ESD-TR-75-301, Volume III

MTR-3067, Volume III

ON-LINE VEHICLE MAINTENANCE DATA MANAGEMENT: MODEL SYSTEM SOFTWARE AND DATA BASE DOCUMENTATION

DECEMBER 1975

Prepared for

DEPUTY FOR COMMAND AND MANAGEMENT SYSTEMS
ELECTRONIC SYSTEMS DIVISION
AIR FORCE SYSTEMS COMMAND
UNITED STATES AIR FORCE
Hanscom Air Force Base, Bedford, Massachusetts

Project No. 522E
Prepared by
THE MITRE CORPORATION
Bedford, Massachusetts
Contract No. F19628-75-C-0001

REVIEW AND APPROVAL

This technical report has been reviewed and is approved for publication.

Alphonso J. Obuchowski Jr.

ALPHONSO J. OBUCHOWSKI, JR., CAPT
Project Engineer

P. R. Veckery

PETER R. VECKERY, GS-14
Project Officer

Frank J. Emma

FRANK J. EMMA, Colonel, USAF
Director, Information Systems
Technology Applications Office
Deputy for Command and Management Systems

UNCLASSIFIED

SECURITY CLASSIFICATION OF THIS PAGE *(When Data Entered)*

REPORT DOCUMENTATION PAGE		READ INSTRUCTIONS BEFORE COMPLETING FORM
1. REPORT NUMBER ESD-TR-75-301, Vol. III	2. GOVT ACCESSION NO.	3. RECIPIENT'S CATALOG NUMBER
4. TITLE *(and Subtitle)* ON-LINE VEHICLE MAINTENANCE DATA MANAGEMENT: MODEL SYSTEM SOFTWARE AND DATA BASE DOCUMENTATION		5. TYPE OF REPORT & PERIOD COVERED 6. PERFORMING ORG. REPORT NUMBER MTR-3067, Vol. III
7. AUTHOR(s) J. B. Czulada J. M. Bergstrom		8. CONTRACT OR GRANT NUMBER(s) F19628-75-C-0001
9. PERFORMING ORGANIZATION NAME AND ADDRESS The MITRE Corporation Box 208 Bedford, MA 01730		10. PROGRAM ELEMENT, PROJECT, TASK AREA & WORK UNIT NUMBERS Project No. 522E
11. CONTROLLING OFFICE NAME AND ADDRESS Deputy for Command and Management Systems Electronic Systems Division, AFSC Hanscom Air Force Base, Bedford, MA 01731		12. REPORT DATE DECEMBER 1975 13. NUMBER OF PAGES 176
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)*

VEHICLE MAINTENANCE DATA MANAGEMENT
VIMS

20. ABSTRACT *(Continue on reverse side if necessary and identify by block number)*

Air Force vehicle maintenace data management is supported by the Vehicle Integrated Management System (VIMS), a standard base-level batch data system. Functions which could profit from on-line access to VIMS were identified and incorporated into an on-line system model in the Data Handling Applications Center. Teams of vehicle maintenance specialists were brought to the Center to evaluate the model. Their comments, together with the model specification, form a requirements and design base for an operational prototype system.

DD FORM 1 JAN 73 1473 EDITION OF 1 NOV 65 IS OBSOLETE

UNCLASSIFIED

20. ABSTRACT (concluded)

Volume II presents the model specification and test results, Volume III documents model software and data base, Volume IV presents prototype development guidelines, and Volume I summarizes these same topics.

TABLE OF CONTENTS

LIST OF CHARTS

LIST OF DIAGRAMS

LIST OF DIAGRAMS (Concluded)

LIST OF FIGURES

LIST OF TABLES

SECTION I

INTRODUCTION

The VIMS development model was implemented in the ESD/MITRE Data Handling Applications Center as a key element leading to an AFDSDC decision on the implementation of an on-line VIMS prototype. The primary objective of this document is to report the work done in creating the model. The secondary objective is to preserve the information required to resurrect, modify, or reconstruct the model and to instruct the user on its operation.

The model simulates the VIMS on-line data management system postulated in ESD-TR-75-1, "Air Force Vehicle Integrated Management System (VIMS) Data Handling Study", and detailed in Volume II of this ESD-TR. The intent was to obtain user feedback on the proposed on-line VIMS design as expediently as possible. Therefore, the model was designed to look realistic to the user, but provides little data processing support beyond that necessary to maintain the appearance of a functioning data management system. Because it is not an operational prototype, any attempts to extend the model design to an operational system should be avoided.

The model supports CRT-printer combinations for the functional areas of Reports and Analysis, Workload Control and Materiel Control, processing all major transactions described in the on-line specifications of Volume II of this ESD-TR. The functions supported are:

Reports and Analysis
- Time card processing
- Fuel-oil issue slip processing
- General file update and edit

Workload Control
- Work order processing
- Deferred maintenance processing
- Monitoring of scheduled maintenance

Materiel Control
- Back-ordered parts processing
- COPARS processing
- Maintenance of the High Cost Bench Stock Master File
- Monitoring of VDP information

The model's data base consists of only those data elements required to produce a realistic user interface at the transactional level. The data on 100 vehicles maintained by the model, along with the associated work orders, fuel-oil issue slips, time cards, and parts information were selected from the Hanscom AFB maintenance data base. Likewise, the model's High Cost Bench Stock Master File is a subset of the Hanscom listing. Employee names, Social Security Account Numbers, and other data that had to be created were made as realistic as possible.

Section II of this document summarizes the design process followed to create the model. The heart of the document, Section III, consists of a description of the model's hardware, software and file structure. The final section is a summary of the lessons learned and some general conclusions. A guide to the model's generation and operation is included in Appendix I. Appendix II is the user's reference manual. Appendix III consists of the graphic aids used to describe the model's software. The utility programs provided for accomplishing basic tasks are introduced in Appendix IV. Finally, Appendix V contains detailed descriptions of the file formats and data elements.

SECTION II

DESIGN APPROACH SUMMARY

The VIMS model was designed using a combination of bottom-up and top-down techniques. The data base and management programs were designed first. Then the system executive was defined. Finally, the transaction and support programs were created to bridge the gap between the executive and the data base.

To determine which files the model would need, the current batch VIMS files were studied. In addition, current VIMS transactions were investigated to determine what additional files the VIMS model would require to put them on-line. Once a system of files was defined, individual data elements were taken from batch VIMS and included as needed in the model's data base. The size and data type of the selected data elements were defined. File formats described in Appendix V were then developed. When all file formats and data elements were described, the interrelations between files were outlined and all file cross references were reconciled. Finally, the physical file structure was developed and the required file access and maintenance routines were created.

When the data base and management system was complete, VIMS model design began with the system I/O functions described in Appendix IV and the top level programs. The executive program was conceived as a large idler loop which carried out top level transaction control and all system bookkeeping. When the VIMS model became too large for the disk, the executive was subdivided into three independent executives, one for each VIMS functional area. The executives are of identical design and could easily be combined into a single program on a larger machine.

The lower level programs were designed one transaction at a time. Transactions with similar designs shared as many program modules as possible. Within each transaction, design was accomplished in levels. The transaction controller was designed first. Whenever the design encountered a specific task to be performed, a program was added to the second level of design to complete it. Only when the top level design was complete did design progress to the second level of programs. Then, as before, each time a more specific task was defined the required program was added at a lower level of nesting. The nesting process continued until all programs required to process each transaction had been specified. This design results in the classic top-down pyramid structure with all programs more general than those a level below them and only as specific as necessary to accomplish their

individual tasks. When this system of programs had been developed the model´s design was complete.

SECTION III

MODEL DESCRIPTION

HARDWARE DESCRIPTION

The VIMS development model operates on a Data General Nova 800 minicomputer with 32,768 16-bit words of core memory. In addition, the model requires a local CRT, local line-printer or high speed line-printer, disk, tape drive, and teletype console (See Figure 1). Although it is not required by the model, a floppy disk unit is available for fast loading of the operating system. The Real-Time Disk Operating System (RDOS) Revision 2 of May 1973 directly supports the system devices listed in Table I. The local CRT and line printer described in Table II are not supported by RDOS and require MITRE-generated software described in Appendix IV.

The model's primary I/O device, the Delta Data CRT, is a buffered unit which provides storage capacity for up to 3072 alphanumeric and control characters. With a screen size of 2160 characters, the buffer's size is large enough to store multiple page formats. This gives the unit the ability to process multiple page transactions rapidly. The CRT also has considerable local hardware capability greatly easing the software burden. Programs may take advantage of hardware cursor, tab, and screen control. There is a format mode with automatic field and tab control to facilitate the input of formatted data. Another valuable control function, the keyboard enable-disable, allows the model to limit user input at the CRT to those times when the CPU is expecting it. This insures that no spurious characters may be input to the CRT while formats are being transmitted and displayed or when the CPU is processing a previous character. The former protects the integrity of the formats, while the latter prevents display of characters which have not been accepted as valid input.

The CRT controls a Centronics 101 Printer connected to it. Because the printer is driven directly from the CRT memory, the CRT cannot be used for the 25 to 30 seconds while printing takes place. When desired, the model allows the user to direct output to the high speed line printer connected directly to the Nova, reducing printout time to about five seconds.

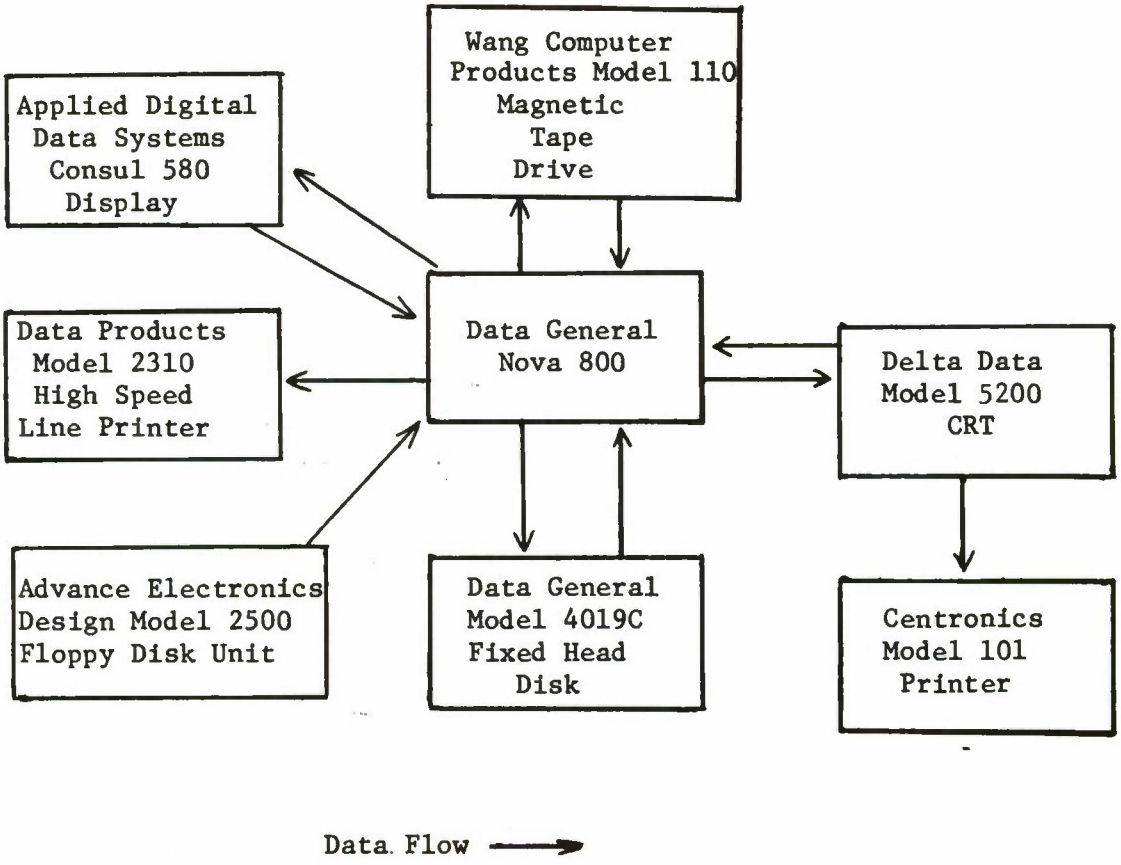


Figure 1. VIMS Development Model Hardware Configuration

Table I

System Devices

Device	Primary Usage
Data General Model 4019C Fixed Head Disk (1/2M bytes, 8.4 milliseconds average access time).	Swapping disk for RDOS and VIMS overlays; storage of VIMS model programs and data files.
Wang Computer Products Model 110 Magnetic Tape Drive (9-track, 45 ips).	Long term storage of RDOS operating system, source code, object code, executable modules, and data for VIMS model; storage for daily test results.
Data Products Model 2310 Line Printer (356 lpm, upper case only, 80 characters per line).	High speed printed output.
Applied Digital Data Systems Consul 580 Display (120 cps) or ASR-33 TTY (10 cps).	Communication with RDOS via a command line interpreter; output device for system messages.

Table II

Non-System Devices

Device	Primary Usage
Delta Data Model 5200 CRT (240 cps).	I/O device for the VIMS functional areas; display of formats and messages; input of data.
Centronics Model 101 Printer (100 cps upper case only, 132 characters per line).	Printed output from CRT screen.
Advanced Electronics Design Model 2500 Floppy Disk Unit (1/4M bytes storage per drive, 400 milliseconds average access time).	Long term storage of RDOS operating system; fast loader for operating system.

APPLICATIONS SOFTWARE DESCRIPTION

The model is programmed in Data General Nova Assembler and Nova Extended Algol and consists of 65 Algol procedures which contain approximately 7500 lines of source code. Development began under RDOS Revision 2 of May 1973. An attempt to change to later revisions was abandoned. They required more core memory for operating system routines, leaving insufficient room for model applications software without some redesign and recoding. The model software is divided into three executable modules, one for each VIMS functional area. Appendix I describes how to generate and operate these modules. Each module is organized top-down using structured programming techniques. The foundation of each module is formed by an executive program and the program utilities discussed in Appendix IV. Each of the executives simulates the remote computer link to a CRT in the corresponding functional area. The executive keeps track of the passage of time, determines transaction processors to be called, and makes any necessary program and overlay calls. For the most part, each transaction is processed by a separate program. Wherever possible, processing logic for a transaction is contained within a single overlay.

Display and maintenance of the CRT formats for the 25 transactions is a major task and time delay for the model. Displaying a large format takes about five seconds which is more than ten times the disk read and processing delay at transaction initialization. However, storing multiple pages in the CRT memory can provide instant redisplay of a format, once the format has been written to the CRT. Wherever possible, reusable parts of formats such as headings and static data fields are retained in the CRT memory, with only variable items being rewritten.

To save main computer memory, to provide flexibility in format design, and to eliminate the need to recompile code when changing formats, large literals comprising display formats are stored separately on the disk instead of being compiled into programs. When a screen format is required it is read into a buffer until used. Each of the VIMS modules requires a different set of formats which are stored in three separate files. Because formats are maintained in standard VIMS files, they may be created, edited, and deleted routinely using VIMS functions described in Appendix III.

Program Descriptions

Model documentation includes Hierarchy Input Processing Output (HIPO) Diagrams and Transfer Vector Charts in place of written program descriptions. These graphics offer a visual description of

the organization and flow of program control, as well as providing a method of locating and analyzing specific programs.

HIPO diagrams included in Appendix III assume two distinct structures. The highest level of HIPO description provides a general overview of program structure indicating organizational relationships, without detailed descriptions of software processes. The tree structure depicted on these diagrams includes one leaf for each general function and demonstrates how functions are subsetted in the model. The lower level HIPO diagrams are descriptions of the actual operations involved in processing the VIMS transactions. Each diagram contains four boxes. The top three describe the inputs, the actual process involved in accomplishing the task specified by the diagram title, and the outputs. Below them, is a description area containing additional information about the specific steps of the process. Implementation details are not given in these descriptions, thus they are free from any specific detail which would bind them to the Algol language or the Nova computer.

The Transfer Vector Charts which also appear in Appendix III show program hierarchy and flow of program control. The charts, like the programs, are organized into the three VIMS functional areas. Each major program in the model is represented by a hexagon and program hierarchy is shown by links between the hexagons.

Source Code Idiosyncrasies

Software development for the model was hampered by a severe Algol compiler problem. Sometimes, Algol programs lost their stack pointers when program flow left a FOR loop. This problem manifested itself in the following three manners. (1) Variables had one set of values within a loop and a different set as soon as the loop was exited. (2) The next statement following the END statement of a FOR loop was not always executed. (3) If the END statements of two nested FOR loops were consecutive statements or were separated only by other END statements, the program crashed or caused a system crash when control left the inner loop.

The last of the three problems occurred every time FOR loops were nested in the manner described. The first two problems occurred only intermittently with no obvious repeatable pattern of occurrence. The vendor suggested switching to a later release of the Algol Compiler and RDOS Revision 3. Preliminary tests indicated that the problems persisted with the new system. In addition, the switch to RDOS Revision 3 would have required recompiling and reworking the programs to fit them into a lesser amount of available core

resulting in a significant time loss. Thus, another way to "fix" the problem was sought.

Observation and experimentation found that the code appears to perform satisfactorily with the following fixes. Placing an Algol SETCURRENT or WRITE statement immediately after the END statement of the loop solves the bad variable-unexecuted statement problem. It is possible that this works because the Algol SETCURRENT and WRITE programs call the Algol SAVE program which restores the stack pointer to the correct value. The nested loop problem may be solved either by inserting intervening statements between the ENDs of the loops or, if that is not possible, by employing the following construction to create the loop.

```
LOOP: body of Loop
      IF BOOL THEN GO TO LOOP;
```

The preceding fixes appear throughout the model, and tampering with them will have unpredictable results.

The Algol disk handler programs created the additional restriction that the length of any file could never be a multiple of 512 bytes, the size of a disk block. If an Algol disk write filled a block exactly, the Algol procedures FILESIZE and POSITION were unable to reposition the file pointer to the end of the file once the pointer had been moved. Instead, the pointer remained at its current position and subsequent writes overwrote the file. To eliminate this problem, the VIMS software checks the file size after each Algol disk write. If the file's length is a multiple of 512, the program appends two bytes to the end of the file before moving the file pointer. These bytes are used only to adjust the size of the file and are overwritten on the next disk write to the end of the file.

DATA FILES

Description

The VIMS development model requires both fixed length record and variable length record files. Each file consists of a header, directories, and data records as depicted in Figures 2 and 3. Records are single indexed by an alphanumeric record key. The read and write functions reference data by logical position within the file as well as by record key. Only complete records are accessible.

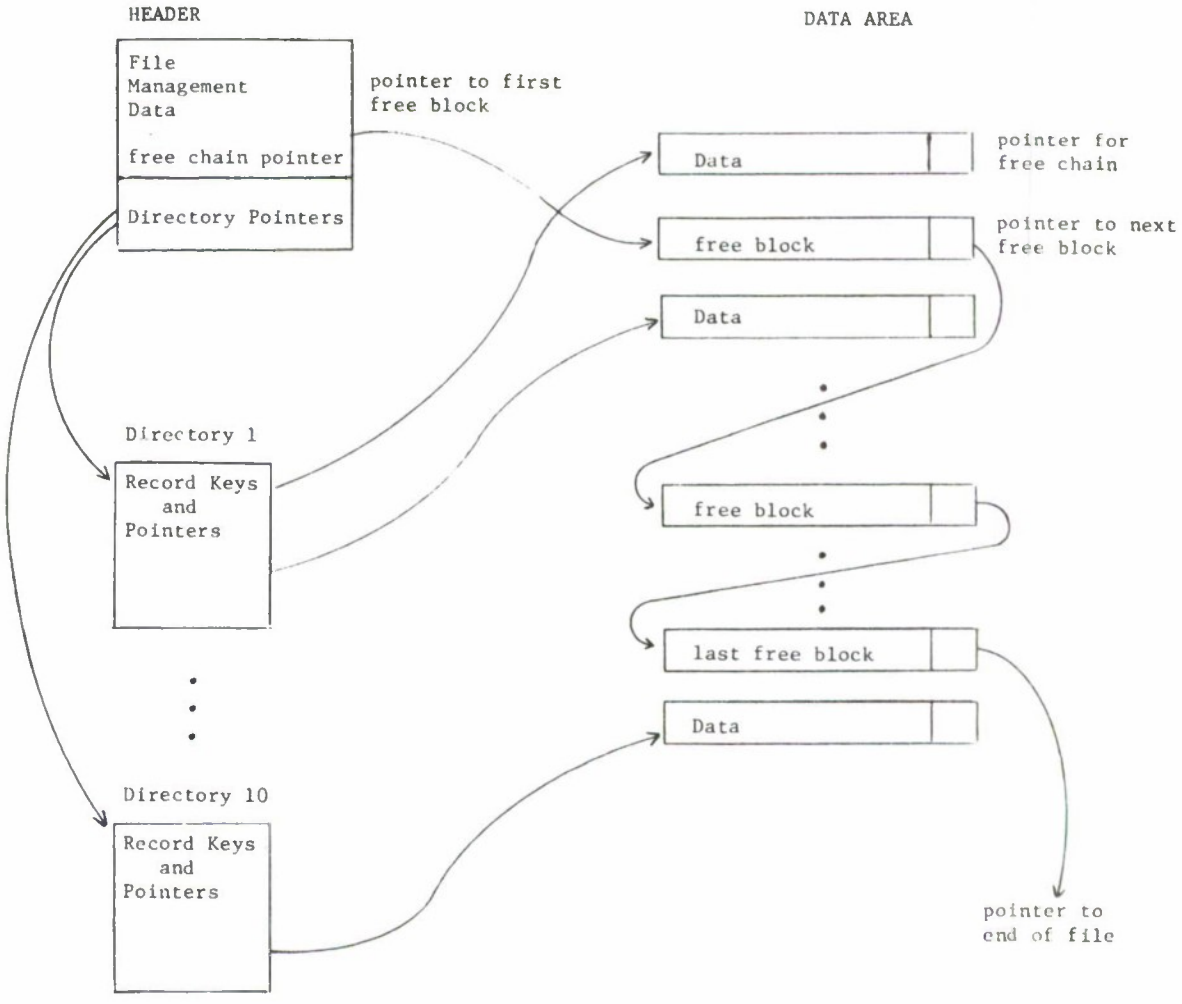


Figure 2. File Structure: Fixed Length Record File

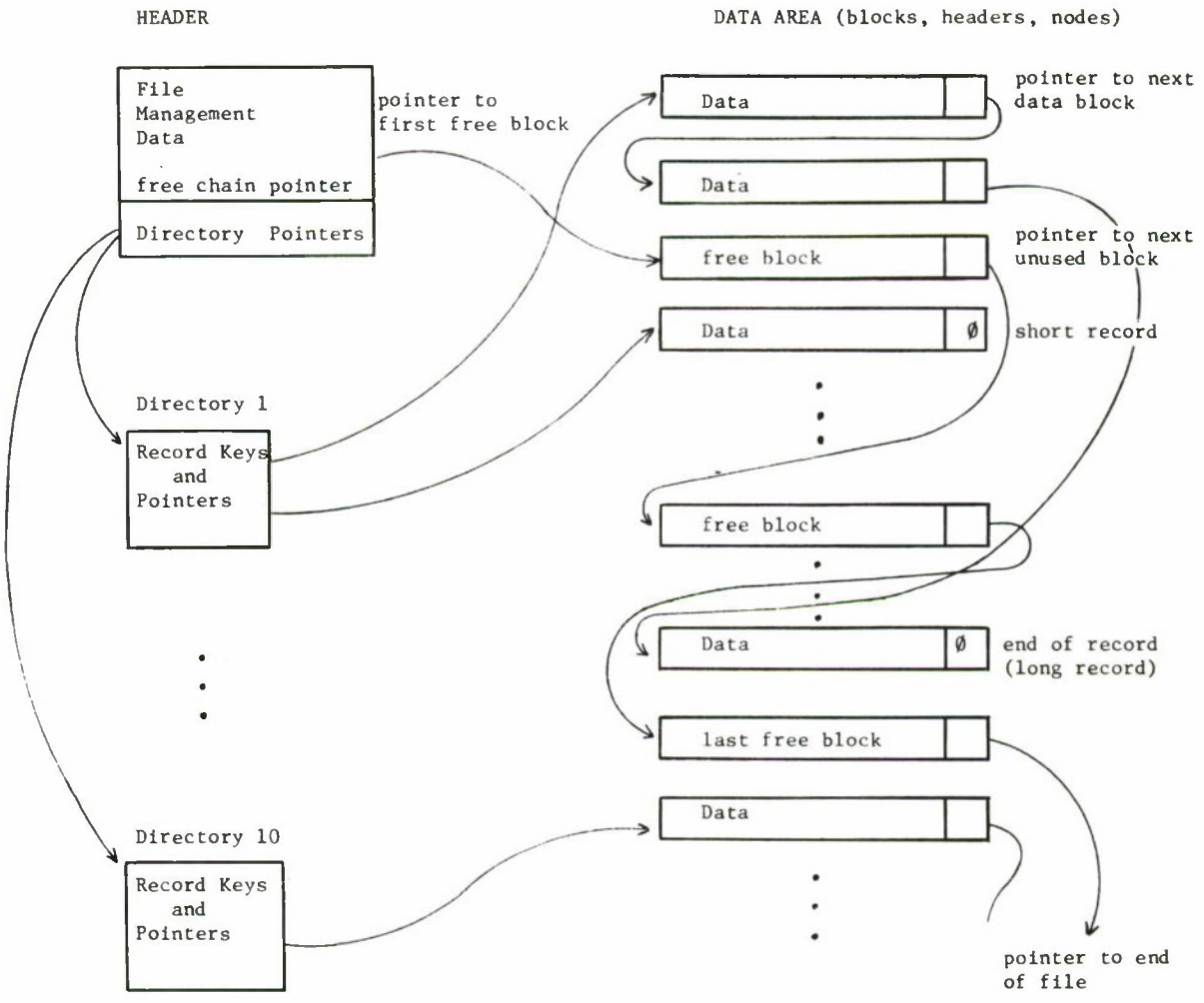


Figure 3. File Structure: Variable Length Record File

Data records are composed of one or more data blocks linked together by integer pointers. To keep the files space-efficent, free chains of unused data blocks are maintained within the file. When data records are edited or deleted, freed data blocks are placed in the free chain. As additional blocks are required they are taken from the free chain. Data blocks are appended to the file only when the free chain is empty.

Header Structure

The file header is created at file initialization time. It contains the file management information required by the read, write, and delete programs. The header consists of the following items:

1. FILE SIZE - the size of the file in bytes (files can be up to 2^{15} bytes in length).

2. MAXIMUM ENTRY - the current number of records in the file.

3. DIRECTORY SIZE - the number of directory elements (record key - two byte pointer) in a directory.

4. ELEMENT SIZE - the size of a directory element.

5. HEADER SIZE - the size of a record header (excludes two byte pointer).

6. NODE SIZE - the size of a variable length record data block (excludes two byte pointer).

7. NEXT NODE - the pointer to the next location where a node may be written.

8. NEXT HEADER - the pointer to the next location where a record header may be written.

9. DIRECTORY POINTERS - a list of ten pointers which point to the (up to ten) directories of the file.

The variable HEADERSIZE is of special importance because it tells the using program which type of file it is reading. The file types are as follows:

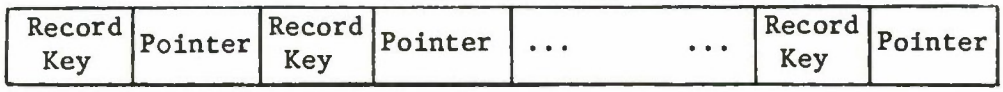
HEADERSIZE < 0	fixed length record file with records whose length is the absolute value of HEADERSIZE.

