PCs and SAS® Software Applied to Public Health Surveillance

Philip J. d'Almada, Battelle Memorial Institute, Atlanta, Georgia

ABSTRACT

Since the cholera epidemic of 1854 in London, England, public health surveillance has incorporated the latest in technological advances. Information technology, in the form of SAS® software, release 6.04, and lap-top personal computers, has led to the development of the Foetal Alcohol Syndrome Applications Facility (FASAF) which was used in the surveillance of foetal alcohol syndrome (FAS). FASAF is an application product that allows end-users with minimal knowledge of SAS to create and maintain SAS relational data sets. FASAF features data entry quality control through warning messages and either on-screen or "pop-up" 'help' windows using SAS/FS® software and Screen Control Language (SCL). Characterised by a menuing system using SAS/AF® software and SCL, FASAF can interact with a dBASE III Plus "stand-alone" product, ANTHRO, to compute and save various paediatric anthropometric measures in the same SAS database.

INTRODUCTION

In the fifth century B.C., Hippocrates, a Greek physician and philosopher, first documented the importance of active surveillance with respect one's physical environment and how that knowledge could be used in the prevention of disease. Then, 2,000 years later in 1662, in London, England, a haberdasher, John Graunt demonstrated the value of routine data collection by manually analyzing weekly births and deaths, establishing the effect of seasonal variations. Some 200 years later still in England in 1839, William Farr, a physician, systematized the reporting of descriptive statistics on routinely collected data, and dealt with issues such as population definition and cause and effect. Finally, about 20 years later, another English physician, John Snow, applied some very contemporary epidemiological techniques in resolving the 1853-54 cholera epidemic in London, England. However, Snow's technique for surveillance, which was to walk door-to-door with a pencil and paper and a single-question "questionnaire"--"Who supplies your water?"--demonstrated little change in data-gathering technology since Hippocrates. Today, in contrast, surveillance could involve electronic technology for data collection and analysis, lengthier software-based questionnaires, and even telephony. For example, in 1993-94, the technological base used to conduct public health surveillance of foetal alcohol syndrome (FAS) included a lap-top personal computer (PC) with an Intel 80386 processor, SAS® software and a dBASE III Plus "stand-alone" product in an MS-DOS environment. Technology to support public health surveillance could include jet aircraft and cars for transportation, and the telephone system for electronic mail and long-distance conversation, in addition to on-site staff to collect data and off-site consultants to develop the software technology.

Known to affect some 4,000-8,000 births annually in the United States, FAS is a behaviour-based consequence. That is, when pregnant women consume inordinate amounts of alcohol, their babies might become physically and/or mentally affected. Growth retardation, facial abnormalities, some degree of mental retardation, and some degree of behavioural problems are expected characteristics of FAS. Other less discernible characteristics associated with in utero exposure to maternal alcohol consumption are termed foetal alcohol effects (FAE). As part of its epidemiological assistance to State governments, the Centers for Disease Control and Prevention (CDC) assisted the State of South Dakota in implementing a surveillance system for FAS among the American Indian population in the region. Battelle Memorial Institute provided the SAS software support under a government contract with CDC.

METHOD

Using the resources at CDC, Battelle constructed the FAS Applications Facility (FASAF) on a PC with an Intel 80486/33m CPU and SAS software release 6.04 loaded into a DOS environment from a local area network. Originally intended for a lap-top PC, the FASAF SAS catalog and SAS data sets were eventually installed with SAS software, release 6.04, on the hard drive of a PC with an Intel 80386 CPU in a DOS environment. FASAF was a menu-driven application for basic database management including data entry, data browsing, and data editing. Two SAS modules, SAS/AF® software and SAS/FS® software, were implemented in the design of FASAF and were supported with SAS screen control language (SCL). FASAF was designed so that non-SAS users, through the menuing system, could easily progress to data entry and also could easily enter data with the aid of on-screen or "pop-up" 'help' windows and/or warning messages that were provided when inconsistent or incorrect entries were keyed into data fields.

