

**TBETAS: Toronto Board of Education Teacher Allocation System
Using SAS Software, EXEC, XEDIT Macros, and COBOL**

Avtar S. Dhanota

Disclaimer: The opinions expressed below are those of the author and do not necessarily reflect a position of the Toronto Board of Education.

ABSTRACT

TBETAS is a menu-driven system developed under the CMS environment using SAS software, EXEC, XEDIT macros, and COBOL. The system is used to allocate teachers, based upon various factors, to 163 of the Toronto Board of Education elementary and secondary schools. About five years ago this system was originally developed using IBM's ISPF dialog management facility, EXEC, and COBOL. The old system did not meet the user's requirements. The user wanted multiple access to the data base and a comprehensive on-line locate/search facility. This was possible to provide using SAS software; therefore, about two years ago the old system was converted from ISPF to SAS software. This system also uses COBOL to access centrally maintained VSAM tables (table of school related parameters) to generate reports instead of stand-alone SAS format codes. In this application SAS software also proved very useful to write programs which in turn write other SAS programs.

SYSTEM OVERVIEW

TBETAS is a stand-alone menu-driven system under CMS. The user is taken directly into the system's main menu (see SCREEN 1) by invoking it through the user's "PROFILE EXEC". When the user selects option 1 from the main menu (only option 1 will be discussed in this paper), the system first takes the user through the primary menu (SAS/AF), data entry/update panel (SAS/FSP), and SAS/AF program screens to capture the user specified parameters. Then the system writes data files on to the disk which are then passed on to system JCL (Appendix A) using EXEC and XEDIT macros. Finally, an exec submits the job to a VSE machine, and on completion returns the user to the main menu.

(NOTE: The SAS and system JCL codes shown are for illustration purpose only; Some of the codes are intentionally omitted to save space. Any participant interested in the complete codes may contact the author).

Primary Menu

When the sign-on is complete, the user's PROFILE EXEC fires the system by executing the following code.

```
PROC DISPLAY CATALOG=ctlg.staff.primary.menu list;
```

As shown in Screen 1, there are four options. In the discussion that follows, I have presented only the first option in this paper i.e., Elementary Staff Allocation.

```

-----
                PRIMARY MENU
-----
1. Elementary Staff Allocation
2. Secondary Staff Allocation
3. Class-size Statistics
4. Human Resources' Statistics
-----

```

SCREEN 1

Anatomy of the Primary Menu

Screen 2 shows (partially) SAS/AF screens that are processed as a result of selecting each task in the primary menu.

```

-----
                PRIMARY MENU
-----
1. Elementary Staff Allocation
2. Secondary Staff Allocation
.....
-----
Command ==>
-----
Parent Name ..... Libref
Screen Type ..... Catalog
-----
Opt. Key Name      Type Lib Catalog
-----
1  _  ELMSTAFF PROGRAM
2  _  SCDSTAFF PROGRAM
.....
-----

```

SCREEN 2

DETAIL

OPTION 1: ELEMENTARY STAFF ALLOCATION

When the user selects option 1 from the main menu (Screen 1), the system displays the following SAS/AF program screen:

```

-----
 DATASET AND REPORT TYPE PANEL
-----
      Dataset number: & (1 - 6)
 Update the dataset?: & (Y / N)
      Detail Report?: & (Y / N)
 Actual/projections?: & (A / P)
-----

```

SCREEN 3

The user has the option of selecting any one of the six datasets residing under SAS/SHARE. These datasets contain students' enrolment data, such as regular class-size, special education students, inner-city enrolment, special education units etc. In the beginning of the year, all the datasets are identical. However, as what-if analysis progresses, the enrolment figures start changing. The user assigns different datasets to different what-if analysis. This minimizes the update time for the successive runs because the enrolment figures for a particular analysis are not changed by another kind of analysis.

In order to keep track of all the parameters (e.g., dataset number, type of reports etc.) specified by the user, the first few pages of the report(s) contain screen images of Screen 3 and Screen 4.

```

/* ----- */
/* Tell the user to hold his/her */
/* horses while SAS is busy */
/* working on user's request. */
/* */
/* MSGWAIT EXEC */
/* The update screen will be */
/* displayed in a moment. */
/* Please stand by..... */
/* ----- */

```

CMS EX MSGWAIT; RUN;

```

/* ----- */
/* Create SAS Macros to capture */
/* the parameters specified in */
/* THE PANEL AS SHOWN IN SCREEN 3*/
/* ----- */

```

```

%let dataset=dataset.elmstaff&field1;
%let datano = &field1;
%let updfg = '&field2';
%let repfor = &field3;
%let retype= &field4;

