Abstract

Teachers in the public school must spend a certain number of hours per year attending inservice workshops. The Educational Service Center is involved in doing many of these presentations. Often, they arrange inservices but recruit professionals in the field to do the actual presentation.

In order to ascertain that they are providing a needed service and to gather data on the effectiveness of individual workshops and instructors, those attending are asked to complete an evaluation form at the end of a session.

The Education Service Center, Region II needed an effective way to analyze and maintain the data collected through these evaluation forms. It was decided to use an IBM XT Microcomputer to process and store this data.

I wrote a menu driven system that will:

a) read data from Chatsworth cards

b) find average evaluation figures for each workshop
c) find attendance totals for each workshop
d) store the attendance and evaluation data on disk
e) produce printed attendance and evaluation reports as workshop data is entered

f) and, produce printed or screen evaluation and attendance reports.
Table of Contents

Introduction..............................................4

Logical Description

Processing of Data.................................6

Retrieval of Data ................................8

Physical Description

INSVANAL.EXE..................................11

REPTRTN.EXE.................................16

INSVRTVL.EXE.................................21

Conclusion.............................................23

Sample Reports.................................Appendix A

Sample Screens.................................Appendix B

Files and Variables..........................Appendix C

Program Listings...............................Appendix D
Introduction

The Education Service Center, Region II was collecting evaluation data from inservice workshops presented by the Education Service Center and attended by school personnel. This data, however, was not being analyzed effectively by hand. They desired a computerized system that would effectively analyze the data.

It was decided to begin using a Chatsworth mark sense card as the evaluation form so that the analysis could be done with a minimum of keyboard input. The Chatsworth mark sense card reader can be programmed to read data of several types. The column spacing is indicated to the card reader by the timing marks along the bottom edge of the card. The Education Service Center, therefore, could have cards printed specific to their use.

The entries made on the card consists of the attendee's social security number, the workshop type and a unique workshop number. Also, it contains the type of program, the person's position, the level of children he works with and the school district where he is employed. Of these the workshop type, workshop number and type of program will be supplied by the instructor of the workshop. Finally, there are six evaluation criteria that are rated one through four.

The cards are then processed to produce evaluation summaries by school district, attendance lists and attendance summaries by school district, and evaluation and attendance averages per workshop. In determining the types of reports to be generated
the desire was to provide versatility. On some occasions the Education Service Center needs to produce a list of those attending to send to the respective school districts. At other times only verification of correctness is required so brevity is desired. For this reason I allow the user to make a choice of the report desired.

Another goal of the ESC was to develop a file of evaluation results. They must be able to show that they are providing a needed service to the schools. These fields were established so that data for any given year is separated from the previous and following years. The data for each workshop is stored as a separate unit. This allows the information to be recalled on a workshop or yearly basis.

The data can be organized in a variety of ways when recalling it from the file. This allows the user to compare the effectiveness and attendance based on several factors and by comparing different report types he can make judgements on what factors help to create an effective workshop.
LOGICAL DESCRIPTION
PROCESSING OF DATA

Incoming data is provided by two methods. Initially, the evaluation data is collected on Chatsworth mark sense cards completed by those attending the inservice workshop. The entries made on the card consists of the attendee’s social security number, the workshop type and a unique workshop number, the type of the program, the person’s position, the level of children he works with, the school district where he is employed, and six evaluation fields.

Secondly, the user enters from the keyboard information appropriate to the workshop. This consists of the workshop number and type, the instructor of the workshop, the date of the workshop and the location of the workshop.

As the cards are read certain fields are verified for accuracy. The workshop number and type must match that entered by the user from the keyboard. There must be a valid school district marked and the evaluations must be valid. The card is not processed if any of this information is invalid. A printed list of these invalid cards is produced. Other information may be invalid and the card will still be processed.

After reading up to 150 cards into memory the cards are processed. Attendance is counted by school district. Each of the six evaluation criteria fields is summed by school district. Attendance by position and level is counted by school district. Each person’s social security number, position, and level is sorted by school district.

The user then may choose what report he wishes to have produced. An attendance report must be chosen in order to verify
the validity of the processing. The user may choose a listing of social security numbers by school district. These reports can be used to inform the individual school administration of attendance from their school. A second choice gives attendance totals by position and level from each school. A third choice will give both reports. All attendance reports are followed by a listing of each of the six evaluation criteria averaged by school and the six evaluation criteria averaged for the entire workshop, a single workshop evaluation average and total attendance.

