

DEVELOPING A SAS BASED DECISION SUPPORT SYSTEM FOR DATABASE CLEANUP

**Yi-Sheng Hwang, Strategic Planning
Child Support Enforcement Division
Office of the Attorney General**

ABSTRACT

Decision support systems (DSS) can assist decision makers in their decision-making process by providing statistics, graphs, models, and simulations. A well-designed and well-maintained decision support system is a powerful tool that can guide and support decision makers.

This paper describes DBCLEAN, a SAS/PC based decision support system for mainframe database cleanup. The system provides for manager determination of skills, numbers, and amount of overtime to be used for cleanup. It allows managers to select criteria that affect the number of cases (records) checked and expected number of errors to be corrected; expected cost of each decision; and expected working days to accomplish the project.

The system was developed under 386 PC and SAS/PC software. Since SAS/AF and SAS/EIS were not available in this organization when the system was designed, this menu-driven system was developed by using WINDOW and DISPLAY statements in data steps and %WINDOW and %DISPLAY statements of SAS macros.

INTRODUCTION

A new automated system is being developed to improve the current Texas Child Support Enforcement Program and to comply with federal guidelines. To increase the data accuracy of the current system and to assure the success of the new system, Database Cleanup Project was proposed to top management. In the proposal, thirty out of hundreds of data elements (fields) were identified as critical to conversion to the new system and must be cleaned before the conversion. Failure to clean the data will result in unsuccessful conversion and operation of the new system. There were another 54 data elements were classified as non-critical but should be cleaned. These elements that are essential to external evaluation of Child Support Enforcement operations, funding and earned revenues, that may result in reduced funding levels, less than deserved performance evaluations, or failure of federal audits but will not cause failure of the new system conversion effort.

To gather essential information in assisting top management and Database Cleanup Steering Committee in making the final decision, a database cleanup test plan was developed to determine:

- . The average time required by a Child Support Officer to clean up a case using the data elements identified by the Steering Committee.
- . The validity of the criteria used to select cases and fields to be cleaned.
- . The estimated manpower required for the accomplishment of the database cleanup effort.
- . The estimated cost and working days required to complete the project based on the resources assigned.

The test was performed on two field offices with randomly-selected sample cases from these two offices. Based on the caseload at that time and at a 95% confidence level, 381 cases were selected.

Also, a Database Cleanup Sub-committee was formed with staffs from several divisions to determine the suspect criteria for each data element (field).

Each of the critical and non-critical data fields would be queried for each of the 381 cases, and classified as "suspect" if the values in that element fall outside the expected logical values, or "non-suspect" otherwise. Each case would be classified as "suspect" if one or more elements are classified as "suspect," and "non-suspect" only if all of the data elements are classified "non-suspect."

A worksheet was generated for each case to identify whether the case was suspect or non-suspect. It also contained the values and indicated which data elements were suspect. Normally, non-suspect cases would not be cleaned. However, to determine how many non-suspect cases should have been cleaned through the test, all the non-suspect cases were checked. During the cleanup, each data element on the worksheet was compared to the actual data contained in the case folder. Then each checked data element was identified as "matched" or "unmatched" on the worksheet.

The test data were collected and analyzed and the result was used in the development of the Database Cleanup Decision Support System.

SYSTEM STRUCTURE

The DBCLEAN was designed to operate on a high performance 386 PC and PC/SAS software. Several window menus were developed to provide a menu-driven environment in helping users working with the system. The structure of the system is shown in Figure 1.