HEADERSIZE = 0	variable length record file with no record header.
HEADERSIZE > 0	variable length record file with record header of length HEADERSIZE.

Directory Structure

A VIMS file contains up to ten directories. The directories are sets of up to 50 directory elements. Each directory element contains an alphanumeric record key and a two byte pointer. Directories are created as needed and are added at the physical end of the file. Once a directory is created it cannot be deleted. When a record is deleted from the file, its directory element is deleted from the directory moving the following elements one position closer to the front of the file. When a record is written, the directories are searched for the record identifier. If the identifier is found the record is updated; otherwise a new directory element is created to accommodate the new record.

Directory

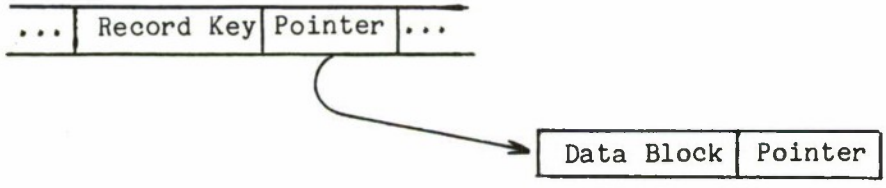


Data Structure

Fixed Length Records

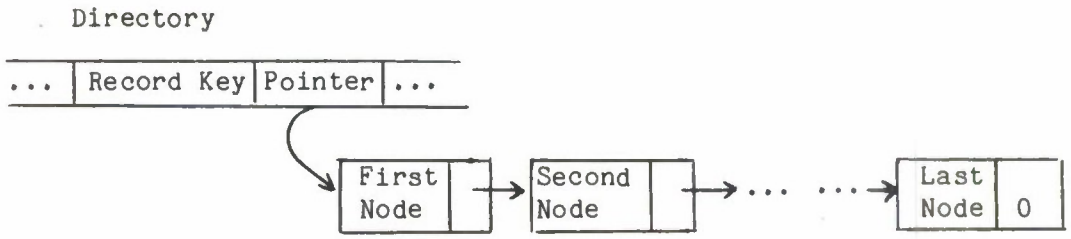
Files which contain static information or data which is stored in a fixed format use fixed length records. A fixed length record may be up to 1000 bytes long. Each record consists of a directory element pointing to a single block of data. The length of the data block is the absolute value of the HEADERSIZE found in the file header. Two bytes following the data block are reserved for a pointer which is used if the data block is ever placed in the free chain of data blocks.

Directory

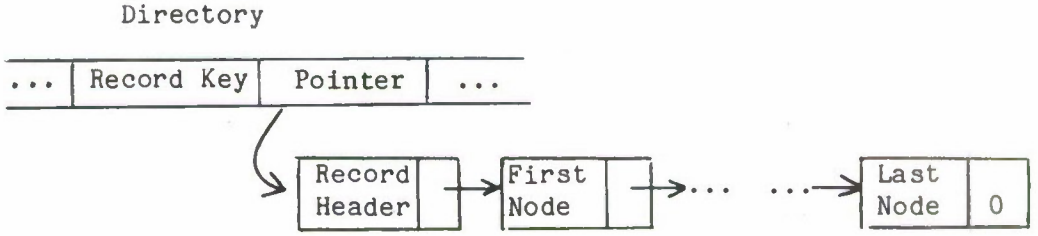


Variable Length Records

Certain VIMS files require records of different lengths which may change as time passes. The simplest type of variable length record consists of a directory element and an associated string of data blocks (nodes). The node string may be any length. Each node has length NODESIZE. Because consecutive nodes need not be contiguous, each node has a two byte pointer to the next node. The pointer following the last node of a record has the value 0. In the event that records or nodes are no longer used, the pointers serve as links in the free chain of data blocks.



More complex variable length record files require record headers which contain static information. These records differ from simple variable length records only because the first block of data is a record header whose length is HEADERSIZE. In general, HEADERSIZE does not equal NODESIZE, the size of the data blocks. Following the header is a two byte pointer to the nodes of the record. This pointer is used as a link in the free chain of headers if the record is deleted. Variable length records with a header may be any length greater than or equal to HEADERSIZE.



SECTION IV

OBSERVATIONS AND LESSONS LEARNED

The most important lesson to be learned from the development of the model is the power of the top-down design and structured programming techniques. Thanks to a top-down design, which stresses modularity and localized control of specific tasks, the model was completed rapidly (less than one and one quarter man-years of effort from initial design to completion) and thanks to structured programming, no programming bugs were discovered during testing by functional personnel.

Block structure and inherent modularity made Nova Algol an excellent medium to implement a top-down design. Algol's procedure-based structure allowed each VIMS transaction to be developed independently, and the model's design allowed each transaction to be added to the model as it was completed. The independence of the modules insured that the entire operating philosophy of a transaction or any module of code could be changed without altering any of the others.

Localizing control of each specific operation, for example reading one line of input from the CRT, in separate program modules kept them small and manageable. Because time and disk space required during compilation are highly dependent on program size, size was of utmost importance on the Nova. Furthermore, small modules allowed a more flexible overlay structure and required less debugging time. In addition, the localization of operations led to more efficient use of each module and ultimately of core and disk space because different programs and programmers shared these modules. For example, all programs shared the system I/O routines and most programs shared a special formatted I/O edit procedure. Finally, localization insured that any change in the way a task was performed had to be made only once and did not alter other programs.

Another supposition reinforced by the implementation of the VIMS model is that the hardware configuration used is a tight constraint on program development and operation. The Data Handling Applications Center <u>had</u> the necessary equipment on hand and one of the ground rules of the project was that the model be <u>bound</u> to use it. Fortunately the primary I/O device, the Delta Data Model 5200 CRT, is a well designed unit with the features discussed in Section III. It is impossible to estimate the amount of code which the hardware features saved the model. On the other hand, the Nova disk's small size was a constant problem during program development. The model is large enough that it had to be divided into three parts

which had to be loaded individually from magnetic tape during the course of the testing. More important, program development required extensive mounting, reading, writing, and dismounting of tapes. For example, any program change required loading source code from magnetic tape and compiling it; dumping the source code with parts of the RDOS system to magnetic tape to save space; loading the remainder of the object code from magnetic tape and creating the executable module; finally, updating all magnetic tape files. With 65 programs in the model it is apparent that such a process was fraught with the danger of losing or destroying data files or blocks of code. A larger disk would have provided for more efficient program development by allowing source and object code to be disk resident at all times.

The composition and small size of the model's development team were important factors in the project's rapid completion. Because the two programmers involved had considerable experience with Nova Algol and with the equipment in the Data Handling Applications Center, no time was required for their indoctrination and training. In addition, both were familiar with the hardware or software problems which might occur and could solve them rapidly. The programmers <u>were</u> the system designers, eliminating the human interface problems which occur when one group designs and another implements. More important, the system was designed with the hardware and software constraints in mind and the programming was accomplished with an understanding of the overall organization of the model. Finally, the model's primary designer had a working knowledge of current batch VIMS which improved the initial design and minimized the amount of work which had to be redone.

To conclude, it must be mentioned that the availability of a computer for extensive hands-on use was almost as important as any other factor in the model's rapid development. Programmers were each allocated two hours of dedicated on-line computer time per day and could always get more time if needed. Since programmers worked interactively with the Nova, turn-around time, which is the major roadblock to rapid software development when using batch systems, was negligible. The computer could be used as a debugging aid to decrease debugging time required, and any changes to the operations or even philosophies of programs could be made rapidly. As the model was being developed new ideas were implemented as soon as they were formulated. More important, during the preliminary testing, the model could be changed rapidly to incorporate suggestions.

APPENDIX I

OPERATOR'S GUIDE

MODEL GENERATION GUIDE

Program Storage

The VIMS development model programs are stored as both source and object code on magnetic tape in the ESD/MITRE Data Handling Applications Center. The source code is primarily Algol, however, some lower level routines are coded in Nova Assembler. The object code associated with each source file is created by the Algol compiler or the Nova assembler of RDOS. Each program consists of a source file PROGRAM. and its object file PROGRAM.RB. The files are organized by VIMS functional area with each area having two tapes assigned to it: a primary tape and a back-up tape. All of the tapes contain the source code, object code, data files, and the special relocatable loader command files required to create executable modules. The primary VIMS tapes also contain copies of the executable modules themselves. A special tape, magnetic tape 6248, contains the VIMS data files, copies of the model's three executable modules, and copies of the stand alone VIMS utilities described in Appendix IV.

Tape and Program Summary

VIMS Functional Area	Tape Number	Program Name	Special Command File Name
Workload Control	6229	VIMS	R
Workload Control Backup	6215		
Materiel Control	6133	VIMSMC	MC
Materiel Control Backup	6247		
Reports and Analysis	6306	VIMSRA	RA
Reports and Analysis Backup	6217		

Program Modification

For programmer convenience and to maintain the integrity of the VIMS storage tapes, any changes to the VIMS source code should be made in the following manner.

1. Load RDOS Revision 2 from magnetic tape 6257 or from the floppy disks.

2. When RDOS has been initialized, load the desired source files from magnetic tape.

   Perform steps 3, 4, 5, and 6 for each source file.

3. Rename the source file to a temporary name with the following RDOS command typed at the console, where ↵ denotes Line Feed and Carriage Return.

   ```
   RENAME FILE TEMP↵
   ```

4. Edit the source file with the RDOS editor. Designate the orginal source file name as the output file name.

5. Delete the temporary file with the following RDOS command typed at the console.

   ```
   DELETE TEMP↵
   ```

6. Compile or assemble the source file by typing one of the following commands on the console.

   ```
   ALGOL/B/E  FILE   $LPT/L↵
   ASM/L/E    FILE   $LPT/L↵
   ```

   This step creates the object file FILE.RB.

7. Update the storage tape in two steps.

   a. Load all untouched source and object files onto the disk.

   b. Dump <u>all</u> files back to the magnetic tape.

   If the untouched files are not loaded onto the disk and then redumped onto the tape they will be lost when the tape is rewritten.

Executable Module Generation

Before the model may be run, the object programs (files with the filename extension .RB) must be resolved into an executable module by the RDOS relocatable loader (reloader). The following procedure creates an executable module.

1. Load RDOS Revision 2 from magnetic tape 6257 or from the floppy disks.

2. When RDOS has been loaded, type the following command string to the console to delete unnecessary programs freeing disk space for the overlay file created by the reloader.

   ```
   CHATR  TCAS.SV  0  FCAS. SV 0  OEDIT. SV 0↵
   DELETE  -.-↵
   ```

   TCAS and FCAS are MITRE generated software.

3. Load the VIMS functional area object files and the special reloader command file from the magnetic tape.

4. To execute the RDOS reloader, type the following command string on the console.

   ```
   @  special command file name @↵
   ```

   The RDOS reloader will create the executable module and the overlay file automatically.

5. Save the executable module PROGRAM.SV and its overlay file PROGRAM.OL on magnetic tape. These two files are considered to be a unit and are referred to as simply "the executable module" throughout the paper.

File Generation

Standard VIMS files must be initialized before they may be used by VIMS programs. The stand-alone program CREAT described in Appendix III may be used to create any necessary files.

The Reports and Analysis module, VIMSRA, of the model contains all user file maintenance capability. To add data records to any file including the Format and Literal File, FORF, use the CREATE option. CREATE insures that a new record can not be established with the same record key as an existing record. Once a record has been CREATEd it may be edited with the EDIT option. EDIT and CREATE require screen formats, which are stored in the FORF file, to input

data. Each format has a two part record key consisting of the filename concatenated with the format page number and may be CREATEd and EDITed in the same manner as any other data record. A DELETE option exists as well to enable the user to clear unused records from any file.

MODEL OPERATING PROCEDURE

The procedure for loading and executing the VIMS model is as follows:

1. Load RDOS Revision 2 from magnetic tape 6257 or from the floppy disks. Be certain to enter the time and date desired for the testing period.

2. When RDOS is loaded, type the following command string to the console to free enough disk space to load the model.

   ```
   CHATR TCAS.SV 0 FCAS.SV 0 OEDIT.SV 0 EDIT.SV 0↵
   DELETE -.-↵
   ```

3. Mount magnetic tape 6248 which contains copies of the three VIMS executable modules and the data files. Load the VIMS functional area module which consists of a NAME.SV and NAME.OL pairing (NAME is VIMS, VIMSMC, or VIMSRA) together with the correct format file FORF. Care must be taken to load FORF from the same tape block as the program module because each of the three modules has its own FORF.

4. Load the data files.

5. Type the program name followed by a carriage return on the console to start the model.

6. Following the model's prompting at the CRT:

   a. Select the high speed line printer if desired.

   b. If the program VIMS is currently loaded, enter all initialization data which includes shift length, hours backlogged, and manpower available for each of the work centers.

When all data has been entered, the model clears the CRT screen. The VIMS model is now waiting for a transaction request. Appendix

II contains a complete description of the user interaction with the VIMS model.

To change the VIMS system time or date at any time during the execution of the model, type "EXIT" followed by the LINE ACCEPT function key at the transaction prompt level. Using the console, set the day and time of day with the RDOS commands SDAY and STOD respectively. Then, restart the model by typing the program name followed by a carriage return on the console.

APPENDIX II

USER'S REFERENCE MANUAL

The purpose of the reference manual is to aid the user in the operation of the VIMS Development Model. It is suggested that the user read Volume II of this ESD-TR to gain a basic understanding of the various transactions before using the Development Model. The reference manual calls out all of the system prompts, error messages, system actions, and user actions for each transaction.

The following is a list of all of the available transactions and the pages on which each transaction is described.

The following is a summary of the function keys and their uses. Each transaction description contains a list of all applicable function keys. Some transactions require additional function keys which are noted in the transaction description.

ATTENTION	System Interrupt, system responds with TRANSACTION ?
BACKPAGE	Display previous page.
DONE	End of transaction.
FWDPAGE	Display next page.
LINE ACCEPT	End of line.
QUIT	Abort transaction, files are not updated.
↑	Up one entry.
↓	Down one entry.

The following is a list of all of the data entry edit function keys and their usage.

IGNORE	Return line to state prior to current data entry.
TAB	Move forward one field.
→	Move forward one formatted position.
←	Move back one formatted position or delete previous character.

A formatted position is denoted by "..." on the CRT screen. Data may be entered in formatted positions only. At the transaction prompt level, "←" deletes the previous character and moves the cursor back one screen position, but during data entry the cursor is simply moved back one formatted position.

The keyboard used in the VIMS Development Model is shown in Figure 4.

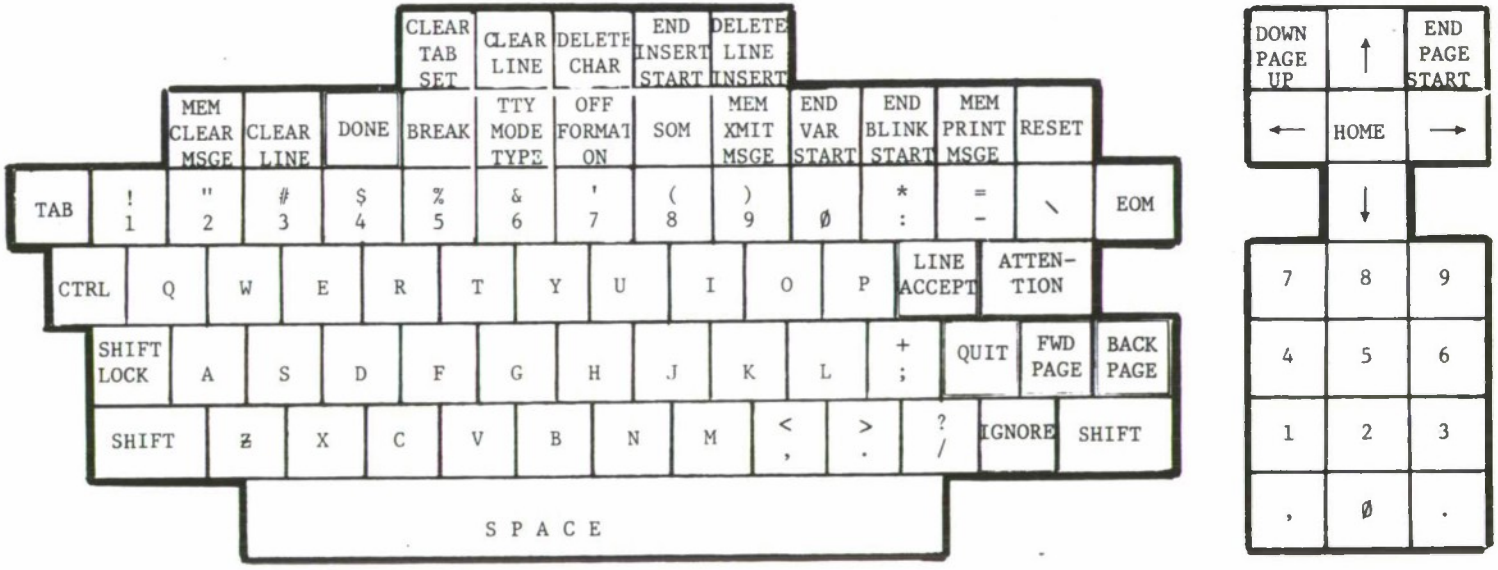


Figure 4. VIMS Model Keyboard

denotes VIMS function keys

OPEN WORK ORDER

1. Press ATTENTION.
   VIMS displays TRANSACTION ?

2. Enter OPEN, OPENA, OPENC, or OPENG. Press LINE ACCEPT.
   VIMS displays VEHICLE ?

3. Enter the vehicle registration number. Press LINE ACCEPT.

   If one of the following messages is displayed, reenter the vehicle registration number or press QUIT.

   UNRECOGNIZED VEHICLE
   VEHICLE ?

   WORK ORDER XXXX ALREADY OPEN THIS VEHICLE
   VEHICLE ?

   WORK ORDER XXXX ALREADY OPEN THIS VEHICLE ON VDP
   VEHICLE ?

   WORK ORDER XXXX SUSPENDED THIS VEHICLE
   VEHICLE ?

4. All scheduled maintenance that is due for the vehicle is displayed as shown in Figure 5.

5. Designate the action to be taken on each job by filling in the Selection Indicator field as follows:

Y	Assign job to work order.
N	Do not assign job to work order.
Space	Do not assign job to work order.

6. Press DONE to terminate the scheduled maintenance portion of the transaction. DONE does not have to be pressed if all Selection Indicators have been filled in.

7. All deferred maintenance for vehicle is displayed as shown in Figure 6.

8. Designate the action to be taken on each job by filling in the Selection Indicator field as follows:

```
                    **** SCHEDULED MAINTENANCE ****

  VEHICLE REGISTRATION NUMBER:  69B01922
       CUMULATIVE MILES/HOURS:  064201

SELECTION  SCHEDULED
INDICATOR  MAINTENANCE DUE        MI/HRS DUE  DATE DUE OVERDUE
---------  --------------------   ----------  -------- -------
   ( )     OIL CHANGE               067111      74348
   ( )     OIL FILTER CHANGE        067111      74348
```

Figure 5. Scheduled Maintenance Display

```
                    **** DEFERRED MAINTENANCE ****

                VEHICLE REG NO: 69B01922

SEL WORK  JOB DEF SYS WRK                                          MATL STD  BIN
IND ORDER NO. CDE CDE CTR JOB DESCRIPTION                          COST HRS  LOC
--- ----- --- --- --- --- ------------------------------------   ---- ---- ---
( )  3897 03  DFP 098 230 REPLACE WATER PUMP & THERMOSTAT
                                                        (74301)     20  4.0 018
```

Figure 6. Deferred Maintenance Display

Y Assign job to work order.

N Do not assign job to work order.

Space Do not assign job to work order.

9. Press DONE to terminate the deferred maintenance portion of the transaction. DONE does not have to be pressed if all Selection Indicators have been filled in.

10. A work order form with all designated jobs is displayed as shown in Figure 7.

11. Enter the Priority, Miles/Hrs, User Phone, and Work Order Type. Press LINE ACCEPT. All of the data entry edit functions, except IGNORE, are available.

12. Fill out the job itemization portion of the work order. The following function keys are available and may be used while positioned in the Primary Action Code field. (Left of /).

A	Assign, add or edit job.
D	Defer job.
K	Cancel job.
↑	Move to previous job.
↓	Move to next job.
FWDPAGE	Display jobs six through ten. Page two is displayed only when the first five jobs have been assigned Primary Action Codes of A or D.
BACKPAGE	Display jobs one through five.
QUIT	Abort transaction.
DONE	End of transction.

The following actions may be performed with the aid of the available function keys.

```
WORK ORDER NO (4227)  VEHICLE REG NO(69801922)  DATE OPENED(74335)  TIME(1343)
MGT CODE(8204)  MAKE/TYPE(P-U CHE)            DATE COMPLETED(     )  TIME(    )
R/O CODE( )  MILEAGE EXCEEDED( )    AGE EXCEEDED( )
PRIORITY(  )  MILES/HRS(      )  USER PHONE(        ) WORK ORDER TYPE( )

 ACTN  JOB SYS      WORK  JOB                                          MATL   STD
 CODE  NO. CODE OPR CTR   DESCRIPTION                                  COST   HRS
--------- ----- --- ----- ---------------------------------------- ------ ------
( /   )01 (361) (O) (220) (OIL CHANGE                             )
       01                 (                                      ) (    ) ( 0.5)
( /   )02 (371) (O) (220) (OIL FILTER CHANGE                       )
       02                 (                                      ) (    ) ( 0.5)
( /OFP)03 (098) (M) (230) (REPLACE WATER PUMP & THERMOSTAT         )
       03                 (                              /BIN 018) ( 20) ( 4.0)
( /   )04 (   ) ( ) (   ) (                                      )
       04                 (                                      ) (    ) (    )
( /   )05 (   ) ( ) (   ) (                                      )
       05                 (                                      ) (    ) (    )
( )PAGE 2

                                  REPAIR ESTIMATES:     LABOR COST($  18)
                                                      INDIRECT COST(   11)
                                                      MATERIAL COST(   20)
            (ONE-TIME REPAIR LIMIT = $  159)        EST TOTAL COST($  49)
```

Figure 7. Work Order Display During Open Transaction

Accept Job	Position to job. Enter A as the Primary Action Code. Press LINE ACCEPT. This action is used only to accept jobs entered on the work order by VIMS, such as scheduled and deferred maintenance.
Add Job	Position to next available job. Enter A as the Primary Action Code. Enter the System Code, O/M Code, Work Center, Job Description, Estimated Material Cost (optional) and Standard Hour Estimate. Press LINE ACCEPT. The job is also assigned to the work order.
Cancel Job	Position to job. Enter K as the Primary Action Code. The job is removed from the work order and all successive jobs are moved forward one job position.
Defer Job	Position to job. Enter D as the Primary Action Code. Enter the appropriate Deferral Code as the Secondary Action Code. (Right of /). Make corrections to job. Press LINE ACCEPT.
Edit Job	Position to job. Enter A as the Primary Action Code. Make corrections to job. Press LINE ACCEPT. The job is also assigned to the work order.

All of the data entry edit functions are available after an A or D has been entered as the Primary Action Code.

13. Press DONE to signal end of work order input. DONE is accepted only when all of the jobs have been assigned a Primary Action Code of A or D.

14. Any possible repeat maintenance is displayed.

15. Press DONE when finished reviewing the Repeat Maintenance display.

16. Workload Control and shop copies of work order are printed.

AMEND WORK ORDER

1. Press ATTENTION.
   VIMS displays TRANSACTION ?

2. Enter AMEND. Press LINE ACCEPT.
   VIMS displays WORK ORDER NUMBER ?

3. Enter the work order number. Press LINE ACCEPT.

   If one of the following messages is displayed, reenter the work order number or press QUIT.

   ```
   VEHICLE ON VDP
   WORK ORDER NUMBER ?

   WORK ORDER SUSPENDED
   WORK ORDER NUMBER ?

   WORK ORDER CLOSED
   WORK ORDER NUMBER ?

   INVALID WORK ORDER NUMBER
   WORK ORDER NUMBER ?
   ```

4. See steps 4 through 9 of Open Work Order.

5. The work order with any newly selected scheduled and deferred maintenance is displayed as shown in Figure 8.

6. See steps 12 through 15 of Open Work Order.

7. VIMS displays NUMBER OF COPIES OF WORK ORDER ?
   Enter the number of copies of the work order to be printed. Press LINE ACCEPT. The desired number of workload control and shop copies of the work order are printed.

```
WORK ORDER NO (4227)  VEHICLE REG NO(69B01922)  DATE OPENED(74335)  TIME(1349)
MGT CODE(B204)  MAKE/TYPE(P-U CHE)            DATE COMPLETED(     )  TIME(    )
R/D CODE( )  MILEAGE EXCEEDED( )    AGE EXCEEDED( )
PRIORITY(Y )  MILES/HRS(067223)  USER PHONE(271-4338) WORK ORDER TYPE(F)

 ACTN  JOB SYS      WORK  JOB                                          MATL   STD
 CODE  NO. CODE OPR CTR   DESCRIPTION                                  COST   HRS
-------- ----- --- ----- ---------------------------------------- ----- ------
(A/   )01 (361) (D) (220) (OIL CHANGE                             )
       01                 (                                      ) (   ) ( 0.5)
(A/   )02 (371) (D) (220) (OIL FILTER CHANGE                       )
       02                 (                                      ) (   ) ( 0.5)
(A/OFP)03 (098) (M) (230) (REPLACE WATER PUMP & THERMOSTAT         )
       03                 (                              /BIN 018) ( 20) ( 4.0)
( /   )04 (   ) ( ) (   ) (                                      )
       04                 (                                      ) (   ) (    )
( /   )05 (   ) ( ) (   ) (                                      )
       05                 (                                      ) (   ) (    )
( )PAGE 2

                                   REPAIR ESTIMATES:     LABOR COST($  18)
                                                       INDIRECT COST(   11)
                                                       MATERIAL COST(   20)
            (ONE-TIME REPAIR LIMIT = $  159)          EST TOTAL COST($  49)
```

Figure 8. Work Order Display During Amend Transaction

RESUME WORK ORDER

1. Press ATTENTION.
   VIMS displays TRANSACTION ?

2. Enter RESUME. Press LINE ACCEPT.
   VIMS displays WORK ORDER NUMBER ?

3. Enter the work order number. Press LINE ACCEPT.

   If one of the following messages if displayed, reenter the work order number or press QUIT.