Information is stored in a file by workshop. For each workshop the following data is stored: all data input from the keyboard by the user; the number of school districts represented at the workshop; for each school district, the school district number, average evaluation one through six for that school district, attendance for that school district and a list of all social security numbers along with the position and level of those attending; finally average evaluations and attendance for the workshop.

All data read from cards and processed are represented by numbers except for the instructor’s name, the date and location of the workshop. These code numbers are matched to the appropriate string descriptor through data values contained in the program and initialized into arrays. This data consists of names for workshop types (component), programs, positions, levels, school districts and evaluation criteria and ratings.

The evaluation system is primarily intended for use by the Education Service Center. However, it may be used by other
organizations and the user is given the opportunity to specify a
different organization at the beginning of the program. It is
assumed that an organization other than the Education Service
Center may not desire the same breakdown of categories as the
Service Center. Therefore, if a special organization is chosen
the user is given the option to change the data descriptors
contained within the program concerning component, school
districts, programs, positions, levels, and evaluation criteria.
This data is stored on the file for use whenever this
organization is processed. Otherwise, the processing of the
evaluation cards is the same.

The data is stored in files based on the year the workshop
took place. The user may choose the year under which he wants
the data to be processed so that it may be changed as necessary.
The information for other organizations is also filed based on
year but the information for each organization is kept in a
separate file.

RETRIEVAL OF DATA

In order to retrieve this data the user is provided with a
second program, INSVRTVL.EXE. This also allows either the Education
Service Center or a special organization to retrieve the data on
file. Again the user will be allowed to determine which year’s
data he wishes to work with. He will be given a number of
choices to arrive at a specific report type.

His first choice will be between a year to date summary of
evaluations and attendance or a regeneration of the workshop
report produced as the data was entered. These reports serve two
purposes. The year to date summaries will produce accumulated data for the entire year. By using these reports the user may compare results achieved by various instructors, the popularity of certain workshops, the attendance variations by season and make other similar comparisons. The regeneration of original data will allow the user to determine if specific workshops were well received or if a particular person was in attendance at a specific workshop.

If the user chooses the year to date summaries, a menu will be displayed listing categories on which these reports may be based. These categories include date, instructor, workshop type, workshop number, and location. He may also choose a grand total evaluation and attendance report in which evaluations and attendance will be totaled for the entire year.

These year to date summaries contain both an evaluation summary by school district with total evaluation averages, and an attendance report by school district with the total attendance. The user may choose whether the attendance report will be divided by position held or level worked with.

If the user chooses a report based on the original format he will be asked to enter a desired workshop number. If he wishes a list of workshop numbers will be generated from which he may choose. After he has determined the appropriate workshop, he may choose a specific school district or may have reports printed from all school districts that attended. If he chooses a single school district only the attendance report for that school district will be printed. If he chooses all districts the report
will be as originally done when entering data. The option of attendance reports by position and level is not given since these reports can be generated through the other option.
INSVANAL.EXE

Initially the program fills the default arrays from data statements. The arrays COMPONENT.D$, PROGRAM.D$, POSITION.D$, LEVEL.D$, SCH.DIST.D$, EVALUATE.D$, and RATINGS.D$ are filled with the values in Appendix I - default values.

The default year routine then allows the user to change the file year to the year of his choice. This routine reads the file DATAYEAR.DAT for the current year file. The user is given an opportunity to change the year, and if changed the new year is filled back in DATAYEAR.DAT for use during program execution. This value remains as the current file year.

The user then must specify if he is processing data for ESC field or a special organization. If processing for a special organization the user is asked to input the name of the organization - ORGANIZATION$. From this name a filename is created using the first four valid characters of the organization name concatenated with the file date. An attempt is made to open this file for input. If the file does not exist the error is trapped and this filename can be used. If the file does exist the first record of the file is read into FILE.ORGANIZATION$. If FILE.ORGANIZATION$ is equal to ORGANIZATION$ then it is determined that data for an organization previously established is being processed and the data will be filed with the already existing file.

If ORGANIZATION$ does not equal FILE.ORGANIZATION$ then the user is informed that there is a problem, given the name of the organization on file and asked if they are the same organization. If the same, it is handled as above. If not the user is asked to
enter the organization in such a way that they will not be confused.

If the organization being processed is already on file, SPEC.FILE$ is reopened and the following arrays are filled by reading the file: FILE.ORGANIZATION$, COMPONENT.E$, SCH.DIST.E$, PROGRAM.E$, EVALUATE.E$, RATINGS$, POSITION.E$, LEVEL.E$. These values will be used instead of default values in data statements. It is assumed that the values will remain unchanged for a single organization whenever information for that organization is being processed.

