ABSTRACT

SAS/SQL-DS software is a powerful new tool for end-user computing in the information center environment. A wide range of possibilities is now available to integrate SQL/DS data into existing SAS applications. The SAS/SQL-DS interface speeds new application development and enables you to take full advantage of the powerful report writing and graphics capabilities of SAS software. This paper discusses the use of this new product, and an example is presented.

INTRODUCTION

Over the past ten years, the information center has evolved into a vital component of the data processing environment. One of the main goals of the information center has been to develop applications quickly that are easily maintained by nontechnical staff members. SAS software has played a key role in the evolution of the information center, and SAS software now provides links to other software tools gaining popularity in this environment. The new SAS/SQL-DS interface to the popular IBM® SQL/DS data base provides such a link.

THE INFORMATION CENTER ENVIRONMENT

The information center was born from the need to eliminate the large backlog of programming requests common to most data processing departments. It was thought that this backlog could be reduced significantly if the users of the data could access it themselves. For this reason, many corporations formed information centers and provided them with technical staff, hardware, and the software tools necessary for users to access data for standard reporting and general decision making.

As time passed, the emphasis within the information center shifted. At the start, educating users in the basics of data processing was stressed. In time, the level of user sophistication increased dramatically. Now, one of the newest responsibilities of the mature information center is to support user-developed applications.

Existing alternatives to developing applications not using the SAS/SQL-DS interface have some definite drawbacks. Such applications are written in a high-level source language (such as PL/I, COBOL, or assembly language) by highly skilled programmers in the company's data processing departments. Any modification or support of these applications relies on these same computer professionals and throws the application back into the thick of the existing computer programming backlog. Enabling the users to code and support their own applications not only minimizes the time that the data processing department needs to budget for application development, but it also cuts back on the invisible backlog of programming requests. This invisible backlog consists of programming requests that the MIS department never even sees because users anticipate delay.

The goal of the information center is to provide corporate data access to the widest variety of people. The SAS/SQL-DS Interface, when used together with the entire product line offered by SAS Institute, provides a totally integrated environment. In this environment, users and support staff can develop and modify applications with a minimum of training and support.

APPLICATION DEVELOPMENT USING SAS SOFTWARE

The new SAS/SQL-DS interface allows for the development of easily maintained applications that use the full capability of SAS software without the high cost and time delays associated with applications developed in a more traditional way. All of SAS software's reporting, graphic, and statistical capabilities can be integrated easily into an application.

SAS/AF software provides the capability to design customized menus as a front end to the application. This is the first step in developing an application that is not only powerful and flexible but useful to any person familiar with SAS software. SAS/AF software enables the user to execute SAS code in noninteractive background job steps based upon menu selections. Any required functions can be built into the SAS/AF system and then modified easily using PROC BUILD.

AN EXAMPLE: HOW MUCH DOUGH IS THERE IN COOKIES?

The following example demonstrates how SAS/GRAPH® SAS/AF, and SAS/SQL-DS software can be combined to provide a simple information center application.

The Creative Cookie Bakers Company is in the retail cookie business. Each of their franchised locations features rich, chewy, chocolate chip cookies. There are several different types of cookies. Some varieties are old standards; others are sold for a trial period and, if profitable enough, are added as a standard selection. Like any retail business, Creative must keep track of inventory and production, income and expenses, and employee information. They chose to use IBM's SQL/DS data base system to manage this information because they want the power of a relational DBMS.

The corporate data base system for Creative Cookie Bakers consists of several SQL/DS tables. The company has various applications written to maintain these tables. In addition, during a period of rapid growth several years ago, the company established an information center (IC). The IC was created to facilitate queries to the data base not covered by existing applications. Many of these queries are for summary or statistical reports of data contained in the SQL/DS tables.

The IC staff uses SAS software to build these kinds of applications quickly and to provide a user-friendly menu-driven interface at the same time. One such application is illustrated here. The illustration does not explain much about using either SAS/AF or SAS/SQL-DS software. Instead, this example highlights how the functions of SAS/AF and SAS/SQL-DS software can be combined to create a self-contained application.
The User's View

The question is, "Which type of cookie has generated the largest profit?" Screen 1 displays the Primary Query Menu in the application. This menu shows that queries can be made into several different aspects of the company's operation.

Screen 1 Primary Query Menu for the Information Center Application

To enter a query about finances, enter PF6 key to select option 6. The next display (Screen 2) enables the user to specify the report to be displayed by choosing data on either Sales or Profits, summarized by either Franchise Location or Cookie Type. Place an X in the field next to the desired option. Screen 2 shows that the user requested a report on profits, summarized by cookie type.

Screen 2 Options Available for Summary Financial Reports

Leave Screen 2 by pressing the END (PF3) key. Screen 3 displays the requested summary. The results show that Original Recipe cookies are the most profitable. The least profitable are the Macadamia Chocolate Chip cookies.

Screen 3 The Summary of the Requested Data

Leave Screen 3 by pressing the END key. Another menu (Screen 4) is used to enable the user to obtain a graphic display of the report.

