Using SAS/EIS® to Front-end and Exploit Your Data Warehouse

James M. Alessio, Executive Director of Enrollment Services, VCU, Richmond, VA
Dennis B. Catley, Systems Analyst, General Injectables & Vaccines, Inc., Radford, VA
Ben T. Cochran, Product Marketing Manager for SAS/EIS, SAS Institute, Cary, NC

Abstract

SAS/EIS Software, with its forty plus pre-written objects available in release 6.11, provides an interactive environment that can be very useful in building many types of applications. These SAS/EIS applications can front-end any part of the SAS System as well as any non-SAS application. This paper focuses on using SAS/EIS to front-end the data warehouse for the purpose of exploiting the resident data from a decision support environment. The application examined in this paper is the Virginia Higher Education Information System.

Introduction

To accomplish its objectives, this paper is divided into four parts. In the first section, the Introduction will focus on the capabilities of the SAS System to meet both the data warehouse needs and the interface needs of users. To do this, a basic warehouse model will be presented along with a general discussion of how to front-end that data warehouse.

In the second section, the actual building of a data warehouse will be examined, along with the specific problems which were overcome.

The third section will be an in-depth look at how this data warehouse was front-ended and, consequently, how the data was exploited to make timely decisions.

In the last section, conclusions will be stated.

The SAS System Data Warehouse Model

In most organizations today, there are two types of data. There is the operational data that corporations use in their production systems, such as accounting, payroll, and inventory. This is data that is critical to the functioning of the corporation. Read, write and update transactions are characteristic of this type of data.
Tier 1 - Accessing and Managing the Data

With the SAS System's unique data access strategy, operational data is accessible regardless of its form, through a single access method. This access method is based on engine technology that provides access to over 50 different relational, hierarchical and network database management systems, external file formats including VSAM, and data gateways and standard API's such as ODBC.

Only the data needed for decision support activities are carried over from the operational environment into the data warehouse. Since operational data are rarely in the form that is needed for decision support purposes, some data management logic must be applied. Data need to be summarized, cleaned, scrubbed, and/or reconciled.

The SAS System also has a rich set of data warehouse management and transformation tools that include:

- Robust 4GL
- Data reduction
- Summarization Procedures
- SQL
- Functions

Tier 2 - Organizing the Data

Once the data have been integrated into a consistent structure, they are ready to be placed into the data repository. Specific characteristics of SAS tables reinforce the SAS System's ability to serve as the data repository for the warehouse. These characteristics include relational features such as:

- Indexing
- Compression of data sets
- Where clause processing

Data can be surfaced to the SAS System in two ways:

- Extracts of data converted into SAS data sets
- Views to other data sources

The SAS System also provides tools for creating metadata, or information that describes how different variables in a warehouse relate to each other. These tools include:

- SAS macro language
- SQL Procedure for creation of data dictionary tables

SAS Institute is continuing to work on ways to improve the creation of the metadata and to make it an easier task in the future.

Tier 3 - Exploiting the Data

A data warehouse project is not complete without an effective set of decision support tools to report against the data. The SAS System provides decision support tools for:

- Developing object oriented applications
- Distributing application logic across multiple hardware environments
- Performing discovery and query functions on data
- Analyzing data multidimensionally
- Visualizing trends and patterns in data

Later in this paper, specific EIS objects that enable exploitation of the data warehouse will be examined.

Building the Warehouse

The State Council of Higher Education for Virginia (SCHEV) is the coordinating agency for over eighty public and private institutions. Along with Virginia's higher education coordinating board and the Office of Institutional Research at Virginia Tech, SCHEV has discussed the need for access to quality management information and enhanced reporting capabilities. Together this group developed an EIS in a client/server mode to exploit the resources in their data warehouse.

The data warehouse was built from operational data supplied by the members of SCHEV and are stored on an RS-6000. This data is in the form of raw ASCII files and SAS data sets. Extracts of personnel and finance data, as well as national higher education data, are made from this operational data and placed in the warehouse as SAS data sets.