If the organization being processed is a new organization the user is allowed to input new values for component, school districts, programs, evaluation criteria, position and level if desired. The default values for these fields are displayed on the screen and the user may change any field he wishes. He may leave some values blank but may not add additional values. No error checking will be done to eliminate data equated with blank fields, however. Changes are made in each case by entering the number of the item to change and then the new information. The user is always allowed to verify and change his input. After each entry he is shown the values as they currently exist and is asked if he would like to change another. Until he answers 'no' to this question he may make as many changes as desired, changing previously changed fields if needed.

For all organizations, special organizations and ESC R II, the user is now asked to enter information specific to the workshop(s) being processed. The number of workshops being
processed will be tallied in WK.SHOP.CNT as the data is entered.

The workshop number will be entered into the array element WK.SHOP.NUM$(WK.SHOP.CNT). It must be a three digit number and the entry will not be accepted if any alpha-numeric characters are present.

The workshop type is entered by choosing one of the existing default values by number. These values are shown on the screen. The user must enter a number from one through six. This value is held in COMPONENT.NUM$.

From the workshop number a filename is created in which this data will be stored. While doing this a check is made to see if the same filename already exists. This check is done by attempting to open the file for input. If it does not exist the error is trapped and the filename is determined to be valid. If it does exist the user will be asked to decide if he wants to write over the data on file or abort the processing. The abort would be chosen if a new workshop number needed to be chosen since the cards would need to be altered also. This filename is held in FILENAME$(WK.SHOP.CNT).

The instructor's name is entered into NME$. The user is instructed to make this entry last name first with no commas. Validity checking on this entry is left to the user and he is given ample opportunity to make corrections.

The date of the workshop is entered into INS.DATE$. The user is asked to enter a date with the following format MM/DD/YY. The validity of the date is checked for numerical entries separated by slashes or hyphens. Months and days with one digit must be preceded with a zero. The date is also compacted into the form
AHMOD and this is the form that will eventually be filed. This
value is held in COMP.DATE$.

The workshop location is entered into LOCATION$. The user
is asked to input the location without commas. The validity of
the location is checked by the user and he is given opportunities
to correct the entry.

When the user has made all entries the information entered
is listed on the screen. At this time the user is asked to
verify that all entries are correct and is given the opportunity
to make additional corrections. To make corrections the user
will enter the number of the item he wishes to change and the
program will reexecute the procedure to enter that item. When no
more corrections are needed he will enter a seven (7) for none.

If data is being processed for ESC R II the values for
component, schools, programs, evaluation criteria, ratings,
positions and levels are stored temporarily in the filename held
in FILENAME$(WK.SHOP.CNT).

For all organizations the values of INS.DATE$, COMP.DATE$,
NAME$, COMPONENT.NUM$, WK.SHOP.NUM$(WK.SHOP.CNT) and LOCATION$ are
stored in FILENAME$(WK.SHOP.CNT). The FILENAME$(WK.SHOP.CNT) is
appended to the list of filenames in YEARFILE$ if ESC R II or
SPECFILE$ if a special organization.

The user may now input information on another workshop if
desired. He may continue entering workshops until data for as
many as fifteen workshops has been entered. When data for all
desired workshops has been entered, then information is written
to the file HOLDFILE.DAT to be held for use after chaining to the
report producing program REPTRTN.EXE. This information includes: SPEC.FILE$ if applicable, WK.SHOP.CNT, all entries in FILENAME$ and all entries in WK.SHOP.NUM$. The program then chains to REPTRTN.EXE for processing the cards.
PHYSICAL DESCRIPTION

REPTRTN.EXE
The card reader is opened as file "CDM1:9600.E". In order to initialize the reader the card format is down loaded to the reader. In this case the card format is indicated by \texttt{CHR$(18) + "P774" + CHR$(18) + "H18" + CHR$(18) + "R1250" + CHR$(18) + "100" + CHR$(18) + "x=000102030400\*" + CHR$(18) + "E"}. Each command to the card reader must be preceded by a Control - R which is a \texttt{CHR$(16)$}. This causes the card reader to recognize the input as a command. Command R is the Hollerith Row Select Command. It is followed by a four digit octal code. This command causes the reader to scan only the enabled rows. The code 7774 allows rows 0 through 9 to be read. The code 1250 allows rows 7,5,3,1 to be read. The H command forces the card reader into the Hollerith to ASCII mode of operation. In this mode all legal Hollerith marks are converted to their equivalent ASCII character. The output will be one ASCII character per timing mark. The H command is followed by 2 digit decimal values that indicates the number of columns to be scanned in the Hollerith to ASCII mode. 00 indicates that all remaining rows should be read. The T command forces the card reader into the Translate Mode. In conjunction with the Command X it gives the computer the capacity of redefining the Hollerith rows. The two decimal digits following the T command indicate the number of columns to scan in this mode. \texttt{X-000102030400\*} redefines rows 1,3,5,7 to the values 1,2,3,4 respectively. The Command E is the end of transmission command. After the card format has been sent the card reader will respond with a G(CR) indicating a successful
Download or a string of ASCII characters followed by (CR) indicating where the error was found. The cards can now be input using an INPUT # statement.

