The end-user system creates and submits an ad hoc batch job coded in SAS® software and the necessary operating system (JCL) statements. Run-time parameters are input by the user, who never sees any code, but instead works with menus with PF key selections, input screens, and when needed, help screens.

The system, which was created using SAS/AF® and SAS/FSP®, was written for an OS/MVS mainframe using TSO.

The amount of time needed to produce results is decreased by putting the computing power directly into the end users' hands, without sacrificing information integrity.

This system and others like it will help to lessen the burden on the limited number of Info Center support personnel who are producing many ad hoc reports. Once more 'canned' systems are in place, the level of support to other areas can increase.

Management's concerns are very important, but above all, the user must be thought of as the customer: Does the system meet the customers' needs? Is the system user-friendly, and is it a valuable resource to the user?

My customer for this system was the Service Operations department, which was specifically concerned with receiving reports on an ad hoc basis that would help them with warranty expense dealer audits. Report selection criteria would change from report to report, and no one in the department had ever used SAS software, nor had anyone any familiarity with the mainframe batch environment and job control requirements.

The user signs on to TSO, then types in a CLIST command that activates the system. Menus are displayed that allow the user to choose which report is needed.

<table>
<thead>
<tr>
<th>Press END to return.</th>
</tr>
</thead>
<tbody>
<tr>
<td>Warranty Claims Tracking Auditor Reporting System</td>
</tr>
<tr>
<td>WHICH REPORT DO YOU WANT TO CREATE?</td>
</tr>
<tr>
<td>PF12 Standard audit: produces four reports</td>
</tr>
<tr>
<td>PF14 Mini-audit: produces seven reports</td>
</tr>
<tr>
<td>PF13 Parts-In Stock audit: NOT AVAILABLE</td>
</tr>
<tr>
<td>To make your selection press the corresponding PF key.</td>
</tr>
<tr>
<td>For more information about each report press PF16.</td>
</tr>
</tbody>
</table>

This system eliminates the duplication of programming effort by allowing many users to use the same system concurrently. Another problem that is avoided is a conflict of information that can occur when different sources or methods are used to get the same information; using this system, the source and the method are the same every time.
Once the user chooses (using the PF keys) a report, an 'OPTIONS SELECTIONS' program screen is displayed offering a selection of variables that can be used as selection criteria for the report. If the user makes a selection, then input screens are opened for the user to type in the values.

Once the user chooses (using the PF keys) a report, an 'OPTIONS SELECTIONS' program screen is displayed offering a selection of variables that can be used as selection criteria for the report. If the user makes a selection, then input screens are opened for the user to type in the values.

A final menu asks the user whether he/she would like to continue to create more reports or to end the SAS session.

The report is generated in an overnight processing queue and is delivered by the next morning. The first page of the report will print an audit of the selection criteria as the user input them.

A series of edit checks and user ERROR and HELP screens are built into the system to assist and guide the user in a friendly way. The user may also exit from any screen (using the Emergency Exit) and terminate the SAS session or start the system over again without any adverse effects to the operation of the system.
HOW THE CODE GENERATOR WORKS

The important points of the code generation process are at the 'OPTIONS SELECTION' screen and the 'REQUIRED SELECTIONS' screen.

When the OPTIONS screen is invoked, a SAS dataset (the options dataset) is created to hold the values input by the user for the variables listed on the screen. When the user makes the choice(s), a routine checks for the logical combination of variables to support business coding schemes.

For each of the variables chosen by the user, the FSEDIT procedure is used to edit the options dataset. One observation is created for each variable chosen (the user enters all possible values of the variable on one FSEDIT input screen).

When the user is finished, the REQUIRED screen will display. The user must input values for the fields shown. The SAS/AF macros for conditional execution are used to check for the completion of those fields.

When the REQUIRED screen is processing and the user input and edit checks are completed, a fixed block, sequential dataset is created. This dataset will hold the batch job code that is generated.

The options dataset is read and a macro containing ARRAY statements and IF/THEN/ELSE DO statements builds code 'modules' of 'IF' statements. These 'modules' are inserted into the hard-coded portion of the report code which is then written (using _NULL_ dataset, FILE and PUT statements) to the sequential dataset.

The job is immediately submitted, using the TSO SUBMIT command, and both the options dataset and the sequential dataset are immediately deleted, using the TSO DELETE command. This is a security feature that will not allow the user to access the actual code that is producing the report so that no alterations can be made.

DATASET SECURITY REQUIREMENTS

The system, stored in a SAS dataset, must be shared by many users at the same time. A system of dataset security should be planned (by naming convention, for example) that would prevent a more sophisticated end user from changing any of the code in the system catalog.

The dataset that this system is stored in is defined in the first-level name qualifier with my TSO user I.D. number. Only those persons who have write access to my datasets may make changes to the system. The end-user group is defined at the second-level qualifier of the dataset name, which is defined for read access only. Defining the end-user level also prevents the system from being used by unauthorized individuals.

HIGHLIGHTS OF THE SYSTEM / CONCLUSION

- It is virtually a 'push-button' system.
- The SAS software coding, JCL coding, TSO commands, and dataset management are transparent to the user.
- Edit checks and logic checks on fields protect the integrity of the information produced.
- The reports are processed in batch, which is more efficient than interactive processing.
- The batch job code is efficient; less experienced SAS programmers have used much more c.p.u. time to produce the same reports.
- Twenty-four hour report turnaround is guaranteed.
- The system is easy for a non-programmer to use, and is user-friendly in its offering of Help and Error Explanation screens.
- Security features protect the integrity of the information produced, as well as protect the system from unauthorized changes.

SAS, SAS/AF, SAS/FSP are registered trademarks of SAS Institute Inc., Cary, NC, USA.