As this data is placed in the warehouse repository, it is summarized to reduce its magnitude. Reduction in size was also accomplished by merging several data sets together. SAS software, in the form of PROC SUMMARY, PROC TABULATE, and DATA step code was used in the management and reduction of data. Summarization was done by institution, department, and a number of different ways so that years of data could fit into the warehouse. (SCHEV has used SAS in one form or another for over twenty years). The result is an extremely robust SAS data warehouse containing more than two decades of valuable Virginia and national educational data.

Because of its unique capabilities to provide a complete end-to-end solution for building a data warehouse, the selection of the SAS System was an easy choice for this project.
Exploiting the Warehouse

SAS/EIS was chosen as the toolset to provide intuitive, transparent access to the data warehouse. Many of the pre-written objects were used to accomplish the tasks involved in the successful exploitation of the warehouse's resources.

The Connectivity Problem

Using the SAS/EIS SIGNON object along with TCP/IP, remote access to the AIX environment was attained quite easily. The SIGNON object calls for the IP address to which you wish to connect, the complete path for the SAS/CONNECT® script file that you wish to use, the communication method to be used, and whether or not you were signing off. With %LET statements in the autoexec.sas file the full path to the TCPUNIX.SCR file was assigned to a macro variable, as was the needed IP address. This was much more preferable than typing C:\SAS\CONNECT\SASLINK\TCPUNIX.SCR in the appropriate SIGNON text entry field, or entering a cryptic IP address. Once a SIGNON object was created and tested, a copy was made with the SIGNOFF option checked and both were successfully tested.

While this approach was successful, it did have some drawbacks. Each user would have to know a common userid and password, and would have to realize that a remote machine was being accessed. This would mean a failure to provide transparent access to remote data.

The signon and signoff functionality was then moved to the EXECUTE object so that accessing a remote machine could be made transparent. This object executes any SAS program. The program below automatically accessed the warehouse on the remote machine.

```sas
/*
ipnode below is the IP address of the remote AIX host and can be assigned the needed value with a %LET statement in the autoexec.sas file. The COMAMID simply specifies TCP/IP as the communication method of choice.
*/
options remote=ipnode comamid=tcp;
/*
The TCPUNIX.SCR file supplied by SAS/CONNECT was copied to TCP2UNIX.SCR. This file was then edited so that the userid and password could be passed directly to the AIX machine without user input.
The LOGUNIX macro variable below resolves to the complete path to the TCP2UNIX.SCR file.
*/
filename rlink &logunix;
signon;
run;
```

With a .SAS file containing the above code, and a second .SAS file containing the same code but with a signoff statement instead of a signon statement, it was now possible to use two EXECUTE objects, one to make the connection, and one to log off. Furthermore, the connection and log off could now occur transparently.

Use of the SAS/EIS Script Object

The SAS/EIS SCRIPT object can be viewed as similar to .BAT files, or .CMD files in OS/2, or to EXEC files in the VM world. It gives the developer a chance to stack a series of 'events' or 'tasks' that the developer wishes to have executed, in the order that is desired. The SCRIPT object calls for a stack of SAS/EIS tasks which the developer has already constructed. In a simple example you can assume that you now have two EXECUTE objects to handle the signing on and signing off to the remote AIX host. The signing on is handled as described above, by an EXECUTE object called UNIXON EXECUTE. The signing off is handled by an EXECUTE object called UNIXOFF EXECUTE. A SAS/EIS SCRIPT object can be built with the first 'event' being the execution of UNIXON EXECUTE and the second 'event' being the execution of UNIXOFF EXECUTE. This SCRIPT object can be called LOGAIX SCRIPT.

When tested it makes a quick connection to the remote AIX platform, then instantly signs off. Coupled with other EIS objects in between the signing on and the signing off, this provides a very powerful client/server foundation. This set up the opportunity to access the superb SAS data warehouse mentioned above, and to keep all access and connections to the remote AIX machine transparent to the users. With transparent access available it was time to turn our attention toward making use of the available state-level data in innovative and analytical ways.

