INTRODUCTION

During the past decade, the SAS(r) System has played an important role in shifting many programming tasks from the data processing professional to the end user. In general, this shift has produced positive results; users have become more involved in specifying and solving their own problems. However, oftentimes, the user's programming function is not supervised - traditional data processing standards are easily ignored. The user department emphasis is on getting the job done and not on producing efficient and well documented programs. This unstructured programming environment eventually leads to a collection of SAS programs and data libraries that is nearly impossible to maintain.

A well designed information system can provide a more controlled environment without limiting user involvement. In fact, more users become involved since less data processing skills are required to access information. In addition, since a system environment enforces data convention standards, users that are familiar with the SAS programming language can make more assumptions when writing application programs.

This paper explains how the tools provided by the SAS System can be used to build an information system. The importance of using information in SAS data sets and format libraries to control and document the system is emphasized. Examples from THECIS [1,2], the clinical information system in use at Lederle Laboratories, are used to illustrate a SAS based information system.

WHEN IS CONTROL NEEDED?

The SAS System has made it relatively easy for professionals without formal data processing training to organize, analyze, and report their data. For single applications, that is, one set of data, one user, and only a few reports, there is little need to implement an information system. For these users, an information center that can provide SAS consulting and training is adequate. However, for applications with large data sets, with many users accessing the same data, and similar reporting needs for different sets of data, the controlled environment provided by an information system is required.

WHY CONTROL?

Improve user efficiency

Improving user efficiency is the primary consideration when introducing an information system environment. Users who once spent a large part of their time organizing data, can more readily get to the information they need. A library of data management tools is available to simplify the the complex data handling tasks and, for many situations, the required report is already available in a library of standard programs. This program library replaces the "copy and modify" technique that is commonly employed when a similar report is required on a different set of data. Finally, on-line access to the data often leads to a simpler solution to a complex query.

Reduce computing cost

Computing costs are also reduced. Since the chance of "getting it right" the first time is quite high, there is much less recycling than is common in the non-controlled environment. Also, less processing is required, since the data structure is more consistent with the reporting needs. Finally, more efficient programming techniques further reduce computing costs.

Improve quality

With less time required to complete the routine tasks, more time is available for data exploration and for improving data presentation. Questions about the data that were once nearly impossible to address are more easily handled.

Minimize risk of serious errors

Finding a "program bug" after a final report has issued is a fear that all users live with. Although never eliminated, the chance of a serious programming error is reduced once coding conventions are introduced and proper programming documentation and validation standards are followed.
A discussion on an information system can be divided into five topics:

1. The database
2. Update methodology
3. Program library
4. Support
5. Security

The database

The center of an information system is the database. Careful planning is required to ensure that the database structure and content meet current user requirements. The database design must be flexible enough to meet future needs.

The SAS System is not a database management system. However, the tools available in the SAS System, data sets, format libraries, and macro libraries, can be used together to construct a database.

SAS data sets are used to store five types of information:

1. Raw data
2. Data attributes
3. Information about the raw data
4. Data status information
5. System information

Raw data

The first type of data set, one that contains raw data, is the type with which most SAS users are familiar. Several concepts must be employed when designing these data sets:

The key variable concept: Each data set has a set of key variables that uniquely identify each observation. These variables define how the data set is sorted.

The level concept: Data are stored only once and at the appropriate level, where level is defined by the last key. For example, in THECIS, the dataset PATIENT is a patient level data set that contains information that is captured once per treatment period (eg study drug and first dosing date) is stored in the data set PTPER with keys P T PT PER. This data set would not contain the date of birth variable.

Data status variables: Each observation in every data set contains status variables that define the data source, date of last modification, and the observation status. These variables are useful for data subsetting as well as explaining different contents of the same standard report.

The unique name concept: With the exception of key variables and system status variables, data elements are assigned unique names across data sets. This convention simplifies the maintenance of a data dictionary and prevents errors when data sets are merged together.

Horizontal and vertical structure: When several data elements are closely related and information on these elements is available for all values of a set of key variables, then a horizontal structure is appropriate. In this structure type, data elements are stored with different (NAME,NAMES) names on the same observation. As an example, in THECIS adverse symptoms are stored in a data set with one observation for each occurrence of an adverse symptom. Related variables include duration, severity, drug relationship, action taken, and patient outcome.

In the second structure type, vertical, data elements are stored on separate observations. The name of the data element is stored as the value of a variable called NAME, and its value, or response, is stored as the variable VALUE. This method of data storage has several advantages:

1. New data elements can be added to the database without modifying the data set structure.
2. Each unique set of key variables can have a different number of data elements (NAMES), with each element having its own set of status variables.
3. A separate analysis for each NAME (BY NAME) requires no data transformation.

In THECIS, laboratory, efficacy, and vital sign data are stored in this vertical structure. It should be noted that data transformations can be used to swap between the two structure types.

Data attributes: The second type of data sets are used for storing the attributes of each data set in the database. Four data sets are needed:

1. Data set names - one observation for each data set with variables that describe the data set, identify the key variables, and the order of the data set (used
when listing the database specifications).

2. Variable names - one observation for each unique variable. This data set contains the variable's attributes: type (character or numeric), length, label, and format. When appropriate, the variable's units are also recorded.

