



MICROCOPY RESOLUTION TEST CHART
NATIONAL BUREAU OF STANDARDS-1963-A

1.2

NOSC TR 937

AD A139560

NOSC TR 937

Technical Report 937

MICROPROCESSOR DATABASE MANAGEMENT SYSTEMS (DBMS):

Evaluation and comparison of three commercial DBMS

Susan S. Duchene

December 1983

SEARCHED
SERIALIZED
MAR 28 1984
A

Approved for public release; distribution unlimited.

DTIC FILE COPY

NOSC

NAVAL OCEAN SYSTEMS CENTER
San Diego, California 92152

84 03 28 016



NAVAL OCEAN SYSTEMS CENTER SAN DIEGO, CA 92162

AN ACTIVITY OF THE NAVAL MATERIAL COMMAND

J.M. PATTON, CAPT, USN
Commander

R.M. HILLYER
Technical Director

ADMINISTRATIVE INFORMATION

The research discussed in this report was done during the period of July 1982 through September 1983 and was funded by Code 51.

Released by
R. A. Buaas, Head
Computer Technology Division

Under authority of
J. A. Gilbreath, Head
Computer Sciences and
Simulation Department

UNCLASSIFIED

SECURITY CLASSIFICATION OF THIS PAGE (When Data Entered)

REPORT DOCUMENTATION PAGE		READ INSTRUCTIONS BEFORE COMPLETING FORM						
1. REPORT NUMBER NOSC TR 937	2. GOVT ACCESSION NO. AD-1139 560	3. RECIPIENT'S CATALOG NUMBER						
4. TITLE (and Subtitle) MICROPROCESSOR DATABASE MANAGEMENT SYSTEMS (DBMS) Evaluation and Comparison of Three Commercial DBMS		5. TYPE OF REPORT & PERIOD COVERED Final Report July 1982-September 1983						
		6. PERFORMING ORG. REPORT NUMBER 84-097						
7. AUTHOR(s) Susan S. Duchene		8. CONTRACT OR GRANT NUMBER(s)						
9. PERFORMING ORGANIZATION NAME AND ADDRESS Naval Ocean Systems Center San Diego, CA 92152		10. PROGRAM ELEMENT, PROJECT, TASK AREA & WORK UNIT NUMBERS						
11. CONTROLLING OFFICE NAME AND ADDRESS		12. REPORT DATE December 1983						
		13. NUMBER OF PAGES 32						
14. MONITORING AGENCY NAME & ADDRESS (if different from Controlling Office)		15. SECURITY CLASS. (of this report) Unclassified						
		15a. DECLASSIFICATION/DOWNGRADING SCHEDULE						
16. DISTRIBUTION STATEMENT (of this Report) Approved for public release; distribution unlimited.								
17. DISTRIBUTION STATEMENT (of the abstract entered in Block 20, if different from Report)								
18. SUPPLEMENTARY NOTES								
19. KEY WORDS (Continue on reverse side if necessary and identify by block number)								
<table> <tr> <td>dBASE II</td> <td>Microcomputer</td> </tr> <tr> <td>DataBase Management Systems (DBMS)</td> <td>Condor 20</td> </tr> <tr> <td>Control Program for Microcomputers (CP/M)</td> <td>FMS-80</td> </tr> </table>			dBASE II	Microcomputer	DataBase Management Systems (DBMS)	Condor 20	Control Program for Microcomputers (CP/M)	FMS-80
dBASE II	Microcomputer							
DataBase Management Systems (DBMS)	Condor 20							
Control Program for Microcomputers (CP/M)	FMS-80							
20. ABSTRACT (Continue on reverse side if necessary and identify by block number)								
<p>This report compares the relevant characteristics of three selected DataBase Management System (DBMS) software packages hosted on a microcomputer running the Control Program for Microcomputers (CP/M) Operating System. The candidate commercial DBMS programs (dBASE II, Condor Series 20, and FMS-80), each available at a cost of \$1000 or less, were identified and evaluated. They were compared on the basis of being simple to use and able to completely satisfy the requirements of the specified application areas of report automation, personnel record management, bibliography modification and printing, and document inventory. After intensive testing and evaluation, dBASE II was determined to be the most appropriate choice for the applications tested.</p>								

DD FORM 1473
1 JAN 73EDITION OF 1 NOV 65 IS OBSOLETE
S/N 0102-LF-014-6601

UNCLASSIFIED

SECURITY CLASSIFICATION OF THIS PAGE (When Data Entered)



Accession For	
MSA	<input checked="" type="checkbox"/>
MSB	<input type="checkbox"/>
MSA announced	<input type="checkbox"/>
Distribution/	
Availability Codes	
Avail and/or	
Special	
A-1	

EXECUTIVE SUMMARY

INTRODUCTION

This report discusses the relevant characteristics of three selected DataBase Management System (DBMS) software packages hosted on a microcomputer running the Control Program for Microcomputers (CP/M) Operating System. The candidate commercial DBMS programs (dBASE II, Condor Series 20, and FMS-80) were identified, evaluated, and compared to determine their strengths and weaknesses in two originally specified application areas. This report includes a summary of applicable features, characteristics, and evaluation results for each DBMS, report conclusions, and recommendations.

BACKGROUND

With the rise in computer use and technology, many offices have found that productivity can be improved substantially through use of computer data management programs. An entire filing system can often be contained on one floppy disk, thereby saving office filing space. With specialized programs such as DBMS, data can be entered and retrieved with enhanced speed and efficiency. Code 51 department personnel were familiar with their microcomputer's CP/M Operating System and WordStar for file editing and storage. In June 1982, they requested assistance in determining which DBMS would be best for the applications they wished to implement. Initially, two applications were specified; two more eventually were added as the comparison effort progressed.

The two originally specified applications included one to update information and automate reporting for Code 51 projects and another to maintain personnel records. The next two applications were needed promptly to satisfy other Code 51 obligations. One application has a format similar to a bibliography card reference concept; the other was created to aid automation of an efficient document inventory system.

The initial task was to survey the commercial market using available literature and resources to find DBM systems compatible with the CP/M Operating System. Three candidate database management systems, each available at a cost of \$1000 or less, were selected and ordered. When each DBMS arrived, the accompanying documentation was read through at least once and then used for reference as the DBMS was exercised extensively in the applicable areas. It was necessary to reach a level of user proficiency to evaluate the systems' capabilities. They were compared on the basis

of being simple to use and able to completely satisfy the requirements of the given database applications. As each DBMS was being evaluated, its software capability was critiqued. After all three systems had been evaluated, an overall comparison of the three was tabulated. One personnel-month was allotted to examine each DBMS with the first two applications and one additional month for the subsequent comparison and report compilation.

CONCLUSION

After intensive testing and evaluation, dBASE II was determined to be the most appropriate DBMS for the tested applications. dBASE II was basically the easiest DBMS with which to set up the applications. It has the best EDITING functions which allow for insert and replace mode toggle, forward and backward record movement, forward and backward character delete keys, and DELETE record toggle. dBASE II was also the most reliable (bug-free) and flexible of the three DBM systems in the area of command options and operation. Even dBASE II has room for improvement, though Ashton-Tate is still persisting in dBASE II enhancement efforts. The relative ease with which the user may create command files for application development was a major deciding point for dBASE II. This, in addition to its widespread distribution/use, continued package updates, improvements, and system support availability, establishes dBASE II as the best choice overall.