   ```
   VEHICLE ON VDP
   WORK ORDER NUMBER ?

   WORK ORDER CLOSED
   WORK ORDER NUMBER ?

   INVALID WORK ORDER NUMBER
   WORK ORDER NUMBER ?
   ```

4. The work order is displayed as shown in Figure 9.

5. See steps 12 through 15 of Open Work Order.

6. See step 7 of Amend Work Order.

```
WORK ORDER NO (4227)  VEHICLE REG NO(69B01922)  DATE OPENED(74335)  TIME(1349)
MGT CODE(B204)  MAKE/TYPE(P-U CHE)            DATE COMPLETED(     )  TIME(    )
R/O CODE( )  MILEAGE EXCEEDED( )    AGE EXCEEDED( )
PRIORITY(Y )  MILES/HRS(067223)  USER PHONE(271-4338) WORK ORDER TYPE(F)

 ACTN  JOB SYS      WORK  JOB                                          MATL   STD
 CODE  NO. CODE OPR CTR   DESCRIPTION                                  COST   HRS
-------- ----- --- ----- ---------------------------------------- ----- -------
(A/    )01 (361) (D) (220) (OIL CHANGE                             )
       01                 (                                       ) (   ) ( 0.5)
(A/    )02 (371) (D) (220) (OIL FILTER CHANGE                      )
       02                 (                                       ) (   ) ( 0.5)
(A/OFP)03 (098) (M) (230) (REPLACE WATER PUMP & THERMOSTAT         )
       03                 (                              /BIN 018) ( 20) ( 4.0)
(A/    )04 (011) (M) (230) (REPLACE VALVES                         )
       04                 (                                       ) ( 80) (10.0)
( /    )05 (   ) ( ) (   ) (                                       )
       05                 (                                       ) (   ) (    )
( )PAGE 2
                                    REPAIR ESTIMATES:     LABOR COST($  62)
                                                        INDIRECT COST(  37)
                                                        MATERIAL COST( 100)
            (ONE-TIME REPAIR LIMIT = $  159)           EST TOTAL COST($ 199)
 X : ONE TIME REPAIR LIMIT EXCEEDED
```

Figure 9. Work Order Display During Resume Transaction

CLOSE WORK ORDER

1. Press ATTENTION.
   VIMS displays TRANSACTION ?

2. Enter CLOSE. Press LINE ACCEPT.
   VIMS displays VEHICLE ?

3. Enter the work order number. Press LINE ACCEPT.

   If one of the following messages is displayed, reenter the work order number or press QUIT.

   WORK ORDER CLOSED
   WORK ORDER NUMBER ?

   VEHICLE ON VDP
   WORK ORDER NUMBER ?

   WORK ORDER SUSPENDED
   WORK ORDER NUMBER ?

4. The work order is displayed as shown in Figure 10.

5. Fill out the job itemization portion of the work order. The following function keys are available and may be used while positioned in the Primary Action Code field.

D	Defer job.
P	Add or post job.
↑	Move to previous job.
↓	Move to next job.
FWDPAGE	Display jobs six through ten. Page two is displayed only if there are five or more jobs on the work order.
BACKPAGE	Display jobs one through five.
QUIT	Abort transaction.
DONE	End of transaction.

```
WORK ORDER NO (4227)  VEHICLE REG NO(69801922)  DATE OPENED(74335)  TIME(1349)
MGT CODE(B204)  MAKE/TYPE(P=U CHE)           DATE COMPLETED(74337)  TIME(1106)
R/D CODE( )  MILEAGE EXCEEDED( )    AGE EXCEEDED( )
PRIORITY(Y )  MILES/HRS(067223)  USER PHONE(271-4338) WORK ORDER TYPE(F)

 ACTN  JOB SYS      WORK  JOB                                          MATL   STD
 CODE  NO. CODE OPR CTR   DESCRIPTION                                  COST   HRS
-------- ----- --- ----- ---------------------------------------- ----- ------
(A/    )01 (361) (O) (220) (OIL CHANGE                             )
       01                 (                                        ) (   ) ( 0.5)
(A/    )02 (371) (O) (220) (OIL FILTER CHANGE                       )
       02                 (                                        ) (   ) ( 0.5)
(A/DFP)03 (098) (M) (230) (REPLACE WATER PUMP & THERMOSTAT          )
       03                 (                              /BIN 018) ( 20) ( 4.0)
(A/    )04 (011) (M) (230) (REPLACE VALVES                          )
       04                 (                                        ) ( 80) (10.0)
( /    )05 (   ) ( ) (   ) (                                       )
       05                 (                                        ) (   ) (    )
( )PAGE 2
```

Figure 10. Work Order at Beginning of Close Transaction

The following actions may be performed with the aid of the available function keys.

Add Job	Position to the next available job. Enter P as the Primary Action Code. Enter the System Code, O/M Code, Work Center, Job Description, Estimated Material Cost (optional), and Standard Hour Estimate. Press LINE ACCEPT. The job is also assigned to the work order.
Defer Job	Position to job. Enter D as the Primary Action Code. Enter the deferral code as the Secondary Action Code. Enter corrections to job. Press LINE ACCEPT.
Post Job	Position to job. Enter P as the Primary Action Code. Enter corrections to job. Press LINE ACCEPT.

All of the data entry edit functions are available after a P or D has been entered as the Primary Action Code.

6. Press DONE to signal the end of work order input. DONE is accepted only when all jobs have be assigned a Primary Action Code of P or D.

7. The Quality Control form is displayed at the bottom of work order as shown in Figure 11.

   If Quality Control data is to be input, enter X.

   For each rejected job, enter the Rejected Job Number and Mechanic Number. Press LINE ACCEPT.

8. Press DONE to signal end of Quality Control input.

9. VIMS displays NUMBER OF COPIES ?
   Enter the number of copies of the work order to be printed.
   Press LINE ACCEPT.

10. The desired number of copies of work order are printed.

11. If jobs are deferred during the Close transaction, a copy of all deferred jobs is printed.

```
WORK ORDER NO (4227)  VEHICLE REG NO(69B01922)  DATE OPENED(74335)  TIME(1349)
MGT CODE(B204)  MAKE/TYPE(P=U CHE)            DATE COMPLETED(74337)  TIME(1106)
R/D CODE( )  MILEAGE EXCEEDED( )    AGE EXCEEDED( )
PRIORITY(Y )  MILES/HRS(067223)  USER PHONE(271-4338) WORK ORDER TYPE(F)

 ACTN  JOB SYS      WORK  JOB                                          MATL   STD
 CODE  NO. CODE OPR CTR   DESCRIPTION                                  COST   HRS
--------- ----- --- ----- ---------------------------------------- ----- ------
(P/   )01 (361) (O) (220) (OIL CHANGE                             )
       01                 (                                       ) (   ) ( 0.5)
(P/   )02 (371) (O) (220) (OIL FILTER CHANGE                      )
       02                 (                                       ) (   ) ( 0.5)
(P/OFP)03 (098) (M) (230) (REPLACE WATER PUMP & THERMOSTAT        )
       03                 (                              /BIN 018) ( 20) ( 4.0)
(P/   )04 (011) (M) (230) (REPLACE VALVES                         )
       04                 (                                       ) ( 80) (10.0)
( /   )05 (   ) ( ) (   ) (                                       )
       05                 (                                       ) (   ) (    )
( )PAGE 2

                          REJECTED
( )  CHECK IF QUALITY     JOB NO.        MECHANIC(S)
     CONTROL INSPECTED    --------       -----------
                            (  )         (   ) (   ) (   )
                            (  )         (   ) (   ) (   )
                            (  )         (   ) (   ) (   )
                            (  )         (   ) (   ) (   )
```

Figure 11. Work Order Display at End of Close Transaction

WORK ORDER FILE SUMMARY

1. Press ATTENTION.
   VIMS displays TRANSACTION ?

2. Enter WO/REVIEW. Press LINE ACCEPT.

3. The first page of the Work Order File Summary is displayed as shown in Figure 12.

4. Review the Work Order File Summary using the following function keys.

   FWDPAGE Display next page of summary.

   BACKPAGE Display previous page of summary.

5. Press DONE to signal end of review.

6. VIMS displays ENTER Y FOR COPY.
   For a copy of the Work Order File Summary, enter Y and press LINE ACCEPT, otherwise press LINE ACCEPT.

7. A copy of the Work Order File Summary is printed if requested.

**** WORK ORDER FILE SUMMARY ****

WORK ORDER	STATUS	NO. JOBS	VEHICLE REG NO.	MAKE/ TYPE	DATE/TIME RECEIVED	DATE/TIME RELEASED	W/O TYPE
4211	CLOSED 1 DAY	03	71802417	P-U DOD	74326/0730	74326/1130	
4212	OPEN	01	64801069	T-T IHC	74326/0735	/	
4213	CLOSED 1 DAY	05	70803908	SEO AMC	74326/0800	74326/1150	
4214	CLOSED 1 DAY	01	69801906	P-U CHE	74326/0810	74326/1030	
4215	CLOSED 1 DAY	02	71803978	8-W FOR	74326/0820	74326/1040	
4216	CLOSED 1 DAY	02	62810760	DUMPSTR	74326/0830	74326/1300	
4217	OPEN	05	69000818	LOR CAS	74326/0835	/	
4218	CLOSED 1 DAY	01	68809928	P-U CHE	74326/0845	74326/1430	
4219	CLOSED 1 DAY	01	72C00135	DUM IHC	74326/0900	74326/1630	
4220	VOP	02	71803552	P-U DOD	74326/0905	/	

PAGE 01 OF 04

Figure 12. Work Order File Summary Display

PLACE VEHICLE ON VDP

1. Press ATTENTION.
   VIMS displays TRANSACTION ?

2. Enter VDP/ON. Press LINE ACCEPT.
   VIMS displays WORK ORDER NUMBER ?

3. Enter the work order number. Press LINE ACCEPT.

   If one of the following messages is displayed, reenter the work order number or press QUIT.

   ```
   WORK ORDER CLOSED
   WORK ORDER NUMBER ?

   VEHICLE ON VDP
   WORK ORDER NUMBER ?

   WORK ORDER SUSPENDED
   WORK ORDER NUMBER ?

   INVALID WORK ORDER NUMBER
   WORK ORDER NUMBER ?
   ```

4. VIMS displays DATE/TIME ON VDP ? Enter the date/time vehicle on VDP. Press LINE ACCEPT. If nothing is entered before pressing LINE ACCEPT, the current date and time are supplied by VIMS.

5. The work order is displayed as shown in Figure 13.

6. Fill out the job itemization portion of the work order. The following function keys are available and may be used while positioned in the Primary Action Code field.

S	Suspend job.
↑	Move to previous job.
↓	Move to next job.
FWDPAGE	Display jobs six through ten. Page two is displayed only if there are five or more jobs on work order.
BACKPAGE	Display jobs one through five.
QUIT	Abort transaction.
DONE	End of transaction.

```
WORK ORDER NO (4227)  VEHICLE REG NO(69B01922)  DATE OPENED(74335)  TIME(1349)
MGT CODE(B204)  MAKE/TYPE(P=U CHE)            DATE SUSPENDED(74335)  TIME(1359)
R/D CODE( )  MILEAGE EXCEEDED( )    AGE EXCEEDED( )
PRIORITY(Y )  MILES/HRS(067223)  USER PHONE(271=4338) WORK ORDER TYPE(F)

 ACTN  JOB SYS      WORK  JOB                                          MATL   STD
 CODE  NO. CODE DPR CTR   DESCRIPTION                                  COST   HRS
--------- ----- --- ----- ---------------------------------------- ----- ------
(A/   )01 (361) (O) (220) (OIL CHANGE                             )
      01                  (                                       ) (   ) ( 0.5)
(A/   )02 (371) (O) (220) (OIL FILTER CHANGE                      )
      02                  (                                       ) (   ) ( 0.5)
(A/DFP)03 (B98) (M) (230) (REPLACE WATER PUMP & THERMOSTAT        )
      03                  (                              /BIN B18) ( 20) ( 4.0)
(A/   )04 (B11) (M) (230) (REPLACE VALVES                         )
      04                  (                                       ) ( 80) (10.0)
( /   )05 (   ) ( ) (   ) (                                       )
      05                  (                                       ) (   ) (    )
( )PAGE 2
```

Figure 13. Work Order Display During Vehicle on VDP Transaction

The following action may be performed with the aid of the available function keys.

Suspend Job	Position to job. Enter S as the Primary Action Code. Make changes to job. Press LINE ACCEPT.

7. Press DONE to signal end of transaction.

TAKE VEHICLE OFF VDP

1. Press ATTENTION.
   VIMS displays TRANSACTION ?

2. Enter VDP/OFF. Press LINE ACCEPT.
   VIMS displays WORK ORDER NUMBER ?

3. Enter the work order number. Press LINE ACCEPT.

   If one of the following messages is displayed, reenter the work order number or press QUIT.

   ```
   WORK ORDER CLOSED
   WORK ORDER NUMBER ?

   WORK ORDER SUSPENDED
   WORK ORDER NUMBER ?

   VEHICLE NOT ON VDP
   WORK ORDER NUMBER ?

   INVALID WORK ORDER NUMBER
   WORK ORDER NUMBER ?
   ```

4. VIMS displays DATE/TIME OFF VDP ? Enter the date/time vehicle off VDP. Press LINE ACCEPT. If nothing is entered before pressing LINE ACCEPT, the current date and time are supplied by VIMS.

5. The work order is displayed as shown in Figure 14 with the date and time that the vehicle was placed on VDP in the Date Completed and Time Completed fields.

6. See steps 5 through 7 of Amend Work Order.

```
WORK ORDER NO (4227)  VEHICLE REG NO(69B01922)  DATE OPENED(74335)  TIME(1349)
MGT CODE(B204)  MAKE/TYPE(P-U CHE)            DATE COMPLETED(74335)  TIME(1359)
R/D CODE( )  MILEAGE EXCEEDED( )    AGE EXCEEDED( )
PRIORITY(Y )  MILES/HRS(067223)  USER PHONE(271-4338) WORK ORDER TYPE(F)

 ACTN  JOB SYS      WORK  JOB                                          MATL   STD
 CODE  NO. CODE DPR CTR   DESCRIPTION                                  COST   HRS
-------- ----- --- ----- ---------------------------------------- ----- ------
(A/   )01 (361) (D) (220) (OIL CHANGE                             )
       01                 (                                      ) (   ) ( 0.5)
(A/   )02 (371) (D) (220) (OIL FILTER CHANGE                      )
       02                 (                                      ) (   ) ( 0.5)
(A/DFP)03 (098) (M) (230) (REPLACE WATER PUMP & THERMOSTAT        )
       03                 (                              /BIN 018) ( 20) ( 4.0)
(A/   )04 (011) (M) (230) (REPLACE VALVES                         )
       04                 (                                      ) ( 80) (10.0)
( /   )05 (   ) ( ) (   ) (                                      )
       05                 (                                      ) (   ) (    )
( )PAGE 2

                                  REPAIR ESTIMATES:     LABOR COST($  62)
                                                      INDIRECT COST(   37)
                                                      MATERIAL COST(  100)
            (ONE-TIME REPAIR LIMIT = $  150)        EST TOTAL COST($ 199)
 X : ONE TIME REPAIR LIMIT EXCEEDED
```

Figure 14. Work Order Display During Vehicle Off VDP Transaction

ADD JOB TO DEFERRED MAINTENANCE FILE

1. Press ATTENTION.
   VIMS displays TRANSACTION ?

2. Enter DEFER/ADD. Press LINE ACCEPT.
   VIMS displays VEHICLE ?

3. Enter the vehicle registration number. Press LINE ACCEPT.

   If UNRECOGNIZED VEHICLE is displayed, reenter the vehicle registration number or press QUIT.

4. All deferred jobs for vehicle are displayed as shown in Figure 15. VIMS enters the Date, Dummy Work Order Number XXXX, and Job Number for the next job entry.

5. Enter the Deferral Code, System Code, O/M Code, Work Center, Job Description, Estimated Material Cost, and Standard Hours Estimate. Press LINE ACCEPT. All of the data entry edit functions are available. When LINE ACCEPT is pressed, the Date, Dummy Work Order Number, and Job Number for the next job entry are entered by VIMS. Repeat this step until all new deferred maintenance is added.

   If VDP is entered as the Deferral Code, INVALID DEFER CODE is displayed in the Job Description field and the Deferral Code field is blanked out. The message is cleared when a valid Deferral Code is entered.

6. Press DONE to signal end of transaction.

7. A list of all deferred jobs for vehicle is printed.

8. A list of all new DFP jobs is printed.

**** DEFERRED MAINTENANCE ****

VEHICLE REG NO: 69B01922

DATE	WORK ORDR	JOB NO.	DEF CDE	SYS CDE	D/M	WRK CTR	JOB DESCRIPTION	MATL COST	STD HRS
74301	3697	03	OFP	098	M	230	REPLACE WATER PUMP & THERMOSTAT	20	4.0
74335	XXXX	01	...	...	.	...	.....	...	....
							.....		

Figure 15. Add Job to Deferred Maintenance File Display

CHANGE JOB IN DEFERRED MAINTENANCE FILE

1. Press ATTENTION.
   VIMS displays TRANSACTION ?

2. Enter DEFER/CHANGE. Press LINE ACCEPT.
   VIMS displays VEHICLE ?

3. Enter the vehicle registration number. Press LINE ACCEPT.

   If UNRECOGNIZED VEHICLE is displayed, reenter the vehicle registration number or press QUIT.

4. All deferred jobs for vehicle are displayed as shown in Figure 6. If there is no deferred maintenance on the vehicle, NO DEFERRED MAINTENANCE ON VEHICLE is displayed.

5. Make the desired changes to the jobs. The following function keys are available and may be used while positioned in the Selection Indicator field.

C	Change job.
↑	Position to previous job.
↓	Position to next job.
Space	Position to next job, clear C if present.
QUIT	Abort transaction.
DONE	End of transaction.

   The following action may be performed with the aid of the available function keys.

Change job	Position to job. Enter C. Make changes to job. Press LINE ACCEPT. All of the data entry edit functions are available after C has been entered as the Selection Indicator.

   The user is not allowed to enter into the Deferral Code field if it is VDP. An error message, ***, is displayed in the Deferral Code field if an attempt is made to change a Deferral Code to

VDP. The user is not allowed to enter into the Work Order Number or Job Number fields.

6. Press DONE to signal end of transaction.

DELETE JOB FROM DEFERRED MAINTENANCE FILE

1. Press ATTENTION.
   VIMS displays TRANSACTION ?

2. Enter DEFER/DELETE. Press LINE ACCEPT.
   VIMS displays VEHICLE ?

3. Enter the vehicle registration number. Press LINE ACCEPT.

   If UNRECOGNIZED VEHICLE is displayed, reenter the vehicle registration number or press QUIT.

4. All deferred jobs for vehicle are displayed as shown in Figure 6. If there is no deferred maintenance on the vehicle, NO DEFERRED MAINTENANCE ON VEHICLE is displayed.

5. Indicate jobs to be deleted. The following function keys are available and may be used while positioned in the Selection Indicator field.

   D Delete job.

   ↑ Position to previous job.

   ↓ Position to next job.

   Space Position to next job, clear D if present.

   QUIT Abort transaction.

   DONE End of transaction.

   The following action may be performed with the aid of the available function keys.

   Delete job Position to job. Enter D. The user is not allowed to delete VDP jobs.

6. Press DONE to signal end of transaction.

REVIEW HIGH COST BENCH STOCK MASTER FILE

1. Press ATTENTION.
   VIMS displays TRANSACTION ?

2. Enter HCBS/REVIEW. Press LINE ACCEPT.

3. The first page of the High Cost Bench Stock Master File is displayed as shown in Figure 16.

4. Review the file using the following function keys.

   FWDPAGE Display next page.

   BACKPAGE Display previous page.

5. Press DONE to signal end of transaction.

*** HIGH COST BENCH STOCK FILE ***

ITEM NO.	FSN	UNIT PRICE	EEIC	CHG CODE	DESCRIPTION
001	2610L0378192835	40.28	609	0	TRAMB9.15X15 REG4PTL
002	2610L0383622835	20.59	609	0	TRTRA7.50X16 RIB6PTT
003	2610L0408022835	42.00	609	0	TRAMB 8.90X15 MS8PTL
004	2610001776877	25.23	609	0	TRTRK800-16.56 PRGTL
005	2610001776881	33.11	609	0	TRTRK800X16.5 MS6PTL
006	2610001777249	16.50	609	0	TRPAS F78-14B4 REGTL
007	2610001777250	15.70	609	0	TRPAS F78-14 4PMSTL
008	2610001777253	37.40	609	0	TRPAS H78-14 4PREGTL
009	2610001777254	36.00	609	0	TRPAS H78-14 MS4PTL
010	2610001777255	18.93	609	0	TRREG TREAD 4PH7815
011	2610001777256	26.00	609	0	TRPAS H78X15 MS4PTL
012	2610002043939	46.57	609	0	TRTB825-2010PN REGTT
013	2610002697383	7.00	609	0	TUBE 900-20TR175APCC
014	2610003509970	52.26	609	0	TRTB 750-20 8P RGTT
015	2610004736705-2	18.87	609	0	TRPAS G78-15 4PMSTL
016	2610004736800	40.00	609	0	TRPAS 915X15 REG8PTL
017	2610004897917	10.46	609	0	TRPAS 845-15 8P RGTL
018	2610004897957	14.12	609	0	TRPAS F78-14 4PREGTL
019	2610004897961	15.77	609	0	TRPAS H78-14 4PREGTL
020	2610004897973	18.68	609	0	TRPAS G78-15 4PREGTL
021	2610004897975	34.28	609	0	TRPAS H78X15 REG4PTL

Figure 16. High Cost Bench Stock File Display

ADD ITEM TO HIGH COST BENCH STOCK MASTER FILE

1. Press ATTENTION.
   VIMS displays TRANSACTION ?

2. Enter HCBS/ADD. Press LINE ACCEPT.
   The form shown in Figure 17 is displayed.

3. Enter the FSN, Unit Price, EEIC Code, Charge Code, Description and Number of Q Cards to be generated. Press LINE ACCEPT.

   The item number is assigned by VIMS. The user is not allowed to enter into the Item Number field. A non-alphanumeric character may not be entered as the first character of the entry. All of the data entry edit functions are available.

4. Repeat step 3 until all items have been added to the file.

5. Press DONE to signal end of transaction.

ADD TO
**** HIGH COST BENCH STOCK FILE ****

ITEM NO.	FSN	UNIT PRICE	EEIC	CHG CODE	DESCRIPTION	Q-CARDS
121	.....	.....	....	.	.....	..

Figure 17. Add Item to High Cost Bench Stock Master File

CHANGE ITEM IN HIGH COST BENCH STOCK MASTER FILE

1. Press ATTENTION.
   VIMS displays TRANSACTION ?

2. Enter HCBS/CHANGE. Press LINE ACCEPT.
   VIMS displays ITEM NUMBER OR X(EXIT).

   If the following message is displayed, reenter the item number or press QUIT.

   INVALID ITEM NUMBER
   ITEM NUMBER OR X(EXIT)

3. Enter the item number. Press LINE ACCEPT.

4. The requested item is displayed as shown in Figure 18.

5. Enter the changes to the item. Press DONE or LINE ACCEPT. The user is not allowed to enter into the Item Number field. All of the data entry edit functions are available.

6. Repeat steps 3 through 5 until all changes have been made.

7. Enter X and press LINE ACCEPT to signal end of transaction. DONE may also be used.

*** HIGH COST BENCH STOCK FILE ***

ITEM NO.	FSN	UNIT PRICE	EEIC	CHG CODE	DESCRIPTION
043	26100519454	11.04	609	0	INNERTUBE,PNEUMATIC

Figure 18. Change Item in High Cost Bench Stock File Display

DELETE ITEM FROM HIGH COST BENCH STOCK MASTER FILE

1. Press ATTENTION.
   VIMS displays TRANSACTION ?

2. Enter HCBS/DELETE. Press LINE ACCEPT.
   VIMS displays ITEM NUMBER OR X(EXIT).

3. Enter the item number. Press LINE ACCEPT.

   ITEM XXXX HAS BEEN DELETED is displayed if the item is found and deleted. If the item is not found, ITEM DOES NOT EXIST is displayed. The item numbers of succeeding items in the file will remain unchanged.

4. Repeat step 3 until all desired items are deleted.

5. Enter X and press LINE ACCEPT to signal end of transaction. DONE may also be used.

ISSUE HIGH COST BENCH STOCK

1. Press ATTENTION.
   VIMS displays TRANSACTION ?

2. Enter HCBS/ISSUE. Press LINE ACCEPT.
   VIMS displays the form shown in Figure 19.

3. Enter the Work Order Number, Item Number, Quantity Issued, Date Issued, and Replenishment Quantity. Press LINE ACCEPT.

   A valid Work Order Number and Item Number must be entered before the operator is allowed to enter any additional data. The messages, INVALID WORK ORDER and INVALID ITEM NUMBER, are displayed whenever an invalid Work Order Number or Item Number is entered. The messages are deleted upon correction of the error. A non-alphanumeric character may not be entered as the first character of the entry. All of the data entry edit functions are available.

4. Repeat step 3 for all High Cost Bench Stock to be issued.

5. Press DONE to signal end of transaction.

**** HIGH COST BENCH STOCK ISSUE ****

WORK ORDER NUMBER	ITEM NUMBER	QUANTITY ISSUED	DATE ISSUED	REPLENISHMENT QUANTITY
....	...	..	.....	..
....	...	..	.....	..
....	...	..	.....	..
....	...	..	.....	..
....	...	..	.....	..
....	...	..	.....	..
....	...	..	.....	..
....	...	..	.....	..
....	...	..	.....	..
....	...	..	.....	..
....	...	..	.....	..
....	...	..	.....	..
....	...	..	.....	..
....	...	..	.....	..
....	...	..	.....	..
....	...	..	.....	..

Figure 19. Issue High Cost Bench Stock Display

INPUT COPARS COST DATA

1. Press ATTENTION.
   VIMS displays TRANSACTION ?

2. Enter COPARS. Press LINE ACCEPT.
   VIMS displays DATE ?

3. Enter the date. Press LINE ACCEPT.
   If DATE ? is displayed, reenter a valid date or press QUIT.

4. VIMS displays WORK ORDER ?

5. Enter the work order number.

   If the following message is displayed, reenter the work order number or press QUIT.

   ```
   INVALID WORK ORDER NUMBER
   WORK ORDER ?
   ```

6. The warranty data is displayed as shown in Figure 20. Press DONE when finished reviewing warranty data.

7. The COPARS sales slip format as shown in Figure 21 is displayed.

8. Enter Part Number, Quantity, List Price, Discount Price, BOP Indicator, Warranty Indicator, Days Warranty, Miles Warranty, and Part Description. Press LINE ACCEPT. A non-alphanumeric character may not be entered as the first character of the entry. VIMS enters the cost. All the data entry edit functions are available.

9. Press DONE to signal end of input. VIMS calculates and displays the total cost.

10. Press DONE to signal end of transaction.

****PARTS WARRANTY****

VEHICLE REG NO,169D00818

PART NO.	PART DESCRIPTION	PART INSTALLED (DATE)	(MILES)	WARRANTY PERIOD (DAYS)	(MILES)
	MANIFOLD	74245		090	
	EXHAUST PIPE	74245		090	

Figure 20. Parts Warranty Display

**** COPARS SALES SLIP ENTRY ****

WORK ORDER NO:14217
DATE:74335

PART NO.	QTY	LIST (EACH)	DISC PCNT	BOP IND	WNTY IND	DAYS WNTY	MILES WNTY	PART DESCRIPTION	COST
.....	...	.....	..	.	.	....	.....	.....	.....
.....	...	.....	..	.	.	....	.....	.....	.....
.....	...	.....	..	.	.	....	.....	.....	.....
.....	...	.....	..	.	.	....	.....	.....	.....
.....	...	.....	..	.	.	....	.....	.....	.....
.....	...	.....	..	.	.	....	.....	.....	.....
.....	...	.....	..	.	.	....	.....	.....	.....
.....	...	.....	..	.	.	....	.....	.....	.....
.....	...	.....	..	.	.	....	.....	.....	.....
.....	...	.....	..	.	.	....	.....	.....	.....

Figure 21. COPARS Sales Slip Display

VDP SUMMARY DISPLAY

1. Press ATTENTION.
   VIMS displays TRANSACTION ?

2. Enter VDP/DISPLAY. Press LINE ACCEPT.

   A form showing all vehicles on VDP is displayed as shown in Figure 22.