The CARD.ARRAY$ is initialized to spaces. A card is read into CARD.IN$. Unless the social security number contains all $'s representing the end-of-file the workshop number, workshop type, school, and six evaluation fields are validated as containing legal values. If there is an error in any of these fields the card image is printed, the type of error indicated and the card is rejected. If no errors are found the value of CARD% is incremented and the card value is stored in CARD.ARRAY$(CARD%).

When up to 150 cards have been read and validated, CARD.ARRAY$ is sorted by workshop number. Cards from each workshop are written to a disk file to be held temporarily for processing. The file names are held in the array TEMPFILE.LIST$. These files will be read back one at a time for further processing.

To process the cards of a workshop the totaling arrays are initialized to zeros. ATTENDANCE() will hold the attendance of each school district, ATTENDANCE.TOTAL$( ) will hold the total attendance of each workshop, EVALUATION() will hold each of the six evaluations for each of the school districts. EVALUATION.TOTAL$( ) will hold the total of each of the six evaluations for each workshop. GRAND.EVAL$( ) will hold the overall evaluation total for each workshop. POSITION() and LEVEL$( ) will hold the attendance by position and level respectively for each school. The first element of the SOC.SEC.ARRAY$( ) is initialized to a "$" which is subsequently used as an end-of-data signal.
This array will hold the social security number, position and level for each person attending. These are kept by school.

If processing is being done for ESC R II the files are read for values for COMPONENT.E$, SCHEDIST.E$, PROGRAM.E$, EVALUATE.E$, RATING$.E$, POSITION.E$, and LEVEL.E$. For all organizations the values for INS.DATE$, COMP.DATE$, NME$, COMPONENT.NUM$, and LOCATION$ are read from the file for the workshop being processed.

Since the information in the file will be stored in a different order than when being held temporarily FILENAME$ is reopened and COMP.DATE$, NME$, COMPONENT.NUM$, WK.SHOP.NUM$ and LOCATION$ are written back out to file. The card data for this workshop is read into CARD.ARRAY$.

The cards are then processed one at a time. The number of the school district is read and the attendance of that school district is increased by one. The six evaluations are read and each is added to the corresponding evaluation total for that school district. Each is also added to the corresponding total evaluation for the workshop. The position is read and attendance for the corresponding position is increased by one. The same is done for the level. Finally, the social security number, position, and level are concatenated into one field and moved to the next free position in the appropriate schools elements of the SOC_SEC.ARRAY$. The next space is filled with a "*" marking the end of the list. When all cards have been processed in this manner the social security numbers for each school are sorted into ascending order.
With this processing completed, the user is then asked to enter the type of report he wishes to have produced. His choices are: 1) A List of Social Security Numbers by School  2) Totals of each School Broken Down by Position and Level  3) Both types of reports. Each report type will include an evaluation summary by school and a total workshop attendance and evaluation report. The user must enter a 1, 2, or 3. His response is tested and the appropriate reports are produced.

To produce a List of Social Security Numbers each school's elements of the SOC.SEC.ARRAY# is tested to see if the first element is a "*" signifying no attendees from that school. If not, a heading is printed with the school's name and the social security numbers are printed in four ordered columns across the page. This continues for each school district until all have been tested.

To produce an attendance report by position or level a page header is first printed. The first element of each school district is checked to see if it contains a "*". If not the position or level name and the attendance in POSITION or LEVEL is printed. The attendance is added to the corresponding TOT.POSITION or TOT.LEVEL field. When attendance for all schools are printed the totals for the workshop are printed.

To produce an evaluation summary report a page header is printed followed by one detail line for each school district attending. For each school where the first element of SOC.SEC.ARRAY# is not a "*" each of the six evaluations is averaged and rounded to the nearest tenth. Each evaluation average and the attendance for the school is added to the total
workshop attendance.

