Abstract

Release 6.06 of the SAS System represents a new software design for the decade of the 1990's. A number of major design goals and software re-engineering in Release 6.06 necessitate changes to existing education offerings on the SAS System as well as suggest new offerings for SAS training programs.

Introduction

Release 6.06 represents the first Version 6 implementation of the SAS System in the mainframe, minicomputer, and OS/2 environments. It represents more than 250 man-years in design and 1000 man-years in programming the approximately 3 million lines of source code.

Release 6.06 provides the ability to support any user, any application, on virtually any machine platform. Implications for training include supporting more users on more hardware platforms and operating systems, and providing them the SAS skills to implement more strategic applications.

Design Goals

Four major goals inherent in the design of Release 6.06 include

- MultiVendor Architecture
- Multiple Engine Architecture
- Increased Functionality
- User Interfaces

These design goals manifest themselves in the use of the SAS System for data access, data management, data analysis, and data presentation. Likewise, these four design goals need to manifest themselves in any education course offerings on the SAS System.

MultiVendor Architecture

MultiVendor Architecture represents SAS Institute's strategy for implementing a single, consistent image of the SAS System on any hardware and operating system.

Education is facilitated because the SAS System has the same look and function as users work in more than one environment. Training skills acquired on the SAS System in one environment should essentially be transferable to other environments. The only exception is host system issues such as file naming conventions, operating systems commands, and so on.

In addition the MultiVendor Architecture design goal supports native windowing so that users who work in a single environment can have the SAS System look and function as other applications they are accustomed to in that same environment. Thus, familiarity with other applications can minimize making the transition to the SAS System and vice versa.

Multiple Engine Architecture

The design goal Multiple Engine Architecture provides the ability to access a variety of different types of files as if they are SAS data sets.

The fact that there now exists different implementations of SAS data sets represents a critical concept fundamental in any SAS courses.

Through internal sets of instructions called engines, the SAS System is able to process data in interface products and other software vendor's files as if they are SAS data sets. Native engines access the different forms of SAS data sets created and maintained by the SAS System. Interface engines support access to different interface products and other software files.

Interface engines in the first Release 6.06 implementation include DB2 and System 2000 in the MVS environment; SQL/DS and System 2000 in the CMS environment; ORACLE and Rdb/VMS in the VMS environment; DBASEII, DBASEIII, DBASEIV, Lotus 1-2-3, and OS/2 Data Base Manager in the OS/2 environment; and SPSS, O SIRIS, and BMP across various operating systems.

Native engines and the SPSS, BMP, and O SIRIS interface engines are assigned when an association is made between a libref and a SAS data library.

The other interface engines are assigned in SAS data views. A SAS data view does not contain data values but rather instructions on how to construct a logical SAS data set from one or more files.

SAS data views can be created by SAS/ACCESS software which creates logical SAS data sets of interface products and other software vendor's files. To what extent SAS/ACCESS software needs to be taught is a function of what role the data base administrator will play in controlling user access to interface product files.

The data base administrator may wish to create all SAS data views for each application or relinquish some control to users to create SAS data views.

SAS data views can also be created by PROC SQL which creates logical SAS data sets of other SAS data sets. PROC SQL implements the industry standard structured query language within the SAS System.

The language represents a very sophisticated language for querying one or more SAS data sets. The retrieved information including the manipulation of existing variables, the creation of new variables, and the manipulation of observations can be placed into a report or a SAS data file.
The language can also be used to update SAS data sets, join two or more SAS data sets, sort SAS data sets, and summarize information in SAS data sets. PROC SQL combines the traditional functionality of a number of DATA and PROC steps into a single step.

PROC SQL usage will be both as an ad-hoc report generator as well as a data base management tool. Both uses will need to be taught.

Additional Functionality

Release 6.06 represents additional functionality across the SAS System. Existing training conducted in any of the following areas would need to include these topics.

<table>
<thead>
<tr>
<th>AREA</th>
<th>TOPIC</th>
</tr>
</thead>
<tbody>
<tr>
<td>Introduction to the SAS System</td>
<td>Base, SAS/GRAPH, SAS/FSP, SAS/ACCESS, SAS/ASSIST</td>
</tr>
<tr>
<td>Report Writing</td>
<td>PROC REPORT</td>
</tr>
<tr>
<td>Data Management</td>
<td>PROC SQL, SAS/ACCESS, SAS Data Files, SAS Data Views</td>
</tr>
<tr>
<td>Efficiencies</td>
<td>Compression, Indexing, Storing Compiled Code, WHERE Processing</td>
</tr>
<tr>
<td>Application</td>
<td>Screen Control Language</td>
</tr>
<tr>
<td>Development</td>
<td>Screen Control Language</td>
</tr>
<tr>
<td>Graphics</td>
<td>SAS/GRAPH System Options, Customize Device Drivers, Computer Graphics Metafile</td>
</tr>
<tr>
<td>Data Entry, Editing, Retrieval</td>
<td>PROC FSVIEW</td>
</tr>
<tr>
<td>Training</td>
<td>Training is needed for existing Version 5 users to make the transition to Release 6.06. Training in this area needs to address the new functionality migration issues, and conversion methodology.</td>
</tr>
</tbody>
</table>
| User Interfaces    | For the experienced users, the SAS Display Manager facility has been enhanced. The SAS Display Manager System is a highly interactive programming environment that enables one to edit, test, and debug programs and applications before moving them into production.