3. Enter the number of copies to be printed and press LINE ACCEPT. If no copies are desired press DONE or LINE ACCEPT to signal end of transaction.

**** VDP SUMMARY ****

VEHICLE REG NO.	WORK ORDR	JOB NO.	TIME/DATE ON VDP	PART NO. (FSN)/NOMENCLATURE	QTY	SC	DATE DUE	RCVD
69B03139	4175	02	0000/74324	CD-3348/CLUTCH PLATE	1	C	74328	NO
69B03139	4175	02	0000/74324	CH-5473/CLUTCH DISK	1	C	74326	NO
69B03139	4175	02	0000/74324	1697C/RELEASE BEARING	1	C	74328	NO
69B03139	4175	03	0000/74324	XS497/REBUILT STARTER	1	C	74328	NO
70E00965	4201	03	0000/74325	1119513/DELCO REGULATOR	1	C	74326	NO
71B03552	4220	01	0000/74326	KING PINS- 71 DODGE PICKUP	4	C	74327	NO
67C01356	4223	03	0000/74326	DUM IHC- COWL GRILLE	1	C	74333	NO
67C01356	4223	04	0000/74326	3010262/RADIATOR	1	C	74333	NO
67B11815	3645	01	0000/74300	267254C91-IHC/LEAF SPRING	1	C	74310	NO
71B00442	4208	02	0000/74325	CHEVY C-ALL DOOR STRIKER	1	C	74328	NO

NUMBER OF COPIES ?

Figure 22. VDP Summary Display

REVIEW BACK-ORDERED PARTS FILE

1. Press ATTENTION.
   VIMS displays TRANSACTION ?

2. Enter PARTS/REVIEW. Press LINE ACCEPT.

3. Page one of the Back-Ordered Parts file is displayed as shown in Figure 23.

4. Review the Back-Ordered Parts file using the following function keys.

   FWDPAGE Display next page.

   BACKPAGE Display previous page.

5. Press DONE to signal end of transaction.

BACK ORDERED PARTS

C C	VEHICLE REG NO.	WORK ORDR	JOB NO.	PART NO.(FSN)NOMENCLATURE	QTY	UNIT COST	M O	SC CO	DATE ORDER	DUE/ RCVD	BIN LOC
	69C00085	XXXX	01	1925103/HEATER CORE	1	33.70	M	C	74301		
	69C00085	4225	02	697-67919/PRINTED CIR BO	1	9.76	M	C	74326		
	69B01922	3897	03	WATER PUMP-69 CHEV PICKUP	1	18.98	M	C	74301	74315	018
	69B01922	4227	04	ENG-VALVES&TAPPETS CHE-PU	8	10.00	M	C	74335	74336	
	66B02546	3990	01	2610004897973/G78-15 TIRE	1	18.68	O	S	74312	74321	TIR
	68D01600	3458	07	WIPER MOTOR- OSH SNO-PLOW	1	42.50	M	C	74239		

Figure 23. Back-Ordered Parts File Display

ADD ITEM TO BACK-ORDERED PARTS FILE

1. Press ATTENTION.
   VIMS displays TRANSACTION ?

2. Enter PARTS/ADD. Press LINE ACCEPT.
   VIMS displays VEHICLE OR X(EXIT).

3. Enter the vehicle registration number. Press LINE ACCEPT.

   If one of the following messages is displayed, reenter the registration number or press QUIT.

   UNRECOGNIZED VEHICLE
   VEHICLE OR X(EXIT)

   INVALID REGISTRATION
   VEHICLE OR X(EXIT)

   All of the entries for the vehicle are displayed as shown in Figure 24.

4. Enter the Work Order Number, Job Number, Part Number, Nomenclature, Quantity, Unit Cost, M/O Code, SC Code, Date Ordered, and Date Due. Press LINE ACCEPT. VIMS enters the vehicle registration number. The messages, INVALID WORK ORDER and INVALID JOB NUMBER, are displayed if errors occur in those fields. The operator is not allowed to continue entering data until the errors are corrected. The messages are cleared when the errors are corrected. A non-alphanumeric character may not be entered as the first character of the entry. All data entry edit functions are available. Repeat this step until all additions are made.

5. Enter X and press LINE ACCEPT to signal end of transaction. DONE may also be used.

BACK ORDERED PARTS

VEHICLE REG NO.	WORK ORDR	JOB NO.	PART NO.(FSN)	NOMENCLATURE	QTY	UNIT COST	M O	SC CD	DATE ORDER	DUE/ RCVD	BIN LOC
69801922	3897	03	WATER PUMP-69 CHEV PICKUP		1	18.98	M	C	74301	74315	018
69801922	....	..	.....	.....	...	.....	.	.	.....	.....	....

Figure 24. Add Item to Back-Ordered Parts File Display

CHANGE ITEM IN BACK-ORDERED PARTS FILE

1. Press ATTENTION.
   VIMS displays TRANSACTION ?

2. Enter PARTS/CHANGE. Press LINE ACCEPT.
   VIMS displays VEHICLE, A(ALL), OR X(EXIT).

Change Item in Record

3a. Enter the vehicle registration number. Press LINE ACCEPT.

If one of the following messages is displayed, reenter the vehicle registration number or press QUIT.

NO BACK ORDERED PARTS FOR VEHICLE
VEHICLE, A(ALL), OR X(EXIT)

INVALID REGISTRATION
VEHICLE, A(ALL), OR X(EXIT)

All Back-Ordered Parts for the vehicle are displayed as shown in Figure 25.

4a. Make changes to the file. The following function keys are available and may be used while positioned in the Control Character field.

C	Change item.
↑	Move to previous item on page.
↓	Move to next item on page.
Space	Move to next item on page, clear C if present.

The following action may be performed with the aid of the available function keys.

Change Item	Position to item. Enter C. Enter changes. Press LINE ACCEPT.

BACK ORDERED PARTS

C	VEHICLE REG NO.	WORK ORDR	JOB NO.	PART NO.(FSN)	NOMENCLATURE	QTY	UNIT COST	M O	SC CD	DATE ORDER	QUE/ RCVD	BIN LOC
C	69B01922	3897	03	WATER PUMP-69 CHEV PICKUP		1	18.98	M	C	74301	74315	018
	69B01922	4227	04	ENG-VALVES&TAPPETS CHE-PU		8	10.00	M	C	74335	74336	

Figure 25. Change or Delete Back-Ordered Parts Display

The operator is not allowed to enter into the Work Order Number and Job Number fields. All of the data entry edit functions are available.

5a. Press DONE to signal end of input.

VIMS displays VEHICLE, A(ALL), OR X(EXIT).

6a. Repeat steps 3a through 5a for each record to be changed.

Change Item in File

3b. Enter A. Press LINE ACCEPT.

Page one of the Back-Ordered Parts file is displayed as shown in Figure 23.

4b. Make changes to the file. The following function keys are available and may be used while positioned in the Control Character field.

C	Change item.
↑	Move to previous item on page.
↓	Move to next item on page.
Space	Move to next item on page, clear C if present.
FWDPAGE	Display next page.
BACKPAGE	Display previous page.

The following action may be performed with the aid of the available function keys.

Change Item	Position to item. Enter C. Enter changes. Press LINE ACCEPT.

The operator is not allowed to enter into the Work Order Number and Job Number fields. All of the data entry edit functions are available.

5b. Press DONE to signal end of input.

VIMS displays PROCESSING to indicate that the files are being updated.
VIMS displays VEHICLE, A(ALL), OR X(EXIT).

6b. Repeat steps 3b through 5b until all changes have been made.

7. Enter X and press LINE ACCEPT to signal end of transaction. DONE may also be used.

DELETE ITEM FROM BACK-ORDERED PARTS FILE

1. Press ATTENTION.
   VIMS displays TRANSACTION ?

2. Enter PARTS/DELETE. Press LINE ACCEPT.
   VIMS displays VEHICLE, A(ALL), OR X(EXIT).

Delete Item from Record

3a. Enter vehicle registration number. Press LINE ACCEPT.

If one of the following messages is displayed, reenter the vehicle registration number or press QUIT.

NO BACK ORDERED PARTS FOR VEHICLE
VEHICLE, A(ALL), OR X(EXIT)

INVALID REGISTRATION
VEHICLE, A(ALL), OR X(EXIT)

All Back-Ordered Parts for vehicle are displayed as shown in Figure 25.

4a. Indicate which items are to be deleted. The following function keys are available and may be used while positioned in the Control Character field.

D	Delete item.
↑	Move to previous item on page.
↓	Move to next item on page.
Space	Move to next item on page, clear D if present.

The following action may be performed with the aid of the available function keys.

Delete Item	Position to item. Enter D.

5a. Press DONE to signal end of input.
   VIMS displays VEHICLE, A(ALL), OR X(EXIT).

6a. Repeat steps 3a through 5a until all desired parts have been deleted from the file.

Delete Item from File

3a. Enter A. Press LINE ACCEPT.

Page one of the Back-Ordered Parts file is displayed as shown in Figure 23.

4b. Designate items to be deleted. The following function keys are available and may be used while positioned in the Control Character field.

D	Delete item.
↑	Move to previous item on page.
↓	Move to next item on page.
Space	Move to next item on page, clear D if present.
FWDPAGE	Display next page.
BACKPAGE	Display previous page.

The following action may be performed with the aid of the available function keys.

Delete Item	Position to Item. Enter D.

5b. Press DONE to signal end of input.
VIMS displays PROCESSING to indicate that files are being updated.
VIMS displays VEHICLE, A(ALL), OR X(EXIT).

6b. Repeat steps 3b through 5b until all desired parts have been deleted from the file.

7. Enter X and press LINE ACCEPT to signal end of transaction. DONE may also be used.

ISSUE BACK-ORDERED PARTS

1. Press ATTENTION.
   VIMS displays TRANSACTION ?

2. Enter PARTS/ISSUE. Press LINE ACCEPT.
   VIMS displays WORK ORDER ISSUED AGAINST OR X(EXIT).

3. Enter the work order number. Press LINE ACCEPT.

   If INVALID WORK ORDER is displayed, reenter the work order number or press QUIT.

4. Page one of the Back-Ordered Parts file will be displayed as shown in Figure 26.

5. Designate the parts to be issued. The following function keys are available and may be used while positioned in the Control Character field.

X	Issue part.
Space	Move to next item, clear X if present.
↑	Move to previous item.
↓	Move to next item.
FWDPAGE	Display next page.
BACKPAGE	Display previous page.

   The following action may be performed with the aid of the available function keys.

   Issue Part Position to item. Enter X.

6. Press DONE to signal end of input. VIMS displays PROCESSING to indicate that the file is being updated.
   VIMS displays WORKORDER ISSUED AGAINST OR X(EXIT).

7. Repeat steps 3 through 6 until all parts are reported.

8. Enter X and press LINE ACCEPT to signal end of transaction. DONE may also be used.

```
WORK ORDER ISSUED AGAINST 4227
                    ***BACK ORDERED PARTS***

C VEHICLE  WORK JOB                                    UNIT  M SC DATE  DUE/   BIN
C REG NO.  ORDR NO. PART NO.(FSN)NOMENCLATURE  QTY    COST  O CD ORDER RCVD   LOC
- --------- ---- --- --------------------------- --- ------- - -- ----- -----  ---
  69C00085 XXXX  01 1925103/HEATER CORE           1   33.70 M  C 74301

  69C00085 4225  02 697-67919/PRINTED CIR BD      1    9.76 M  C 74326


  69B01922 3897  03 WATER PUMP-69 CHEV PICKUP     1   18.98 M  C 74301 74315  018

  69B01922 4227  04 ENG-VALVES&TAPPETS CHE-PU     8   10.00 M  C 74335 74336


  66B02546 3990  01 2610004897973/G78-15 TIRE     1   18.68 O  S 74312 74321  TIR


  68001600 3458  07 WIPER MOTOR- OSH SNO-PLOW     1   42.50 M  C 74239
```

Figure 26. Issue Back-Ordered Parts Display

INPUT TIME CARDS

1. Press ATTENTION.
   VIMS displays TRANSACTION ?

2. Enter TIME/INPUT. Press LINE ACCEPT.
   VIMS displays WORKCENTER, SSAN, OR X(EXIT).

Input by Employee

3a. Enter the social security account number. Press LINE ACCEPT.

    If the following message is displayed, reenter the social security account number or press QUIT.

    INVALID SSAN
    WORKCENTER, SSAN, OR X(EXIT)

    The time card form is displayed as shown in Figure 27.

4a. Enter the time card data. Press DONE to terminate data input for employee.

    The following function keys may be used during the input of time card data.

↑	Move up one position.
↓	Move down one position.
→	Move right one formatted position.
←	Move left one formatted psoition.
"	Enter data from corresponding field on preceeding line. Does not enter data on screen. Must be first character entered in field.
IGNORE	Terminate entry. No data is entered in file.
TAB	Move to next formatted field.
LINE ACCEPT	Move to next line.

5a. VIMS displays WORKCENTER, SSAN, OR X(EXIT).

EMPLOYEE NAME:METZGER E H
SSAN:021436096
WORK CENTER:17200

DATE	WORK ORDER NUMBER	JOB NUMBER OR LABOR CODE	WORK SHIFT CODE	TIME (HOURS)
.....	....	..	.	....
.....	....	..	.	....
.....	....	..	.	....
.....	....	..	.	....
.....	....	..	.	....
.....	....	..	.	....
.....	....	..	.	....
.....	....	..	.	....
.....	....	..	.	....
.....	....	..	.	....
.....	....	..	.	....
.....	....	..	.	....
.....	....	..	.	....
.....	....	..	.	....
.....	....	..	.	....
.....	....	..	.	....

Figure 27. Time Card Input Display

6a. Repeat steps 3a through 5a for each employee.

7a. Enter X and press LINE ACCEPT to signal end of transaction. DONE may also be used. PROCESSING is displayed to indicate that files are being updated.

8a. If an Error Suspense File has already been created and no new entries are made, VIMS displays COPY OF ERROR SUSPENSE FILE ?, otherwise the Error Suspense File is printed. If the message is displayed, enter Y and press LINE ACCEPT for copy or enter N and press LINE ACCEPT if no copy is needed.

Input by Work Center

3b. Enter the work center number. Press LINE ACCEPT.

If the following message is displayed, reenter the work center number or press QUIT.

```
INVALID INPUT
WORKCENTER, SSAN, OR X(EXIT)
```

VIMS displays

```
NEXT EMPLOYEE IS NNNNNNNN
(N)NEXT (S)SKIP (P)PREVIOUS (X)EXIT
```

4b. Select Employee and enter the time card data.

Next employee	Enter N. Press LINE ACCEPT. See step 4a.
Skip employee	Enter S. Press LINE ACCEPT.
Previous employee	Enter P. Press LINE ACCEPT. VIMS displays EMPLOYEE SSAN. See steps 3a and 4a. The next employee is the same as when P was entered.

5b. VIMS displays

```
NEXT EMPLOYEE IS NNNNNNNN
(N)NEXT (S)SKIP (P)PREVIOUS (X)EXIT
```

Repeat steps 4b and 5b for each employee in work center.

6b. Enter X and press LINE ACCEPT to signal end of input for work center. DONE may also be used.

7b. VIMS displays WORKCENTER, SSAN, OR X(EXIT).

8b. Repeat steps 3b through 7b until all time card data is input.

9b. Enter X and press LINE ACCEPT to signal end of transaction. DONE may also be used. PROCESSING is displayed to indicate that files are being updated.

10b. If an error Suspense File has already be been created and no new entries are made, VIMS diplays COPY OF ERROR SUSPENSE FILE ?, otherwise the Error Suspense File is printed. If the message is displayed, enter Y and press LINE ACCEPT for copy or enter N and press LINE ACCEPT if no copy is needed.

EDIT TIME CARDS

1. Press ATTENTION.
   VIMS displays TRANSACTION ?

2. Enter TIME/EDIT. Press LINE ACCEPT.
   VIMS displays ENTRY, OR X(EXIT).

3. Enter the entry number. Press LINE ACCEPT.

   If one of the following messages is displayed, reenter the entry number or press QUIT.

   INVALID ENTRY NUMBER
   ENTRY, OR X(EXIT)

   ENTRY HAS BEEN PROCESSED
   ENTRY, OR X(EXIT)

4. Time card data is displayed as shown in Figure 28.

5. Enter corrections to time card data. See step 4a of Input Time Card.

6. VIMS displays ENTRY IS NOW CORRECT or ENTRY STILL INCORRECT.

7. Repeat steps 3 through 6 until all corrections are made.

8. Enter X and press LINE ACCEPT to signal end of transaction. DONE may also be used.

9. VIMS displays COPY OF ERROR SUSPENSE FILE ?
   Enter Y and press LINE ACCEPT for copy.
   Enter N and press LINE ACCEPT if no copy is needed.

EMPLOYEE NAME:METZGER E H
SSAN:021436096
WORK CENTER:17200

DATE	WORK ORDER NUMBER	JOB NUMBER OR LABOR CODE	WORK SHIFT CODE	TIME (HOURS)
74335	4211	01	1	01.2
74335	4211	02	1	00.8
74335	4212	*03	1	02.0
74335	4212	*04	1	01.3
74335	4217	02	1	00.5
74335	9999	54	1	*01.0

Figure 28. Time Card Edit Display

CREATE RECORD

1. Press ATTENTION.
   VIMS displays TRANSACTION ?

2. Enter CREATE. Press LINE ACCEPT.
   VIMS displays FILENAME OR X(EXIT).

3. Enter the filename. Press LINE ACCEPT.

   Reenter the filename or press QUIT if FILENAME OR X(EXIT) is redisplayed.

4. VIMS displays ENTRYID ?

5. Enter the record ID. Press LINE ACCEPT.

6. VIMS displays ENTRY ALREADY EXISTS or format for data entry.

7. The following function keys may be used during the creation of data records and formats.

↑	Move up one space.
↓	Move down one space.
←	Move left one space.
→	Move right one space.

   The following functions are available only when creating data records

CNTL/A FWDPAGE	Display page two.
CNTL/A BACKPAGE	Display page one.
TAB	Forward one field.

8. Press DONE to signal end of data input.
   Press XMIT to signal end of data input when creating a format.

9. VIMS displays SAVE ?

10. Enter Y and press LINE ACCEPT to save record. Press LINE ACCEPT if record is not to be saved.

11. If record is not to be saved
    VIMS displays REEDIT ?

12. Enter Y and press LINE ACCEPT to reedit, otherwise press LINE ACCEPT.

13. If record is to be reedited repeat steps 6 through 12.

14. Enter X and press LINE ACCEPT to signal end of transaction. DONE may also be used.

EDIT RECORD

1. Press ATTENTION.
   VIMS displays TRANSACTION ?

2. Enter EDIT. Press LINE ACCEPT.
   VIMS displays FILENAME OR X(EXIT).

3. Enter the filename. Press LINE ACCEPT.

   Reenter the filename or press QUIT if FILENAME OR X(EXIT) is redisplayed.

4. VIMS displays ENTRYID ?

5. Enter the record ID. Press LINE ACCEPT.

6. VIMS displays ENTRY DOES NOT EXIST TYPE Y TO CONFIRM or displays data record.

7. Enter Y and press LINE ACCEPT if new record is to be created, otherwise press LINE ACCEPT.

8. See steps 7 through 14 of Create Record.

DELETE RECORD

1. Press ATTENTION.
   VIMS displays TRANSACTION ?

2. Enter DELETE. Press LINE ACCEPT.
   VIMS displays FILENAME OR X(EXIT).

3. Enter the filename. Press LINE ACCEPT.

   Reenter the filename or press QUIT if FILENAME OR X(EXIT) is redisplayed.

4. VIMS displays ENTRYID ?

5. Enter the record ID. Press LINE ACCEPT.

   VIMS displays ENTRY nnnn HAS BEEN DELETED or ENTRY DOES NOT EXIST.

6. Enter X and press LINE ACCEPT to signal end of transaction. DONE also may be used.

FUEL/OIL ISSUE

1. Press ATTENTION.
   VIMS displays TRANSACTION ?

2. Enter FUEL. Press LINE ACCEPT.
   The Fuel/Oil Issue format is displayed as shown in Figure 29.

3. Enter the Issuing Organization, Vehicle Registration Number, Fuel(gallons), Oil(quarts) and Date. Press LINE ACCEPT.

   When one of the following messages is displayed, the operator may either correct the appropriate field or override the message by pressing LINE ACCEPT. The error messages are cleared when LINE ACCEPT is pressed after the error is corrected.

   UNRECOGNIZED VEHICLE

   INVALID FUEL ENTRY

   INVALID OIL ENTRY

   INVALID DATE

   All of the date entry edit functions are available. In addition, a ditto key (") is available. This allows the operator to enter the data in the corresponding field of the preceeding line by entering ("). Partial fields may also be entered in this manner. When the form is full, the format is cleared and the operator starts at the top of the page. The last line of input is remembered, so that ditto key can be used.

4. Press DONE to signal the end of transaction.

**** FUEL/OIL ISSUE ****

ISSUING ORG	VEHICLE REG. NO.	FUEL (GAL)	OIL (QT)	DATE	
••	•••••••	••••	•	•••••	••••••••••••••••••••
••	•••••••	••••	•	•••••	••••••••••••••••••••
••	•••••••	••••	•	•••••	••••••••••••••••••••
••	•••••••	••••	•	•••••	••••••••••••••••••••
••	•••••••	••••	•	•••••	••••••••••••••••••••
••	•••••••	••••	•	•••••	••••••••••••••••••••
••	•••••••	••••	•	•••••	••••••••••••••••••••
••	•••••••	••••	•	•••••	••••••••••••••••••••
••	•••••••	••••	•	•••••	••••••••••••••••••••
••	•••••••	••••	•	•••••	••••••••••••••••••••
••	•••••••	••••	•	•••••	••••••••••••••••••••
••	•••••••	••••	•	•••••	••••••••••••••••••••
••	•••••••	••••	•	•••••	••••••••••••••••••••
••	•••••••	••••	•	•••••	••••••••••••••••••••
••	•••••••	••••	•	•••••	••••••••••••••••••••
••	•••••••	••••	•	•••••	••••••••••••••••••••
••	•••••••	••••	•	•••••	••••••••••••••••••••
••	•••••••	••••	•	•••••	••••••••••••••••••••
••	•••••••	••••	•	•••••	••••••••••••••••••••

Figure 29. Fuel/Oil Issue Display

CHANGE PERSONNEL PARAMETERS

1. Press ATTENTION.
   VIMS displays TRANSACTION ?

2. Enter PARAM. Press LINE ACCEPT.
   VIMS displays NUMBER OF PERSONNEL ON DUTY AT WORK CTR 17220 IS XXX ENTER NEW NUMBER.

3. Enter the number of personnel as a positive real number. Press LINE ACCEPT.

   VIMS displays NUMBER OF PERSONNEL ON DUTY AT WORK CTR 17230 IS XXX ENTER NEW NUMBER.

4. Enter the number of personnel as a positive real number. Press LINE ACCEPT.

SELECT PRINTER

1. Press ATTENTION.
   VIMS displays TRANSACTION ?

2. Enter LPT and press LINE ACCEPT.
   VIMS displays HIGH SPEED PRINTER ?

3. Enter Y and press LINE ACCEPT for high speed printer.
   Enter N and press LINE ACCEPT for Centronics printer.

CHANGE TIME

1. Press ATTENTION.
   VIMS displays TRANSACTION ?

2. Enter TCHANGE and press LINE ACCEPT.
   VIMS displays ENTER THE NUMBER OF HOURS YOU WISH TO ADVANCE.

3. Enter the number of hours to be advanced as a positive real number. Press LINE ACCEPT.

CHANGE SCALE

1. Press ATTENTION.
   VIMS displays TRANSACTION ?

2. Enter SCALE. Press LINE ACCEPT.
   VIMS displays ENTER THE DESIRED CHANGE IN TIME SCALE.

3. Enter the new time scale as a positive real number. Press LINE ACCEPT.

PRINT FILE

1. Press ATTENTION.
   VIMS displays TRANSACTION ?

2. Enter PRINT. Press LINE ACCEPT.
   VIMS displays FILENAME ? or PLEASE ENTER FILENAME.

3. Enter filename. Press LINE ACCEPT.

4. File is printed on the high speed line printer.

APPENDIX III

PROGRAM DESCRIPTIONS

GENERAL

Model documentation includes Hierarchy Input Processing Output (HIPO) Diagrams and Transfer Vector Charts. These graphics offer a visual description of the organization and flow of program control, as well as providing a method of locating and analyzing specific programs.

Diagram 1 is an overview of the entire model. Diagrams 2, 3, and 4 demonstrate how transaction processing is allocated to each VIMS functional area. The final 32 diagrams are lower level descriptions of the actual operations involved in processing the VIMS transactions. Each diagram contains four boxes. The left-hand box defines the inputs to the process. The middle box provides a step-by-step description of the process involved. The right-hand box shows all outputs of a process. Arrows trace the data flow through the steps of the process with large arrows showing the most important data paths. Outputs of one step which are inputs to later steps are shown by an arrow returning to the process box from the output box. Below the input, process, and output boxes is a description area containing more detailed step descriptions where needed. The program or set of programs which accomplish each step are listed to the right of that step description.

Transfer Vector Charts, like the programs, are organized into the three VIMS functional areas. Each major program in the model is represented by a hexagon. For simplicity, the VIMS utilities discussed in Appendix IV do not appear on the charts.

USING THE GRAPHIC AIDS

The graphic aids provide a quick means to find which program accomplishes a specific task.

Refer to the HIPO diagrams for Steps 1 through 4.

Step 1: Locate the desired VIMS functional area in Diagram 1, the VIMS overview.

Step 2: Turn to the functional area overviews, Diagram 2, 3, or 4, and trace the path to the transaction desired.

Step 3: Turn to the diagram listed in the box for the transaction and find the correct step in the process.

Step 3a: If the step description mentions another diagram, turn to it and again find the desired step in the process.

Step 4: Read the program name opposite the step number in the description box at the bottom of the page. Once the program has been located, the Transfer Vector Charts may be used to find the routines which call that program and which routines, if any, the program calls.

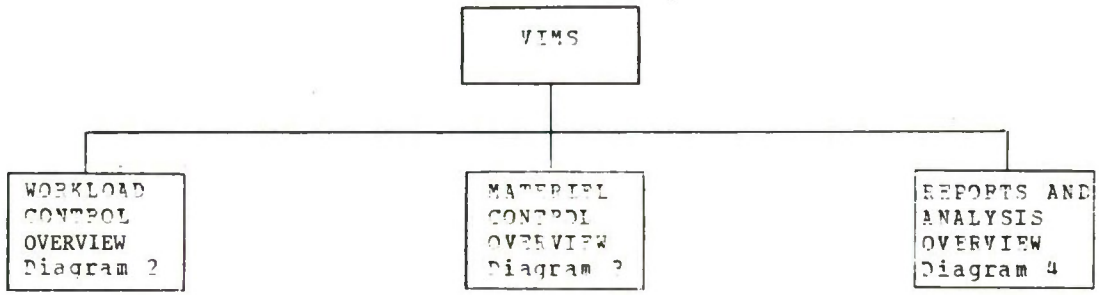
Refer to the Transfer Vector Charts for steps 5 through 7.

Step 5: Locate the correct chart for the VIMS functional area.

Step 6: Find the hexagon which bears the program name. Because some programs appear more than once on the chart, insure that the name located is in the proper transaction.

Step 7: Use the chart as demonstrated in the legend on page 134 to find the desired routines.