CONTENTS

	PAGE
1. INTRODUCTION	1
1.1 PURPOSE	1
1.2 BACKGROUND	1
1.3 FOUR TESTED APPLICATIONS	1
1.3.1 Project Report Update and Personnel Records	2
1.3.2 Bibliography Cards and Document Inventory	2
2. APPROACH	3
2.1 GENERAL PROCEDURE OUTLINE	3
2.2 CRITERIA FOR SELECTION OF THE THREE DBM SYSTEMS	3
2.2.1 The Benchmark Microcomputer System	3
2.2.2 Microcomputer Database Structures	3
2.3 SELECTION PROCEDURE FOR THE THREE DBM SYSTEMS	4
3. DBMS OVERVIEW AND DESCRIPTIONS	5
3.1 dBASE II EVALUATION and DATABASE DEVELOPMENT STRATEGIES	5
3.2 Condor 20 EVALUATION	8
3.3 FMS-80 EVALUATION	10
4. OBSERVATIONS AND COMPARISON RESULTS	12
4.1 DBMS PROGRAM LANGUAGE CAPABILITIES	12
4.1.1 dBASE II Advanced Development Language (ADL)	12
4.1.2 Condor 20 Programming Capabilities	13
4.1.3 FMS-80 Extended File Management (EFM) Language	13
4.2 RESEMBLANCE BETWEEN Condor 20 and dBASE II	14
4.3 DBMS CATEGORIES FOR FINAL SELECTION	14
5. CONCLUSIONS AND RECOMMENDATIONS	16
5.1 DBASE II	16
5.2 Condor 20	16
5.3 FMS-80	17
5.4 SUGGESTED DBMS FOR THE TESTED APPLICATIONS	17
BIBLIOGRAPHY	18

CONTENTS (Cont.)

	PAGE
APPENDIX A - GENERAL COMMANDS/FEATURES SUPPORTED BY THE THREE DBM SYSTEMS	
dBASE II - List of Commands	19
dBASE II Functions	20
Condor 20 - Summary of Commands	21
FMS-80 Command Summary	22
EFM Language Features	23
LIMITATIONS and CONSTRAINTS of Each DBMS	24
APPENDIX B - SUPPLEMENTAL DATA: THE FOUR APPLICATION DATABASES	
BIBLIOG Screen Format and Database Structure	25
DOCS Screen Format and Database Structure	26
51PEOPLE Screen Format and Database Structure	27
51PROJ - Two Database Structures: 51PAR1, 51PAR2	28
51PROJ Command File for 51report.cmd	28
TABLES	
1. Basic DBMS Specifications	4
2. DBMS Fitness Analysis - Major considerations	15

1. INTRODUCTION

1.1 PURPOSE

This report discusses the relevant characteristics of three selected DataBase Management System (DBMS) software packages hosted on a microcomputer running the Control Program for Microcomputers (CP/M) Operating System. The candidate commercial DBMS programs (dBASE II, Condor Series 20, and FMS-80) were identified, evaluated, and compared to determine their strengths and weaknesses in two originally specified application areas. From these three systems, one is recommended as the most appropriate DBMS package to be used by the Code 51 department office personnel and its management.

This report includes a summary of applicable features, characteristics, and evaluation results for each DBMS, report conclusions, and recommendations.

1.2 BACKGROUND

With the rise in computer use and technology, many offices have found that productivity can be substantially improved through use of computer data management programs. An entire filing system can often be contained on one floppy disk, thereby saving office filing space. With specialized programs such as DBMS, any desired data can be entered and retrieved more rapidly and efficiently. Code 51 Department personnel were already familiar with their microcomputer's CP/M Operating System and WordStar for file editing and storage. In June 1982, they requested help to determine which DBMS would be best for the applications they wished to implement. Initially, two applications were specified; two more were added eventually as the comparison effort progressed.

1.3 FOUR TESTED APPLICATIONS

The two originally specified applications were 51PROJ, to update information and automate reporting for Code 51 projects; and 51PEOPLE, to maintain personnel records. The next two applications were needed promptly to satisfy other Code 51 obligations: the BIBLIOG format is similar to a bibliography card reference concept; DOCS was created to help automate an efficient document inventory system. These last two applications were not officially requested for the comparison, but testing them offered additional support in determining the most appropriate DBMS.

1.3.1 Project Report Update and Personnel Records

51PROJ

The first application involved data for Code 51 projects and their related needs: personnel, management, and cost projections. The information for database input was obtained from the FY 1982 report which had been hand-typed. Approximately the same report format was requested to be automated with the report generation facility on the selected DBMS. Subsequent reports could then be prepared simply by modifying the existing database information where necessary and running the already created report forms.

51PEOPLE

The second application required monitoring personnel data for each Code 51 employee: physical location, college education, promotions, years of service, security clearance level, salary, and other pertinent information. The code also requested the capability to report or list selected sorted data. A screen format, which appears on the terminal as a form to fill in or modify, is a convenient way to view this type of information.

1.3.2 Bibliography Cards and Document Inventory

BIBLIOG

In April 1983, Code 51 needed to update an existing bibliography for an upcoming report. A DBMS was required for this application so that existing DBMS software could be used to conduct queries and obtain selective lists and reports on specified fields or records. A "C" computer language program was developed to compile all the records into a final report for the specified bibliography style format. None of the DBMS report commands could generate the style of report required for this application.

DOCS

In July and August 1983, Code 51 personnel in San Diego and Hawaii asked for assistance in developing DBMS applications for document inventory control. They requested items such as report number, date, author, and description. By this time, it was apparent that a screen format was desirable for data entry, update, and printing. No unique report format was specified, otherwise this application was similar to BIBLIOG.

2. APPROACH

2.1 GENERAL PROCEDURE OUTLINE

The initial task was to survey the commercial market using available literature and resources to find DBM systems compatible with the CP/M Operating System. Three candidate database management systems, each available at a cost of \$1000 or less, were selected and ordered. When each DBMS arrived, the accompanying documentation was read through at least once and then used for reference as the DBMS was exercised extensively in the applicable areas. It was necessary to reach a level of user proficiency to evaluate the systems' capabilities. The systems were compared on the basis of being simple to use and able to completely satisfy the requirements of the given database applications. As each DBMS was evaluated, its software capability was critiqued. After all three DBM systems had been evaluated, an overall comparison of the three systems was tabulated. One personnel-month was allotted for the examination of each DBMS with the first two applications and for the subsequent comparison and report compilation.

2.2 CRITERIA FOR SELECTION OF THE THREE DBM SYSTEMS

2.2.1 The Benchmark Microcomputer System

The existing computer equipment for Code 51 determined the prototype benchmark for this comparison. The testing, therefore, was accomplished on a Zobex Zilog Z-80 series microprocessor with 64K memory hosting the CP/M Version 2.2 Operating System. The terminal and printer types used by the Code were the Zenith Z-19 (Heathkit H-19) terminal and the DIABLO 630 daisy wheel printer. The writer's benchmark system was identical to the one described above except for the printer, an EPSON model MX-80IIIF/T. DBMS printing applications and directives were tested with each printer to assure comparable effects.

2.2.2 Microcomputer Database Structures

The term "structure" refers to the organization of a database system. The simplest system to understand and implement is the "file" type system. The file is the database and all records that can be used will belong to this single file. A "multifile" system allows one or more files to be used simultaneously.

Certain advanced structures are normally associated with software systems classifiable as true DBMS. The three main approaches are hierarchical, network, and

relational. Hierarchical and network systems are both organized on a tree structure concept involving one-to-many relationships. The relational data model was first applied to the DBMS by Dr. E.F. Codd in 1970. This approach represents database information in two dimensional tables made up of rows (tuples or records) and columns (attributes or fields). In the relational data model, no element is owned by any other element and no permanent associations exist between them.