Since an iterative development process was employed to construct FASAF, the FASAF database comprised two data sets including the medical-chart abstraction data set and the growth-chart data set. These data sets opened into windows of 17 screens and one screen, respectively. The user could access the growth-chart data set either directly from the main menu or during medical-chart data entry through a window variable in Screen 10 whereby the user could choose to enter growth-chart data in a separate data-entry window then return to continue medical-chart data entry. The growth chart data set was constructed separately for four reasons: (i) the growth-chart information was submitted at the penultimate development stage; (ii) the main data-entry instrument, the medical-chart data-entry window already contained so many screens; (iii) the medical-chart data set had already become too large for an efficient, PC-based, analytic data set; (iv) the

In this paper, text-imbedded bold type was used to represent SAS SCL code, variable names, data set names, window names, frame names, and, in two instances, subheadings. Italicised lower-case bold type was used to represent generic nomenclature. Italicised upper-case bold type was used to represent generic nomenclature but in the case as controlled by the SAS System.
growth-chart information was intended to be located after Screen 10 of the medical-chart information thus creating the need to transpose numerous growth records per case to many additional fields per case and to interpose these fields in the screen flow of the medical-chart form. Thus, SAS SCL in SAS/FSP software was composed to overcome these reasons and to open the growth-chart data-entry window from Screen 10 of the medical-chart data-entry window. (Author's note: In the following sections as in this paragraph, a user perspective is first presented followed by a technical discussion.)

SAS/AF software and SCL

Upon initiating a SAS session, the user also would be transparently initiating FASAF and the FASAF main menu would be displayed (Figure 1). From the FASAF main MENU

Figure 1. A MENU DISPLAY window--FASAF main menu.

DISPLAY window (Figure 1), the user could choose from among four selections in the 'Data forms' section, to do data entry and browsing. These selections would open 'confirmation' or PROGRAM DISPLAY windows (Figure 2) that would provide the user with additional information on corresponding menu selections and to allow a change of mind regarding the implementation of menu selections. During data entry, various HELP DISPLAY windows would provide the user with correct data values for specific fields. Selections from the 'Other information' section of the main menu (Figure 1), would open MENU DISPLAY windows of submenus that allowed user access to background information on FASAF, to key instructions on some of the features of FASAF and implementation of FASAF, and to an analytic procedure that interfaced with a dBASE III Plus "stand-alone" product, ANTHRO.

SAS/AF software was used to build the menuing system which included menu windows, 'confirmation' windows, and 'help' windows, and these components were stored as MENU, PROGRAM, and HELP entries, respectively, in a SAS catalog.

Figure 2. A PROGRAM DISPLAY window--main menu selection 4.

Submenu MENU DISPLAY windows and PROGRAM DISPLAY windows were linked to the MENU DISPLAY window of the main menu through its menu attribute or MENU ATTR window (Figure 3). Similarly, HELP DISPLAY windows containing background information and key notes or instructions on FASAF, and other PROGRAM DISPLAY windows were linked to the MENU DISPLAY windows of the submenus through their respective submenu attribute or MENU ATTR windows.

Figure 3. The MENU ATTR window for the main menu.
The PROGRAM ATTR window was primarily used to define a suitable "Alias" or name for the window variable field provided for user response in the corresponding PROGRAM DISPLAY window. This "Alias" was used in the SCL composed in the PROGRAM SOURCE window (see below). Normally used to manage the general attributes or appearances of DISPLAY windows such as names, positions, and sizes, GATTR windows of the MENU, PROGRAM, and HELP, catalog entries were only used to provide suitable names for the respective MENU, PROGRAM, and HELP DISPLAY windows. SAS SCL was composed and compiled in user-transparent PROGRAM SOURCE windows to control the implementation of the main-menu selection from the PROGRAM DISPLAY or 'confirmation window in response to the user entry in the window variable field provided in the display. Thus, for example (as shown in Figure 2), through the user-response window variables, GCBS (allowing growth chart browsing in "data-entry" format) and GCBM (allowing growth chart browsing in tabular display), the following code would execute so as to ensure that only one appropriate response was made and, with one affirmative response keyed, either the following SCL code

```sas
IF GCBS = 'Y' & GCBM = 'Y' THEN DO;
  ALARM; CURSOR GCBS;
  _MSG_ = 'Make one selection only.);
END;
```