DIAGRAM 1 VIMS DEVELOPMENT MODEL OVERVIEW



LEGEND

Filenames and program names appear in capital letters. (For file abbreviations see Appendix V)

Inputs to and outputs from CRT appear in brackets. [input or output]

Entry into diagram

DIAGRAM 2 WORKLOAD CONTROL OVERVIEW

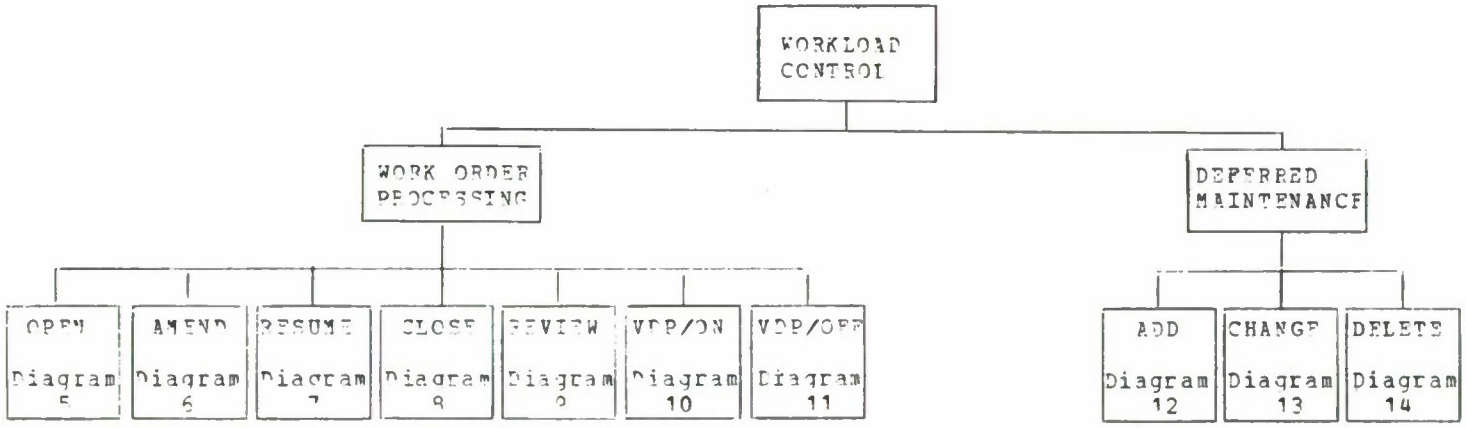


DIAGRAM 3 MATERIEL CONTROL OVERVIEW

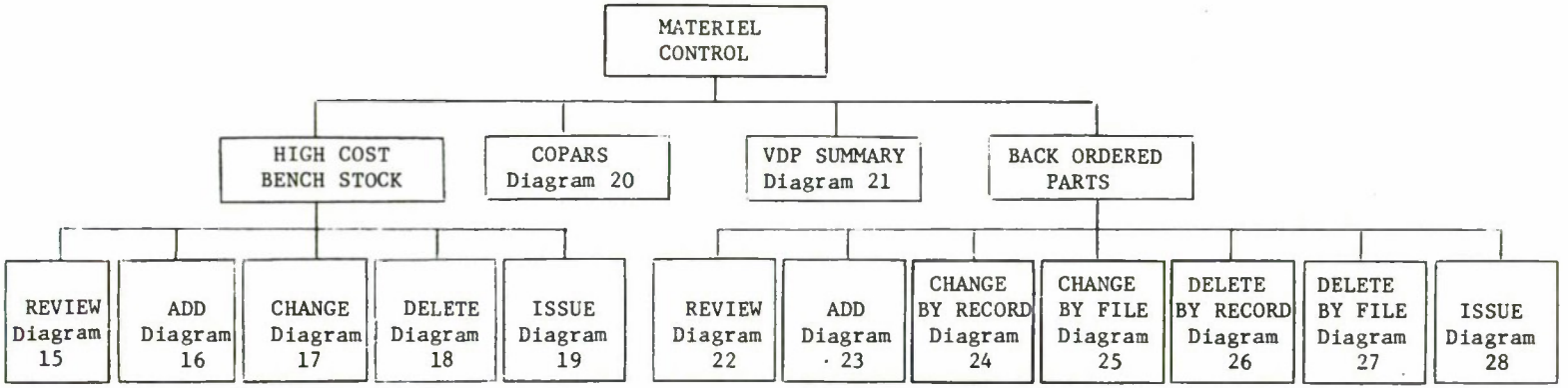


DIAGRAM 4 REPORTS AND ANALYSIS OVERVIEW

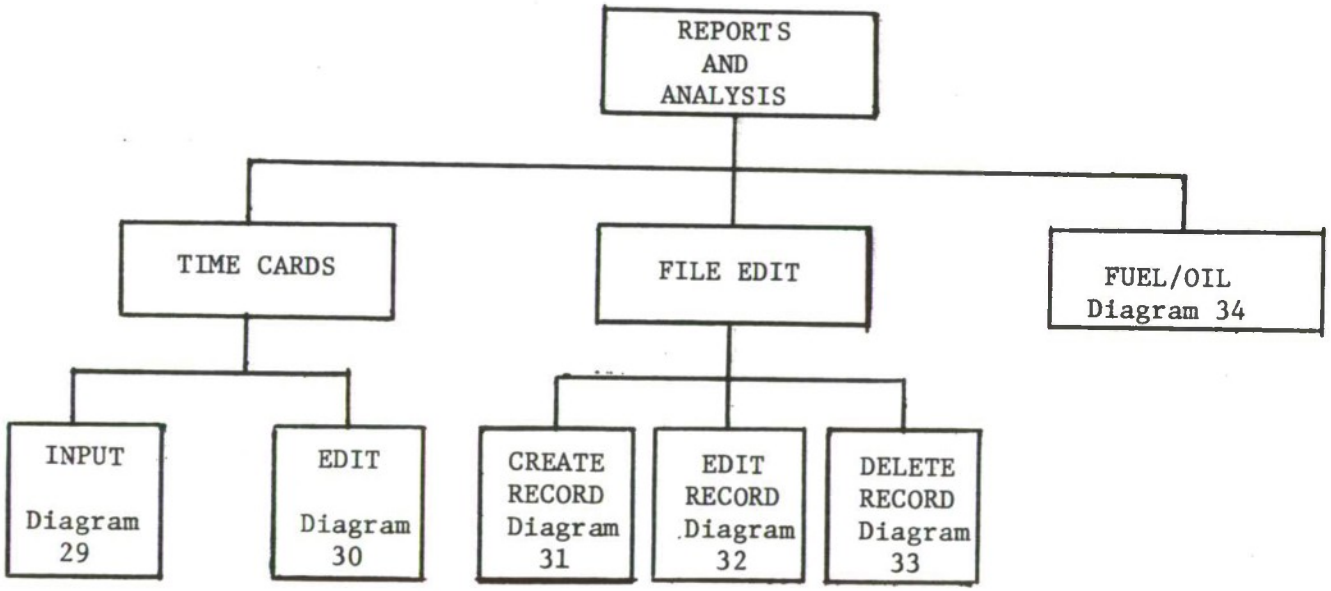
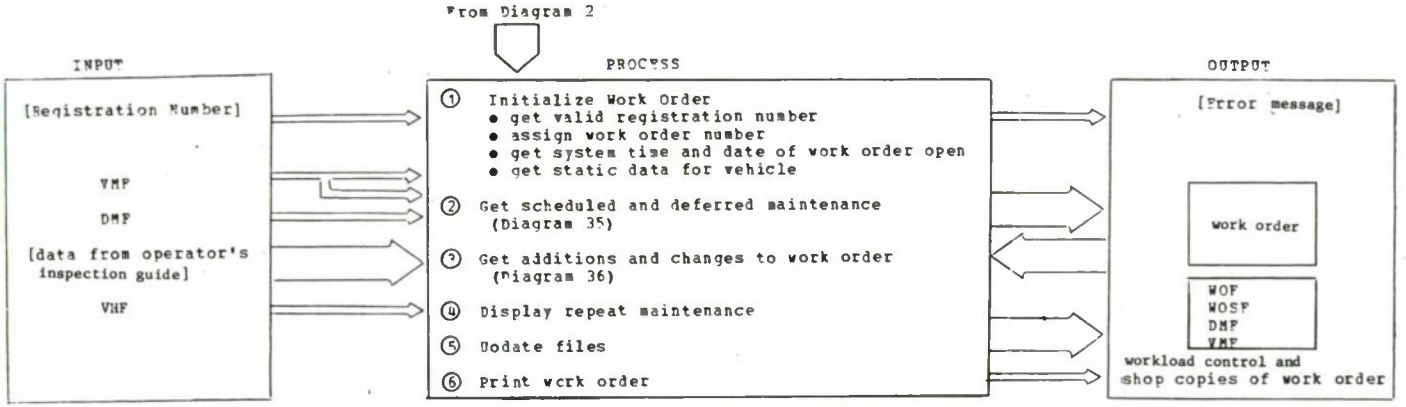
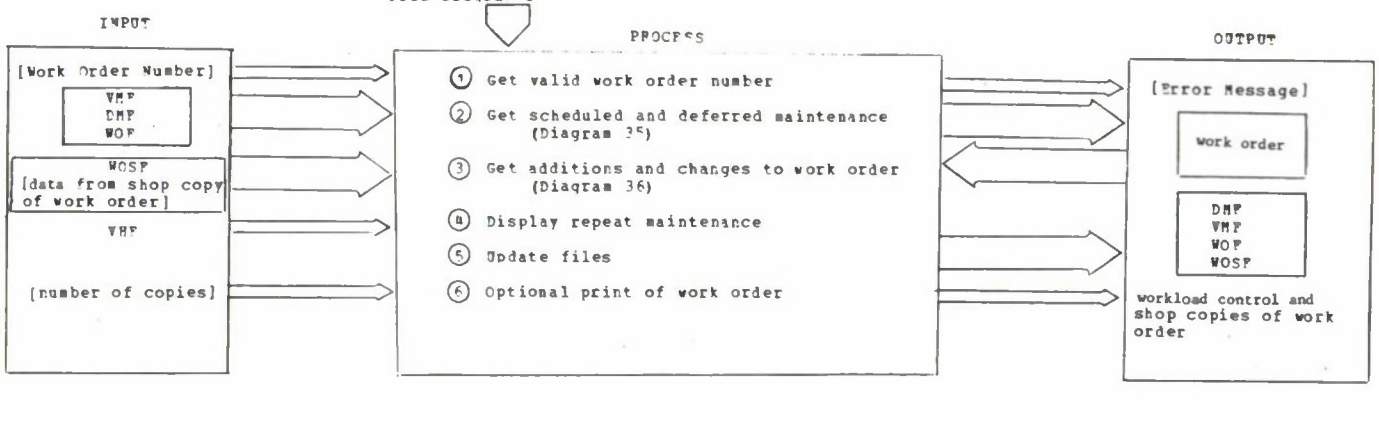


DIAGRAM 5 - OPEN WORK ORDER



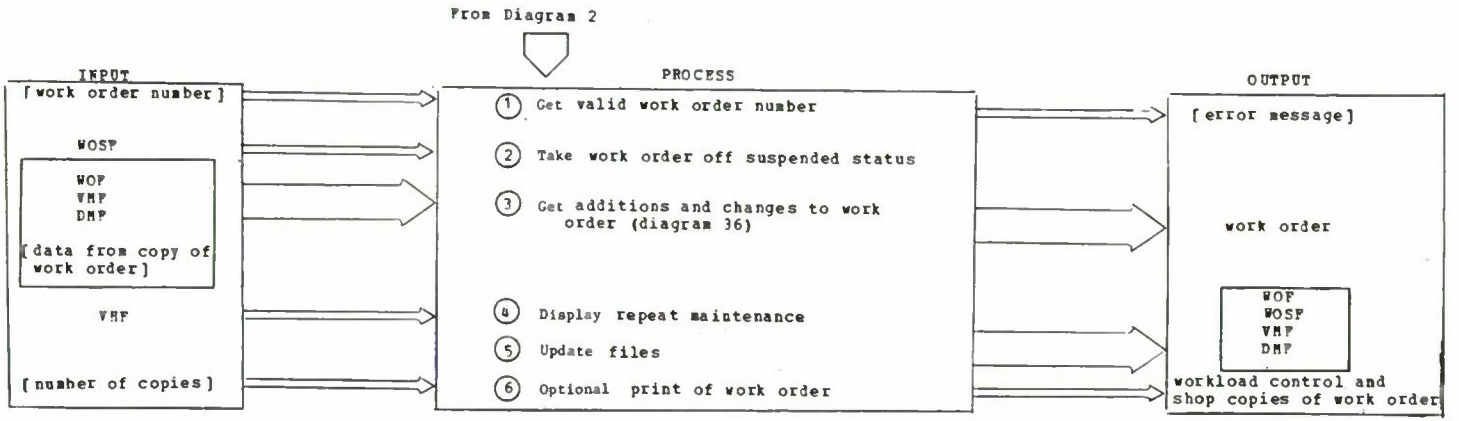
STEP	DESCRIPTION	PROGRAM
①	Display error message if vehicle not in fleet	WOPROCESS
②		DISPMAINT
③		WORKORDER
④	Redisplay work order if there is repeat maintenance	WORKORDER
⑤		WORKORDER
⑥	Deferred jobs are not printed	WOPRINT

DIAGRAM 6 AMEND WORK ORDER



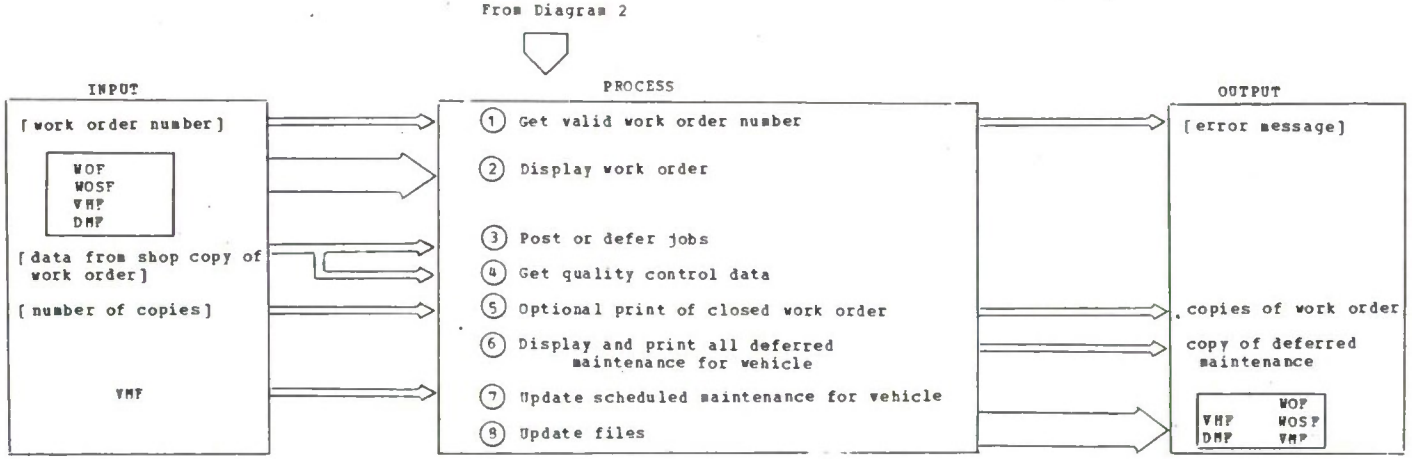
STEP	DESCRIPTION	PROGRAM
1	Display error message if work order not in open status	WOPROCESS
2		WORKORDER
3		WORKORDER
4	Redisplay work order if there is repeat maintenance	WORKORDER
5		WORKORDER
6	Deferred jobs are not printed	WOPRINT

DIAGRAM 7 RESUME WORK ORDER



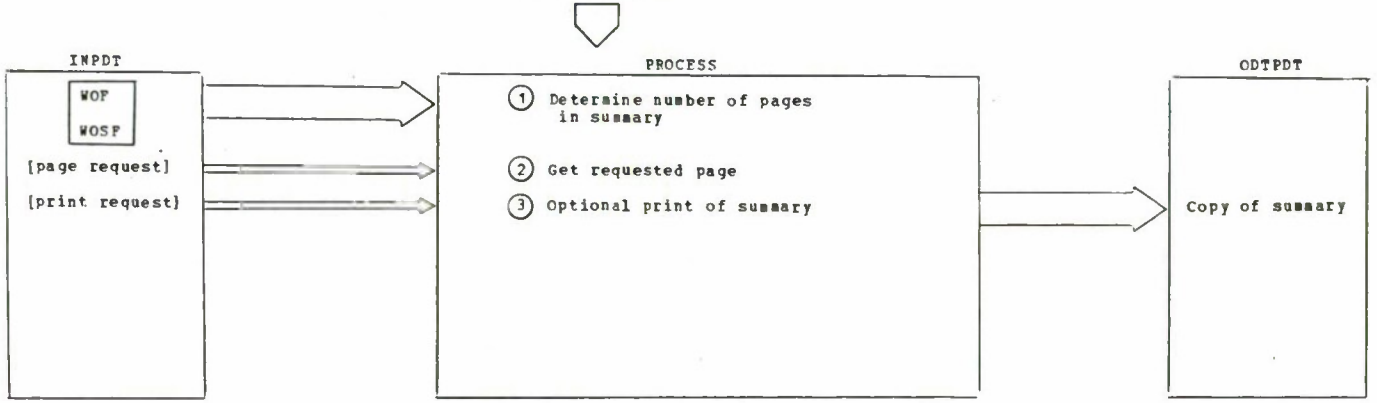
STEP	DESCRIPTION	PROGRAM
1	Display error message if work order not in snspended status	WOPROCESS
2		WORKORDER
3		WORKORDER
4	Redisplay work order if there is repeat maintenance	WORKORDER
5		WORKORDER
6	Deferred jobs are not printed	WOPRINT

DIAGRAM 8 CLOSE WORK ORDER



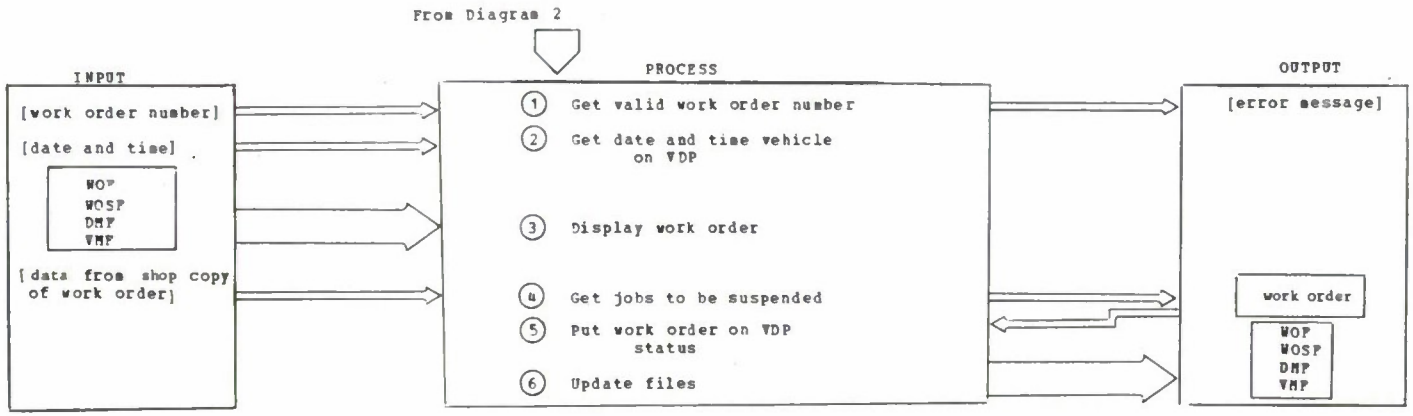
STEP	DESCRIPTION	PROGRAM
1	Display error message if work order not in open status	WOPROCESS
2		WOCLOSE
3	The vehicle historical record is updated as jobs are posted	CLOSEPROC
4		WOCLOSE
5		WOPRINT
6		LINEPRINT
7		WOCLOSE
8		WOCLOSE

DIAGRAM 9 REVIEW WORK ORDERS



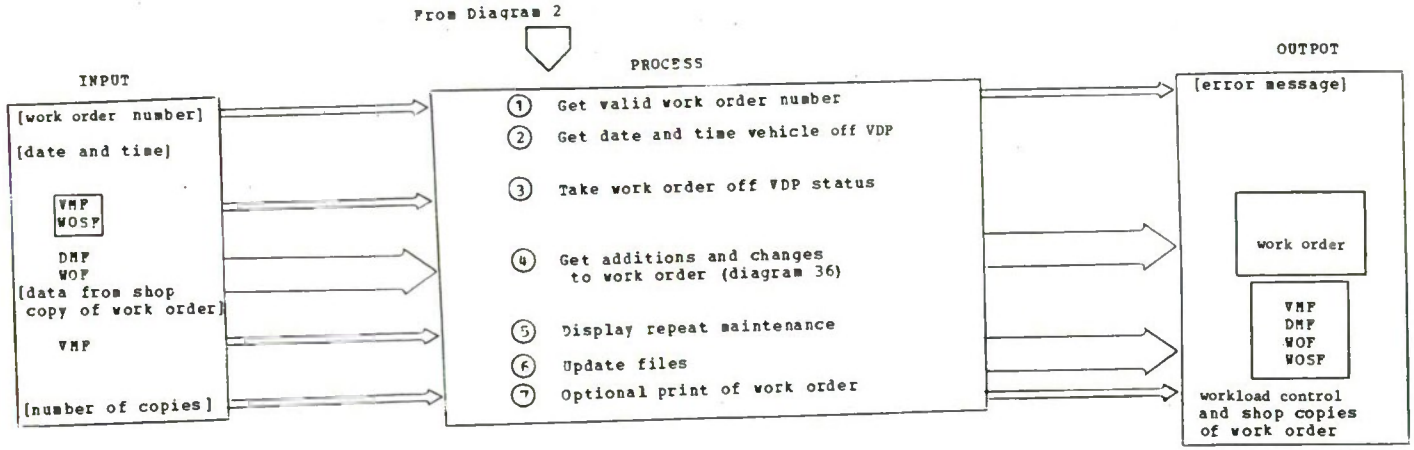
STEP	DESCRIPTION	PROGRAM
1		WOSUMMARY
2	User may step forward or backward one page at a time. Additional pages are created as the user pages forward through summary. Previously created pages are retrieved as he pages backward.	GETPAGE
3		WOSUMMARY

DIAGRAM 10 PLACE VEHICLE ON VDP



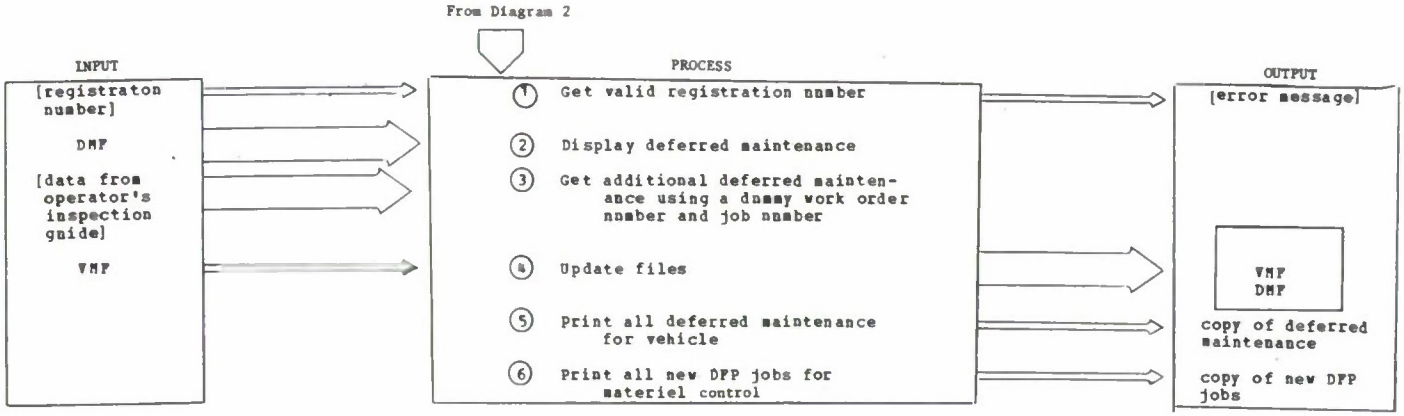
STEP	DESCRIPTION	PROGRAM
1	Display error message if work order not in open status	WOPROCESS
2	If user does not enter date and time, supply system date and time	WOCLOSE
3		WOCLOSE
4	All suspended jobs are deferred	CLOSEPROC
5		WOCLOSE
6		WOCLOSE

DIAGRAM 11 TAKE VEHICLE OFF VDP



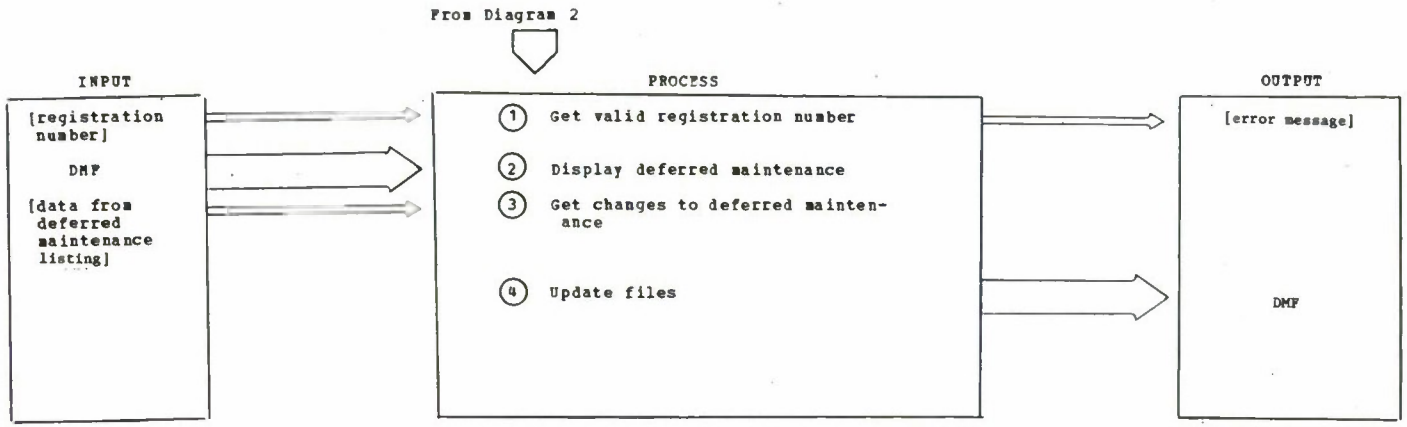
STEP	DESCRIPTION	PROGRAM
1	Display error message if vehicle not on VDP	WOPROCESS
2	If user does not enter date and time supply system date and time	WOCLOSE
3		WOCLOSE
4		WORKORDER
5	Redisplay work order if there is repeat maintenance	WORKORDER
6		WORKORDER
7	Deferred jobs are not printed	WOPRINT