2.3 SELECTION PROCEDURE FOR THE THREE DBM SYSTEMS

In selecting the three DBM systems to examine for this comparison, the author originally concluded that, if possible, they all should use the same structure approach. The comparison then would not discuss only the pros and cons of network or hierarchical versus relational, for example. dBASE II, selected first because of its well-known reputation, and Condor 20, discovered through the DBMS research, are both relational DBM systems. FMS-80 was chosen, though multifile in structure, since a third relational candidate was not discovered. It was selected through reviewing comparison charts and advertisements in computer periodicals and by discussing options with computer store operators, vendors and colleagues. The FMS-80 advertisement sounded appealing and seemed to present the system as a good candidate for the comparison. The capabilities listed for the system appeared similar to those available for the other two DBM systems. The Table 1 shows fundamental information regarding the version of each DBMS.

Table 1. Basic DBMS Specifications

	dBASE II	Condor Series 20	FMS-80
Ordered	8-25-82	9-22-82	9-21-82
Arrived	9-14-82	10-21-82	2-14-83
NOSC Cost	\$700	\$995	\$995
Structure	relational	relational	multifile, ISAM
Produced By	Ashton-Tate Los Angeles, CA 90010	Condor Computer Corp Ann Arbor, MI 48107	Systems Plus Palo Alto, CA 94303
DBMS Sent From	Martian Technologies Spring Valley, CA	Condor Computer Corp Ann Arbor, MI 48107	Computer Image San Diego, CA
Version # + Date(s)	Version 2.3C 22 FEB 1982	Version 2.09 C. 1980, 1981, 1982	Release # 2.20 C. 1978, 1979, 1980

3. DBMS OVERVIEW AND DESCRIPTIONS

Since dBASE II was the first DBMS to arrive for testing, it became a benchmark with which to compare the others when they arrived. Before working with dBASE II, the author had been exposed to database theory and structures on large-scale computers, but had never developed any applications with DBMS software on microcomputers. The three DBMS systems have comparable commands available for developing the given applications, but the manner in which these are executed differ in style and level of difficulty. Experimentation with dBASE II established an introductory view of some standard database development considerations, techniques, and limitations. Beginning with dBASE II, operation of the applicable features offered by the three selected DBMS systems will be described individually, where possible, in the order of their arrival, and then the evaluations will be compared.

3.1 dBASE II EVALUATION and DATABASE DEVELOPMENT STRATEGIES

With dBASE II, as with all the DBMSs, the documentation was the first thing to be considered. The dBASE II documentation, two manuals in one binder, was easy to read and contained many examples to illustrate command usage and special functions. The first manual was divided logically into several sections for users in various stages of database development skill and user experience and was written by a dBASE II user. The second was written by the DBMS system developer and serves as a command reference guide. The comprehensive index for the two manuals was also an invaluable tool for quick reference. An additional section or index reference would be helpful to advise new users how to suspend execution of an erroneous command temporarily or permanently after giving the command. (Pressing the "ESC" key will terminate the execution of a dBASE II ADL command file described later.)

To initiate installation or database development for any of the DBM systems, the operator inserts a floppy diskette containing the DBMS software into the appropriate disk drive and directs the computer to read that drive.

The installation procedure for the dBASE II package involves finding a preconfigured terminal type compatible with the user's terminal. This is a one time procedure as it is for all of the DBM systems. But some DBMS installations are easier and less time-consuming than others. dBASE II was reasonably straight-forward, since one of the given terminal types, Heath 89, matched the ZENITH Z-19, that was used for this comparison.

The user needs only to enter the command "dbase" or "DBASE" to invoke dBASE II from CP/M. Then the user is prompted to enter the current date or a carriage return. The user may type "DBASE DOCS" as an alternate invocation method, e.g., where DOCS.COM is a "canned" file containing a series of dBASE II commands to be executed. In this case, dBASE II immediately executes the file named and skips the current date request.

To develop an application, the user must know all possible information categories desired for data entry and how the data will be used, whether for calculations or character manipulation. Data type must be chosen for this purpose. dBASE II provides three types: Character, Numeric, and Logical. After determining the field type, the user must consider field length and record length. The 51PEOPLE application described earlier is a good example to use to understand a typical database structure. A company may have a "record" for each employee containing information items or "fields" such as name, address, or salary. The field length is the maximum number of bytes or characters allowed for the item or field description. The number of records in this database is determined by the number of employees in the company. A database can then be considered a collection of records with identical field categories and corresponding lengths defined.

Maximum specifications for dBASE II include:

254	characters/field
32	fields/record
1000	characters/record
65535	records/database file

The database developer will probably wish to implement an application so it requires only one database structure to contain all the needed information. For databases with longer records, this may be impossible because of the DBMS's record length constraint. For this reason, a single database structure is not feasible to implement with dBASE II on the 51PROJ application. After reviewing the report data, we found the sum of the maximum number of characters required by each field in the "worst" case created a need for a total of over 1130 characters to define an adequately large database structure. (dBASE I has a maximum number of 1000 characters/record.) In this case, two database structures were required to complete this application for the 51PROJ report. (NOTE: Condor 20's limit is 1024 characters/record so a similar dual structure was feasible. FMS-80's record size limit is 40K bytes so implementation of a single database structure for this application was possible, but not necessary for satisfactory implementation.)

The EDIT and BROWSE commands, used for data entry and update, both accept a subset of standard WordStar control character sequences, including CTRL V, a toggle for insert/replace modes. EDIT views one record at a time with all fields usually visible if the number of fields defined is 20 or less. The BROWSE command can be a useful alternative to the EDIT command, enabling the user to view the corresponding fields for 20 records simultaneously, but it is also more accommodating when the field length is 80 characters or less.

Summary of dBASE II editing functions:
 (^R = press "CTRL" and "r" keys simultaneously)

^R	Positions to previous record
^C	Positions to Next record
^V	Insert/Replace Mode Toggle
^U	Delete current record Toggle
^D	Forward one characters w/o modifying
^Y	Clears current field
^G	Deletes character to right of cursor
RUB or DEL	Deletes character to left of cursor

DELETED records can be RECALLED from deleted status until the PACK command is issued. PACK automatically cleans out records marked for deletion and the database is overwritten.

The 51PROJ application gave an excellent opportunity to test the report generator functions available for each DBMS. For dBASE II, a text editor was used successfully to modify report form (.FRM) files after their initial creation with the REPORT command although this method was not recommended by the developer. This allowed the writer to make minor modifications to an already existing REPORT format file without having to operate the REPORT command to re-enter all the report specifications again.

Initially the programming utilities provided by each system were avoided so that each DBMS could be judged by its capabilities for the non-programmer users. Condor 20 allows the user to develop a screen format without requiring any special programming skills. After working with this capability, the writer applied dBASE II ADL programming to develop screen formats/menus for several of the applications. (The programming utilities provided by each DBMS are examined in the Section 4.)

3.2 Condor 20 EVALUATION

Condor 20 does not refer to its documentation as two manuals, but the structure is similar to that of dBASE II since a large appendix is used for the command reference guide. The Condor 20 manual is clear to read and provides examples, though not to the same extent as dBASE II. All the DBM systems supply a table of contents, but Condor 20 is the only one lacking the additional index.

