INTRODUCTION

Self Serve is a computer system developed at York University to assist the production of reports as required by the university executive to assist them in the decision making process. Its initial development was in response to the need to provide information quickly to the senior administrators. As its name suggests, Self Serve is intended for use by those who may have little to no computer programming expertise. One of the primary benefits of this system is that it provides the end user with a certain amount of freedom from the traditional dependency on systems analysts and programmers.

SYSTEM OVERVIEW

The initial version of Self Serve was implemented using the Roscoe RPF language and of course SAS. Five menus as described in detail below, were developed to assist the user in building SAS programs which in turn were submitted for processing in the MVS batch environment. These SAS programs would access SAS databases prepared from the student records files. Once the processing was completed the output could be viewed through Roscoe or printed on a line printer.

The first menu presented the Self Serve Logo. The second menu presented the user with a selection of databases that could be accessed. The third menu was used to specify the selection criteria and the sort criteria. In the selection criteria the user through IF statements and DELETE statements would specify which records to be excluded from the report. It was decided to allow actual SAS statements to be used rather than attempt to build macros, since it would be difficult to anticipate all the macros that would be needed. Also exposing the user to SAS would allow them to gain some expertise in SAS programming over time. All the data set control statements such as the DATA and SET statements are provided automatically and need not be specified. On the same menu the sort criteria could be specified if required, by listing the variables in the order that the report is to be sorted. This statement would have the same format as the BY statement in a regular SAS program.

The fourth menu was used to select the report type; listing or frequency. If a listing was to be produced (via Proc Print) then the user would enter the names of the variables that were to appear in the listing. Optionally, one could specify control variables if special paging was required. If a frequency was required then the top half of the screen was used. The user would enter the names of the variables to be used in the crosstabulation. Optionally, variables could be weighted by using the WEIGHT statement or paging and report layout controlled via the BY statement. Once the fourth menu was completed the SAS program was created using the information provided and prepared for submission for processing in the batch environment.

The fifth menu would present the default job control information, which would be suitable for most programs. However, as the user grew more experienced they could alter the settings to suit their requirements. The information presented on this screen would consist of job control information, routing information, name of the database to be accessed and most importantly the title to be included as part of the report. Once the ENTER key was pressed the SAS program would be submitted for processing. Once ready, the output could be viewed at the terminal using the Roscoe output facility or simply printed on the line-printer and picked up later.

This version of Self Serve was quite primitive and limited in its flexibility. However, it did permit reports to be produced quickly and efficiently without the involvement of systems analysts and programmers, who could now concentrate on larger and more complex requests. Appendix A contains examples of the menus used in this initial version of Self Serve.

The current version of Self Serve looks quite different from the initial version. A decision was made to allow the user access to the full roscoe editor, something which the previous Self Serve did not allow. Secondly, rather than provide panels to guide the user in program creation, a decision was made to provide model programs. Through menus the user selects the database to be accessed and a sample program is displayed. At this point the full roscoe editor is available for program creation. The user is expected to enter the selection criteria, sort criteria, the proc statements, and the title statements. After the program statements have been entered a program function key is pressed to call an action menu to further process the newly created SAS program. Via the action menu the program can be saved in a librarian file external to Roscoe, or can be submitted for processing in the batch environment.

Through the revised menus the user could retrieve sample SAS programs or custom written programs, alter them as required and then submit them for processing. This added feature now allows for entire systems to be scheduled and submitted for processing by non data processing personnel. In fact the Enrolment Reporting Officer at York has access to all the SAS programs that are used to prepare the enrolment
reporting tape for our government. He can select and submit an individual program or through what we call the express lane feature, submit a whole series of programs at once.

Appendix B has examples of some of the menus available in the current Self Serve. The Self Serve Logo is the first screen to be presented. Here the user can select one of three options. The first option will display the SAS program selection menu, the second option will cause an index of the contents of the user's librarian file to be displayed. The third option will invoke the services of the Self Serve output processor used to review SAS program output, once jobs have been processed in the batch environment.

The SAS program selection menu consists of one screen divided into 2 windows. In the first window is shown a list of the databases or categories of programs available to the user; for example, enrollment reporting, finance, budgeting, SHF processing, student records. The user presses the appropriate Program Function Key to indicate the selection. Now the programs that are available within that selection are listed in window 2. If there are more than 12 programs, then the user can scroll through the list by using PFkeys 12 or 1. If the selected area is further broken into sub-categories, then window 2 can be used to display these programs as well. The user selects the desired SAS program in window 2 by pressing the appropriate PF key. The program is retrieved and displayed to the user. The full power of the Roscoe editor can now be used in developing the SAS program. If the program selected has been custom written and requires no modification then the user presses the PF2 key and the action request screen is displayed. The program currently being created or reviewed can now be saved or submitted for processing.

