CALL EXECUTE. Next it will create a label statement for each variable using data from tblMetadata to populate the details of CALL EXECUTE commands.

```
*** create a library
libname sasout 'c:\mysasfiles';

*** add labels to datasets ***;
%macro addlabel(dsn);
      data _null_;
            *** use call execute to create dataset on fly ***;
            call execute ("data sasout.&dsn; set work.&dsn;");

            *** read records from label source dataset where
            *** formname is appropriate dataset;
            do until(end_tbllabel);
                  set work.tblMetadata (where=(table="&dsn"))end=end_tbllabel;
                  *** create a label statment (max len=200 char) for each record
                  *** in label source dataset;
                  call execute( 'label ' !! variable !! ' = ' !!
quote(substr(trim(label),1,200)) !! ';');
            end;

            *** close virtual datastep ***;
            call execute ("run;");
            stop;
      run;
%mend
```

We have found this code to be very useful in our work to ensure that SAS programmers and DBAs are always looking at identical data structures. In the Enterprise Guide project that creates extracts, we abstain from joins and instead focus on a one to one relationship between SAS datasets and database tables or views. This simplifies trouble shooting and delineation of programming tasks.

CONCLUSION

This paper has demonstrated an easy approach to real-time data access to a database using SAS Enterprise Guide. Using functionality available in Enterprise Guide, along with the instructions presented in this paper, programmers have access to data residing in databases with minimal effort.

CONTACT INFORMATION

Richard F Pless
Director, Data Operations
Ovation Research Group, a division of ICON Clinical
600 Central Avenue, #265
Highland Park, IL 60035
rpless@ovation.org

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. SQL Server is a registered trademark of the Microsoft Corporation. ® indicates USA registration.

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