The configuration for the Zenith Z-19 terminal type was not a pre-installed Condor 20 option so it was necessary to determine the required cursor positioning sequences for the Zenith before completing Condor 20's DBMS installation.

Typing "SUBMIT START" is the standard procedure to invoke Condor 20 from CP/M. In this way, the license number, date, and terminal type are requested each time the user starts up Condor 20. The submit file "START.SUB" originally contains just the three commands: DBMS, DATE, and TERM. A way to simplify this procedure is to edit the file similarly to read as follows:

DBMS 123456 DATE 01/01/01 TERM ZENI	(This file format reminds the user of the license number to type in, then the computer enters a dummy date, and the selected terminal type.)
--	--

Without this modification many users consider it an inconvenience to type this information at each invocation.

Condor 20 offers the greatest variety of field data types:

AN	Alphanumeric	a - z, A - Z, 0 to 9, symbols.
A	Alphabetic	a - z, A - Z, space, " ' ", " . ", " - "
N	Numeric	+/- 2148373647, integer.
\$	Dollar	+/- \$21,483,736.47, 2 decimal places.
J	Julian date	Format mm/dd/yy, for input and output.
R	Required entry	Used with any of the above types.

Condor 20's Julian date type is in a convenient standard format and obtains the correct chronological order when used as a key or index for SORTing. The same chronological results can be obtained in dBASE II only if the field containing date information is defined as a Character field type and date entered in yy/mm/dd format or, if defined as Numeric type, with date entered in yymmdd format. Both of these dBASE II procedures are often undesirable from a human engineering point of view.

Maximum specifications for Condor 20 include:

127	characters/field (for certain data types)
127	fields/record
1024	characters/record
32767	records/database file

Practical file size limitations are determined by the amount of diskette space remaining for the database files and DBMS command restraints. Condor 20, for example, can handle only a maximum file size of 128K bytes for the SORT command though the advertised file size limitation is 32,767 records. With 1024 bytes/record this logically could translate to 32,767K bytes, which is impossible for the floppy diskettes currently available.

The database is defined in two steps:

1.	Create the screen format and data item names file (.FRM)
2.	Create the data item definitions file (.DBF) and the data file (.DAT), and add the database name to the dictionary (DATA.DIC)

The concept of a screen format works well once established and debugged, but Condor 20 commands provided for this have some problems. The number of underscores typed during the screen formatting process, FORMAT, must match the number defined during the data definition procedure, DEFINE. The user is not informed during FORMAT if these lengths are in error. The user may discover this when attempting to ENTER data. Condor 20 was also unreliable during FORMAT in other ways. It does not always return to the beginning of the next line after typing the 80th character on the screen. FORMAT sometimes moves characters to undesired areas after screen refresh or exit from the routine. This makes additional editing within FORMAT necessary to attempt to remove unwanted characters. Also, accessing the database is impossible if the database name has not been entered in the DATA.DIC file, e.g., after being renamed.

New records are appended to the end of the database with the ENTER command. (The Code 51 department office discovered that ENTER did not always display a new form when additional records were requested.) The UPDATE command is used to modify existing records. The user-designed screen format is displayed for both of these commands, but they both offer only the replace editing mode for field data modification. Condor 20 does let the user interactively specify search conditions for DISPLAY or UPDATE to find records satisfying the conditions while maintaining the screen format. Once a record is DELETED (after confirmation) it cannot be accessed, but it still occupies diskette space until the user copies the database to a new name and then overwrites the original database without the DELETED records.

The report generator allows the user to create a page format and to revise it if desired. The user is required to define the purpose for each word or field included in the designed report form. A bug was encountered when attempting to print the report on screen or paper. Condor 20 cleared the screen and typed "BUSY" there, as expected, but then exited to CP/M or completely stopped the system and required turning off the microcomputer to restart.

3.3 FMS-80 EVALUATION

The File Management System (FMS-80) has some powerful capabilities, but applying them easily is another story. It was determined that FMS-80 is too difficult to use for the purposes intended and not competitive with the other two DBM systems. Because of these findings, its capabilities were not tested as extensively as the other two systems. The following comments are included for interested readers.

The documentation for FMS-80 usually gives enough information to operate the commands and menus, but the writing style and printing quality are the least intelligible and least friendly of the three DBM systems. The manual often supplies too many unnecessary details and lacks examples where needed. The manual has an index, unlike Condor 20, but it is not as extensive as that of dBASE II.

System installation for FMS-80 involved testing each ".PRM" parameter file on the MASTER diskette until one was found that would correctly configure the terminal type. Fortunately, "DIRECT.PRM" sufficiently matched the ZENITH type. Then this file was renamed PARM.SYS as instructed and all other ".PRM" files on the working disk were deleted.

A customization process available at any time during development is the ability to modify the "LOCATE.SYS" file. This special file keeps track of all files used with FMS-80 programs. The users' manual states: "Each entry in LOCATE.SYS consists of a three-character extension followed by the letter designating the drive on which the files with that extension are to be found." LOCATE.SYS is also "used to communicate miscellaneous configuration information to FMS-80." Three date formats are available: MM/DD/YY, YY/MM/DD, and DD/MM/YY. But this customization applies only to the system date which is requested at time of invocation of FMS-80, since there is no Julian date type as in Condor 20. Print spacing, page width, page length, and adjustment of other defaults must be controlled by modifying this file. Both dBASE II and Condor 20 are more flexible in their management of print controls since they are handled separately for each individual report by calling the REPORT generator.

Typing "FMS" is the general method for invoking FMS-80. The date can then be

entered in the format shown or a carriage return will keep the date setting as it was last set. Then control passes automatically to the FMS-80 Main Menu. FMS-80 is basically a menu-driven system. A menu appears on the screen allowing the user to select one of the commands described, move to another menu, or exit from FMS-80 back to CP/M. The commands can also be operated separately by name; however, this takes more practice because the command names generally are not as meaningful as their equivalent counterparts in the other two DBM systems.

FMS-80 provides three data types for field definition:

A	Alphanumeric	Any characters allowed
D	Decimal	Only decimal digits
V	Variable-length	Any characters allowed

The Variable-length type is similar to Alphanumeric, but the actual length of the field in each record is determined by a CR LF (carriage return, line feed) character sequence that is automatically placed at the end of the entered data. Only one Variable-type field is allowed for each record and it must be the last field in the record. Even the manual relates: "Furthermore, direct update cannot be performed on a file that contains V fields." The usefulness of this particular field type remained unclear throughout the application implementation and testing.

The following specifications are very generous, but a file size (records/database) limit is not mentioned. Maximum specifications for FMS-80 include:

255	characters/field
999	fields/record
40k	characters/record (bytes)

The data definition process utilizes a type of screen formatting, but the method of specifying cursor position for items is hard to use within the available programs.

FMS-80, like Condor 20, offers only the replace mode to enter or modify data. Forward and backward record positioning is provided, unlike Condor 20, but not as easily as with dBASE II.

The report generation process uses several modules to specify the report format. The report definition is first created using EDITRD (EDIT Report Definition). Next REPORT is called to specify page and line information that is requested on the screen. Then a report specified may be obtained. The type of reports available in FMS-80 are similar to that of the other DBMSs.

4. OBSERVATIONS and COMPARISON RESULTS

4.1 DBMS PROGRAM LANGUAGE CAPABILITIES

The standard commands for each DBMS, especially those commands for data modification, are usually designed with a particular execution style for general use. These commands may have some limitations for users who desire varied forms on which to operate the commands. Extended programming language features supply enhanced flexibility with which to fulfill users' needs for customized DBMS procedures, additional screen formats, and frequently used "canned" command sequences.