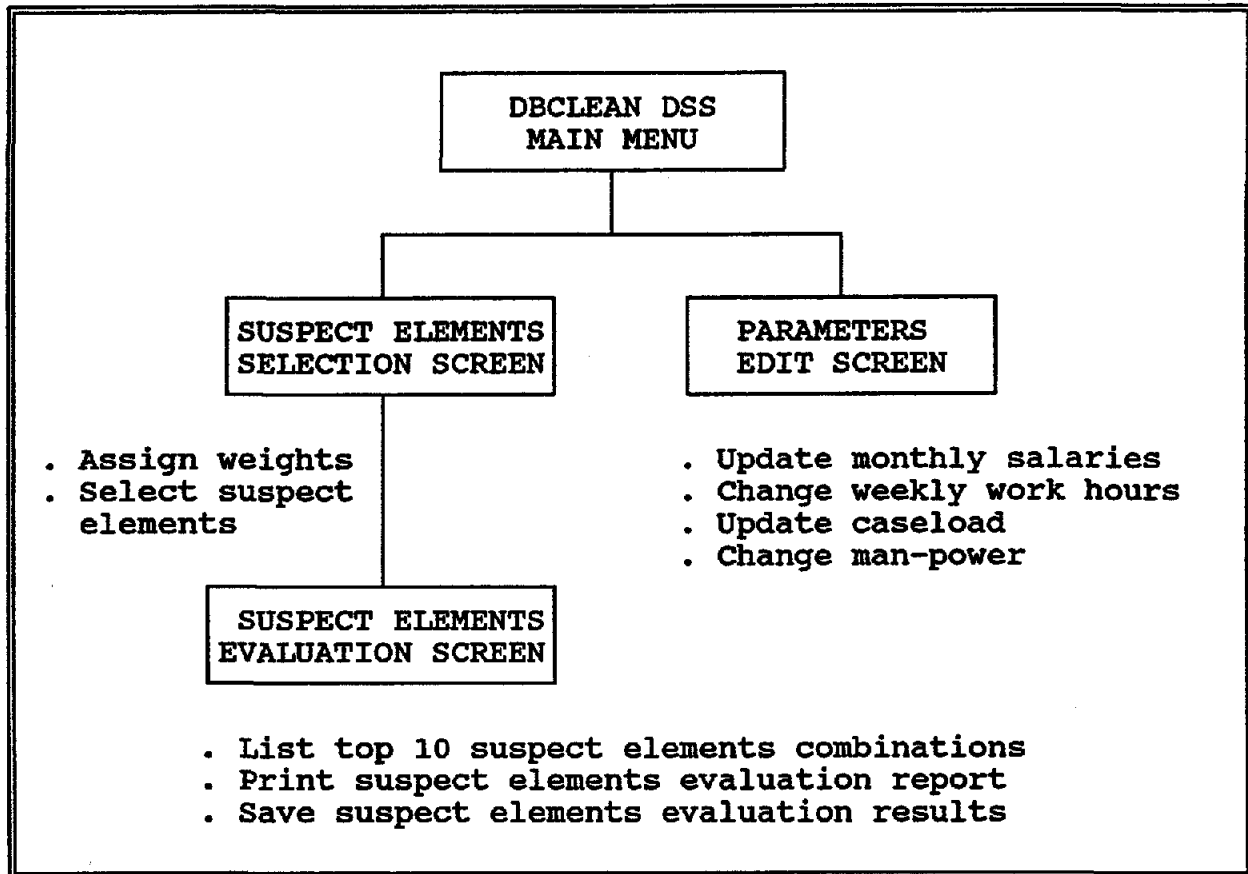


Figure 1

SYSTEM OPERATIONS AND PROGRAMMING TECHNIQUES

Main Menu

The DBCLEAN starts with a main menu with two options and exit under the menu (see Figure 2). User uses the TAB key move the cursor between options and place an "X" under the selection then press ENTER to activate the following window screen.

```

MAIN
Command ==>

DATA BASE CLEANUP DECISION SUPPORT SYSTEM
MAIN MENU

Suspect Elements      Parameters Edit
Selection             Screen
-                     X

EXIT
-

Place an "X" under your selection and press ENTER.

```

Figure 2

If any character other than "X" is entered under the selection, the system will display a small red window that contains an error message (see Figure 3). The original menu redispays with all fields reset to null after user presses ENTER.

```

MAIN
Command ==>

DATA BASE CLEANUP DECISION SUPPORT SYSTEM
MAIN MENU

Suspect Elements      Parameters Edit
Selection             Screen
-                     C

ERROR
Command ==>

ERROR:  Incorrect value entered in select field
CORRECT: Place an X in the appropriate field.
Press ENTER to continue.

Place an "X" under your selection and press ENTER.

```

Figure 3

The SAS code used to generate the main menu was adapted from Tina Keene's paper[1]. A number of macro variables were defined to make the menu change much easier. In the codes below, %WINDOW statement was used to define two windows: MAIN and ERROR.

```
%LET CA1=C=MAGENTA ATTR=REV_VIDEO;
%LET C1=16;
%LET C2=41;
%LET C3=30;

%WINDOW MAIN
#1 @17 'DATA BASE CLEANUP DECISION SUPPORT SYSTEM' C=YELLOW
#2 @17 '          MAIN MENU          ' C=YELLOW
#5 @&C1 '          ' &CA1
   @&C2 '          ' &CA1
#6 @&C1 ' ' &CA1 'Suspect Element' C=WHITE ' ' &CA1
   @&C2 ' ' &CA1 'Parameters Edit' C=WHITE ' ' &CA1
#7 @&C1 ' ' &CA1 'Selection' C=WHITE ' ' &CA1
   @&C2 ' ' &CA1 'Screen' C=WHITE ' ' &CA1
#8 @&C1 '          ' &CA1
   @&C2 '          ' &CA1
#13 @&C3 '          ' &CA1
#14 @&C3 ' ' &CA1 'EXIT' C=WHITE ' ' &CA1
#15 @&C3 '          ' &CA1
#9 @25 SUSPECT 1 AUTO=YES
   @50 EDIT 1 AUTO=YES
#16 @37 EXIT 1 AUTO=YES
#20 @15 'Place an "X" under your selection and press ENTER.' C=GREEN;

%WINDOW ERROR IROW=15 ROWS=6 ICOLUMN=5 COLUMNS=60 COLOR=RED
#1 @5 'ERROR: Incorrect value entered in selecting field'
#2 @5 'CORRECT: Place an X in the appropriate field.'
#3 @5 '          Press ENTER to continue.';
```