with the MODIFY command from the FSEDIT DATASETNAME window. From the FSEDIT Menu window (Figure 4) four of five options were invoked in the following order. First, using the SAS text editor in the FSEDIT Modify window, data-entry screens were modified by rearranging field locations and changing variable labels of corresponding fields so that the automated instrument more closely resembled the hard-copy surveillance instrument. Fields that were computed or that were repeated from screen to screen were listed in the FSEDIT Names window. Then relocated, repeated, and computational fields were identified in the FSEDIT Identify window. Second, general parameters were modified through the FSEDITParms window. Parameters controlled were the color and attribute of various areas of the data-entry screen(s), the override on errors, and the automatic save after a select number of observations were entered. Third, special attributes to data-entry fields were assigned through the FSEDIT Attribute window. This window was composed of a number of frames, one frame per attribute. The attributes assigned to various data-entry fields were controlled with only the MAXIMUM, MINIMUM, REQUIRED, and PROTECT frames. Fourth, SAS SCL program statements were composed, edited, and compiled in the FSEDIT Program window. SAS SCL was used together with specified field attributes to control variable ranges and to provide quality control during data entry. That is, field attributes and SCL were used to maintain data integrity during data entry. Validation of values of certain fields and consistency in responses among other

```sas
CALL FSEDIT('datASETNAME','screenentry','BROWSE');
```

where `datASETNAME` was the data set to be opened, and `screenentry` was the catalog screen entry that controlled the display of `datASETNAME`. (See also the SAS SCL code listed in SAS/AF software and SCL, above.) The FSEDIT DATASETNAME window was the display where the data entry screen layout was designed and where the data were keyed. This window was customized through the selections in the FSEDIT Menu window (Figure 4) which was opened

![Figure 4. The FSEDIT Menu window.](image-url)
fields were the checks for which SCL was composed. SAS SCL was also composed so that when an affirmative response was keyed in the appropriate window variable field of Screen 10 (Figure 5) of the medical-chart data-entry window and the

```
FSEDIT ABSTFORM
Command:  FSEDIT ABSTFORM
Obs:  0
MARKED: No observations on data set. Please press ESC or ALTER. Screen 10
INCIDENT: 010-9 search code: ___ Entry #: ___

SECTION 3: FAS DIAGNOSTICS-GROWTH DEFICIENCY
In growth deficiency mentioned in the chart: [Yes, No]
Was failure-to-thrive (FTT) diagnosed: [Yes, No]
Was growth deficiency calculated: [Yes, No]
Do you wish to enter growth chart data now? [Yes, then press ENTER; (If 'No', then leave blank.)
Otherwise, ...

To access the growth chart data independent of the chart abstraction form, return to the main menu (press F10) then select option 2.
Press F3 to leave any screen. All data entries or changes will be saved. This menu selection will be terminated. You will return to the main menu.

[Screen 1-9: SWF 17; Screen 11-17: SWF 16]
```

Figure 5. Screen 10 of the FSEDIT ABSTFORM window.

ENTER key pressed, SAS SCL would execute, and the growth-chart data-entry window would open. The SAS SCL statement used was

```
CALL FSEDIT ('lr.go', 'lr.fascat.gcscar');
```

where lr, go, fascat, and gcscar were as defined in the previous section. All features designed, developed, and composed by implementing MODIFY in the FSEDIT DATASETNAME window were saved in the FASAF SAS catalog as a screen entry. Thus, one screen entry per data set was created and saved in the FASAF SAS catalog. The remaining FSEDIT Menu option was a Help Modify window that provided details on screen modification.