In one example of implementing the above methodology a third EXECUTE object was placed as 'event' number two, in between signing on and signing off. A small PROC TABULATE program was created and saved to a file called faculty.sas. The code is as follows:

```sas
rsubmit;
libname unixdata 'usr\faculty';
proc tabulate data = unixdata.fac95;
class rank gender;
var salary;
table gender*rank , salary*(n mean) / rts = 25;
```
where face = '3759' and year = '1995';
title 'Faculty Salary Statistics for 1995';
endsubmit;

This third EXECUTE object named FACULTY EXECUTE, was placed in the SCRIPT object after UNIXON EXECUTE and before UNIXOFF EXECUTE. This allowed for a transparent signon, a block of SAS code remotely submitted, and after the user was finished viewing the results, a transparent signoff occurs. This provided incredibly valuable transparent remote compute facilities but offered very little flexibility to the user who may wish to view a year other than 1995 or the faculty salary statistics of a peer institution. The need for user flexibility was accomplished with SAS/AF®, in particular the FRAME entry.

SAS/AF Objects in SAS/EIS

An application can be written in SAS/AF and then be executed using the AF object within SAS/EIS. Using the FRAME technology of SAS/AF provides the capability for user selectivity. Providing a radio box that offered the user a choice of which year to view, and another radio box which allowed any institution to be chosen, added great value and was accomplished in very little time. The block of tabulate code mentioned above was then altered to include macro variables which acquired their values from the user input. These quick successes were energizing and the project was becoming more fun and exciting by the day! The easy success adding our initial FRAME entries prompted increased use of list boxes, push buttons, icons, and radio boxes.

GRAPH Objects in SAS/EIS with 'Hot-Spotting'

There was quite a bit of interest in seeing student residency data graphically. PROC GMAP processing was handled by the local CPU. Input was received from users that while viewing GMAP output of the residency of students it would be wonderful to have instant access to the supporting dataset in the form of text reports and possibly other graphics displays. This led us to realize the great facility of HOT SPOTS. Several of the objects in SAS/EIS offer ‘HOT SPOTTING’ capability. The GRAPH object is one of those. We created a ‘HOT SPOT’ on the graph’s title. Users clicked anywhere on the title and a popup menu, using the object, offered users a stack of related choices. Some examples of related choices included intuitive access to both SAS/ASSIST® and SAS/INSIGHT®, direct access to jumping from SAS into Microsoft WORD® or EXCEL®, other graphics displays, and related reports. The access to SAS/ASSIST and SAS/INSIGHT® was accomplished using the SASCMND object in SAS/EIS. This object accepts a SAS command as simple as ASSIST, and when executed will access SAS/ASSIST and then return the user right back to the SAS/EIS application. The access to Word and EXCEL is accomplished using the HOST COMMAND or SYSCMD object. The SYSCMD object accepts the command needed to access another application outside of SAS. In the case of EXCEL for instance C:EXCELEXCEL.EXE. Again, when you then leave EXCEL you are returned to the SAS/EIS application. This gave users the comfort level of having their customary ‘toolset’ easily accessible.

When it was determined that there was a periphery of informational needs, or even simply that it was possible that a user may have informational needs based on the display he was viewing and desire direct access to more supporting or related data, objects which supported hotspots were chosen. We wanted users to get anywhere they desired analytically, from anywhere that they found themselves.