A macro (%DBCLEAN) was defined to control the flow of displaying main menu and error window, branching to the next screen, and validating the user's input. The conditional macro statements %DO %UNITL and %IF %THEN were utilized to achieve the functions.

```
%LET SUSPECT=;
%LET EDITPAR=;
%LET EXIT=;

%MACRO DBCLEAN;
  %DO %UNTIL(%UPCASE(%EXIT)=X);
    %DISPLAY MAIN;
    %IF %UPCASE(%EXIT)=X AND %UPCASE(%EDIT)= AND %UPCASE(%SUSPECT)=
      %THEN %GOTO OUT;
    %ELSE %IF %UPCASE(%SUSPECT)=X AND %UPCASE(%EDIT)= AND %UPCASE(%EXIT)=
      %THEN %DO;
        %INCLUDE 'D:\SAS\CLEANUP\CLN1.SAS';
        %LET SUSPECT=;
      %END;
    %ELSE %IF %UPCASE(%EDIT)=X AND %UPCASE(%SUSPECT)= AND %UPCASE(%EXIT)=
      %THEN %DO;
        %INCLUDE 'D:\SAS\CLEANUP\CLN2.SAS';
        %LET EDIT=;
      %END;
  %END;
```

```

XEND;
XELSE XDO;
  XDISPLAY ERROR;
  XLET SUSPECT=;
  XLET EDIT=;
  XLET EXIT=;
XEND;
XEND;

XOUT: XDO;
  XLET SUSPECT=;
  XLET EDIT=;
  XLET EXIT=;
XEND;
XMEND DBCLEAN;

```

Parameters Edit Screen

Selecting the Parameters Edit Screen will lead to a window with all the current data that used in the evaluation calculation (Figure 4). These parameters include average time spend on cleaning each case, working hours per week per person, monthly salaries for different skill levels, and incentive rate. The screen also allows users to update the caseloads and manpowers in each region. User can make changes on these parameters and save the changes.

```

EDITST-----
Command ==>

                PARAMETERS EDIT SCREEN - STATE

Average Time/Folder: 10 minutes      Week Hours/Person: 40

Monthly Salary: Manager=$3,223  CSO=$1,849  Ad_Tech=$1,622  Other=$1,350

Incentive (Salary Premium): 1.5
-----
                ***** Manpower *****
                Manager    CSO    Ad_Tech    Other
Region 1      55,898        3        17        14
Region 2      41,055        2        12        12
Region 3      40,211        2        9         6
Region 4      31,137        4        12         7
Region 5      19,150        2        9         5
Region 6      22,995        3        13         8
Region 7      34,028        2        12         7
State       395,272       18       87       59

Type over to change data;      Save (Y/N): Y

```

Figure 4

The above screen was defined by using WINDOW statements within DATA steps. Each parameter on the screen was assigned to a variable and kept in a permanent dataset (CLNPAST). A macro SAVEYN was defined to decide whether the changed values will be saved or not.

```

XLET CA1=C=MAGENTA ATTR=REV_VIDEO AUTOSKIP=YES;

DATA CLNPAST;
  SET DBCLN.CLNPAST;
  WINDOW EDITST
  #1 @25 'PARAMETER EDIT SCREEN - STATE' COLOR=YELLOW
  #3 @2 'Average Time/Folder(minutes):' +1 AVG_TIME 2. &CA1
  @45 'Week Hours/Person: ' WK_HRS 2. &CA1
  #5 @2 'Monthly Salary: Manager' +1 SLR_MGR 5. &CA1 +3
  'CSO' +1 SLR_CSI 5. &CA1 +3 'Ad_Tech' +1 SLR_TEC 5.
  &CA1 +3 'Other' +1 SLR_OTH 5. &CA1
  #7 @2 'Incentive (Salary Premium):' +1 INCENT 3.1 &CA1
  #8 @2 '-----'
  #9 @14 '***** Manpower *****'
  #10 @14 'Caseload      Manager      CSO      Ad_Tech      Other '
  #11 @2 'Area 1' +6 CSLD1 8. &CA1 +7 MP_MGR1 6. &CA1 +5
  MP_CSI1 6. &CA1 +5 MP_TEC1 6. &CA1 +5 MP_OTH1 6. &CA1
  #12 @2 'Area 2' +6 CSLD2 8. &CA1 +7 MP_MGR2 6. &CA1 +5
  MP_CSI2 6. &CA1 +5 MP_TEC2 6. &CA1 +5 MP_OTH2 6. &CA1
  #13 @2 'Area 3' +6 CSLD1 8. &CA1 +7 MP_MGR3 6. &CA1 +5
  MP_CSI3 6. &CA1 +5 MP_TEC3 6. &CA1 +5 MP_OTH3 6. &CA1
  #14 @2 'Area 4' +6 CSLD4 8. &CA1 +7 MP_MGR4 6. &CA1 +5
  MP_CSI4 6. &CA1 +5 MP_TEC4 6. &CA1 +5 MP_OTH4 6. &CA1
  #15 @2 'Area 5' +6 CSLD5 8. &CA1 +7 MP_MGR5 6. &CA1 +5
  MP_CSI5 6. &CA1 +5 MP_TEC5 6. &CA1 +5 MP_OTH5 6. &CA1
  #16 @2 'Area 6' +6 CSLD6 8. &CA1 +7 MP_MGR6 6. &CA1 +5
  MP_CSI6 6. &CA1 +5 MP_TEC6 6. &CA1 +5 MP_OTH6 6. &CA1
  #17 @2 'Area 7' +6 CSLD7 8. &CA1 +7 MP_MGR7 6. &CA1 +5
  MP_CSI7 6. &CA1 +5 MP_TEC7 6. &CA1 +5 MP_OTH7 6. &CA1
  #18 @2 'State ' +6 CSLD 8. &CA1 +7 MP_MGR 6. &CA1 +5
  MP_CSI 6. &CA1 +5 MP_TEC 6. &CA1 +5 MP_OTH 6. &CA1
  #20 @2 'Type over to change data;' +5 'Save (Y/N):' +1 SAVE $1. &CA1;
  DISPLAY EDITST;
  CALL SYMPUT('SAVEYN',SAVE);
RUN;

%MACRO SAVEYN;
  %IF &SAVEYN=Y %THEN %DO;
    DATA DBCLN.CLNPAST;
      SET CLNPAST;
    %END;
  %MEND SAVEYN;