SCL for warnings and windows

Whenever the user would key invalid or inconsistent values and press the ENTER key, SCL in the screen catalog entry for the current data set would execute, and any errors would be detected. Errors were indicated to the user when an alarm would sound, the cursor would relocate to the field in error, and either a warning/error message would be displayed below the command line of the screen, or a HELP DISPLAY window would be opened so as not to obscure the field in question, and would display valid field values.

SCL statements used to effect these error responses were either

```
ALARM;
CURSOR fieldname;
_MSG_ = 'message';
```

or

```
ALARM;
CURSOR fieldname;
CALL DISPLAY ('displayname.HELP');
```

where fieldname was the field to which the cursor was relocated, and displayname.HELP was the catalog entry name for the HELP DISPLAY window bearing valid field values. The positioning of messages was controlled by default by the SAS System whereas the positioning of the HELP DISPLAY window was controlled by design during development in SAS/AF software.

SCL for inter-data-set integrity

A key feature of FASAF in terms of maintaining a high-quality database was its capacity to pass identification numbers from the medical-chart abstraction data set to the growth-chart data set without user control. Thus, when the user would elect to enter growth-chart data from the medical-chart data-entry window by responding in the affirmative through the appropriate window variable field in Screen 10 (Figure 5), the value in the "Entry #" field in Figure 5 would be displayed in the "Entry #" field in the growth-chart data-entry window when that window would open as shown in Figure 6.

```
Instructions for growth chart data entry:
--- Note --- --- Note ---
Age will be computed in years and months. Percentile fascia head circumference will be computed. Enter length and head circumference with up to 2 decimal places.

--- Data Entry by Individual Record and Date of Measurement ---
This option allows for entry/insertion/deletion of weight, length/height, and head circumference. Measurements are entered for one date at a time.

FSEDIT GRTHCHRT
Command menu

--- GROWTH CHART DATA ENTRY SCREEN ---
Entry #:
Screen Observation Number: ___ Data: ___ Year: ___ Age: ___ yrs. ---
Height: ___ in. ___ ft. ___ in. --- Head Circumference: ___ in. ---
Length/Height: ___ in. ___ ft. ___ in. --- [If applicable] (If applicable) enter blank. (To add another record, press NEW at the 'Command menu' time. To exit: F10.)
```

Figure 6. The FSEDIT GRCCHRT window with legend over the preceding PROGRAM DISPLAY window.

Programmatically, the SAS macro variable assignment routines, CALL SYMPUT ('macrovarname', 'varname') for character variables and CALL SYMPUTN ('macrovarname', 'varname') for numeric variables, and the SAS retrieval functions, SYMGET ('macrovarname') for character values and SYMGETN ('macrovarname') for numeric variables, were used to preserve integrity of character and numeric identification numbers across data sets. In these routines and functions, macrovarname was the macro variable to which the value of the character/numeric variable varname was assigned, and from which that value was retrieved. These routines and functions were included in the SAS SCL code.


A number of computed fields for a frequency distribution were built into Screen 13 of the medical-chart data-entry window (Figure 7). After keying data values the user could press the ENTER key to compute and display the frequency values. These computed values would also display if the user would choose to review previously keyed records prior to keying new records.

Selected SAS features were used to complete the computed fields for this on-screen frequency distribution. Those SAS features included ARRAY statements, nested DO loops, and incrementing operations such as varname + 1. The following SAS SCL code.

```
ARRAY PMDXNUM {10} PMDX1-PMDX10;
/* --Produce on-screen distribution of previous mentions of FAS/FAE-- */
ARRAY CNTARR {9} CNT1-CNT9;
DIFFDX = 0;
DO J = 1 TO 9;
    CNTARR{J} = 0;
END;

/* --Produce 1 response on Screen 17 from Screen 13-- */
IF CNTARR(J) GT 0 THEN DIFFDX = 1;
ELSE DIFFDX = 0;
/* --Produce 1 response on Screen 17 from Screen 13-- */
IF CNTARR(J) GT 0 THEN DIFFDX = 1;
ELSE DIFFDX = 0;
```