One major advantage of Condor 20 over dBASE II is its user developed screen format. This format is automatically applied to the commands UPDATE (edit), ENTER (append new records), and DISPLAY (selective search, print option and display). The DBMS operator can be assured that the data in each record will appear in the designed format when using these three commands. While dBASE II offers the insert mode for data modification and some other superior features, Condor 20's screen format, Julian date field type, and easily operated SORT command held Condor 20 in a close race with dBASE II. A discussion of the programming capabilities available in each DBMS became necessary to help break the apparent stalemate that had developed.

4.1.1 dBASE II Advanced Development Language (ADL).

Condor 20's general concept for data entry via the user formatted screen and a small accompanying menu works well for many applications. While dBASE II offers the EDIT and BROWSE options, another format resembling the type possible with Condor 20 was attempted. It was discovered that similar results, perhaps better than Condor 20, could be achieved through programming with the dBASE II Advanced Development Language (ADL). Although this takes more time and programming knowledge, the product is very satisfactory.

The author's first significant ADL command package was developed for the BIBLIOG application. This package produces a formatted screen with 3 to 4 lines at the bottom reserved for process menu and system messages for the user. The screen format is almost identical to that which was initially developed with the Condor 20 DBMS, and essentially combines Condor 20's ENTER, UPDATE, and DISPLAY commands while adding extra record positioning functions. By entering a letter corresponding to the desired menu selection, the dBASE II user can conduct edit, enter(insert), display, search, delete, print, and record positioning functions for the record (or bibliography card entry) currently appearing on the screen. Programming

and testing for this BIBLIOG application using the dBASE II ADL took two weeks. This includes the time to learn specialized functions with the ADL. Techniques and program ideas learned from this first effort made it possible to complete similar implementations for the next application, 51PEOPLE, in four hours, and the third, DOCS, in two hours. dBASE II provides the ZIP program to simplify the process of developing ADL command files with cursor control. Its operation is similar to Condor 20's FORMAT command concept, but adds the ability to insert vertical or horizontal lines at any given position during its execution. During testing, ZIP was determined to be more foolproof than Condor 20's FORMAT command.

A program is needed to automate the setup of the command files so that it is as transparent to the user as it is in Condor 20 (after the screen format is developed). Even though the screen format works on the terminal screen, when using ADL commands to print the screen as seen, problems occur for fields that wrap-around to the next line (s). But even this situation can be dealt with by assigning a memory string variable for each part of the field to be printed. The actual problem here was deceptive, but the answer was not hard to implement.

4.1.2 Condor 20 Programming Capabilities

Standard Condor 20 commands may be combined in desired sequence with a CP/M compatible editor to produce "canned" routines that may be executed within Condor 20 to save the user the effort of entering the same sequence repetitively. Condor 20 is the only DBMS of the three that does not supply a complete programming capability. Only the IF/ENDIF construct is implemented and no reiterative language capabilities like the DO/WHILE or even GOTO are offered. A message capability lacking cursor control is available.

4.1.3 FMS-80 Extended File Management (EFM) Language

An improved screen layout was attempted through manipulation of the FMS-80 EFM language. The users' manual states: "EFM (Extended File Management) is the part of FMS-80 that provides the most flexibility and also requires the most skill to use." The EFM language has most of the programming language capabilities that are available in dBASE II ADL, but EFM has some major faults and differences. Referring to fields by number (i.e., their relative position in the data definition) is more difficult than referring to them by name as in dBASE II. The programmer must learn an entirely new set of commands because those offered in EFM are neither employed nor accessible for interactive testing in the standard FMS-80 package. Neither does EFM provide the many special manipulation functions available with dBASE II ADL. These differences are tolerable and have advantages over writing in BASIC, but the overall

effect is a programming atmosphere inferior to that of dBASE II.

4.2 RESEMBLANCE BETWEEN Condor 20 and dBASE II

Literature and colleagues were inclined to favor the dBASE II software, but, through comparison with Condor 20, certain dBASE II deficiencies have become apparent. However, these did not disqualify the overall superiority of dBASE II. Research during the comparison revealed that similarities occurring between dBASE II and Condor 20 are not accidental. Ashton-Tate bought the rights to the Condor 20 database design several years ago and used it to develop the version of dBASE II available today. Condor 20's emphasis appears to have remained on the non-technical (non-programmer) user. dBASE II removed the standard screen format process and turned to supplying greater manipulation abilities through use of specialized functions and the evolution of the ADL for programming.

4.3 DBMS CATEGORIES FOR FINAL SELECTION

As previously stated, the three DBMS were compared on the basis of being simple to use and able to satisfy the requirements of the given database applications. User friendliness is especially important for non-programmer users who need the system's utilities to be easy to operate. However, even if the DBMS requires programming experience, some organizations may have personnel qualified to tailor applications so that non-programmers can operate them. The accuracy and completeness of the DBMS documentation is essential for the initial learning and the proficient operation of the accompanying system software. Deficiencies in several areas may seriously affect the general operation of the DBMS. Table 2 summarizes many of the positive (+) and negative (-) attributes discovered from this DBMS comparison. Condor 20 and dBASE II, for example, offer a similar feature not supported in FMS-80. The Condor 20 Command Line Analyzer executes any programs with ".COM" extensions, by preceding the command name with a dollar "\$" sign, e.g., \$d b: or \$ws docs.cmd or \$a: pip b:=a:.dat. The Condor DBMS never exits to accomplish this. dBASE II, however, supports the "QUIT TO <.com file list>" command which exits dBASE and then chains to other ".COM" programs. The <.com file list>, like the following example, is executed in sequence by CP/M:

```
. QUIT TO 'DIR B:','PIP A:=DOCS.TXT','DBASE CMDFILE'
```

The dBASE II QUIT command exits the DBMS and takes longer to execute the ".COM" files than the "\$" feature for Condor 20. So Condor 20 received a (++) and dBASE II received a (+) for these corresponding features.

Table 2. DBMS fitness analysis - major considerations.

Category	dBASE II	Condor 30	FMS-80
Documentation	(++)Excellent	(+)Reasonably good	(-)Hard to read and understand.
Set-up	(++), most reliable field type/length definition. Then (+) instantly ready for data entry. (-) Format is not automatically user designed but (+) can use ZIP +/or ADL programming to build desired screen.	(+)Acceptable method for field type/ length definition. Then user must develop screen format that works well (+) once debugged though FORMAT program for developing screen (-) can be unreliable or inconsistent.	(-)Complicated method for field type and length assignment. (-)Also hard to implement a satisfactory screen format with supplied commands.
Ease of Use	(+) EDIT and BROWSE use some WordStar Function keys, ^Y ^R ^C ^G DEL (+) ^V=INSERT/ REPLACE (+)^U =DELETE/ RECALL toggles. PACK option. EDIT lines fields on left side of screen. (+)QUIT TO .COM file	(+)ENTER, UPDATE, and DISPLAY use screen format. UPDATE/ DISPLAY use search conditions. (-) Only Replace mode. (-) When record is DELETED it still uses disk space. (-) (NO PACK) (+) J-Date field type.	(-)Menu can be cumbersome and commands are not named in a memorable fashion either. (-) Only replace mode for data modification.
Reports ->	Best REPORT command.	(++) \$<.com file line> Good method but bugs.	(-) Not Available. Again difficult.
SORT	(-)Sort of fields for date is problem unless YYMMDD format Sorts 1 field per command. Must use cascading sorts in reverse desired order (minor->major).	(+)Properly sorts date in any format. (+)Sorts up to 32 fields with up to 128 total key length in one command.	(-) Not easy to use.
Command files	(++)Wonderful for programmer. Many functions available, cursor control, has 'do while', 'case', 'if', 'goto'... This ADL language lets dBASE get the best of both worlds for those who can learn how to use it.	Used to put together ' canned ' routines, so that user needn't type long commands and sequences. Only offers 'if' construct. Message capability, but no true cursor control.	(+/-) Has EFM programming language. It appears to have some similar functions to dBASE II ADL, but it is again NOT as friendly.