```

Suspect Elements Selecting Screen

Choosing Suspect Elements Selection on Main Menu will display a selection screen that contains four weighting fields for evaluation variables and thirty critical elements (see Figure 5). A dataset was created to store the weighting values and elements' markings, i.e. the "X"s, from the previous selection. Users can overwrite the current weight values and select elements that they like to use as suspect criteria to identify the suspect cases. The selections of the suspect elements were limited to 8. Typing "Y" at the bottom will start the calculation.

SUSPECT
Command ==>

SUSPECT ELEMENTS SELECTION SCREEN - STATE

WEIGHT FOR CS_CHECKED: 10	WEIGHT FOR CS_CLEANED: 10
WEIGHT FOR ERR_CLEANED: 10	WEIGHT FOR EFFICIENCY: 10

- NCP Last Name	- Application on File	- Child Last Name
- NCP First Name	- Agreement Type	- Child Gender
- NCP SSN	- PP1 Amount	- Child SSN
- NCP Date of Birth	- PP1 Frequency	- Child Date of Birth
- CP Last Name	- Payment Start Date	- AFDC Case No.
- CP First Name	- PP2 Amount	- Legal Action
- CP SSN	- PP2 Frequency	- Date Filed
- Child BOW	- CP Date of Birth	- Date of Service
- URESA	- CP/NCP Relationship	- Date of Disposed
- FIPS Code	- Child First Name	- Result

Change Weights: Overwrite the current numbers;
 Select Suspect Elements: Mark X in front of elements (up to 8);
 Start the calculateion ? (Y/N) :

Figure 5

DBCLEAN was designed to be the most effective system in helping decision makers during the suspect elements selection process. When the user selects one suspect element the system will automatically use that element, combine it with each of the remaining 29 unselected elements, and calculate each combination. If two suspect elements are marked, the system will calculate 28 different combinations, i.e. 2 selected elements combined with the remaining 28 unselected elements. Instead of selecting the suspect elements combination one by one and then comparing the results, the system provides the results for all the possible combinations. Without this feature the selecting process would become a very tedious task.