SCL for computed fields

A number of computed fields for a frequency distribution were built into Screen 13 of the medical-chart data-entry window (Figure 7). After keying data values the user could press the

```
ENTRY key in the window variable field, GCDE in Screen 10 (Figure 5), the following SCL code

IF GCDE = 'Y' THEN DO;
    CALL SYMPUT ('XPTDOB',PTDOB);
    CALL SYMPUT ('XENTNO',ENTNO);
    CALL SYMPUT ('XSEX',SEX);
    CALL FSEDIT ('hr.gc','hr.fascat.gscf');
END;
```

in the medical-chart screen entry would be invoked, which would assign macro variables to the patient’s date of birth, record-entry identification number, and sex. The SCL code in the growth-chart screen entry that would retrieve the values of the macro variables was

```
IF ENTO = ' ' THEN ENTO = SYMGET ('XENTNO');
BDAT = SYMGETN ('XPmOB');
SEX = SYMGET ('XSEX');
```

and this code would execute when the growth-chart data-entry window would open. The patient’s date of birth and sex would not be displayed but would be used analytically.

SCL for SAS-dBASE interface

As mentioned earlier, selection '6' from the main menu (Figure 1) was an analytic option and supported the growth-chart data set. That is, growth-chart data included height and weight measurements; however, accurate population percentiles for all growth measurements on an individual were not easy to obtain, and since manual estimation of percentiles from growth curves would introduce measurement error, FASAF was complemented with a dBASE III Plus "stand-alone" product, ANTHRO, to provide these percentiles. ANTHRO, which was developed by CDC and the World Health Organization (WHO), was designed to compute various paediatric anthropometric measurements. Further, using the window variable field provided, the user could also "flag" existing growth-chart records for recomputation of percentiles if any previously keyed data were changed. Thus, through selection '6' from the main menu (Figure 1), a submenu would open (Figure 8) from which the user either could transparently prepare the growth-chart data set for computation of percentiles using the window variable field in the corresponding PROGRAM DISPLAY window, or the user could transparently update the growth-chart data set after the computation of percentiles through the window variable field in that corresponding PROGRAM DISPLAY window. In this version of FASAF, following the preparation of the growth-chart data set for computation of percentiles, the user would terminate the FASAF session then implement ANTHRO software where the user would be required to actively bypass certain options. Once the percentiles were computed, the user would terminate ANTHRO then start another SAS-FASAF
Programmatically, FASAF was interfaced with ANTHRO through SAS Sel, which was composed in SAS/AF software in two PROGRAM SOURCE windows. Figure 9 shows the relevant portion of SAS Sel code that would execute following a user-choice of selection '1' from the analytic submenu (Figure 8), and an affirmative user-response in the window variable field, PCTCALC. in the corresponding PROGRAM DISPLAY or 'confirmation' window. Similarly, Figure 10 shows the relevant portion of SAS SCL code that would execute following a user-choice of selection '2' from the analytic submenu (Figure 8), and an affirmative user-response in the window variable field, PCTRET. in the corresponding PROGRAM DISPLAY or 'confirmation' window. One feature of SCL employed in Figures 9 and 10 was the SUBMIT IMMEDIATE; block which allowed control to return to the submenu from which either selection was made following execution of the SUBMIT IMMEDIATE; block. A subset of growth-chart records would be set apart for computation of percentiles if growth measurements were missing in one percentile variable (Figure 9). Other growth-chart records simultaneously added to that subset for recomputation of percentiles would be records for which percentiles were previously computed but in which one or more variables were edited, and also, for which the user had keyed a response in the window variable field, RECOMP, to flag the record for recomputation of percentiles (Figure 9). Data set preparation was completed as weight values were converted from grams to kilograms, and a KEEP statement was used to restrict the list of variables passed to ANTHRO (Figure 9). This list included variables passed transparent to the user from the medical-chart data set to the growth-chart data set. (See also SCL for inter-data-set integrity, above.) PROC DBF was used to create GCPCTOLD.DBF from a SAS data set (Figure 9). During the ANTHRO session, the input dBASE file generated by SAS would be GCPCTOLD.DBF, and the output dBASE file to be read from by SAS would be GCPCTNEW.DBF. To retrieve percentiles, PROC DBF was used to read GCPCTNEW.DBF into a SAS data set (Figure 10). Since the ROUND(,.) function was not available in SAS software, release 6.04, SAS macroprogramming and the INT(,.) function were used in a DATA step to round off computed percentiles to the nearest integer (Figure 10). The RENAME = (oldvarname = newvarname ...) function was used in the same DATA step to change the dBASE III Plus variable names to the designated SAS variable names for the computed percentiles, and a KEEP statement was used to...
restrict the variable list passed to FASAF (Figure 10). PROC DATASETS was used to format the percentile variables, and the UPDATE statement was used in the final DATA step to add new percentiles to or to replace existing percentiles in the growth-chart data set, and to "switch off" the "recompute" flag by setting RECOMP to a blank missing value. This use of SCL allowed the user to be transparent to the preparation of data for computing missing percentiles and recomputing incorrect percentiles, and to be transparent to the retrieval of computed and recomputed percentiles and the updating of the database.