To produce the total workshop attendance report each of the six workshop evaluation totals is averaged and each is added to the overall evaluation total. This overall total is averaged. A page header is printed followed by the six evaluation averages, the overall evaluation average and the overall attendance.

The data for this workshop is filed when the reports have been printed. Written to FILENAME$ is the number of school districts, for each of the school districts attending the school district identification number, the six evaluation totals, the attendance and a list of the elements in SOC.SEC.ARRAY$ (social security number, position and level). After all schools have been filed the six workshop evaluation averages, the overall evaluation average and the overall attendance are filed.

The user is allowed to make a separate report choice for each workshop being processed.
PHYSICAL DESCRIPTION

INSVRTVL.EXE
The user must first indicate if he is retrieving data for ESC R II or a special organization. If his choice is a special organization he is asked to enter the name of the organization. In the same manner as when processing data a filename is created from the organization name and the date (SPEC.FILE$). From this file the value for the arrays COMPONENT$, DISTRICT$, PROGRAM$, QUESTION$, RATING$, POSITION$ and LEVEL$ are read. LISTCH$ is read from the DATA statements. LISTCH$ holds the report categories to be chosen. The filenames of the workshop files are read into the array FILEARRAY$.

If ESC R II is chosen the arrays of LISTCH$, COMPONENT$, PROGRAM$, DISTRICT$, QUESTION$, RATING$, POSITION$, and LEVEL$ are filled from DATA STATEMENTS. The YEARFILE$ is opened and all filenames of the workshop files are read into FILEARRAY$. For both choices each FILEARRAY$ is opened and the first six records in each file are read into GENARRAY$.

The user then makes a choice of the report he wants. His first choice is between 1) Year to Date Summary of Evaluation and Attendance and 2) Individual Workshop Report Including a List of Social Security Numbers.

If choice one is made each workshop file is read into SDDATA$. As the workshop's data is read the attendance by position is totaled, the attendance by level is totaled, overall totals of position and levels are also made. The evaluation totals are read from the file into GENARRAY$ and the position and level totals are stored in GENARRAY$.

The user will then choose the basis for this report. The
report can be based on Date, Instructor, Component, Workshop Number, Location or a Grand Total Evaluation and Attendance report. If one of the first five is chosen GENARRAY$ is sorted by the category chosen. These categories are in the first five fields of the array. SDDATA$ is sorted by school district. The user now chooses whether he wants the attendance to be reported by position or level.

Next the user will choose between having all reports in the category or a specific one. If he chooses specific he must enter a valid argument so a search can be made. If all reports are chosen they will be produced in order through the file.

By using a Control variable and a field in both GENARRAY$ and SDDATA$ that indicates the workshop with which each school district is associated, the six evaluations for each school is totaled, TEMPRECORD$, the six evaluations for each category is totaled, TOTALSARRAY, and for the entire year, GTOTALARRAY.

When all school districts have been tested the report for the first Control break is printed on the screen. The user is given the option to have a hard copy printed. If a single report was chosen the user is returned to the category selection menu. If all reports were chosen the next report is produced in the same manner.

If the user choose report type 2) Individual Workshop Report the user must enter a desired workshop. He may enter a specific school or have reports listed for all school districts in the workshop. These reports are generated exactly as when the data was originally entered.
CONCLUSION

The major goals of this system have been reached. The menu driven programs provide the user with a simple to use method of processing in-service workshop evaluation forms. In addition it provides the user with a number of options in both the processing of incoming data and in the retrieval of stored data.

The choice of using BASIC, however, created problems with memory size. I found the BASIC interpreter is only capable of addressing 64k of memory so that no advantage is gained by having a machine with a large memory. This 64k must hold all the program code as well as the data. The BASIC compiler uses memory in a different manner. A segment of 64k is set aside for program code. Data is stored in a different segment which allows the data to grow to fill the entire memory. Therefore, I was able to overcome the memory access problem by compiling the programs. Compilation makes the programs execute much faster which is an advantage in the final version. However, the need to run a compiled version at all times negated the advantages of using an interpreted language during the writing and debugging of the program.

In addition there are many features of Microsoft BASIC that are not supported by the compiler. One of these is the ability to dynamically dimension arrays. In the data retrieval program this would be a great advantage since the size of the array required would vary considerably depending on the amount of data currently stored and the type of report being generated. The possibility of rewriting the data retrieval program to run in
chained sections not requiring compilation still is being considered. The speed of execution remains a prime consideration and this would need to be weighed carefully against the advantages gained by dynamically dimensioned arrays.