Similar to the Parameters Edit Screen, a WINDOW statement was used to produce the Suspect Elements Selection Screen.

```
%LET CA1=C=MAGENTA A=REV_VIDEO AUTO=YES;

%MACRO SUSPWIN;
  DATA CLNSLST;
    SET DBCLN.CLNSLST;
  WINDOW SUSPECT
  #1 @20 'SUSPECT ELEMENTS SELECTION SCREEN - STATE' C=YELLOW
  #3 @2 'WEIGHT FOR CS_CHECKED:' +2 WT_CHK 3. &CA1
    @45 'WEIGHT FOR CS_CLEANED:' +1 WT_CLN 3. &CA1
  #4 @2 'WEIGHT FOR ERR_CLEANED:' +1 WT_ERR 3. &CA1
    @45 'WEIGHT FOR EFFICIENCY:' +1 WT_EFF 3. &CA1
  #6 @3 NCP_LAST $1. &CA1 +1 '- NCP Last Name'
    @28 APP_ON_F $1. &CA1 +1 '- Application on File'
    @53 CH_LAST $1. &CA1 +1 '- Child Last Name'
  #7 @3 NCPFIRST $1. &CA1 +1 '- NCP First Name'
    @28 AGR_TYPE $1. &CA1 +1 '- Agreement Type'
```

```

      @53 CH_GNDR $1. &CA1 +1 '- Child Gender'
#8  @3 NCP_SSN $1. &CA1 +1 '- NCP SSN'
      @28 PP1_AMT $1. &CA1 +1 '- PP1 Amount'
      @53 CH_SSN $1. &CA1 +1 '- Child SSN'
#9  @3 NCP_DOB $1. &CA1 +1 '- NCP Date of Birth'
      @28 PP1_FREQ $1. &CA1 +1 '- PP1 Frequency'
      @53 CH_DOB $1. &CA1 +1 '- Child Date of Birth'
#10 @3 CP_LAST $1. &CA1 +1 '- CP Last Name'
      @28 PAY_STRT $1. &CA1 +1 '- Payment Start Date'
      @53 AFDC $1. &CA1 +1 '- AFDC Case No'
#11 @3 CP_FIRST $1. &CA1 +1 '- CP First Name'
      @28 PP2_AMT $1. &CA1 +1 '- PP2 Amount'
      @53 LEG_ACT $1. &CA1 +1 '- Legal Action'
#12 @3 CP_SSN $1. &CA1 +1 '- CP SSN'
      @28 PP2_FREQ $1. &CA1 +1 '- PP2 Frequency'
      @53 DT_FILED $1. &CA1 +1 '- Date Filed'
#13 @3 CH_BOW $1. &CA1 +1 '- Child BOW'
      @28 CP_DOB $1. &CA1 +1 '- CP Date of Birth'
      @53 DT_SERVE $1. &CA1 +1 '- Date of Service'
#14 @3 URESA $1. &CA1 +1 '- URESA'
      @28 CP_NCP_R $1. &CA1 +1 '- NC/NCP Relationship'
      @53 DT_DISP $1. &CA1 +1 '- Date of Disposed'
#15 @3 FIPS $1. &CA1 +1 '- FIPS Code'
      @28 CH_FIRST $1. &CA1 +1 '- Child First Name'
      @53 RESULT $1. &CA1 +1 '- Result'
#18 @2 'Change Weights: Overwrite the current numbers;'
#19 @2 'Select Suspect Elements: Mark X in front of elements (up to 8
      elements);'
#20 @2 'Start the calculation ? (Y/N) ' CALC $1. &CA1;
      DISPLAY SUSPECT;
      CALL SYMPUT('CALYN',CALC);
RUN;
%MEND SUSPWIN;

```

To find out what and how many suspect elements were marked, PROC TRANSPOSE was used to generate dataset ELEMENT and transform 30 selecting elements(variables) into observations under new variable _NAME_. Meanwhile, the only original observation, which kept the selecting marks, was automatically assigned to variable COL1. Then a PROC SORT statement put those selected variables at the top of the observations. Finally, in DATA _NULL_ step, the program assigned all elements to macro variables with E1 through E30 by using CALL SYMPUT statements. Also, the program counted the number of selected elements and assigned the number to macro variable NO.

```

%MACRO CALYN;
  %IF &CALYN=Y %THEN %DO;
    DATA DBCLN.CLNSLST;
      SET CLNSLST;
    DATA CLNSLST(DROP=WT_CLN WT_CHK WT_ERR WT_EFF CALC);
      SET CLNSLST;
      CALL SYMPUT('WCLN',LEFT(WT_CLN));
      CALL SYMPUT('WCHK',LEFT(WT_CHK));
      CALL SYMPUT('WERR',LEFT(WT_ERR));
      CALL SYMPUT('WEFF',LEFT(WT_EFF));
    RUN;
    PROC TRANSPOSE DATA=CLNSLST OUT=ELEMENT;
      VAR NCP_LAST APP_ON_F CH_LAST NCPFIRST AGR_TYPE CH_GNDR NCP_SSN
          PP1_AMT CH_SSN NCP_DOB PP1_FREQ CH_DOB CP_LAST PAY_STRT
          AFDC CP_FIRST PP2_AMT LEG_ACT CP_SSN PP2_FREQ DT_FILED

```

```

CH_BOW CP_DOB DT_SERVE URESA CP_NCP_R DT_DISP FIPS
CH_FIRST RESULT;
RUN;

PROC SORT DATA=ELEMENT;
  BY DESCENDING COL1 _NAME_;
RUN;

DATA _NULL_;
  SET ELEMENT(RENAM=( _NAME_=VAR_NAME COL1=SUSP_SEL)) END=EOF;
  IF SUSP_SEL='X' THEN N+1;
  CALL SYMPUT('E'||LEFT(_N_),VAR_NAME);
  IF EOF THEN CALL SYMPUT('NO',TRIM(LEFT(N)));
RUN;