5. CONCLUSIONS AND RECOMMENDATIONS

This study was undertaken to determine which DBMS to use for specific applications. It was not intended to be a comprehensive comparison of the three DBM systems. They were exercised only in the areas necessary to successfully implement the given applications. No numerical accounting operations were required, for example. The major considerations for the final selection involved users' convenience in operating the features provided by each DBMS. Appendix A lists the features supported by each DBMS, regardless of the feature's importance to the final selection or whether it was tested for this comparison.

5.1 dBASE II

dBASE II was basically the easiest DBMS with which to set up the applications (not including extra programming efforts with ZIP and ADL to set up user screen format and other related routines). It has the best EDITing functions which allow for insert and replace mode toggle, forward and backward record movement, forward and backward character delete keys, and DELETE record toggle. dBASE II was also the most reliable (bug-free) and flexible of the three DBMSs in the area of command options and operation. Even dBASE II is not perfect, but Ashton-Tate continues to improve this DBMS product. The relative ease with which the user may create command files by employing ZIP with the ADL was a major deciding point for dBASE II. This, in addition to its widespread distribution/use, continued package updates, improvements, and system support availability, establishes dBASE II as the best choice overall.

5.2 Condor 20

Condor 20 generally has all the necessary database functions. Once its screen format is created, it is a great plus. Sorting on several keys in one command and SORT's proper treatment of Julian date fields are other advantages over dBASE. The report generator has potential to accomplish complicated reports once the bug described earlier is resolved. Experience obtained during Condor 20 testing was very enlightening. It does have noticeable bugs, more so than dBASE II. However, until the dBASE II ADL was applied, Condor 20 was a close contender, though only replace mode was offered for data modification. Condor 20 is a good DBMS for users with little programming experience and limited access to colleagues who could aid them in this area.

5.3 FMS-80

FMS-80 can eventually achieve most of the functions necessary for the given applications, but numerous design and operation deficiencies were encountered in several phases of database definition, modification, development, and report generation. FMS-80 was a major contender as the most powerful and one of the first DBMS available before 1980. The tools for unleashing its power are available, but use of FMS-80 is not recommended because of its many shortcomings in the areas of command operation and user friendliness.

5.4 SUGGESTED DBMS FOR THE TESTED APPLICATIONS

The 51PEOPLE, BIBLIOG, and DOCS applications all work well with the screen format concept available through Condor 20 without programming skills and through dBASE II's ADL. Modifying the larger database fields is less time-consuming when insert mode is provided for editing, but this feature is offered only by dBASE II. Presently, the actual report production for the 51PROJ application is best performed with the dBASE II REPORT generator command. The following example contrasts a section of the BIBLIOG screen format implemented in Condor 20 and dBASE II.

```
Title : .....
..... t2 : .....
.....
Remainder : .....
..... r2 : .....
```

The 127 characters/field limit for Condor 20 made it necessary to divide the Title and Remainder fields into two fields each as shown above. The dBASE II screen format for the BIBLIOG application is identical to Condor 20's except that dBASE II's 256 characters/field limit allows the Title and Remainder fields to be continuous as seen below.

```
Title : .....
.....
Remainder : .....
```

NOTE: All database structures and corresponding screen formats for the four applications are illustrated in Appendix B as implemented in dBASE II, the DBMS recommended overall.

BIBLIOGRAPHY

Barley, Kathryn S., and Driscoll, James R., "A Survey of Data-Base Management Systems for Microcomputers," *BYTE*, Volume 6, Number 11, BYTE Publications Inc., Peterborough, NH, November 1981.

"Data-base management systems are becoming a popular software item. Check this survey for the one that interests you." (This article surveyed 20 DBM systems, including Condor 20 and FMS-80, but not dBASE II. The entire issue focuses on Data Base Management Systems.)

Martin, James, *Computer Data-base Organization*, 2nd Edition, Prentice-Hall Inc., Englewood Cliffs, NJ, 1977.

"A guide to data base design, design objectives; principles of data base software; a survey of alternate logical and physical structures and their trade-offs."

Micro computer Hosted DBMS for C3 Project Management, System Development Corporation (SDC), San Diego, CA, December 31, 1982. (This report was prepared for Naval Ocean Systems Center.)

This SDC report surveys all three DBMS described in this Technical Report and several others that also operate with the CP/M Operating System on a Z-80 based microprocessor.

Stonebreaker, Michael, "Evaluation Criteria for Data Base Management Systems," *UNIX Softalk*, International Data Services, Inc., Sunnyvale, CA, August 1983.

The article refers specifically to minicomputer and multiuser systems, but suggests useful considerations for evaluating DBMS in general. ©.in-0.5i

APPENDIX A

GENERAL COMMANDS/FEATURES SUPPORTED BY THE THREE DBM SYSTEMS

dBASE II - List of Commands

? <exp> [.list]
@ <coord> [SAY <exp> [USING '<picture>']] [GET <var> [PICTURE '<picture>']]
ACCEPT ['<cstring>'] TO <memvar>
APPEND [FROM <file> [SDF] [DELIMITED] [FOR <exp>]] or [BLANK]
BROWSE
CANCEL
CHANGE FIELD <list> [<scope>] [FOR <exp>]
CLEAR [GETS]
CONTINUE
COPY TO <file> [<scope>] [FIELD <list>] [FOR <exp>]
[SDF] [DELIMITED [WITH <delimiter>]] or [STRUCTURE]
COUNT [<scope>] [FOR <exp>] [TO <memvar>]
CREATE [<filename>]
DELETE [<scope>] [FOR <exp>] DELETE FILE <file>
DISPLAY [<scope>] [FOR <exp>] [<exp> list] [OFF]
DISPLAY STRUCTURE DISPLAY MEMORY
DISPLAY FILES [ON <disk drive>] [LIKE <skeleton>]
DO <file>
DO WHILE <exp>
EDIT
EJECT
ELSE
ENDDO
ENDIF
ERASE
FIND <key>
GO or GO TO [RECORD], or [TOP], or [BOTTOM], <>
IF <exp>
INDEX ON <char string expression> TO <index file name>
INPUT ['<cstring>'] TO <memvar>
INSERT [BEFORE], or [BLANK]
JOIN TO <file> FOR <expression> [FIELDS <field list>]
LIST
LOCATE [<scope>] [FOR <exp>]
LOOP
MODIFY STRUCTURE MODIFY COMMAND <command file>
NOTE or *
PACK
QUIT [TO <list of CP/M level commands or .COM files>]
READ
RECALL [<scope>] [FOR <exp>]
RELEASE [<memvar list>], or [ALL]
REMARK
RENAME <current file name> TO <new file name>
REPLACE [<scope>] <field> WITH <exp> [AND <field> WITH <exp>]

dBASE II - List of Commands (Cont.)