DIAGRAM 12 ADD DEFERRED JOBS TO FILE



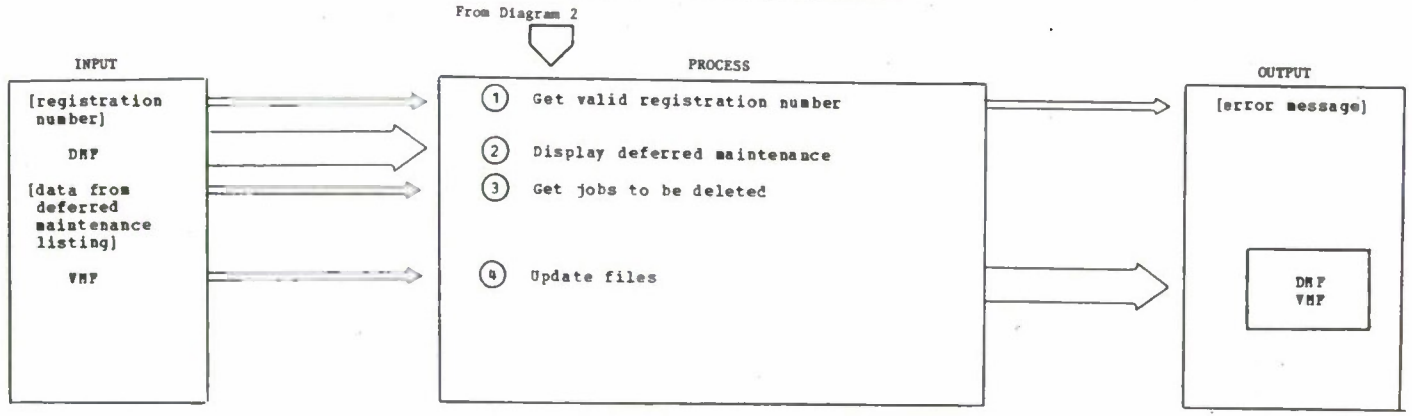
STEP	DESCRIPTION	PROGRAM
1	Display error message if vehicle not in fleet	DDEFER
2		DDEFER
3	Do not allow the addition of VDP jobs	EDITJOB
4		DDEFER
5		LINEPRINT
6		LINEPRINT

DIAGRAM 13 CHANGE DEFERRED JOBS



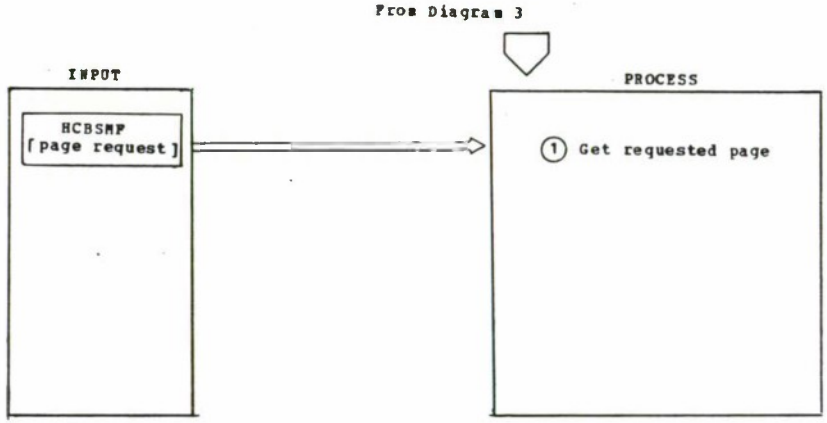
STEP	DESCRIPTION	PROGRAM
1	Display error message if no deferred maintenance on vehicle	DDEFER PARTSUPDATE
2		PARTSUPDATE
3	Protect work order number, job number, and action code "VDP" from change. Do not allow entry of action code "VDP".	EDITJOB
4		PARTSUPDATE

DIAGRAM 14 DELETE DEFERRED JOBS



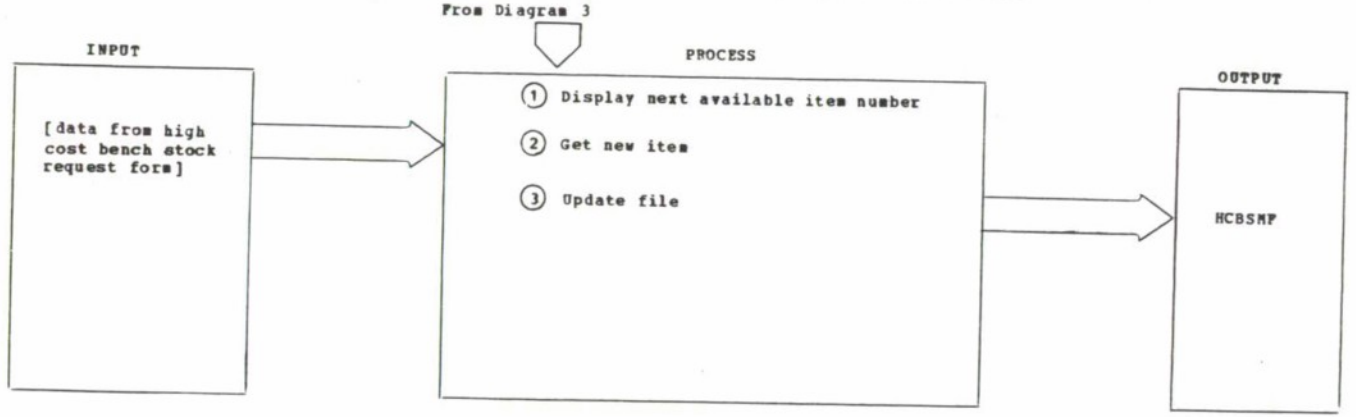
STEP	DESCRIPTION	PROGRAM
1	Display error message if no deferred maintenance on vehicle	EDEFER PARTSUPDATE
2		PARTSUPDATE
3		EDITJOB
4		PARTSUPDATE

DIAGRAM 15 REVIEW HIGH COST BENCH STOCK MASTER FILE



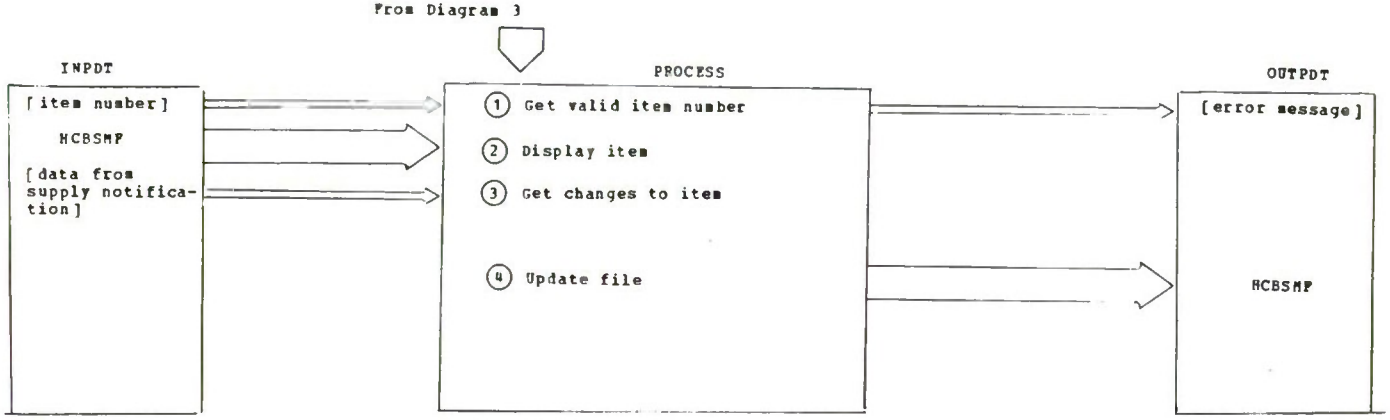
STEP	DESCRIPTION	PROGRAM
1	Display file one page at a time. User may step forward or backward one page at a time.	HREVIEW DOPAGE

DIAGRAM 16 ADD ITEM TO HIGH COST BENCH STOCK MASTER FILE



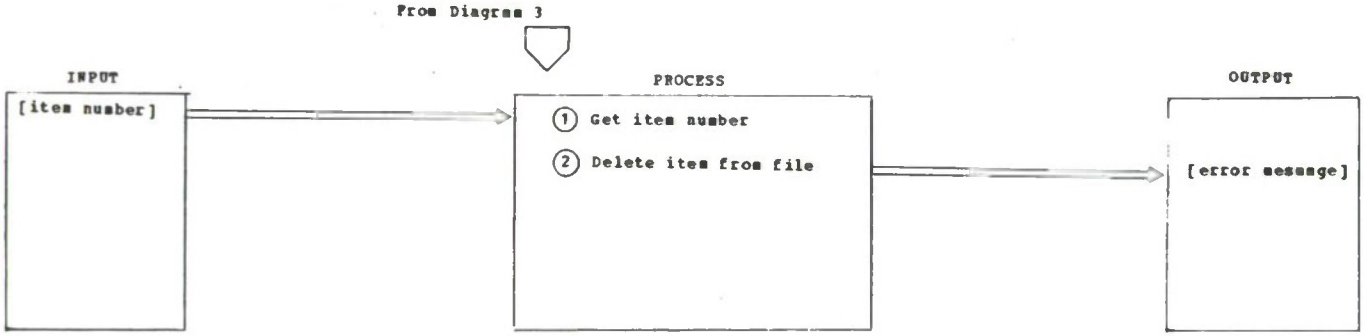
STEP	DESCRIPTION	PROGRAM
1		HADD
2		GETPART
3		HADD

DIAGRAM 17 CHANGE ITEM IN HIGH COST BENCH STOCK MASTER FILE



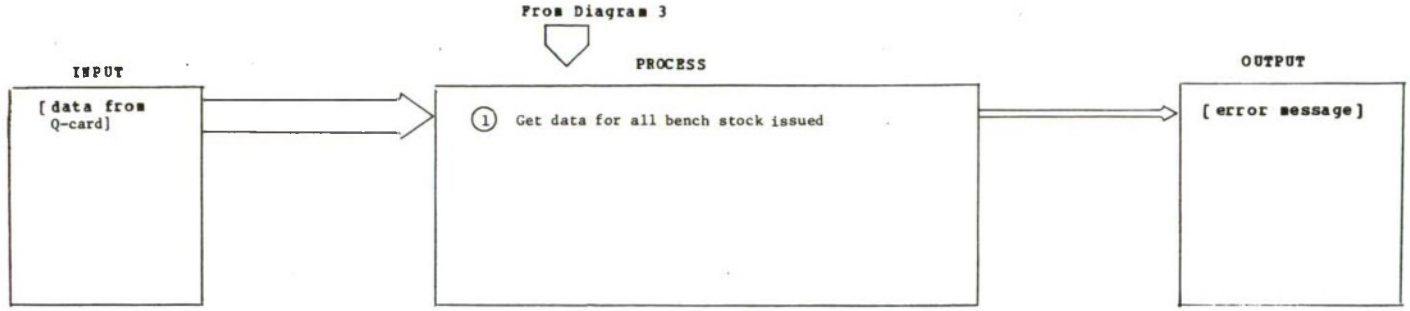
STEP	DESCRIPTION	PROGRAM
1	Display error message if item does not exist	HCHANGE
2		HCHANGE
3		GETPART
4		HCHANGE

DIAGRAM 18 DELETE ITEM FROM HIGH COST BENCH STOCK MASTER FILE



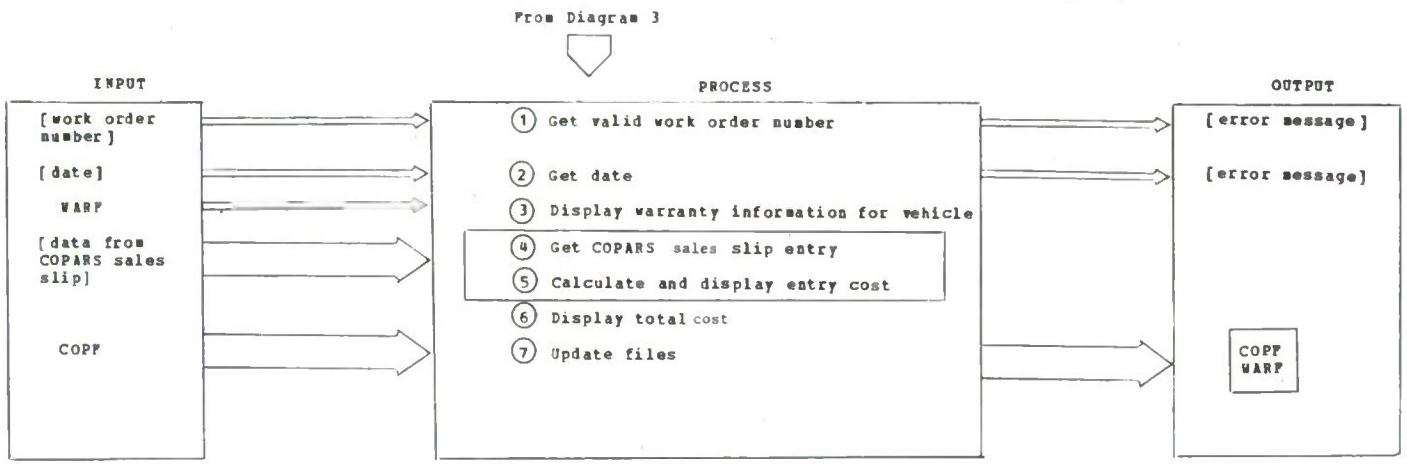
STEP	DESCRIPTION	PROGRAM
1		MDELETE
2	Display error message if item not found	MDELETE

DIAGRAM 19 REPORT ISSUE OF HIGH COST BENCH STOCK



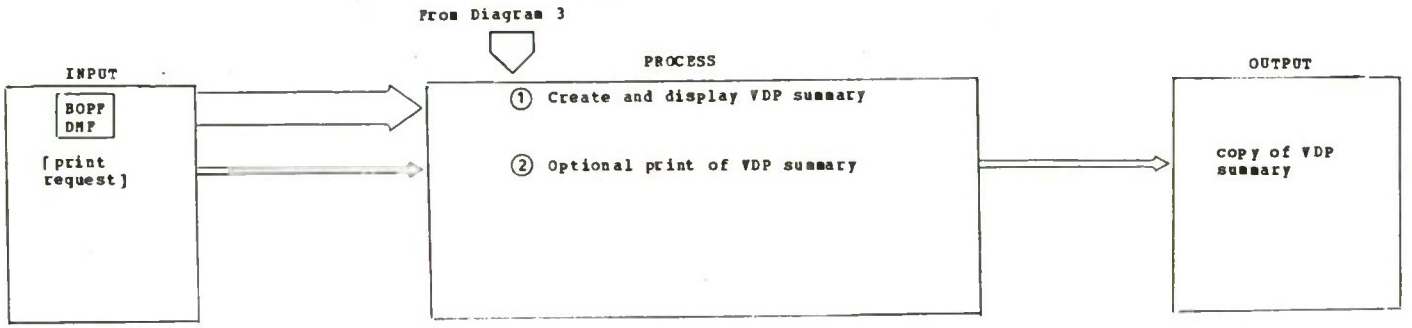
STEP	DESCRIPTION	PROGRAM
1	Display error message for invalid work order number or invalid HCHSMF item number	HISSUE

DIAGRAM 20 ENTER COPARS SALES SLIP DATA



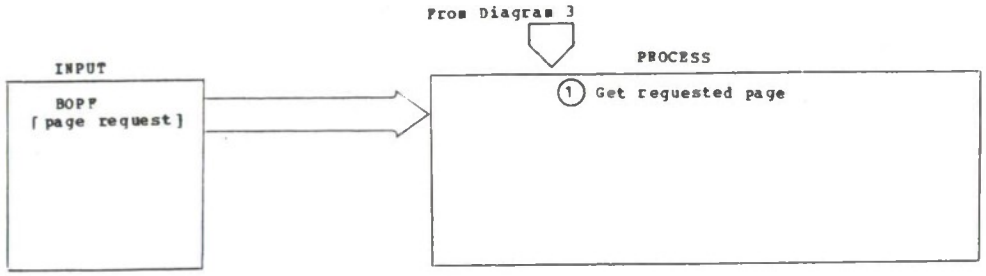
STEP	DESCRIPTION	PROGRAM
1	Display error message if work order does not exist	COPARS
2	Display error message if invalid date	COPARS
3		COPARS
4		GETPART
5	Calculate cost from number of items, list price, and discount percentage	COPARS
6		COPARS
7		COPARS

DIAGRAM 21 CREATE VDP SUMMARY



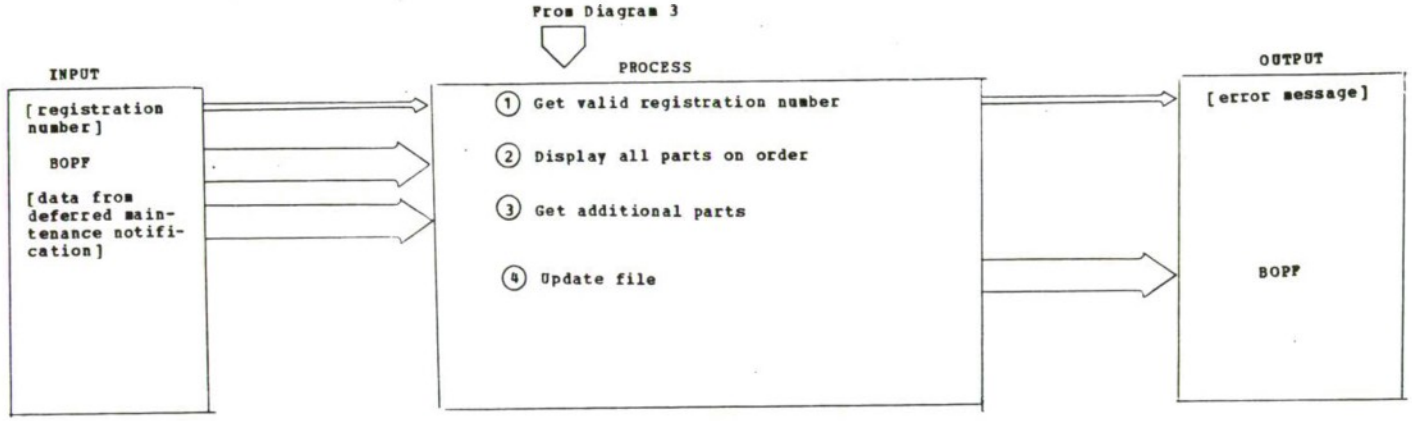
STEP	DESCRIPTION	PROGRAM
①	Check DMF for VDP jobs, accumulate parts information from BOPF	SUMMARY
②		LINEPRINT

DIAGRAM 22 REVIEW BACK-ORDERED PARTS FILE



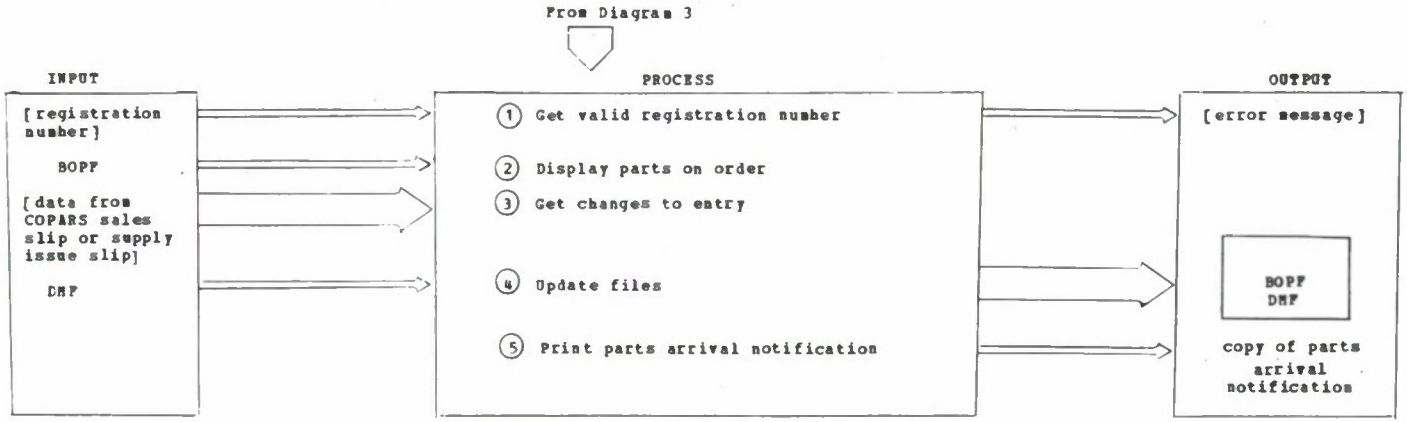
STEP	DESCRIPTION	PROGRAM
1	User may step forward or backward one page at a time. Additional pages containing full entries are created dynamically as the user pages forward through file. Previously created pages are retrieved as user pages backward.	DISPFILE DISPPAGE READPAGE PAGEWRITE

DIAGRAM 23 ADD PART TO BACK-ORDERED PARTS FILE



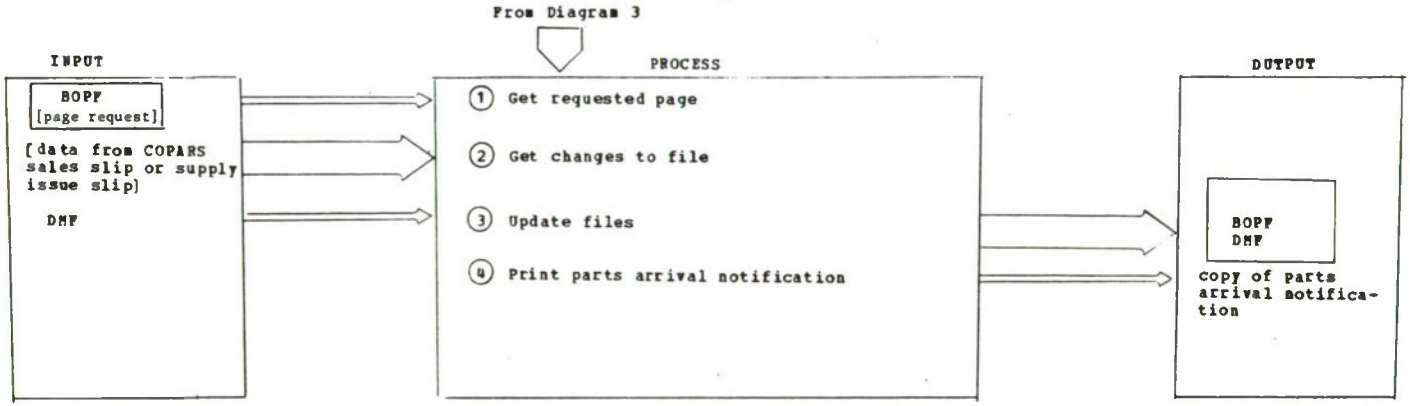
STEP	DESCRIPTION	PROGRAM
1	Display error message if vehicle not in fleet	PADD
2		PADD
3	Each entry is checked for valid work order number and valid job number	GETPART
4		PADD

DIAGRAM 24 CHANGE BACK-ORDERED PARTS FILE RECORD



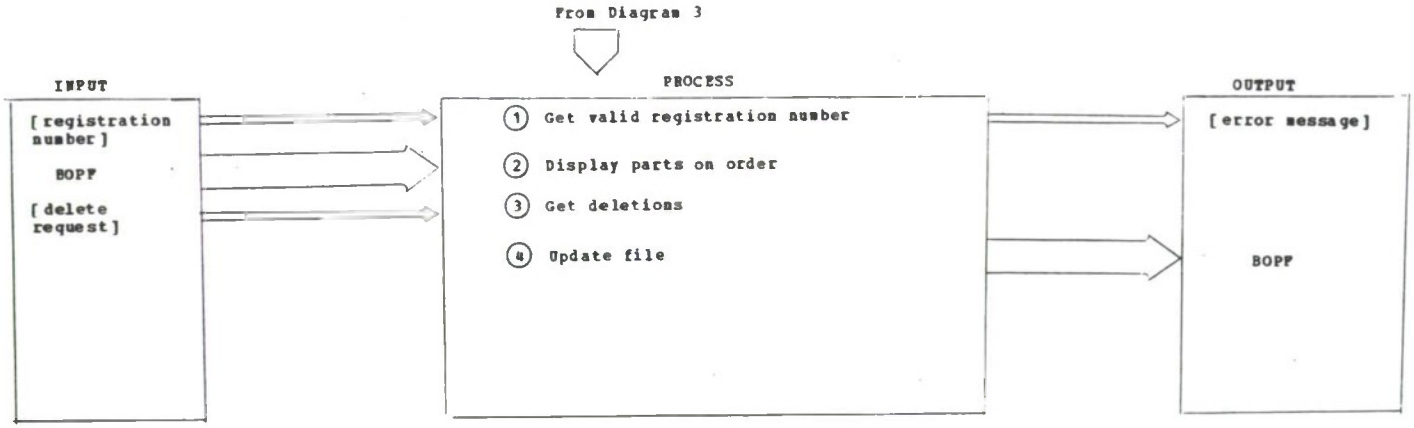
STEP	DESCRIPTION	PROCESS
1	Display error message if no parts on order for vehicle	PARTS
2		PCHANGE
3	Protect work order number and job number fields from changes	GETPART
a	Report jobs for which all parts have arrived. Check for VDP jobs.	PARRIVAL

DIAGRAM 25 CHANGE BACK-ORDERED PARTS FILE



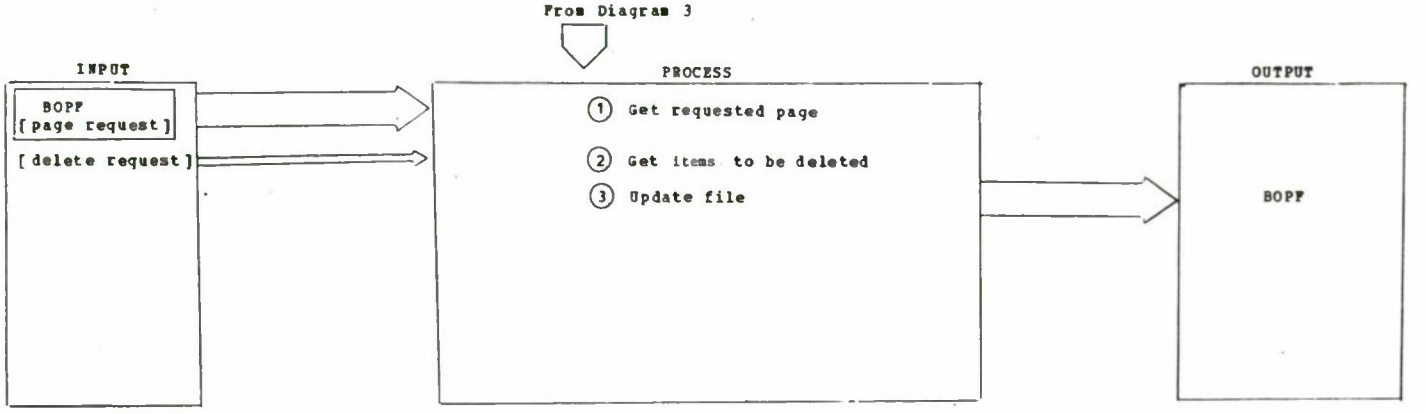
STEP	DESCRIPTION	PROGRAM
1	User may step forward or backward one page at a time. Additional pages containing full records are created dynamically as the user pages forward through file. Previously created pages are retrieved as he pages backwards.	DISPFILE DISPPAGE READPAGE PAGEWRITE
2	Protect work order number and job number from changes	GETPART
3		PROCPAGE
4	Report jobs for which all parts have arrived. Check for VDP jobs.	PARRIVAL