```

Because there are thousands of different selections user can make and for each selection the system has to combine the selected elements with unselected ones, this becomes a sophisticated programming problem. The following SAS codes illustrate how the problem was solved by using nested macro %DO loops. A macro %CALCULATE was defined in the program to compute the evaluation parameters.

```

%IF "&NO"="0" %THEN %DO;
  DATA SUSP(DROP=&E1 CORRECTC);
    SET DBCLN.CLEANUP(KEEP=&E1 CORRECTC) END=EOF;
    RETAIN MCHK MCLN MERR 0;
    IF &E1='*' THEN DO;
      MCHK+1; MERR+CORRECTC; END;
    IF &E1='*' AND CORRECTC=0 THEN MCLN+1;
    IF EOF THEN DO;
      S10=SYMGET('E1');
      WCLN=SYMGET('WCLN'); WCHK=SYMGET('WCHK');
      WERR=SYMGET('WERR'); WEFF=SYMGET('WEFF');
      OUTPUT;
    END;
  %DO I=2 %TO 30;
    DATA SUSPX(DROP=&E&I CORRECTC);
      SET DBCLN.CLEANUP(KEEP=&E&I CORRECTC) END=EOF;
      RETAIN MCHK MCLN MERR 0;
      IF &E&I='*' THEN DO;
        MCHK+1; MERR+CORRECTC; END;
      IF &E&I='*' AND CORRECTC=0 THEN MCLN+1;
      IF EOF THEN DO;
        S10="&E&I";
        WCLN=SYMGET('WCLN'); WCHK=SYMGET('WCHK');
        WERR=SYMGET('WERR'); WEFF=SYMGET('WEFF');
        OUTPUT;
      END;
    DATA SUSP; SET
      SUSP SUSPX;
  %END;;

  DATA CLNPAST;
    SET DBCLN.CLNPAST(KEEP=CSLD SLR_MGR SLR_CSI SLR_TEC SLR_OTH
      MP_MGR MP_CSI MP_TEC MP_OTH INCENT AVG_TIME WK_HRS);
    %DO I=1 %TO 30; OUTPUT; %END;
  DATA SUSP; MERGE
    SUSP CLNPAST;
  %CALCULATE;

```

```

KEEP S10 WCLN WCHK WERR WEFF PCLN PCHK
PERR PEFF OVRALL COST DAYS;
FORMAT PCHK PCLN PERR PEFF OVRALL 4.2;
FORMAT AVG_SLR DOLLAR6. COST DOLLAR12.;
RUN;
XEND;

%IF "&NO"="0" %THEN %DO;
%LET NO1=%EVAL(&NO+1);
DATA SUSP(DROP= %DO I=1 %TO &NO1; %&E&I %END; CORRECTC);
SET DBCLN.CLEANUP(KEEP= %DO I=1 %TO &NO1; %&E&I %END; CORRECTC)
END=EOF;
RETAIN MCHK MCLN MERR 0;
IF %DO I=1 %TO &NO; %&E&I=%' OR %END; %&E&NO1=%' THEN DO;
MCHK+1; MERR+CORRECTC; END;
IF (%DO I=1 %TO &NO; %&E&I=%' OR %END; %&E&NO1=%') AND
CORRECTC=0 THEN MCLN+1;
IF EOF THEN DO;
%DO I=1 %TO &NO; S&I="%&E&I"; %END; S10="%&E&NO1";
OUTPUT;
END;
%DO I=%EVAL(&NO+2) %TO 30;
DATA SUSPX(DROP= %DO J=1 %TO &NO; %&E&J %END; %&E&I CORRECTC);
SET DBCLN.CLEANUP(KEEP= %DO J=1 %TO &NO; %&E&J %END; %&E&I CORRECTC)
END=EOF;
RETAIN MCHK MCLN MERR 0;
IF %DO J=1 %TO &NO; %&E&J=%' OR %END; %&E&I=%' THEN DO;
MCHK+1; MERR+CORRECTC; END;
IF (%DO J=1 %TO &NO; %&E&J=%' OR %END; %&E&I=%') AND
CORRECTC=0 THEN MCLN+1;
IF EOF THEN DO;
%DO J=1 %TO &NO; S&J="%&E&J"; %END; S10="%&E&I";
OUTPUT;
END;
PROC APPEND BASE=SUSP DATA=SUSPX;
XEND;;

DATA CLNPAST;
SET DBCLN.CLNPAST(KEEP=CSLD SLR_MGR SLR_CSI SLR_TEC SLR_OTH
MP_MGR MP_CSI MP_TEC MP_OTH INCENT AVG_TIME WK_HRS);
%DO I=1 %TO %EVAL(30-&NO); OUTPUT; %END;
DATA SUSP; MERGE
SUSP CLNPAST;
WCLN=SYMGET('WCLN'); WCHK=SYMGET('WCHK');
WERR=SYMGET('WERR'); WEFF=SYMGET('WEFF');
%CALCULAT;
KEEP %DO I=1 %TO &NO; S&I %END; S10 WCLN WCHK WERR WEFF
PCLN PCHK PERR PEFF OVRALL COST DAYS;
%DO I=%EVAL(&NO+1) %TO 8; S&I=' '; %END;
FORMAT PCHK PCLN PERR PEFF OVRALL 4.2;
FORMAT AVG_SLR DOLLAR6. COST DOLLAR12.;
RUN;
XEND;;

```

Evaluation Screen

After completing the calculation, the system will list the top 10 combinations and their evaluation information on the screen (Figure 6). The ranking of the evaluation was based on Weighted Overall that considers four factors, the percentage of cases checked (CS_CHECKED RATIO), the percentage of cases cleaned (CS_CLEAVED RATIO), the percentage of errors cleaned (ER_CLEAVED RATIO), and criteria efficiency, altogether. Under the screen user can choose print or save the evaluation results or both.