Last year we began to use SAS under CMS. Here we have done away with the menu driven approach and instead simply provide the users with the SAS database and model programs. The database is created according to the user's specifications and includes only those records and variables that are required. The model programs include all the CMS control statements and SAS statements required to produce a simple report. A data dictionary is included in the model program using comment statements. Armed with the model program and data dictionary the users can experiment until the desired report is produced. The communication features provided by CMS allow for the sharing of programs and simplify debugging. When a user encounters an error in the SAS syntax or the logic of their program they simply send a copy of the SASLOG to me. I can review the log and send then an electronic note describing how to correct the error.

The full screen procedures available with SAS/CMS make it possible to maintain small specific databases which in the past would have required a major expenditure of time and money. Now with FSEDIT, a new database can be created within hours or days and is very inexpensive to maintain and easily accessible by the user. The CMS version of Self Serve has not replaced the previous ROSCOE versions but rather they complement each other. The programs that extract the data from large administrative databases and create the smaller databases are available to the user on the ROSCOE based SELF SERVE. The file is then transferred to CMS, but at the user's discretion.

In January of this year we received our copy of SAS/PC and began to explore its use as a Self Serve tool. Within a very short period of time we were able to transfer some of the smaller database extracts to the IBM/AT and produce useful reports in the same fashion as on CMS. In fact now we can transfer the SAS databases to the high density floppy available on the IBM/AT and distribute model programs and data to the users on these. SAS/PC has many attractive features and will make a valuable addition to our Self Serve facility. There are two new features that we are quite pleased with. The linesize option allows for line lengths greater than 133 characters and the output from SAS/PC is placed in an ASCII file that can be easily processed by a word processor. So now the user can use SAS/PC to produce reports and then easily include them in documents.

Another feature of SAS/PC that we have found very promising is the SAS/AF facility which provides menu driven assistance for the various SAS procedures. This will allow the users to explore and utilize the power of the various SAS procedures by simply filling in the blanks on the SAS/AF menu screens.

SELF SERVE PHILOSOPHY

The primary purpose of Self Serve is to allow the end user to access his/her own data quickly and directly without having to rely on others. There is no reason why a manager, director, or vice-president cannot obtain information quickly and when required. The person that needs the information is probably the best one to sit at a terminal and program the request. The ability to do so provides the user with a feeling of independence and encourages experimentation. This combined with the user's familiarity with the data and a good sense of what is required proves in many instances to be very productive and rewarding for that user.

The user learns to think through a request and gains an appreciation for what the computer can and cannot do. In the long run a Self Serve user gains a fair bit of programming experience and also helps when dealing with data processing personnel. Being this close to the data has other benefits in that the user can search for data anomalies resulting in a more accurate database.

280
CONCLUSION

Self Serve has been in use at York University for the past 3 years and has met with favorable response and results and is now being used by many offices within the university.

The Alumni office uses Self Serve in trying to maintain an accurate alumni file. They will run SAS programs searching for data anomalies. The Finance department uses Self Serve in preparing government reports and in analyzing spending trends throughout the University. The Personnel office uses Self Serve to produce reports about the University's staffing. The Registrar's office uses Self Serve to produce reports for the various student offices across the campus.

The Enrolment Reporting and Statistics officer can be considered the biggest user of the Self Serve system. He prepares almost all of his own reports and has access to all of the programs that are used in an enrolment reporting exercise. Weeks before any exercise is to be run, he will run the entire exercise himself in order to test the programs and check for any data anomalies.

The Associate Vice-president, who is a very strong supporter of the Self Serve system, has over the past year become a SAS/CMS expert and can be seen between meetings running reports and trying to master PROC TABULATE.

The Self Serve system at York has proven to be a useful and productive tool. With such a tool in place the offices and individuals mentioned above have been given access to their data and can freely produce reports without having to rely on others. This has enabled the Data processing departments to spend more time on developing new large scale systems and databases, and also the time to investigate new hardware.

ACKNOWLEDGEMENTS

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

The author can be contacted at:
York University
South 911 Ross Building,
4700 Keele St., North York, Ontario,
M3J 1P3, Canada.