3. Variable into data set mapping - one observation for each data set and variable within the data set. This data set controls which variables are associated with each data set as well as their positioning within the data set.

4. Names - one observation for each possible value of the 'NAME' variable in the vertically structured data sets. The following information is recorded: name description or label, data set that contains the name, the order for positioning the name on reports, format, units, and improvement direction. The improvement direction variable is used for evaluating changes in a name's response. For example, a value of 'I' indicates that an increase in the name's response would be considered an improvement.

Information data sets: Information data sets are the third type of data sets that are used in a SAS database. These data sets contain information that help describe the content of the raw data sets. The number, structure, and content of these data sets depend upon the database application. In addition to describing the contents of the database, information data sets are used to query the database. In THECIS, information data sets contain data on study design, dosage schedules, and visit schedules.

Status data sets: Status data sets contain 'how many' and 'when' information. Like information data sets, the types of status data sets depend upon the database application. THECIS examples include a contents data set that contains the number of observations, the number patients, and the last update date for each data set and study in the database. A visit inventory data set contains visit dates for each patient in the database.

System data sets: System data sets are created for the purpose of controlling the system. They are not normally directly accessed by the end user. A data set that contains the observation numbers for a set of key variables is an example of a system data set. It would be used by utility programs to more efficiently select a subset of data from a raw data set.

Format Libraries

The second component in a SAS database is the format library. A format library is a permanent collection of formats that is available to all users of the database. In their most common use, formats are used to decode data values. For example, the format SEV could be used to map the value '1' into 'MILD', the value '2' into 'MODERATE', and the value '3' into 'SEVERE'. A useful extension to the formats for data values is the 'C' format, which maps a code into a code plus a label. For example, the 'C' format for SEV is called SEVC and maps the values '1', '2', and '3' into '1 MILD', '2 MODERATE', '3 SEVERE', respectively. In THECIS, the 'C' formats are stored with the data. Users can easily see how to construct queries after browsing the data.

Although decoding data is the most common use of SAS format libraries, two other types of formats are equally important, utility and system. Utility formats contain information about a data value, not the data value's decode. In THECIS, the formats 'VISITL' and 'VISITH' map the value of the variable 'VISIT' into a lower and upper allowable day range, respectively. These limits can be obtained in any data step, without first sorting the data set by 'VISIT'.

System formats, like system data sets, are used to control the system. They are not used directly by the end user. A format that maps variable names and values of the NAME variable into the appropriate data set is a system format. It would be used by the system to assist the user in building a query.

Macro Libraries

The final component of a SAS database is a library of data management macros. These macros may derive additional variables, restructure a data set, combine two or more data sets, or select one or more observations from a data set. This library of data management 'tools', together with the SAS language, form a very powerful method to access information in the database.
Update Methodology

The next topic in a discussion of an information system is the methodology used to update the database. This includes adding new data as well as changing existing data. The update process may also involve extensive data management if the raw data are captured in a different format than they are stored in the database, or if complicated derivations are required. This is the area of the information system where control is most needed. Most of the resources that are required in setting up an information system involve developing programs that are used in the update process. These include batch programs and programs that control interactive menu-driven sessions. There is no standard set of programs that will work in all applications. Only a set of guidelines can be given.

1. No errors, either by the user or in the raw data, may lead to the violation of the database specifications.
2. The update process does not interfere with other database users. An overnight batch routine is recommended.
3. The update process is specific. That is, only data sets with changes or additions are altered. The value of the source status variable can be used to determine which data to replace.
4. The updating of the status and system data sets is an integral part of the update process.
5. Previous versions of changed observations are stored in an audit trail data set. This permits the system to recover from incorrect edits.
6. Data error checks are part of the update process. Values of data elements that are not possible should be handled as missing data.

Program Library

The program library provides the means for getting information out of the database. Programs list and summarize the raw data and provide information about the database contents and status. These programs are menu driven so that the necessary options can be explained and the user input to the options can be controlled.

A properly written application program takes full advantage of the data sets, format libraries, and macro libraries that make up the database. The format of the output produced when a program is executed depends upon the information in the database as well as the options specified by the user.

Each program in the library must be properly documented and validated. Changes to a production program must be under the same control as the database update procedure.

Support and Documentation

In order to keep an information system running smoothly, support and documentation is necessary. Support includes user training and both system and application programming support. New users must be trained and users familiar with the system need to have questions answered promptly and correctly. System programming support is required to maintain, enhance, and expand the system. Application programmers are required to maintain and expand the macro and program libraries and to help answer the difficult ad hoc queries.

Documentation includes information on how the system works as well as how to work the system. System documentation is needed by the system programmers that maintain the system. User documentation includes training manuals, data convention standards, and application programming coding standards.

Security

All components of the information system must be protected against accidental or intentional alteration or deletion. This includes SAS data sets, SAS catalogs, format libraries, and code libraries.

SUMMARY

We have shown that an information system environment can be used to help control the SAS System and its users.

To create this environment requires an initial commitment of resources to set up a system as well as a continuing commitment to maintenance and training.

It must be emphasized that controlling the user programming environment places no restrictions on user involvement or creativity. Instead, the user has more
time to further explore the data. Those users who development new application programs and data management tools have a system in place for easily sharing their contributions with all users.

SAS is the registered trademarks of SAS Institute Inc., Cary, NC, USA.

References