Drill-down Capability in SAS/EIS

An early effort in employing the SAS/EIS graphics object to examine the average salary of faculty by GENDER resulted in user input being received as to some value which could be added to the display. When users were viewing average faculty salary information by GENDER they often found an analytical need to also examine the average salary by the other very common class variables in the world of faculty data, RANK, TENURE, and ETHNICITY. This request resulted in a fantastic introduction to the ‘drill-down’ capability offered within SAS/EIS software. By registering the above four variables to the METABASE as drill variables, with GENDER being drill level 1, ETHNICITY drill level 2, TENURE drill level 3, and RANK drill level 4, (which took about a minute to do), the bar chart of average salary by faculty gender took on incredible new functionality. The display presented itself as before with average salary by gender but offered a yellow drill-down arrow in the top right corner of the
display. Clicking on this arrow instantly toggled the class variable to ETHNIC and presented the average salary by ETHNICITY. The arrow at this point offers both an up and down direction, i.e., a return to GENDER or another level down to an analysis of average salary by TENURE. All of this without ANY SAS coding. Other added default benefits within the graphics object are worth mentioning. At the click of a mouse the graph type can be changed to a pie chart. At the click of a mouse the statistical option can be toggled.

When we realized a need for users to rapidly traverse hierarchies of data, or examine statistics based on several class variables, our strategy was the employment of the drill-down capability. Analytically this often gave the users more than they asked for. Programmatically the facility provided multiple directions with no additional programmer cost in terms of coding.

The RANGE Object

The RANGE object exists to be coupled with several other objects. The RANGE object accepts a lower and upper numeric boundary and a color you wish associated with that range. For instance you can have 0 to 30 associated with red, 31 to 34 associated with yellow, 35 to 37 associated with green, 38 to 40 associated with yellow, and 41 above associated with red. This is an example of ‘traffic-lighting’ but there is no demand that only red, green, and yellow be used. All colors available to SAS can be employed within the RANGE object. One of the objects that the RANGE object is used for is the CRITICAL SUCCESS FACTOR object, also known as CSF. The CSF objects uses one numeric variable from a SAS dataset as input, and presents that numeric value in an ‘odometer-like’ graphic presentation.

An anecdotal example of how we made use of both the RANGE object and the CSF object may further clarify both. When developing the early prototype we had a chance to demonstrate the product to the Admissions Office staff and administration. One of their ‘driver’ statistics is enrollment rate which is the number of applicants accepting an enrollment offer divided by the number of applicants that were offered the chance to attend. This is a key piece of information for a number of reasons. One important reason is that an institution has a finite number of dormitory rooms, and a finite number of beds in those rooms. They also typically have some capital debt that housing revenue will help to retire. Thus, in bringing in a first-time freshmen class there is an optimal range that a school may shoot for. We theorized that if 31 to 34 percent of those applicants who were accepted were agreeing to attend, it would be borderline on the low side. Below 31 percent could be a serious problem. The range from 35 through 37 would be optimal. The range 38 to 40 percent would be again borderline problem on the high side, and above 40 percent could be a serious situation. We had the necessary statistic calculated during an overnight processing run which populated a local SAS data warehouse with needed reports and datasets. (as an aside both the rapidity of development and the speed of altering or making changes to all of the SAS/EIS object truly supports the terms rapid application development). Admissions employees could then see where the needle was on each day and make the necessary decisions.

The RANGE object was used in conjunction with two other objects as well. The REPORT object accepts the path down to a text file as its only needed information. You reference a text file and that text file is then displayed when chosen from a menu. The VIEWFILE object is very similar to the REPORT object. The REPORT object accepts the RANGE object as value-added. A REPORT object can make use of a RANGE object to visually enhance a column of numbers. In displaying a numeric report, using the RANGE object can visually do wonders for users. Each column of a given REPORT can be coupled with a different RANGE object. We had a faculty tabulates display that had the N of instructional faculty at each in-state Virginia college and university, along with columns for the percent female, the percent Black, and the percent PhD. RANGE objects were created and customized to each of those columns so that users could easily, at a glance, have ‘traffic-lighting’ communicate where our university stood relative to our in-state peers across several key measurement areas. The VIEWFILE object does not support use of the RANGE object.

Menu Objects in SAS/EIS

All of the above is only as valuable as the intuitive ability to find the data or display that you need. Menu objects provide the ability to group related choices and provide easily understood pathways into the capabilities of the system.