DIAGRAM 26 DELETE PARTS FROM BACK-ORDERED PARTS FILE RECORD



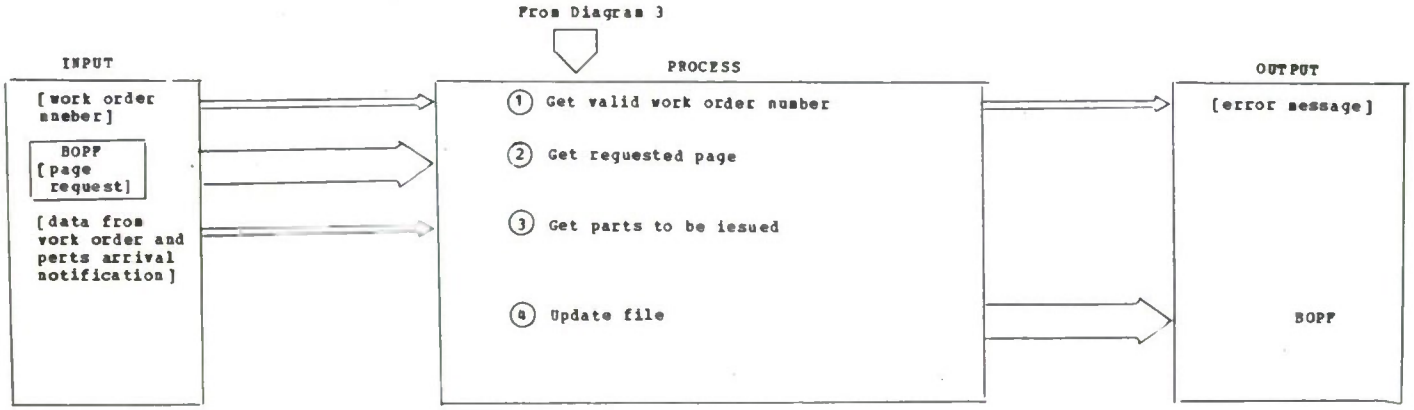
STEP	DESCRIPTION	PROGRAM
1	Display error message if no parts on order for vehicle	PARTS
2		PCHANGE
3		PCHANGE
4		PCHANGE

DIAGRAM 27 DELETE PARTS FROM BACK-ORDERED PARTS FILE



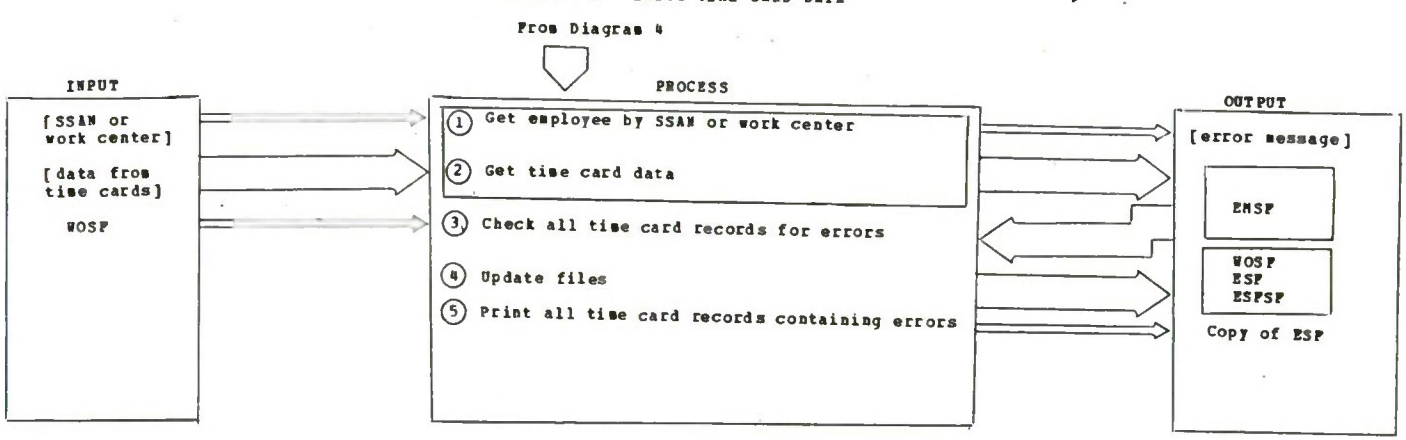
STEP	DESCRIPTION	PROGRAM
1	User may step forward or backward one page at a time. Additional pages containing full records are created dynamically as the user pages forward through file. Previously created pages are retrieved as user pages backward.	DISPFILE DISPPAGE READPAGE PAGEWRITE
2		DISPFILE
3		PROCPAGE

DIAGRAM 28 REPORT ISSUE OF BACK-ORDERED PARTS



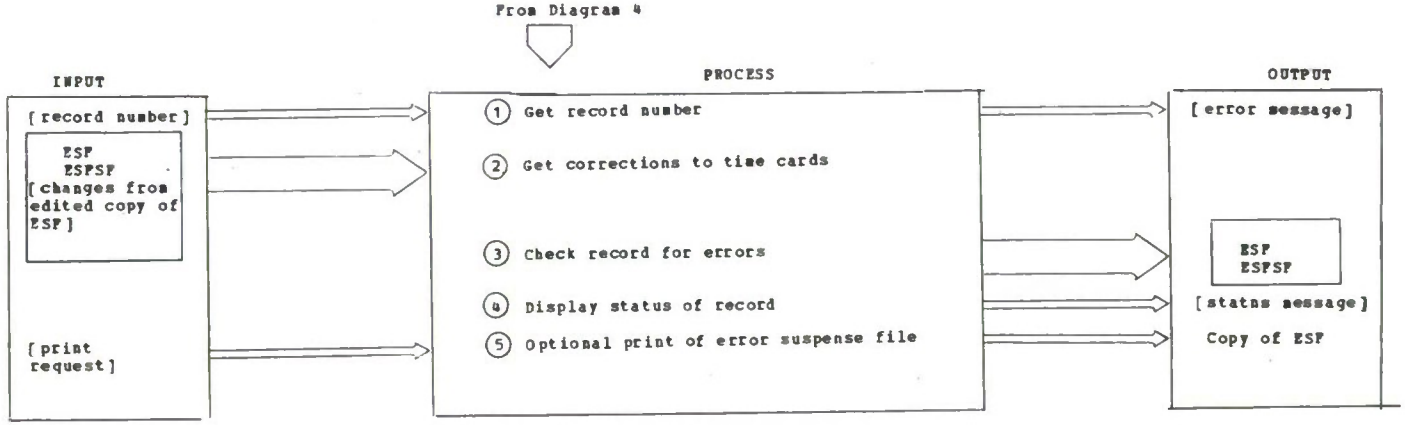
STEP	DESCRIPTION	PROGRAM
1	Display error message if work order does not exist	PARTS
2	User may step forward or backward one page at a time. Additional pages containing full records are created dynamically as the user pages forward through file. Previously created pages are retrieved as he pages backward.	DISPFILE DISPPAGE READPAGE PAGEWRITE
3		DISPFILE
4		DISPFILE

DIAGRAM 29 INPUT TIME CARD DATA



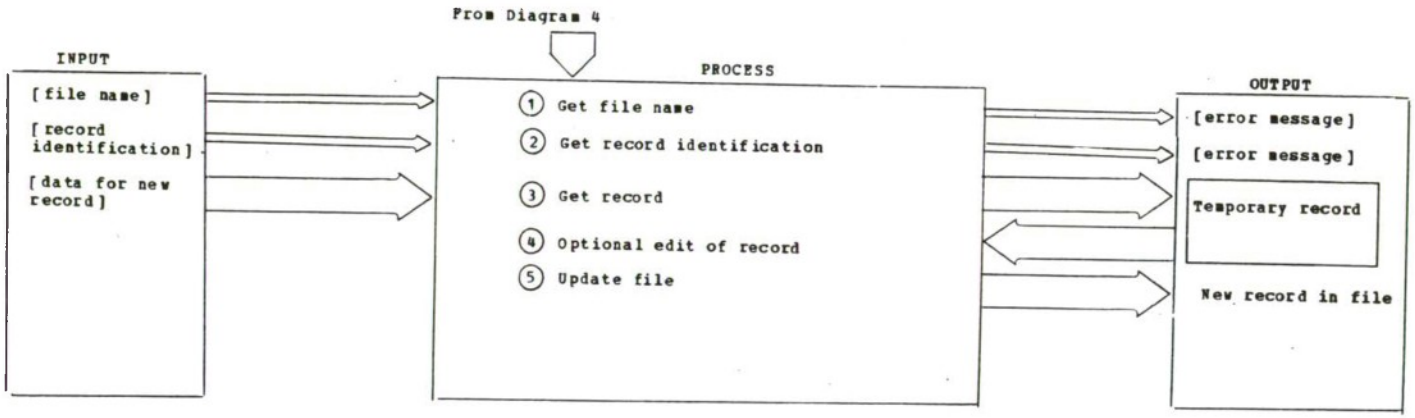
STEP	DESCRIPTION	PROGRAM
1	Get all employees assigned to work center in alphabetical order. Display error message if unrecognized employee or work center.	TIMEPROC
2		PROCESS
3	Check for errors in work order number, job number, shift code, and time	PENTRY
4	Update work order cost information for records with no errors	PENTRY
5		TIMEPROC

DIAGRAM 30 EDIT TIME CARD DATA



STEP	DESCRIPTION	PROGRAM
1	Display error message if record has been corrected or does not exist	TIMEPROC
2		PROCESS
3	Check record for errors in work order number, job number, shift code, and time	PENTRY
		PROCESS
		TIMEPROC
5		

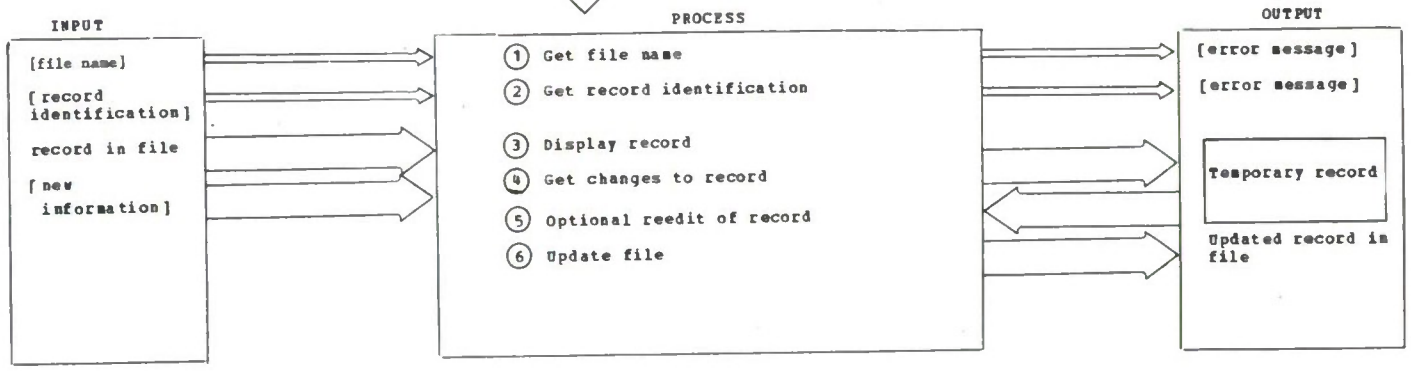
DIAGRAM 31 CREATE DATA RECORD



STEP	DESCRIPTION	PROGRAM
1	Display error message if invalid file name	EDIT
2	Display error message if record already exists	EDIT
3		GETSTRING
4		EDIT
5		EDIT

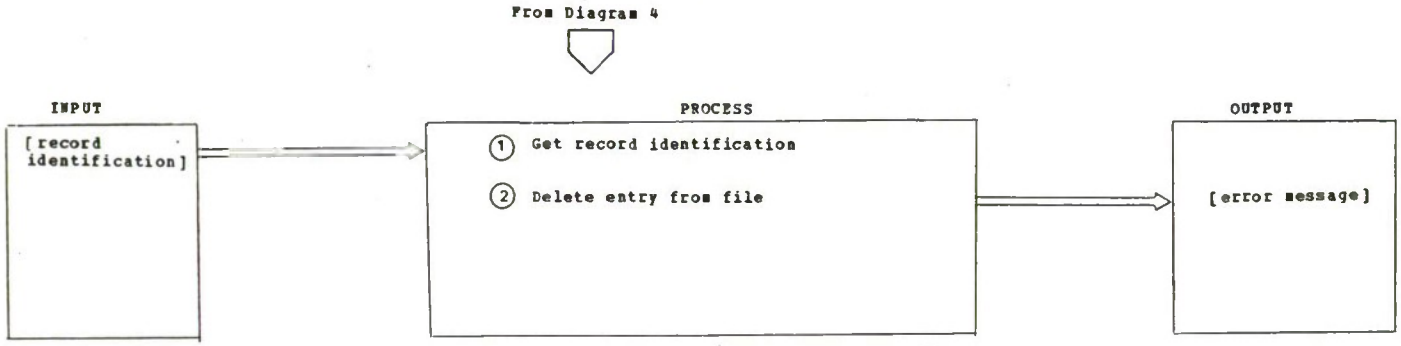
DIAGRAM 32 EDIT DATA RECORD

From Diagram 4



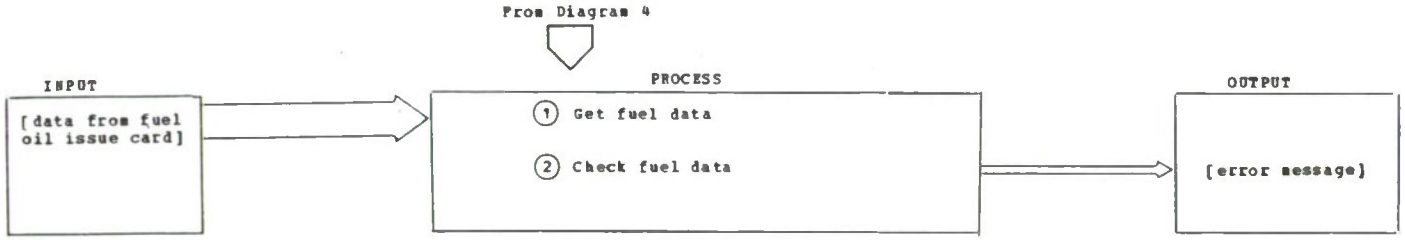
STEP	DESCRIPTION	PROGRAM
1	Display error message if invalid file name	EDIT
2	Display error message if record does not exist	EDIT
3		GETSTRING
4		GETSTRING
5		EDIT
6		EDIT

DIAGRAM 33 DELETE DATA RECORD



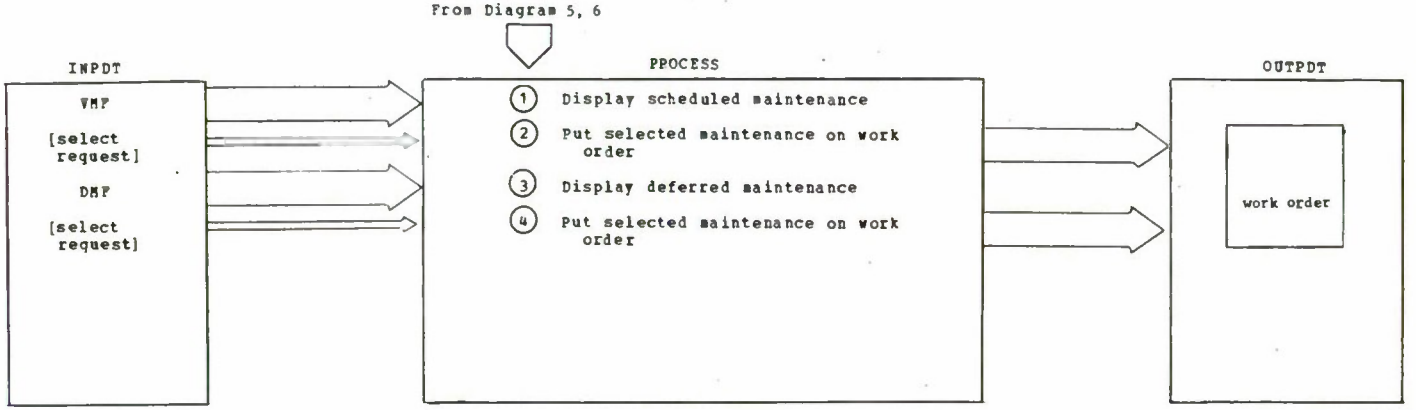
STEP	DESCRIPTION	PROGRAM
1	Record identification is either record key (name) or record number (logical location in file)	DELMASTER
2	Display error message if record does not exist	MDELETE

DIAGRAM 34 PROCESS FUEL/OIL ISSUE CARDS



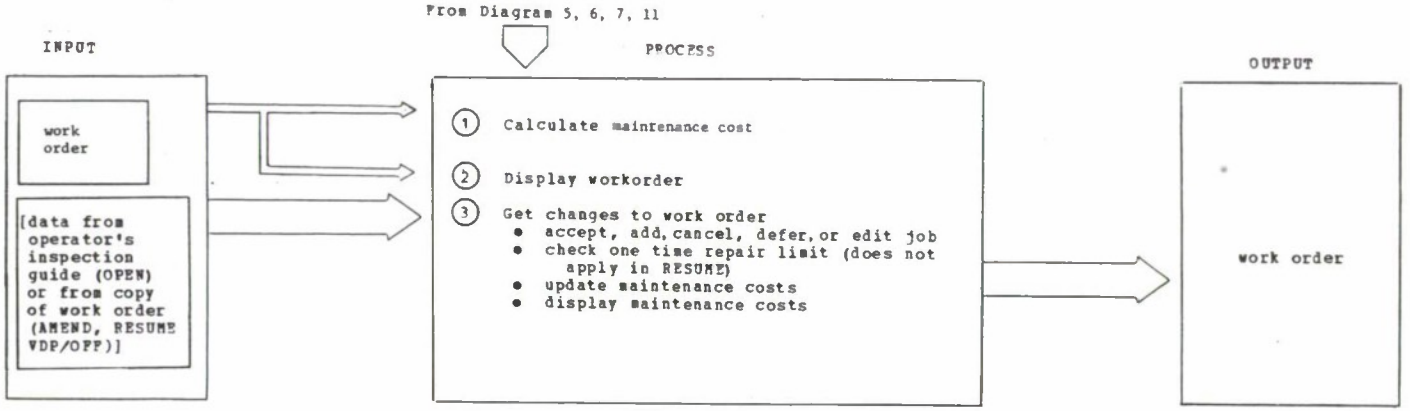
STEP	DESCRIPTION	PROGRAM
1	Get data card by card	GETFUEL
2	Check if vehicle in fleet, valid date, fuel and oil	GETFUEL

DIAGRAM 35 GET SCHEDULED AND DEFERRED MAINTENANCE



STEP	DESCRIPTION	PROGRAM
1	Display any due and overdue scheduled maintenance not previously assigned to work order	DISPMAINT
2	Add selected jobs to work order	DISPMAINT
3	Display any deferred maintenance on vehicle not previously on work order	DISPMAINT
4	Add selected jobs to work order	DISPMAINT

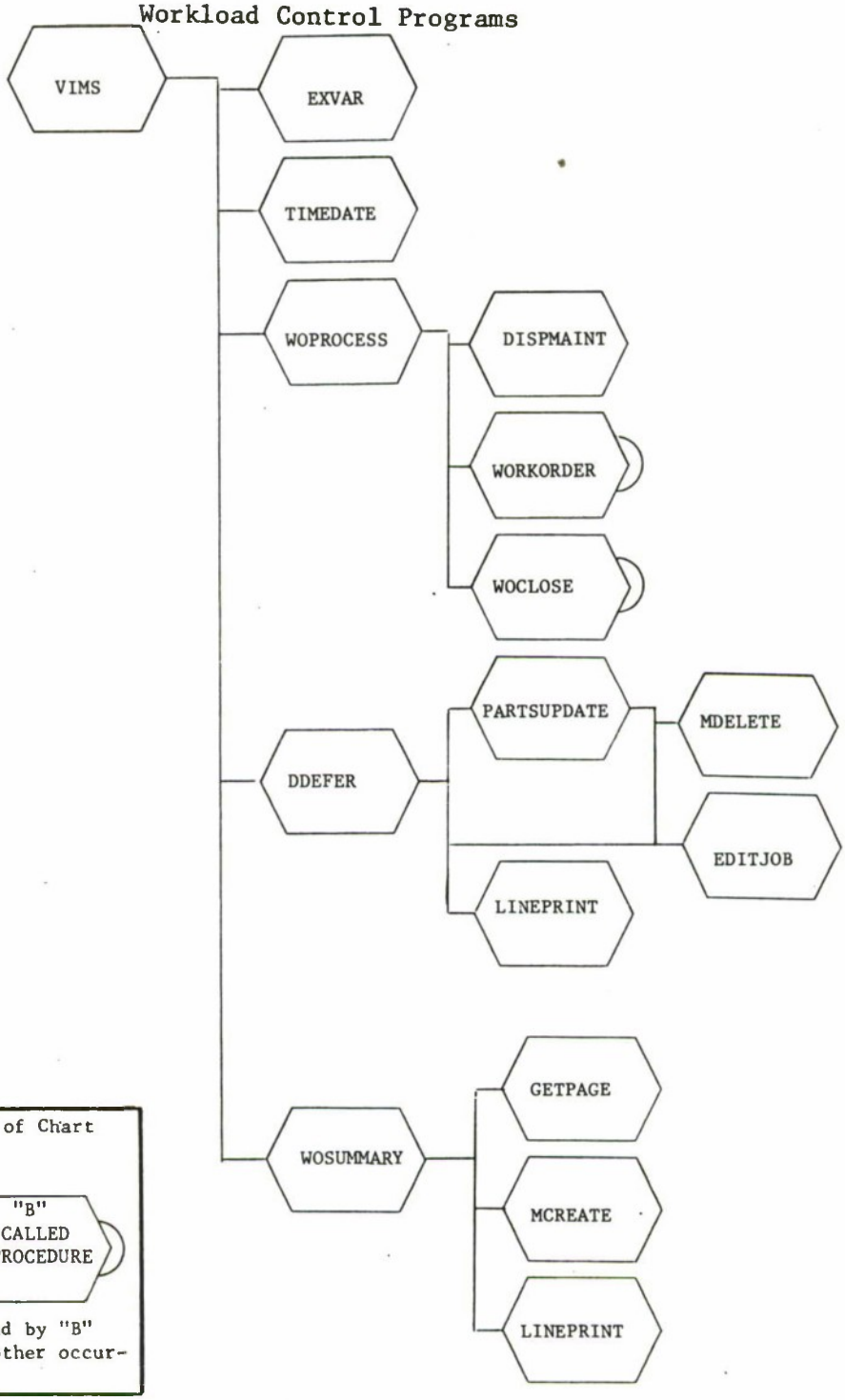
DIAGRAM 36 GET ADDITIONS AND CHANGES TO WORK ORDER



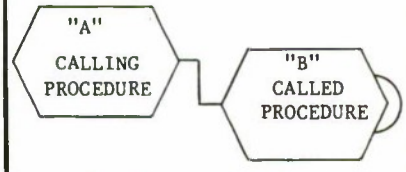
STEP	DESCRIPTION	PROGRAM
1	Calculate maintenance costs nsing material costs, standard hours, and wage rate for each job on work order	UPDATECOSTS
2	Display work order format, static data for vehicle, work order, and maintenance costs	WORKORDER
3	Pepeat until user signals ead of trans-action	PROCESSJOB EDITJOB

Chart I

Workload Control Programs



Explanation of Chart



Procedures called by "B" are shown at another occurrence of "B".