REPORT [<scope>] [FORM <form file>] [TO PRINT] [FOR <exp>]
RESTORE
RETURN
SAVE TO <file>
SELECT [PRIMARY or SECONDARY]
SET <parm> [ON], or [OFF] SET ALTERNATE TO <file>
SET DEFAULT TO <drive> SET DATE TO <string>
SET FORMAT TO <format file name> SET HEADING TO <string>
SET INDEX TO <index file> SET MARGIN TO <n>
 other SET options: TALK, PRINT, CONSOLE, SCREEN, STEP, ECHO, DEBUG, BELL ...
SKIP </-> [<n>]
SORT ON <field> TO <file> [ASCENDING], or [DESCENDING]
STORE <exp> TO <memvar>
SUM <field> [<scope>] [TO <memvar list>] [FOR <exp>]
TOTAL TO <file> ON <key variable> [FIELDS <field list>]
UPDATE FORM <file> ON <key variable> [ADD <field list>]
 [REPLACE <field list>]
USE <file> [INDEX <index file name>]
WAIT [TO <memvar>]

The dBASE II ADL consists of all dBASE II Commands and Functions.

dBASE II - Functions

@(<string1>,<string2>)	AT function
*	deleted record function
#	record number function
!(<char string>)	upper case function
\$(<char string>,<start>,<length>)	substring function
<string1>\$<string2>	substring search
CHR(<numeric expression>)	numeric to ASCII
DATE()	system date function
EOF	end-of-file function
FILE(<file>)	existence function
INT(<numeric expression>)	integer function
LEN(<char string>)	length function
STR(<numeric expression>,<width>[,<decimals>])	string function
VAL(<char string>)	value function
TRIM(<char string>)	trims strings
TYPE(<exp>)	supplies data type

Condor 20 - Summary of Commands

COMMAND	DESCRIPTION
&	Signals Command Line Analyzer a .COM CP/M command line follows
ABORT	Stops a RUN command
APPEND	Attach records of one database to another
CHANGE	Change data item values in a database
COMBINE	Attach records of two databases, creating a RESULT database
COMPARE	Compare data item values in databases for (not) matching conditions and create a RESULT database
COMPUTE	Computer data item values in a database
COPY	Copy a database or file
DATE	View or enter date
DBMS	Loads CONDOR SERIES 20 DBMS system
DEFINE	Create new database, redefine a database, describe a database
DELETE	Delete records of a database meeting specified conditions
DESTROY	Eliminate a database or file
DIC	View entries in the data dictionary
DIR	View the list of files in the disk directory
DISPLAY	View selected records of a database
EMPTY	Eliminate all records in a database
ENTER	Insert new records into a database
FORMAT	Create or revise a form or HELP screen
HELP	Assist operator in selecting procedures
INDEX	Create or rebuild a database index
JOIN	Attach data items of two databases by matching data item values
LIST	View records of a database in sequential order
LOG	Log a new disk in the computer
POST	Update data item values in one database with those from another
PRINT	Print records of a database in sequential order
PROJECT	Create a RESULT database from selected data items of a database
READ	Transfer records from an ASCII file to an existing database
RENAME	Change the name of a database or file
REORG	Reorganize structure of a database, adding or deleting data items
REPORT	Create, revise print a report
RESTART	Continue processing of an interrupted command procedure
RUN	Start processing a command procedure having directives
SAVE	Save a RESULT database
SELECT	Select database records meeting specified conditions, creating a RESULT database
SET	Set DBMS operating parameters
SORT	Sort database records by data item values
STAT	View or print statistics of data item values
SYSTEM	Exit from Condor 20 DBMS
TABULATE	Summarize and print specified data items
TERM	Defines system video terminal
TITLE	Print report headings
UPDATE	Change data items in a database meeting specified conditions
WRITE	Transfer records from a database to an ASCII sequential file

FMS-80 - Command Summary

FD = File Definition RD = Report Definition
MD = Menu Definition SD = Screen Definition

APPLY - part of the UPDATE facility that controls the update process beyond the work done by TRANSACT.

APPLY1 - merges a transaction file with an existing data file, or creates a data file from a transaction file.

BATCH - provides the ability to execute a predefined batch command file, or to dynamically create and execute a batch command stream.

DATE - displays the FMS-80 title screen, asks for current date, and validates it. Control then passes to MASTER or FMS-80 Main Menu.

DEFSORT - creates a file with extension .CTL, containing key definitions associated with the FD named.

DO - DO, the EFM interpreter, executes a EFM file compiled w/ PREPARE.

EDITFD - used to create or change a FD (the heart of the FMS-80 system).

EDITMD - editor for creating and changing Custom Menus.

EDITRD - used to create or change a Report Description ().

EDITSD - used to create or change Screen Definition, which allows the screen to be set up for data entry and retrieval.

FMS - The FMS-80 Shell, which is a version of Shell-80 customized for use with FMS-80, performs command line decoding and execution, input redirection, argument management, & batch stream execution.

FMS80A - FMS-80 File Definitions Menu.

FMS80B - FMS-80 File Maintenance Menu. FMS80C is the FMS-80 Reports Menu.

GLOSSARY - provides documentation of the contents of a File Definition (FD).

HELP - provides quick access to a text file containing explanations of selected key phrases.

HITCOUNT - counts the records in the given data file that pass the selection definition provided.

INDEX - builds an FMS-80 Index, which is used to provide access to any record in a file. A secondary index may be used to access a file by fields other than its master keys.

MENU - executes a Custom Menu defined by EDITMD, but cannot be invoked stand-alone, since it needs the Shell for invoking other programs.

PREPARE - compiles EFM language programs that can then be executed with DO.

CPRINT - prints a Control Definition in a form suitable for framing.

MDPRINT - prints a Custom Menu defined by EDITMD.

RDPRINT - produces a printed recap of a RD created by EDITRD.

PRINT - prints a data file according to the FD with which it was created.

SDPRINT - provides a printed recap of a screen definition created by EDITSD.

SPRINT - prints a formatted recap of a SD created by SELECT.

QUERY - allows rapid retrieval and update of data in an indexed file.

REPORT - runs a report which was defined by EDITRD.

SELECT - creates a SD (.SEL) used by PRINT,REPORT,SUBFILE, SORT, or APPLY1.

SORT - puts a file into a different order.

SUBFILE - creates a new file, and an FD to go with it, that contain only the records and fields defined in the SD provided.

TRANSACT - creates a transaction file for use by APPLY1 to update an FMS-80 data file.

EFM Language Features

Data Types: Field, Field Group, Variable, Numeric Literal, Header, Date, Character Literal, Key, Complex

All Statement names are **RESERVED WORDS**.

remove <dest>,<source>; **RECORD MOVE STATEMENT**

LABEL:

goto <label>;

call <label>;

return;

File I/O Statements:

Sequential I/O:

read <file>;

write <file>;

Random I/O:

aread <file> <record>;

aread next <file>;

aread prev <file>;

kread <file>;

KEYED READ

nwrite <file> <record>;

kwrite <file>;

KEYED WRITE

rewrite <file>;

Rewrites the record most recently read

by any random access statement from the designated file.

flush <file>;

Console I/O Statements:

clear; clears screen and positions cursor at HOME

clearln; line is cleared from cursor position to end of line

curse <line>,<column>; positions cursor to designated line/column

display <rval> [<rval> ...];

enter <ival>;

enteru <ival>;

enterr <ival>;

enterur <ival>;

print [<rval> ...] [at(<rval>) [<rval> ...]];

skip [<rval>]; sends one return-linefeed combination to the printer.

eject;

auto; enables automatic reading of File 1;

noauto; disables automatic reading of File 1;

shell <func> <arg>;

end; indicates end of program. Compiler ignores anything following.