DBCLEAN not only provides a powerful tool for management in the decision process of the database cleanup project, it also assists them in finding a cost-effective selection to accomplish the project. The following procedures are suggested when using the system to reach a final decision. First, make a run without selecting any suspect element and by giving the same weight to each evaluation factors. This will cause the system to calculate all the evaluation parameters for each element and generate "one-element report." Then user can select a suspect element from the 30 available based on the percentage of cases cleaned, weighted overall index, or importance of the element. The system will use the selected element to combine with the remaining elements individually and produce a "two-element report." User can continue to choose more suspects from the selection screen and change the weights to see the results and the cost for completing the project.

RESULTS							
Command ==>							
SUSPECT ELEMENTS EVALUATION - TOP 10 OPTIONS							
DATA ELEMENT	CS_CHECKED RATIO	CS_CLEAVED RATIO	ER_CLEAVED RATIO	CRITERIA EFFICIENCY	WEIGHT OVRALL	TOTAL COST	WORK DAYS
CP_NCP_R	0.75	0.80	0.84	0.59	0.65	1,191,307	116
CP_DOB	0.78	0.81	0.83	0.57	0.65	1,239,932	121
CH_SSN	0.67	0.69	0.66	0.56	0.64	1,069,745	104
PP1_AMT	0.79	0.76	0.73	0.52	0.62	1,264,245	123
PP1_FREQ	0.79	0.76	0.73	0.52	0.62	1,264,245	123
AGR_TYPE	0.72	0.68	0.63	0.52	0.62	1,142,683	112
DT_FILED	0.70	0.66	0.62	0.51	0.62	1,118,370	109
PAY_STRT	0.66	0.61	0.57	0.50	0.62	1,061,641	104
CH_DOB	0.48	0.45	0.49	0.51	0.62	761,788	74
AFDC	0.47	0.43	0.45	0.50	0.62	745,580	73
ELEMENT 1=NCP_LAST ELEMENT 5= CS_CHECKED WEIGHT=10 ELEMENT 2=NCP_SSN ELEMENT 6= CS_CLEAVED WEIGHT=10 ELEMENT 3= ELEMENT 7= ERR_CLEAVED WEIGHT=10 ELEMENT 4= ELEMENT 8= EFFICIENCY WEIGHT=10							
☒ - Print Report ☒ - Save Report ☒ - Exit							

Figure 6

EVALUATION METHODOLOGY

An evaluative model with weighted multiple criteria was used to calculate the weighted overall in the system. The model includes four factors: percentage of cases checked, percentage of cases cleaned, percentage of errors cleaned, and criteria efficiency of the selected elements. The criteria efficiency was defined as a ratio of the number of cases cleaned to the number of cases checked. Each factor has a weight assigned to it. By giving a zero weight to a factor, users can eliminate the factor from the overall index calculation. The formula for computing a weighted overall index is:

$$\text{Weighted Overall} = \frac{((1-P_CHK)*W_CHK + P_CLN*W_CLN + P_ERR*W_ERR + P_EFF*W_EFF)}{(W_CHK + W_CLN + W_ERR + W_EFF)}$$

where P_CHK = percentage of total cases need to be checked if selected elements are used to identify suspect cases.

P_CLN = percentage of total erroneous cases that will be cleaned if selected elements are used to identify suspect cases.

P_ERR = percentage of total erroneous data that will be cleaned if selected elements are used to identify suspect cases.

P_EFF = criteria efficiency of the selected suspect elements. This percentage is calculated from the number of cases cleaned divided by the number of cases checked.

W_CHK, W_CLN, W_ERR, and W_EFF are weights assigned to P_CHK, P_CLN, P_ERR, and P_EFF respectively.

OUTPUT REPORTS

The format of the printing report is similar to the top 10 combinations listed on the screen. However, the report lists all the possible combinations that use the selected element(s) with each of the remainings. The weights assigned to the factors and all the parameters (salaries, caseload, etc.) used in the calculation also are included in the report (Figure 7).