(416) 6676362

APPENDIX A

SELF SERVE REPORTING

REPORT CATEGORY
1. STATCAN - YORK OFFICIAL REPORTING SYSTEM
2. DATABASE - POST ADMISSIONS
3. DATABASE - PRE ADMISSIONS
4. DATABASE - COURSE ANALYSIS
5. DATABASE - POST ADMISSIONS -- LABELS
X. EXIT SELF SERVE

PLEASE ENTER SELECTION (1,2,3,5,X) :

MENU 2

SELF SERVE REPORTING

MODE = DATABASE - POST ADMISSIONS REPORTING

SELECTION CRITERIA
LINE 1 : if faculty ne 'as' then delete;
LINE 2 :
LINE 3 :
LINE 4 :
LINE 5 :
LINE 6 :
LINE 7 :
LINE 8 :
LINE 9 :
LINE 10:

SORT CRITERIA
1. 2. 3.
4. 5. 6.

MENU 3

SELF SERVICE REPORTING

FREQUENCY TABLES

TABLES year * major :

OR

TABLES * * :

BY :

WEIGHT :

------- PRINT PROCEDURE -----

VAR :

BY :

MENU 4
SELF SERVE REPORTING

FACILITY

MODULE NAME: IAPSTRB
CATEGORY: SELF SERVE
DESC: TABLES YEAR & MAJOR:
DATA SET USED: YORK.LEVY.QPART.PROD(O)
JOB PARGS: 2,5,LOCK2,CLASS=A
AUTH. CODE: xxxxxxxx
OUTPUT TO RETURN TO ROSECOE: N
ABORT RUN (Y/N): N

TITLE1 'YORK UNIVERSITY';
TITLE2 'POST ADMISSIONS FILE ANALYSIS';
TITLE3 'USER GENERATED TITLE';
TITLE4 'USER GENERATED TITLE';

MENU 5

APPENDIX B

YORK UNIVERSITY -- SELF SERVE FACILITY

--- WINDOW 1 --- WINDOW 2 ---

SERVICE JUNE ENROLMENT REPORTING

SELECTED:

PF1 - JUNE 30
PF2 - JUNE 17
PF3 - EXIT
PF4 - AUG 26
PF5 - NOV 1
PF6 - FEB 1
PF7 - MAR 1
PF8 - JUNE 6000 PROGRAM
PF9 - JUNE 7000 PROGRAM
PF10 - JUNE 8000 PROGRAM
PF11 - JUNE SPECIAL REPORTS
PF12 - NEXT PAGE
PA2 - WINDOW 3

USE THE ENTER KEY TO SWITCH WINDOWS

MENU 2

YORK UNIVERSITY -- SELF SERVE FACILITY

--- WINDOW 1 --- WINDOW 2 ---

SERVICE JULY REPORTING -- PAGE 2

SELECTED:

PF1 - JULY 30
PF2 - JULY 17
PF3 - EXIT
PF4 - AUG 26
PF5 - NOV 1
PF6 - FEB 1
PF7 - MAR 1
PF8 - JUNE 6000 PROGRAM
PF9 - JUNE 7000 PROGRAM
PF10 - JUNE 8000 PROGRAM
PF11 - JUNE SPECIAL REPORTS
PF12 - NEXT PAGE
PA2 - WINDOW 3

USE THE ENTER KEY TO SWITCH WINDOWS

MENU 4

282
SELF SERVE MENU

PF1 -- SAVE (WITH OPTIONAL SUBMIT)
PF2 -- SUBMIT (WITH OPTIONAL SAVE)
PF3 -- EXIT
PF4 -- SUPPLEMENTARY COMPONENTS
PF5 -- REPORT COMPONENTS
PF6 -- RETURN TO MAIN MENU

PRESS THE APPROPRIATE PF KEY TO INDICATE SELECTION

SAVE MENU

SELF SERVE SUBMIT PROCESS

JOB NAME : XXLN5026
TIME : 4
LINES : 7
PROGRAMMER : LOCKER B
ROUTE : ROSCOE (Y/N) Y
CMS USER: AT NODE:
JOB CLASS : A
ACCOUNT : xxxxxxxx

ENTER PASSWORD:

PRESS PF3 TO EXIT

PRESS PF5 TO SUBMIT TO IBM FOR PROCESSING
SELF SERVE LOGO