CONCLUSION
The CDC, contracting with Battelle Memorial Institute, provided the State of South Dakota with FASAF, a menu-driven, PC-based application developed in SAS software, release 6.04, which assisted in the surveillance of foetal alcohol syndrome. This application was iteratively developed over a 6-month to 1-year period when the interface to ANTHRO, a dBASE III Plus "stand-alone" product, was built into FASAF to complete growth-chart data with computed percentiles. FASAF, which combined the best available hardware and software technology at CDC, was developed by using the SAS System on a PC with an Intel 80486 CPU in Georgia and installed in a PC with an Intel 80386 CPU in South Dakota. A database, accurate from the moment of data entry, was provided to researchers and was also readily accessible for analysis with the SAS System. Further development of FASAF at CDC was halted as priorities and funding changed. However, a portable automated instrument was developed and successfully implemented by Battelle for CDC and offered the non-SAS user a simple means of creating an accurate SAS database.

REFERENCES
CDC (1993), "Preventing Fetal Alcohol Syndrome and Other Alcohol-Related Developmental Disabilities," FAS Prevention Section, Centers for Disease Control and Prevention, Atlanta, Georgia.


Hennekens, Charles H. and Julie E. Buring (1987), Epidemiology in Medicine, Boston/Toronto: Little, Brown and Company.


ACKNOWLEDGEMENTS
The author thanks Valborg Kvigne, Battelle Memorial Institute, and Michele D. Strachan, M.D., formerly of the Indian Health Service, for their contributions to the successful composition of SCL and design of FASAF. The author also thanks the Developmental Disabilities Branch, National Center for Environmental Health, Centers for Disease Control and Prevention for the physical facilities whereby this paper was prepared. The author thanks Dr. Joe Hollowell, Branch Chief, and Dr. Pierre Decoufle, Deputy Branch Chief, both of the Developmental Disabilities Branch, for administering the contract under which FASAF was developed, and for the time to document a portion of that work in this paper. The author appreciates the devotion of his wife, Shelley, and children, Lesli, Seth, Sammy, and Abby.

SAS, SAS/AF and SAS/FSP are registered trademarks or trademarks of SAS Institute Inc. in the USA and other countries. ® indicates USA registration.

Other brand and product names are registered trademarks or trademarks of their respective companies.

DISCLAIMER
Use of trade names is for identification only and does not imply endorsement by the Public Health Service or the U.S. Department of Health and Human Services.

Contact author at:
c/o CDC, 4770 Buford Hwy., Mail Stop F-15, Atlanta, GA 30341-3724.
404-488-7283