```

```

/* ----- */
/* Based upon the user's */
/* RESPONSES CREATE SAS JOB STREAM*/
/* TO INVOKE PROC FSEDIT */
/* ----- */

```

OPTION DQUOTE;
CMS FILEDEF PROGRAM DISK
PROGRAM SAS A;

```

DATA _NULL_;
FILE PROGRAM;
UPDFG=&UPDFG;
/* Code SAS program if requested */
IF UPDFG='Y' THEN DO;
PUT "LIBNAME dataset
SERVER=SASSERVS;";
PUT "PROC FSEDIT DATA=&DATASET (read=xxxx)
SCREEN=SCREEN.EA_ENRL;";
PUT 'RUN;';
PUT 'CMS EX MSGWAIT;';
END;
PUT '* DUMMY STATEMENT;';

```

```

/* ----- */
/* Assuming that the user has */
/* chosen to update the dataset, */
/* then execute PROGRAM SAS A */
/* written above with FILEDEF */
/* ----- */

```

%INCLUDE PROGRAM;

```

/* ----- */
/* After exiting FSEDIT, go */
/* initialize staff allocation */
/* factors the user wants to apply*/
/* (user has the option to change)*/
/* and display SAS/AF program */
/* screen (Screen 4). */
/* ----- */

```

```

%LET AJK = 0.05000;
%LET ASK = 0.04110;
%LET APRM = 0.04010;

```

/* 20 more factors*/

```
PROC DISPLAY
C=ctlg.staff.elmfctr.program;
```

```
-----
STAFF ALLOCATION FORMULA
ITEM      FACTOR
Run Number ==> &rn
           J.K. ==> &ajk___
           S.K. ==> &ask___
Primary(1-3) ==> &aprm___
Junior(4-6) ==> &ajr___
.....
(18 more factors)
-----
SCREEN 4
```

When the user exits from FSEDIT and SAS/AF program screens, the following data files are written onto a mini disk (codes omitted intentionally).

(1) **Elementy Data A:** Enrolment data file which was updated through the FSEDIT screen.

(2) **Factor Data A:** Staff allocation formula file (data captured through Screen 4) which will be used to calculate the number of teachers for each school.

The aforementioned data files are passed on to the system JCL (Appendix A) using EXECs and XEDIT macros as described in the following text.

UPSI Switches:

Now create a numeric value for the UPSI switches, later to be used by the COBOL program in order to print the type of report the user has requested (see Screen 3).

```
OPTION DQUOTE;
DATA NULL;
REPFOR="&REPFOR"; /* see Screen 3 */
IF REPFOR='N' THEN
UPSISW='00000000'; /* Summary Report */
ELSE
UPSISW='10000000'; /* Detail Report */
```

```
/* ----- */
/* Hold UPSI. It may disappear. We */
/* need it later to specify the */
/* type of report the user has */
/* requested. */
/* ----- */
```

```
CALL SYMPUT('UPSI',UPSISW);
OPTION DQUOTE;
CMS FILEDEF PROGRAM2
DISK PROGRAM2 SAS A;
DATA NULL;
FILE PROGRAM2;
```

/* write SAS codes to be executed later*/

```
PUT "CMS EXEC CHANGE
      **UPSI** &UPSI
      110INFO JCL A;"
```

```
/* ----- */
/* Make a copy of the MASTER file */
/* of the system JCL which sits */
/* in the R/O minidisk "G" */
/* ----- */
```

```
CMS COPY 110INFO COPY G
        110INFO JCL A;
```

```
/*----- */
/* INCLUDE ENROLMENT DATA IN JCL */
/* ----- */
/* 1. Call XEDIT macro using */
/* INSERT EXEC */
/* 2. Look for text string FLAG */
/* in system JCL (APPENDIX A) */
/* 3. Insert enrolment data */
/* (e.g., ELEMENTY DATA A ) */
/*----- */
```

```
CMS EXEC INSERT FLAG
      ELEMENTY DATA A 110INFO JCL A;
```