Screen 4 Graphics Can Be Viewed After The Summary

Choose option 1 to display a pie chart (Screen 5) that clearly illustrates the high profitability of Original Recipe cookies and the marginal profits of the Macadamia Chocolate Chip cookies. The next menu (not shown) enables the user to obtain a hardcopy of the pie chart.
Having examined the application from a user's point of view, you can take a more detailed look at the application's construction. The application uses two of the tables in the corporate database: The PRODCTN table contains records of the cookies baked at each franchise location. (The actual unit is trays of thirty-six cookies; this is simpler than using single cookies.) The CCOST table contains a record for each type of cookie, indicating the cost of producing a tray of cookies. (This cost estimate is computed using data from other tables regarding the cost of ingredients, labor, and so on.) The columns for each table are described in Table 1. The application described here uses SAS/AF, SAS/SQL-DS, and SAS/GRAPH software to allow queries about information drawn from both of these tables.

### Table 1: Tables and Columns Used

<table>
<thead>
<tr>
<th>Table</th>
<th>Column Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>PRODCTN</td>
<td>FRANLOC</td>
<td>Franchise location</td>
</tr>
<tr>
<td>PRODCTN</td>
<td>DATE</td>
<td>Date of transaction</td>
</tr>
<tr>
<td>PRODCTN</td>
<td>CTYPE</td>
<td>Cookie type</td>
</tr>
<tr>
<td>PRODCTN</td>
<td>NTRAYS</td>
<td>Number of trays sold</td>
</tr>
<tr>
<td>PRODCTN</td>
<td>PRICEPER</td>
<td>Price per tray for this type</td>
</tr>
<tr>
<td>PRODCTN</td>
<td>MGRNAME</td>
<td>Shift manager</td>
</tr>
<tr>
<td>CCOST</td>
<td>CTYPE</td>
<td>Cookie type</td>
</tr>
<tr>
<td>CCOST</td>
<td>CCOST</td>
<td>Cost per tray for this type</td>
</tr>
</tbody>
</table>

The SQLEXT procedure provides a way to extract data from one or more SOL/DS tables and to place them in a SAS data set. The major part of this process is identifying the tables and the columns to be used in the extraction. In the context of a simple query, the table names are known, and only a small number of columns are of interest. In this case, SAS/AF MENU and PROGRAM screens are used to prompt and validate the columns to be extracted. Then, SAS macros are used to generate the SAS code needed to extract, summarize, and report the data. The displayed portion of the PROGRAM screen used for this application is shown in Screen 6. This screen contains fields that are associated with SAS macro variables. A macro that uses these variables to build an invocation of the SQLEXT procedure is displayed in Figure 1.

Other macros are used to sort and analyze the data set created by PROC SQLEXT. Once the requested report has been generated, other SAS/AF screens and SAS macros are used to generate graphic output using SAS/GRAPH software.

### Figure 1: SAS Macro That Uses Input from Screen 6

```sas
/* This macro used to validate the fields on the screen.*/
macro chekem;
/* Get sales totals.*/
if field1="then
   ntrays*priceper; /* Get sales totals.*/
else
   ntrays*(priceper-ccost); /* Compute profits.*/
from prodctn p, ccost c
where p.ctpe=c.ctpe;
fmt 2='Scfmt.' 3='6.2' '+='6.2' 5='6.2' 6='6.2';
if field1="then
   rename 6=sales;
else
   rename 6=profits;
list all;
run;
macro done;
```

**SOME ADVANTAGES OF USING SAS SOFTWARE**

Applications that use SAS/SQL-DS and other SAS software have several advantages over applications that interface to the DBMS directly. One of these advantages is the ability to associate a format with a variable. Often, the values stored in a database are codes or abbreviations, such as C1 and C2 for Cookie Type in the above example. These shorter values use less space but make it more difficult for users to understand the values. However, if the data are extracted to a SAS data set, the variables can use formats that provide a more readable representation of the value. Furthermore, if the data are extracted to a SAS data set, the variables can use formats that provide a more readable representation of the value. For instance, C1 becomes Original Recipe. There are a variety of formats supplied with the SAS System, and, in addition, you can define your own formats.

Another advantage (mentioned earlier) is that the applications written using SAS software require much less technical expertise to develop and maintain than those in traditional application languages. Information centers are finding that their users want not only training in how to use computing facilities but also want to be able to develop their own applications. SAS software makes this easier.
only training in how to use computing facilities but also want to be able to develop their own applications. SAS software makes this easier.

SUMMARY

As you can see, the SAS/SQL-DS interface used with other SAS software provides a powerful new tool for end-user computing. This tool is totally adaptable to a wide variety of data processing situations encountered in the typical information center environment. The user of corporate data can use SAS software to develop applications that are easy to support and maintain. These features do much to eliminate the invisible backlog of data processing requests experienced by today's data processing departments.

REFERENCE


SAS, SAS/AF, and SAS/GRAPH are registered trademarks of SAS Institute Inc., Cary, NC.

SAS/SQL-DS is a trademark of SAS Institute Inc., Cary, NC.

IBM is a registered trademark of International Business Machines Corporation.