GRAPHICS MENU BUILDER objects gave us the ability to use SAS/GRAHP output within icons. The region attributes of each icon provide the ability to title and box the icon. We found that the GRAPHICS MENU could become too visually ‘busy’ and evolved a preference for a different menu object called the BLOCK MENU.

BLOCK MENU objects give a very professional appearance. Users provided feedback to us that they liked the look and feel much better. SAS supplies an icon library that covered our needs quite well. We settled on using the icons which included numbers, with the idea that responding to users’ requests with an instruction such as ‘click on FACULTY on the opening menu, then click on choice 2, then choice 3’ would be very understandable.

LISTMENU objects were used extensively. The LISTMENU provides the developer the opportunity to
stack a list of choices. That stack of choices can include any other SAS/EIS objects. For instance, within a given LISTMENU you can provide access to ASSIST as the first choice, access to a subservient BLOCK MENU as the second choice, access a SCRIPT object which remotely submits a block of code as choice number 3, an 'out' to WORD as choice number 4, and a popup FRAME entry for choice number 5. Providing access to a LISTMENU from on-display 'hot-spots' proved invaluable.

Conclusion

The SAS/EIS tools set proved to be superb. There were some minor conceptual hurdles due in a large part to the fact that this project was an introduction to the OOP world. Overall the product thrilled the developers and resulted in a great deal of positive feedback from users. Some advice for those who may embark on similar endeavors follows.

Before beginning an endeavor which will attempt to cut across information domains in any organization it is essential that you have an executive sponsor or executive champion. Pick up any EIS/DSS textbook and it will say the same. The reasons are quite evident. You WILL have domain struggles. It is not just a possibility. Instead, count on it occurring and plan accordingly as to how you will deal with these domain problems. You had better have someone to call upon to expeditiously resolve them.

If you are employed in a university setting enlist on-campus EIS/DSS expertise from the Management Science Department or the MBA program. We had regular consultations with three Management Science faculty members who were invaluable. They added great value via their input. We also enlisted an Associate Registrar whose expertise was Cartography. After finding flaws with our choice of colors in early GMAP student residency displays she was drafted as our mapping expert. In short involve users and you will see positive 'buy-in' and reap large rewards from their inputs and expertise.

SAS assistance is available from the authors via the email addresses which are included. Even more SAS expertise is available via the Internet SAS discussion group known as SAS-L. See the authors or ask around at SUGI if an introduction to this incredibly valuable resource is desired. I can not overstate the value of the SAS-L group in solving or helping to solve both conceptual and syntactical problems.

The technical support staff at SAS is unparalleled in their responsiveness and abilities. Call them when the need arises.

As far as manuals, I have always favored the course notes which can be purchased through SAS. The course notes are the same as what you receive if you were to take the SAS-taught EIS course. They typically are a gentle and methodical approach to learning a new SAS component such as SAS/EIS.

Acknowledgements

The authors greatly appreciate all of the help and input from the SAS Institute, Inc. Data Warehouse Project Team. Special thanks to Becky Smith-Brown.

The authors can be contacted at:

**Jim Alessio**
Executive Director of Enrollment Services
Virginia Commonwealth University
Richmond, VA 23284
Phone (804) 828-9944
FAX (804) 828-1464
Email: jaillesio@ues1.ues.vcu.edu

**Dennis Catley**
Systems Analyst
General Insectables & Vaccines, Inc.
Radford, VA

**Ben Cochran**
Product Marketing Manager for SAS/EIS
SAS Institute Inc.
SAS Campus Drive
Cary, NC 27513
Phone (919) 677-8000
FAX (919) 677-4444
Email sasbtc@unx.sas.com

References


SAS, SAS/ASSIST, SAS/AF, SAS/CONNECT, SAS/EIS, SAS/INSIGHT are registered trademarks of the SAS Institute Inc. in the USA and other countries. ® indicates USA registration.