Online help is available from each window. The help is layered and is as detailed as statement syntax and function.

SAS/ASSIST is an interactive, full-function interface that supports common tasks including data access, data management, data analysis, and data presentation. It allows those users who have no SAS expertise to immediately become productive with the SAS System. As user prompts are completed, SAS programs are constructed to perform the task. The source code programs are fully commented and can be stored for subsequent execution.

From the Primary Menu a tutorial can be taken which discusses how to use and navigate the SAS/ASSIST screens. SAS/ASSIST software was completely written using SAS/AF software. You may use the SAS/ASSIST screens as models for building your own applications.

SAS/ASSIST software also contains an Executive Information System (EIS) builder.

Each of these enhancements and the windowing implementation of many SAS procedures provides for a more user-friendly windowing environment. The windowing environment in the SAS System is completely compatible with the window architecture and standards for the hardware platforms the SAS System runs on.

Introduction to the SAS System

The SAS System represents an integrated system that can handle many different strategic applications. A user's first exposure to the SAS System must include a broad orientation to its capabilities. Too narrow a focus will severely restrict one's understanding of its applications. Minimal exposure should at least technically address certain components in Base, SAS/GRAPH, SAS/TSP, SAS/ACCESS, and SAS/ASSIST software with exposure to the other software products.

Report Writing

Report writing represents a frequent application. Two new procedures that need to be discussed in any related courses include PROC SQL and PROC REPORT.

PROC SQL implements structured query language within the SAS System. Structured query language can be used to produce very sophisticated reports from one or more SAS data sets including selectively choosing certain variables and observations, computing new variables, sorting, performing BY-group processing, formatting the report, and summarization.

PROC REPORT constructs reports. The report definition template can be painted in a full-screen mode. The template definition can be stored for subsequent use when producing the production report. Although the data
which comprises the report is changing, the template definition to reconstruct the report structure remains the same.

Data Management

SAS/ACCESS software, using multiple engine architecture, permits different implementations of SAS data sets. Other interface products and software vendor's files can logically be modeled as SAS data sets. This represents a very efficient technique since the data resides in its native file structure and no intermediate SAS data file is constructed.

PROC SQL, the procedure which implements structured query language, provides data base management support too. Two or more SAS data set implementations can be joined, indexed, updated, sorted, and manipulated to construct new SAS data files and SAS data views.

Efficiencies

Release 6.06 represents many performing tuning enhancements. These enhancements typically affect disk usage, I/O count, CPU time, and elapsed time.

Compression removes redundancy from SAS data files thus minimizing the amount of disk space needed to store the SAS data file. The SAS data file is stored as variable length records.

Indexing variables in SAS data files provides fast retrieval to subsets of observations in a SAS data file. In addition indexing eliminates the need for PROC SORT steps when performing BY-group processing since the values returned via the index are in collating sequence order.

DATA step source code can be stored in compiled form. CPU resources are saved since the source code does not have to be recompiled each time it is executed. In addition the compiled code affords appropriate protection and security in a production environment.

WHERE processing provides the ability to subset SAS data sets within the PROC step rather than using the DATA step to create a subsetted data set.

Application Development

Screen control language represents a new application development language that controls the interaction between the user and PROGRAM entries in SAS/AF applications.

The language is typically used to validate user entries, create and manipulate variables, create window variables, access SAS data sets and external files, define error conditions and customize error handling, control branching and opening other windows, and control the application environment.

The language is DATA step like and contains most of the statements and functions in the DATA step. In addition it also has a rich library of additional statements and functions.

A thorough understanding of screen control language is critical to any application development.

Graphics

Specific enhancements to SAS/GRAPH software include additional system options, enhancements to global statements, the ability to customize device drivers, and support for sharing graphics output among different vendor's software using computer graphics metafiles.

Data Entry, Editing, and Retrieval

The PROC FSVIEW procedure is new and provides the ability to browse/edit a SAS data set as a table of rows and columns. During the FSVIEW session, new window variables can be computed and their formulas stored for subsequent use, the data set can be sorted, and new SAS data files created.

Screen control language represents a new application development language that sits between the user and PROC FSEDIT, PROC FSBROWSE, and PROC FSVIEW applications.

The language is typically used to validate user entries, create and manipulate variables, create window variables, access SAS data sets and external files, define error conditions and customize error handling, control branching and opening other windows, and control the application environment.

The language is DATA step like. It contains not only most of the statements and functions in the DATA step but also has a rich library of additional statements and functions.

Statistics/Decision Support

SAS/STAT, SAS/QC, SAS/ETS, and SAS/OR software contain enhancements to existing procedures and new procedures. These enhancements will need to be communicated to those who already have the expertise to use these tools.

Summary

Release 6.06 represents a major re-engineering of the SAS System. The increased windowing, user interfaces, additional functionality, and multiple engine architecture need to manifest themselves in any SAS courses taught. The multiple vendor architecture design goal facilitates learning the SAS System and making the transition to the SAS System on other hardware platforms and operations systems.