DATABASE CLEANUP SUSPECT ELEMENTS EVALUATION

DATA ELEMENT	CS_CHECKED RATIO	CS_CLEANED RATIO	ER_CLEANED RATIO	CRITERIA EFFICIENCY	WEIGHT OVRALL	TOTAL COST	WORK DAYS
CP_NCP_R	0.75	0.80	0.84	0.59	0.62	1,191,307	116
CP_DOB	0.78	0.81	0.83	0.57	0.61	1,239,932	121
CH_SSN	0.67	0.69	0.66	0.56	0.56	1,069,745	104
PP1_AMT	0.79	0.76	0.73	0.52	0.55	1,264,245	123
PP1_FREQ	0.79	0.76	0.73	0.52	0.55	1,264,245	123
AGR_TYPE	0.72	0.68	0.63	0.52	0.53	1,142,683	112
DT_FILED	0.70	0.66	0.62	0.51	0.53	1,118,370	109
PAY_STRT	0.66	0.61	0.57	0.50	0.50	1,061,641	104
CH_DOB	0.48	0.45	0.49	0.51	0.49	761,788	74
AFDC	0.47	0.43	0.45	0.50	0.48	745,580	73
CH_FIRST	0.47	0.43	0.45	0.50	0.48	745,580	73
CH_GNDR	0.47	0.43	0.45	0.50	0.48	745,580	73
CH_LAST	0.47	0.43	0.45	0.50	0.48	745,580	73
DT_DISP	0.49	0.46	0.42	0.51	0.47	777,997	76
DT_SERVE	0.43	0.38	0.36	0.49	0.45	680,747	66
NCP_DOB	0.40	0.35	0.33	0.47	0.44	640,226	63
LEG_ACT	0.41	0.34	0.34	0.44	0.43	656,435	64
RESULT	0.41	0.34	0.34	0.44	0.43	656,435	64
CH_BOW	0.43	0.36	0.33	0.45	0.43	680,747	66
CP_SSN	0.41	0.34	0.29	0.44	0.41	656,435	64
URESA	0.37	0.30	0.29	0.44	0.41	591,602	58
PP2_AMT	0.37	0.29	0.27	0.43	0.41	583,498	57
FIPS	0.37	0.29	0.27	0.43	0.41	583,498	57
APP_ON_F	0.36	0.28	0.25	0.43	0.40	567,289	55
CP_FIRST	0.36	0.28	0.25	0.43	0.40	567,289	55
CP_LAST	0.36	0.28	0.25	0.43	0.40	567,289	55
NCPFIRST	0.36	0.28	0.25	0.43	0.40	567,289	55
PP2_FREQ	0.36	0.28	0.25	0.43	0.40	567,289	55

ELEMENT 1=NCP_LAST	ELEMENT 5=	CS_CHECKED	WEIGHT=10
ELEMENT 2=NCP_SSN	ELEMENT 6=	CS_CLEANED	WEIGHT=10
ELEMENT 3=	ELEMENT 7=	ERR_CLEANED	WEIGHT=10
ELEMENT 4=	ELEMENT 8=	EFFICIENCY	WEIGHT=10

Average Time/Case=16 Minutes

Week Hour/Person=10

Monthly Salary: Manager=\$3,223 CSO=\$1,849 Ad_Tech=\$1,622 Other=\$1,350

Incentive (Salary Premium): 1.5

	Caseload	Manager	CSO	Ad_Tech	Other
Region 1	55,898	3	33	17	14
Region 2	61,055	2	27	14	12
Region 3	40,211	2	18	9	6
Region 4	81,737	4	24	12	7
Region 5	49,150	2	18	9	5
Region 6	22,995	3	30	15	8
Region 7	36,028	2	24	12	7
State	395,272	18	174	87	59

Figure 7

CONCLUSION

This paper presents a simple but useful SAS application that provides the decision makers with a flexible and effective tool in the decision-making process. The user friendly menu-driven environment not only allows user to change all the parameters involved in the calculation but provides an evaluation report immediately with the significant information needed for making a decision. The future plans are to add an automatic function to update caseload by submitting a mainframe program to find out the most current caseload and breakdown this system by region that will allow regional administrators to make their own regional decisions.

REFERENCES

- [1] Keene, Tina (1991). Macro Windows: Creating a User-Friendly Application. Proceedings of the Sixteenth Annual SAS Users Group International Conference. SAS Institute Inc., Cary, NC. pp. 223-228.
- [2] SAS Institute Inc. (1990). SAS Language: Reference, Version 6, First Edition. SAS Institute Inc., Cary, NC. pp. 322-323, 505-515.
- [3] SAS Institute Inc. (1987). SAS Guide to Macro Processing, Version 6, First Edition. SAS Institute Inc., Cary, NC. pp. 64, 105-113, 135-155.
- [4] SAS Institute Inc. (1988). SAS Language Guide for Personal Computers, Release 6.03 Edition. SAS Institute Inc., Cary, NC.
- [5] DiIorio, Frank C. (1991). SAS Applications Programming: A Gentle Introduction. PWS-KENT Publishing Company, Boston, MA. pp. 550-556.
- [6] Aster, Rick & Seidman, Rhena (1991). Professional SAS Programming Secrets. Windchest/McGraw-Hill, Inc., Blue Ridge Summit, PA. pp. 473-489.

CONTACT ADDRESS

The author can be contacted regarding the material presented in this paper at the following address:

Yi-Sheng Hwang
Strategic Planning, Child Support Enforcement Division
Office of the Attorney General
210 Barton Springs Rd.
Austin, TX 78704
(512) 463-2181 Ext. 2250