Conditional Constructs

if <relexpr>

if <relexpr>

switch <rval>

<group>

<group>

case <rval>:

endif

else

<group>

<group>

[break;]

endif

[more cases]

[default: <group>]

endswitch

LIMITATIONS and CONSTRAINTS of Each DBMS

dBASE II

32 fields per record	64 memory variables
1000 characters per record	254 characters per command line
65535 records per database	5 expressions in SUM command
254 characters per field or string	254 characters in REPORT header
10 digit accuracy of numeric fields	64 pending GETS
1.8×10^{63} approx = largest number	100 characters in index key
1.0×10^{-63} approx = smallest number	16 files open at one time

Condor Series 20

127 fields per record	127 characters per command line
1024 characters per record	8 Data Items in an INDEX key
32767 records per database	32 Data Items in a SORT key
127 characters per field	127 bytes combined length of Key
1 to 10 digits for Numeric or \$ data	Data Items for INDEX or SORT
*/-2148373647 = Numeric (integer) range	
*/- \$21,483,736.47 = Dollar (\$) range	128K bytes max file size for SORT

FMS-80

999 fields per record	
40K characters per record	250 bytes combined length of Key
32767 records per database	Data Items for SORT
255 characters per field	
1 to 40 digits for "Decimal" data	
$10^{40} - 1$ (40 nines) = max value that can be added or subtracted	
$2^{32} - 1$ = max value that can be multiplied or divided	

APPENDIX B

SUPPLEMENTAL DATA: THE FOUR APPLICATION DATABASES (Implemented with dBASE II)

BIBLIOG Screen Format and Database Structure

```
.....  
SectionNo: :— • ANNOTATED BIBLIOGRAPHY of PUBLICATIONS • (dBASE II)  
              • FROM THE US NAVY'S MARINE MAMMAL PROGRAM •  
Author1 : _____  
Others : _____  
_____ yr:article : _____  
  
Title : _____  
_____  
_____  
Remainder : _____  
_____  
_____  
  
s1 : _____  
s2 : _____  
s3 : _____  
s4 : _____  
s5 : _____  
s6 : _____  
CompressFlag :—  
.....
```

```
STRUCTURE FOR FILE: BIBLIOG.DBF  
NUMBER OF RECORDS: 00026  
DATE OF LAST UPDATE: 00/00/00  
PRIMARY USE DATABASE  
FLD      NAME      TYPE WIDTH  DEC  
001     SECTIONNO  C      002  
002     AUTHOR1    C      030  
003     OTHERS     C      110  
004     YR:ARTICLE  C      005  
005     TITLE      C      215  
006     REMAINDER  C      210  
007     S1         C      075  
008     S2         C      075  
009     S3         C      075  
010     S4         C      075  
011     S5         C      075  
012     S6         C      051  
013     COMPRESFLG  C      001  
.. TOTAL ..          01000
```


51PEOPLE Screen Format and Database Structure

```

-----
Code 51 Personnel Information System
-----
Code :_____ | Employee Record | Location :____
-----+-----
Name (Last) :_____ (First) :_____ (MI) :__
Title :_____ GradeLevel :_____ Salary:yr :_____
Salary:hr :_____
Birthdate :_____ Vet:Pref points :__ ServCompDat :_____
Tenure:Grp :_____ FLSAstatus :_____ NPstatus :__ Clearance :__
Degree1 :_____
Degree2 :_____
Degree3 :_____
Lstssp:sa :_____ MajAwards :_____ LstPromDat :_____
LstPerfPt :__ LstPerfCOL :__
-----
    
```

STRUCTURE FOR FILE: 51PEOPLE.DBF
 NUMBER OF RECORDS: 00007
 DATE OF LAST UPDATE: 00/00/00
 PRIMARY USE DATABASE

FLD	NAME	TYPE	WIDTH	DEC	FLD	NAME	TYPE	WIDTH
001	LAST:NAME	C	020					
002	FIRST:NAME	C	016					
003	MI	C	001					
004	CODE	C	005					
005	BIRTHDATE	C	008					
006	VET:PREF	C	001					
007	SRVCOMPDAT	C	008					
008	TENURE:GRP	C	006					
009	FLSASTATUS	C	010					
010	LOCATION	C	003		020	LSTSSP:SA	C	006
011	NPSTATUS	C	001		021	MAJAWARDS	C	010
012	CLEARANCE	C	001		022	LSTPRONDAT	C	006
013	DEGREE1	C	060		023	LTPERFPT	N	001
014	DEGREE2	C	060		024	LTPERFCOL	N	001
015	DEGREES	C	060		025	BDATE	N	006
016	TITLE	C	023		026	SRVDATE	N	006
017	GRADELEVEL	C	006		027	LSSPDATNO	N	006
018	SALARY:YR	N	005		028	LPRONDATNO	N	006
019	SALARY:HR	N	005	002	** TOTAL **			00352

51PROJ - Two Database Structures: 51PAR1, 51PAR2

STRUCTURE FOR FILE: 51PAR1.DBF
 NUMBER OF RECORDS: 00011
 DATE OF LAST UPDATE: 00/00/00
 PRIMARY USE DATABASE

FLD	NAME	TYPE	WIDTH
001	RESRCHTITL	N	001
002	RESRCHAREA	C	036
003	OPNAV:SPON	C	009
004	CONTR:NAME	C	040
005	CONTR:ROLE	C	045
006	FY82:FUND	C	008
007	CONTR:FLAG	N	001
008	PROGMG:CODE	C	025
009	FOTOVIEWNO	C	018
010	XTRA:NEEDS	C	050
011	ACHIEV:2YR	C	254
012	NOSCINVOLV	C	120
013	BLANK	C	001
** TOTAL **			00808

Report Forms Developed:

51par1p1.frm
 51par1p2.frm
 51par2p3.frm
 51par2p4.frm
 51par1p5.frm
 51par2p6.frm

STRUCTURE FOR FILE: 51PAR2.DBF
 NUMBER OF RECORDS: 00011
 DATE OF LAST UPDATE: 00/00/00
 PRIMARY USE DATABASE

FLD	NAME	TYPE	WIDTH
001	RESRCHAREA	C	033
002	BLANK	C	001
003	NOSCPROJNO	C	018
004	SHORTDESCR	C	200
005	PRDUCTAREA	C	003
006	SYSCOMSPON	C	015
007	FUNDCATAPP	C	040
008	EXPProgCHG	C	020
009	MAJMILESTN	C	145
010	FY81:MY	C	004
011	FY81:K	C	005
012	FY82:MY	C	004
013	FY82:K	C	005
014	FY83:MY	C	004
015	FY83:K	C	005
016	FY84:MY	C	004
017	FY84:K	C	005
018	FY85:MY	C	004
019	FY85:K	C	005
020	FY86:MY	C	004
021	FY86:K	C	005
** TOTAL **			00530

The following "canned" Command File, "51report.cmd" will execute the six 51PROJ report pages sequentially when the user types:
 do 51report

```

** 51REPORT.CMD
SET TALK OFF
use 51par1
report form 51par1p1 to print plain
report form 51par1p2 to print plain
use 51par2
report form 51par2p3 to print plain
report form 51par2p4 to print plain
use 51par1
report form 51par1p5 to print plain
use 51par2
report form 51par2p6 to print plain
return
    
```

END

FILMED

4-84

DTIC