/*----- INSERT EXEC -----*/

```
&TRACE OFF
&STACK MACRO INSERT &1 &2 &3 &4
XEDIT &5 &6 &7
```

```
/*-----INSERT XEDIT MACRO-----*/
```

```
&TRACE OFF
COMMAND FIND &1
&IF &RETCODE = 0 &GOTO -FOUND
CP MESSAGE - UNABLE TO FIND &1
&GOTO -FILEIT
-FOUND
COMMAND DELETE 1
COMMAND UP 1
COMMAND GET &2 &3 &4
&IF &RETCODE = 0 &GOTO -FILEIT
CP MESSAGE - UNABLE TO MERGE &2 &3 &4
-FILEIT
COMMAND FILE
```

```
/* ----- */
/* INCLUDE FACTORS IN SYSTEM JCL */
/* Now pass on staff allocation */
/* factors to system JCL by */
/* executing INSERT EXEC again. */
/* ----- */
```

```
CMS EXEC INSERT FLAG
FACTOR DATA A
110INFO JCL A;
```

```
/* ----- */
/*PROGRAM2, written internally, */
/*contains the following codes */
/*CMS EXEC CHANGE **UPSI** &UPSI */
/* 110INFO JCL A*/
/* ----- */
```

```
%INCLUDE PROGRAM2;
```

```
/*----- CHANGE EXEC -----*/
```

```
/* ----- */
/* CHANGE EXEC will change UPSI */
/* switch in the system Note that */
/* the number value e.g., 00000000*/
/* in this case has already been */
/* captured (see SYMPUT statement)*/
/* ----- */
```

```
&TRACE OFF
&STACK MACRO CHANGE &1 &2 &3 &4 &5
XEDIT CHANGE &3 &4 &5
```

```
/*-----CHANGE XEDIT MACRO-----*/
```

```
&TRACE OFF
```

```
* &1 = text of the flag to be located
* &2 = text of the flag to be replaced with
```

```
&CH = &CONCAT OF C/ &1 / &2 /*
&CH
COMMAND FILE
```

```
/*----- */
/* The system JCL is now ready to */
/* to be fired to VSE */
/*----- */
```

```
CMS EXEC PNCH 110INFO JCL A VSE31 ;
```

```
/* Now back to square 1 - get */
/* rid of the garbage */
```

```
CMS ERASE 110INFO JCL A;
CMS ERASE ELEMENTY DATA A;
CMS ERASE FACTOR DATA A;
CMS ERASE PROGRAM SAS A;
CMS ERASE PROGRAM2 SAS A;
```

```
*****
/* Voila! */
/* Time to return to the main menu */
*****
```

```
PROC DISPLAY
CATALOG=ctlg.staff.primary.menu list;
```

Reports

This system produces two kinds of reports -- a detail report by school, and a summary report for complete school system. For both of these reports this system uses VSAM files to capture school attributes such as, school name, inner-city rank, school administrative area and city ward. The reports are sorted using various combinations of these attributes.

For ease of identification, we capture all the input from screen 3 and screen 4 and print the exact screen images on the leading pages of each report.

APPENDIX A

```
// JOB 740S.A
// OPTION NOLOG
* CREATE STAFF ALLOCATION DISK FILE
// ASSGN SYS002,X'00C'
// DLBL SORTOUT,'740S.A.ALLOC.ELEM.FILE',N,R,T=100,SYS001
// DLBL SORTWK1,,N,R,T=10,SYS002,D
// EXEC SORT,SIZE=(AUTO,64K)
```

----- more JCL codes -----

FLAG Enrolment file (ELEMENTY DATA A) goes here

```
/* -- END SORT --
// DLBL CATFFF,'TBEFFF.USERCAT',,VSAM
// DLBL SYS011,'SCHOOL.MASTER.VSAM',,VSAM,CAT=CATFFF
// DLBL SYS002,'740S.A.ALLOC.ELEM.FILE',0
// DLBL SYS010,'740S.A.PROJ.ENROL',N,R,T=02
// EXEC IS001EA,SIZE=(AUTO,100K)
// DLBL SORTIN1,'740S.A.PROJ.ENROL',0,SYS002
// DLBL SORTWK1,,N,R,T=02,D,SYS003
// DLBL SORTWK2,,N,R,T=02,D,SYS004
// DLBL SORTOUT,'740S.A.PROJ.ENROL.S',N,R,T=02,SYS001
// EXEC SORT
```

```
* $$ LST CLASS=T,JSEP=0,RBS=0,LST=SYS008
// DLBL SYS010,'740S.A.PROJ.ENROL.S',0
// OPTION PARTDUMP
// UPSI **UPSI** ( UPSI value is captured in the SAS/AF panel)
```

FLAG Factor data file (FACTOR DATA A) goes here

```
/* -
/&
```

Acknowledgements:

The author wishes to thank Cheung Lui, programmer, at the Toronto Board of Education for his help in developing the system JCL and COBOL programs.

SAS is the registered trademark of SAS Institute Inc., Cary, NC, USA. ISPF is a product of IBM Corporation.

Contact:

Avtar S. Dhanota
Dept. of Computer Services
Toronto Board of Education
155 College St., Toronto, Ontario
Canada M5T 1P6
Tel: (416) 591-8163