Chart I (Continued)

Workload Control Programs

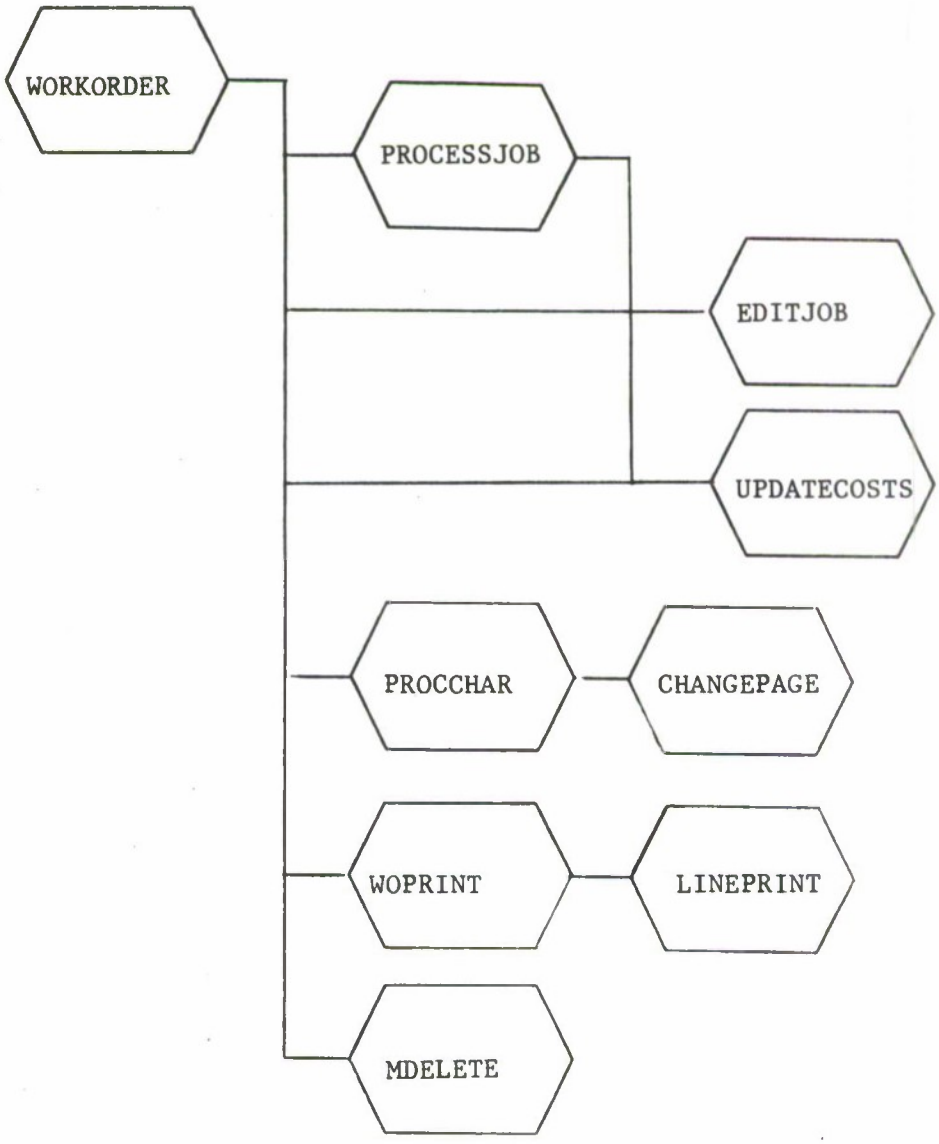


Chart I (Concluded)

Workload Control Programs

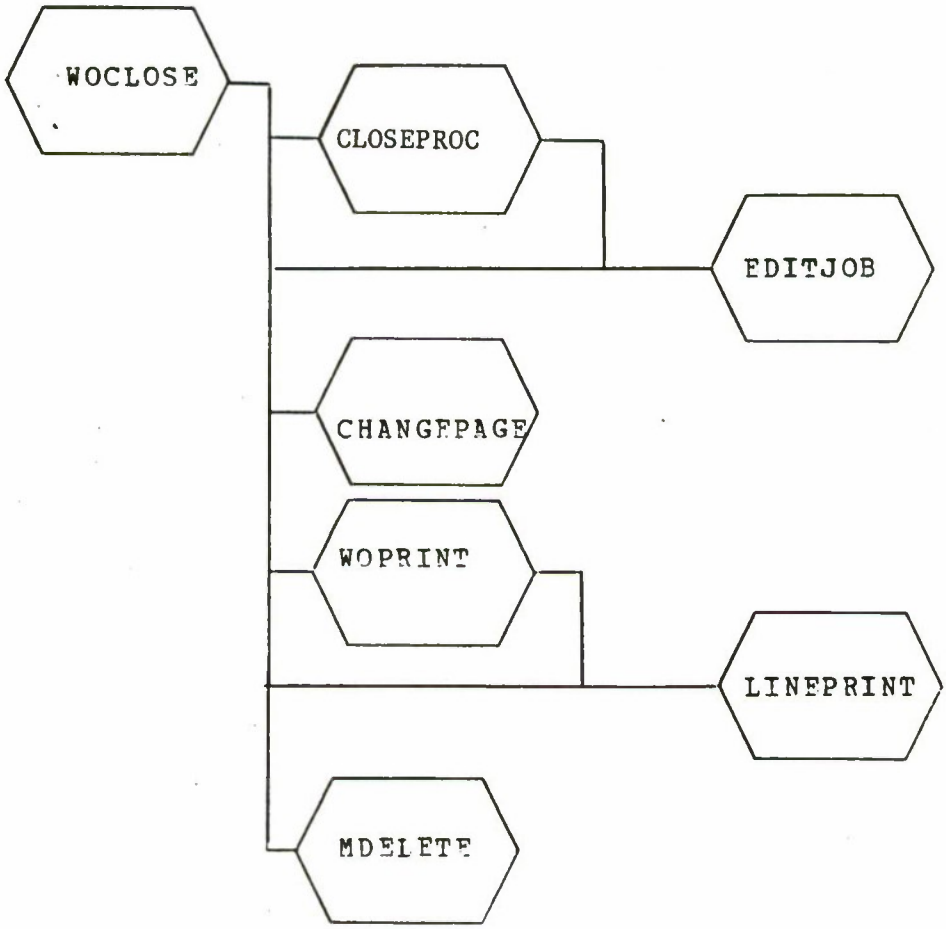


Chart II

Materiel Control Programs

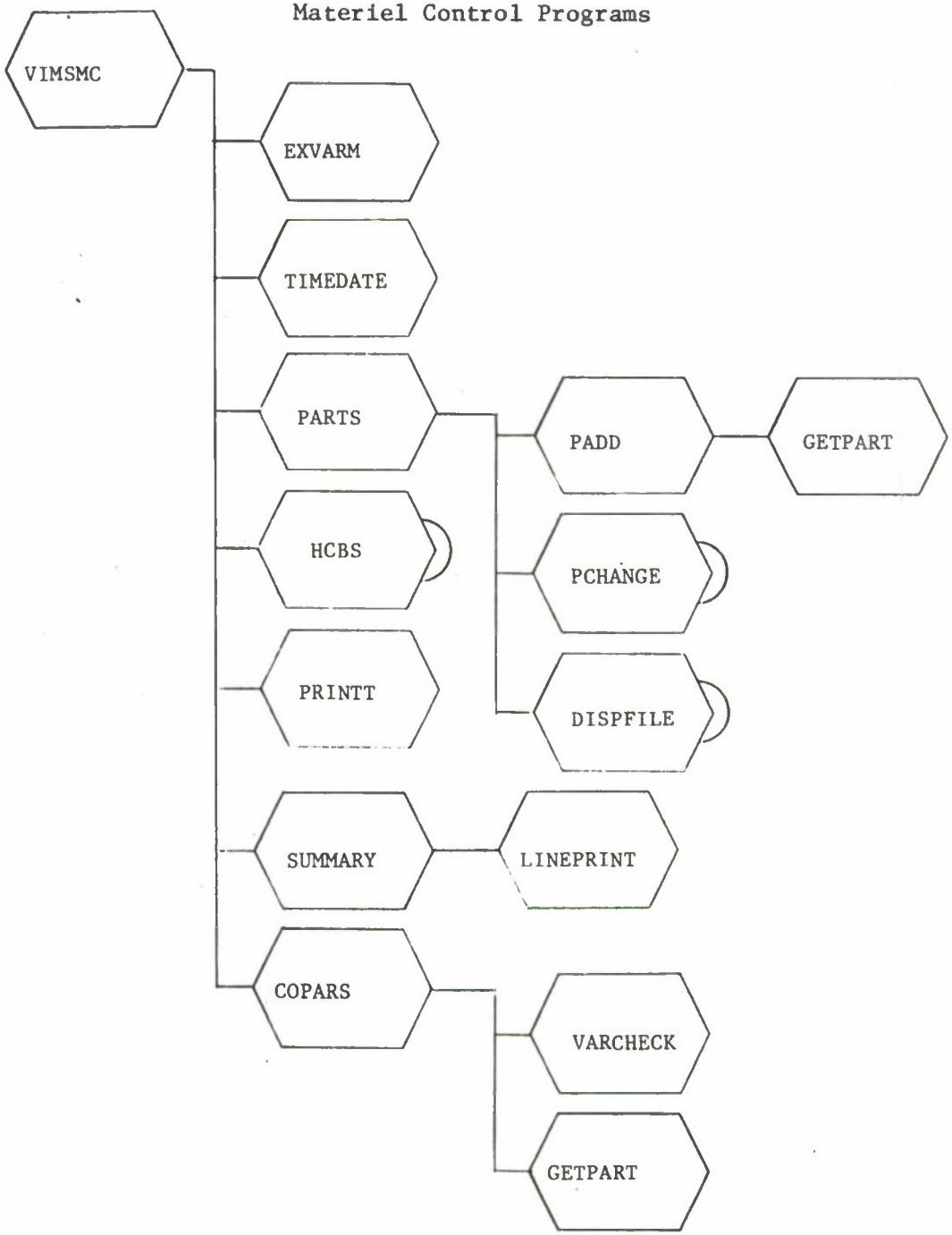


Chart II (Continued)

Materiel Control Programs

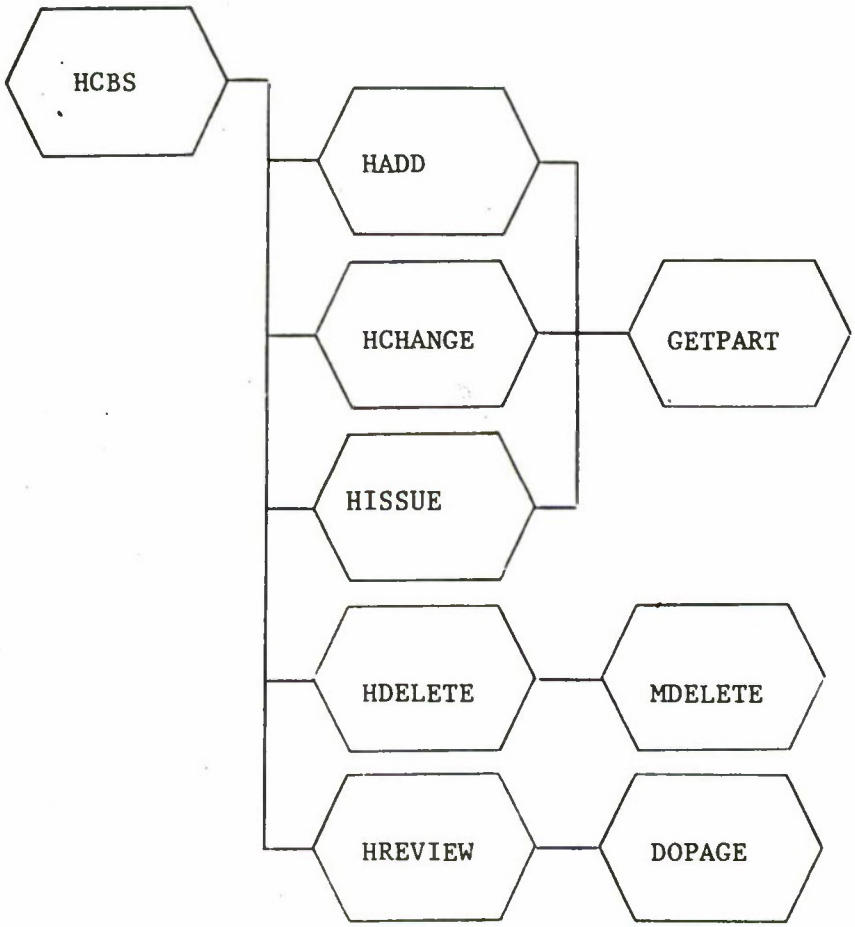


Chart II (Continued)

Materiel Control Programs

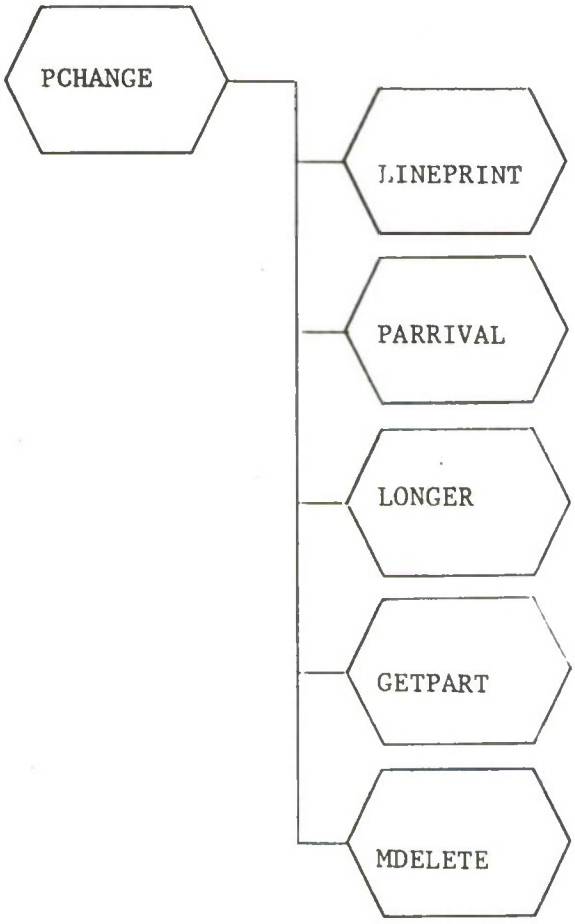


Chart II (Concluded)

Materiel Control Programs

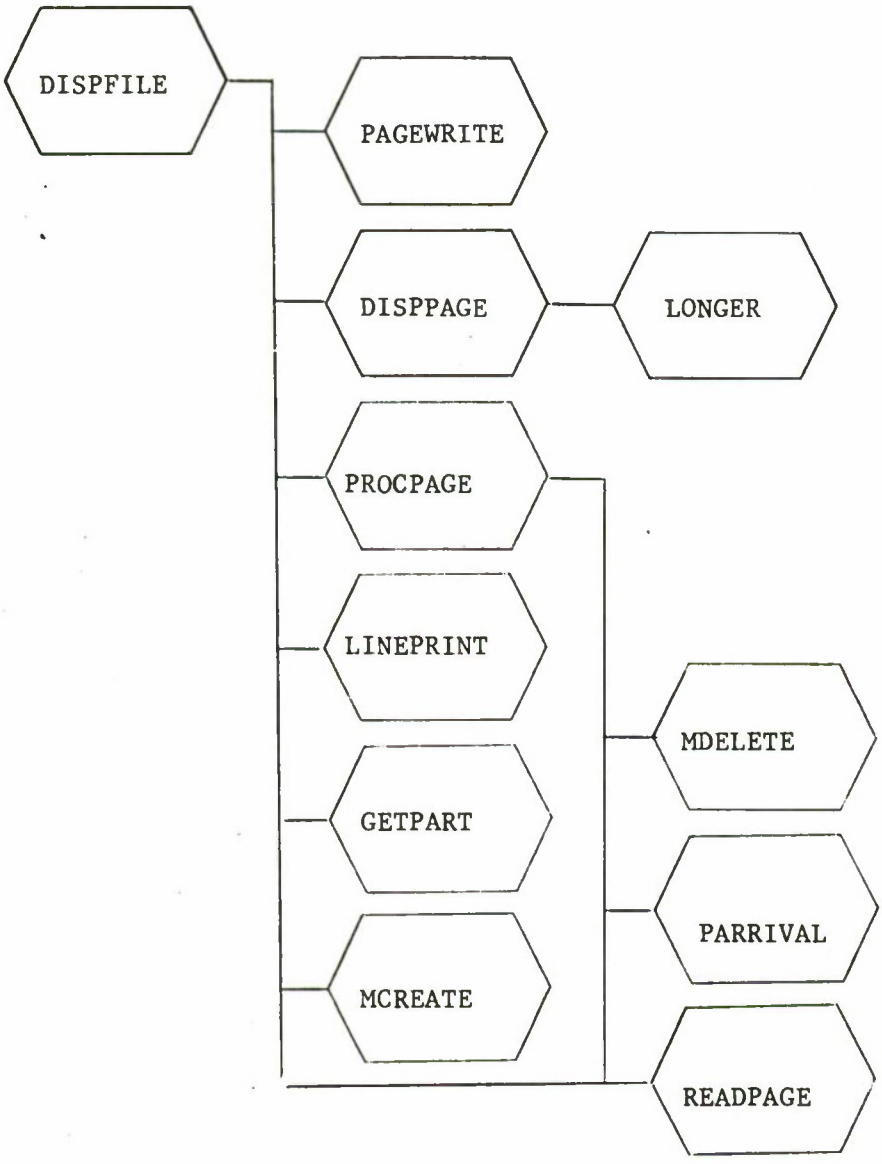
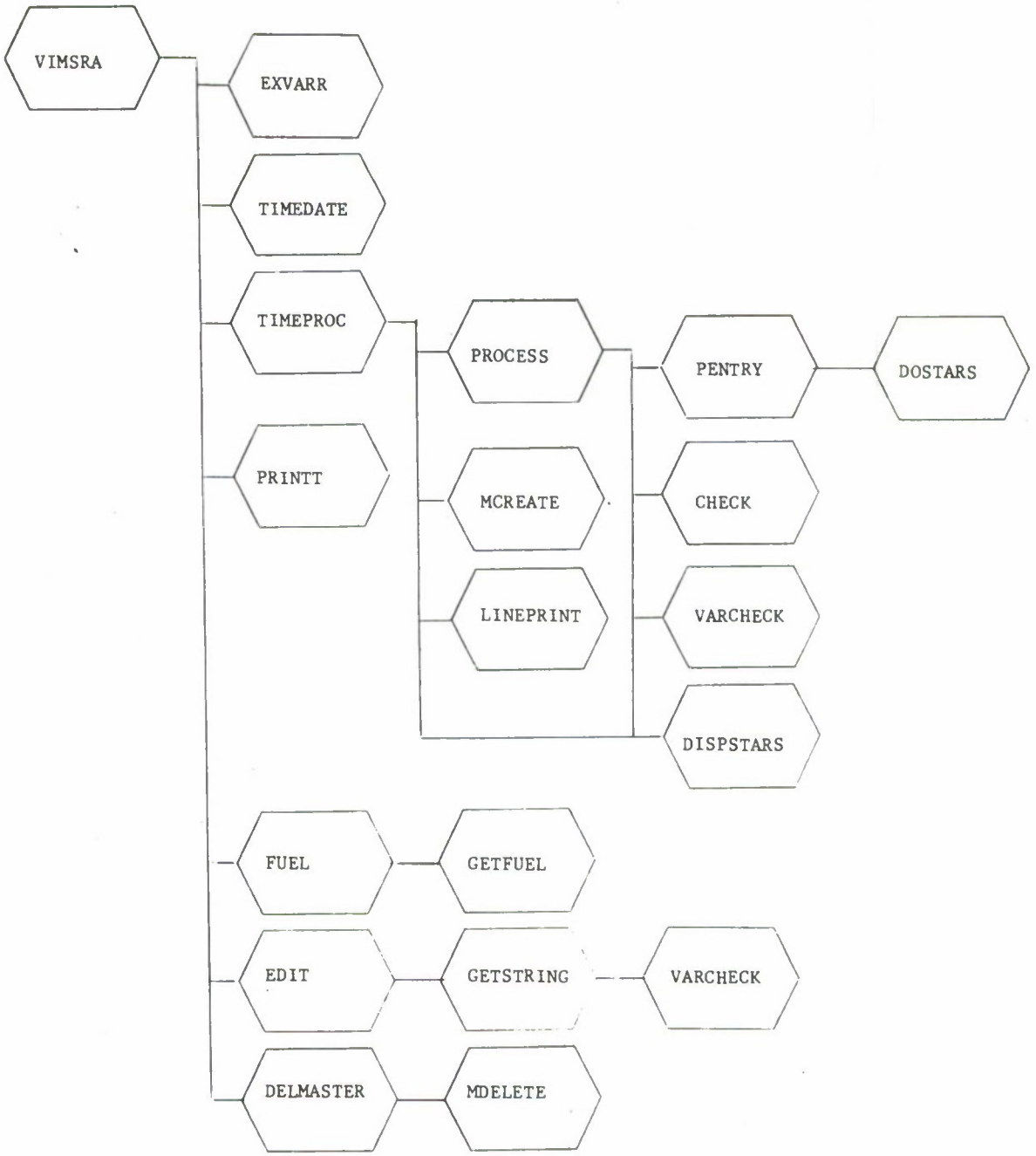


Chart III

Reports and Analysis Programs



APPENDIX IV

VIMS MODEL UTILITY PROGRAMS

The following utility programs are used by the VIMS transaction programs to accomplish basic system generation, I/O, and file maintenance functions.

Function Name: BOPFWRITE

Purpose: To print a formatted copy of the Back-Ordered Parts File suitable for use by Materiel Control as a daily summary.

Arguments: None.

Notes: This program is a stand-alone module which is called by typing BOPFWRITE at the TTY console.

Function Name: CCPX

Purpose: To determine cursor position on CRT screen.

Arguments:

ADDRESS(X) integer; pointer to an integer variable to receive character position (Column: 1-80).

ADDRESS(Y) integer; pointer to an integer variable to receive line position (Row: 1-27).

Return Values:

0 initiated.
1 not in configuration.
2 in use.

Notes: The cursor may be positioned by command string to CWR. CCPX is a query function only. (1,1) is the HOME position.

Function Name: CDEF

Purpose: To define line termination characters for TTY mode CRT read commands.

Arguments:

ADDRESS(LIST)	integer; address of string, or pointer to first character of list of delimiters.
CT	integer; number of delimiters, or length of string.

Return Values:

0 defined.
1 not in configuration.
2 in use.

Notes: A count of zero may be given to remove all previous terminator characters defined and to rely solely on maximum count exhaustion for CRT TTY mode reads.

Function Name: CGET

Purpose: To force a memory read and unconditional CRT memory transfer without operator action.

Arguments:

ADDRESS(BUF) integer; pointer to buffer to receive characters.

ADDRESS(CT) integer; pointer to variable to receive count of characters received.

Return Values:

0 initiated.
1 not in configuration.
2 in use.

DCHK Return Values:

4 no response.
5 no STX; wrong mode?

Notes: No parity; packed; keyboard locked while XMIT under way, as with all Delta Data operations.

Function Name: CPOS

Purpose: To position the cursor on the CRT.

Arguments:

X integer; x coordinate.

Y integer; y coordinate.

Function Name: CREAD

Purpose: To allow entry of characters to the CRT screen in teletype mode.

Arguments:

STG	string; name of the string to receive the CRT input.
MAX	integer; maximum number of characters to be read.

Notes: MAX is usually 1 because CREAD is used to monitor the entry of control characters from the CRT.

Function Name: CREAT

Purpose: To create a VIMS file header.

Arguments:

FNAME	string; name given to the file to be CREATed.
HEADERSIZE	integer; size of the header blocks within the file.
NODESIZE	integer; size of the nodes within the file.
DIRSIZE	integer; size of the file directories.
ELSIZE	integer; size of the directory elements.

Notes: CREAT must initialize a new file before the file may be accessed by the VIMS file management programs. This program is a stand-alone module which is called by typing the following command at the teletype console.

CREAT FNAME HEADERSIZE NODESIZE DIRSIZE ELSIZE ↵

Function Name: CTRE

Purpose: To read a line (defined by CDEF) in TTY mode, half duplex; to get character-by-character inputs from the CRT (keyboard) until an end-of-line character is received or the maximum count is exhausted.

Arguments:

ADDRESS(BUFFER)	integer; pointer to buffer to receive characters.
ADDRESS(CT)	integer; pointer to variable to receive count of characters actually received.
MAX	integer; maximum number of characters to accept.

Return Values:

0 initiated.
1 not in configuration.
2 in use.

Function Name: CWR[5]

Purpose: To write a string to the CRT, including any control characters.

Arguments:

ADDRESS(BUFFER)	integer; address of character string, or pointer to first character to be transmitted.
CT	integer; count of characters or length of string.

Return Values:

0 initiated.
1 not in configuration.
2 in use.

Notes: All CRT functions may be performed via CWR, in either TTY or TYPE mode.

Function Name: DADD

Purpose: To add a device to the non-system configuration, and identify its interrupt service routine to the operating system. Analogous to a file open command.

Arguments: D integer; a device number.

Return Values:
0 device added.
1 device not available.
2 device already in configuration.

Notes: I:= DADD(1); adds the CRT station.

Function Name: DCHK

Purpose: To synchronize completion of I/O requests with further processing. Checks if the outstanding request on a device is complete and suspends the user until it is. Returns immediately if no I/O requests are pending.

Arguments: D integer; a device number.

Return Values:
0 I/O is complete; arguments used in the transfer request are well-defined and usable.
1 device not in configuration.
2 a device-specific error condition has occurred (listed by device).

Notes: Task suspension is explained in Data General RDOS documentation; basically, a task is suspended when it must wait for I/O completion before continuing to allow overlapped tasks or other users to share the processor.

Function Name:	DISPLAY
Purpose:	To display the specified string on the CRT screen.
Argument:	STG string; the string to be displayed.
Notes:	Uses function CWR.

Function Name:	EXVAR, EVVARR, EXVARM
Purpose:	To allocate space for the external variables used in VIMS, VIMSRA, and VIMSMC respectively.
Arguments:	None.
Notes:	EXVAR, EXVARR, EXVARM must be called before any external variables may be referenced by the main programs.

Function Name:	GETRESP
Purpose:	To get an edited line of input from the CRT.
Arguments:	RESPONSE string; the resulting input string.
Notes:	Keyboard is enabled only when program expects user's input. The user may edit string by hitting the cursor left (←) key to delete the last character input. Uses function CTRE.

Function Name: HCBSWRITE

Purpose: To print a formatted copy of the High Cost Bench Stock Master File suitable for use as a Materiel Control reference document.

Arguments: None.

Notes: This program is a stand alone module which is called by typing HCBSWRITE at the TTY console.

Function Name: LINEPRINT

Purpose: To print a CRT screen image at the high speed line printer.

Arguments: FF string; form feed request.

Function Name: MDELETE

Purpose: To delete a specified record from the selected disk file.

Arguments: FNAME string; name of the file.

ENTRYID string; identification of the record to be deleted.

Return Values: BOOL boolean; delete indicator.

Notes: ENTRYID may be a record key (name) or a record number (logical position in file).

Function Name: MREAD

Purpose: To read a selected record from a disk file into a working area.

Arguments:

FNAME	string; name of the input file.
ENTRYID	string; identifier of the record to be input.
STG	string; name of the string to receive the input.

Notes: ENTRYID may be a record key (name) or a record number (logical position in file).

Function Name: MWRITE

Purpose: To write a specified string to a selected disk file.

Arguments:

FNAME	string; name of the output file.
ENTRYID	string; identifier of the output record.
STG	string; name of the string to be written.

Notes: ENTRYID may be a record key (name) or a record number (logical position in file).

Function Name: OLLD

Purpose: To conditionally load an overlay node.

Arguments: ARG1 integer; overlay number.

ARG2 integer; node number.

Notes: Uses RDOS system call .OVLOD.

Function Name: OLOP

Purpose: To open the overlay channel.

Arguments: None.

Notes: Uses RDOS system call .OVOPN.

Function Name: PRINT

Purpose: To print a formatted copy of the desired file at the high speed printer.

Arguments: FNAME string; name of the file to be printed.

Function Name: TIMEDATE

Purpose: To read the system time-of-day clock and calendar providing the equivalent of RDOS GTOD and GDAY commands to an ALGOL program.

Arguments:

HR integer; variable to receive hours (0-23).

MIN integer; variable to receive minutes (0-59).

SEC integer; variable to receive seconds (0-59).

DAY integer; variable to receive day (1-31).

MO integer; variable to receive month (1-12)

YR integer; variable to receive year, e.g. 75.

Notes: Uses RDOS calls .GTOD and .GDAY; Year is adjusted to normal A.D. from the RDOS form given in the system call, thus corresponding to the time entered at bootstrap time or via the RDOS .STOD.

Function Name: WAIT

Purpose: To suspend the user for a specific duration.

Arguments: TIME integer; time to delay in seconds.

Notes: Uses RDOS call .DELAY.

APPENDIX V

DATA FILE FORMATS

This appendix contains descriptions of the files used in the VIMS development model. Below is a list of the file names, the file name abbreviations, and the page numbers where the files are described in the appendix.

Filename: Back-Ordered Parts File

Description: This file contains information on parts that have been ordered to complete deferred jobs. Parts in this file correspond to jobs in the Deferred Maintenance file.

File Type: Variable length record
Header size: 0 bytes
Node size: 57 bytes

Record Key: Vehicle registration number

Record Format: Each node of the record has the following format.

Data Element	Length in bytes
work order number	(4)
"A"	(1)
job number	(2)
part description	(26)
quantity	(3)
unit cost	(6)
maintenance or operations indicator	(1)
source code	(1)
date ordered	(5)
date due or received	(5)
bin location	(3)

Filename: COPARS File

Description: This file contains COPARS sales slip data.

File type: Variable length record
Header size: 0 bytes
Node size: 57 bytes

Record Key: Work order number

Record Format: Each node of the record has the following format.

Data Element	Length in bytes
part number	(10)
quantity	(3)
list price	(6)
discount percentage	(2)
back-ordered part indicator	(1)
warranty indicator	(1)
days warranty	(3)
miles warranty	(5)
part description	(20)
cost	(6)

Filename: Deferred Maintenance File

Description: This file contains all deferred and VDP job information for the vehicle fleet.

File Type: Variable length record
Header size: 0 bytes
Node size: 107 bytes

Record Key: Vehicle registration number

Record Format: Each node of the record has the following format.

Data Element	Length in bytes
work order number	(4)
job number	(2)
action code	(3)
system code	(3)
charge code	(1)
work center	(3)
job description	(76)
material cost	(3)
standard hours	(4)
bin location	(3)
date deferred	(5)

Filename: Employee Master File

Description: This file contains all static information about employees working at the maintenance facilities.

File Type: Fixed length record
Record size: 39 bytes

Record Key: Employee SSAN

Record Format:

Data Element	Length in bytes
Employee name	(25)
Employee SSAN	(9)
work center	(5)

Filename: Employee Master Suspense File

Description: This file holds all time card input until the end of a TIME/INPUT session. Then the EMSF is error checked, time updates are accomplished, and the EMSF is cleared for reuse.

File Type: Variable length record
Header size: 39 bytes
Node size: 80 bytes

Record Key: Employee SSAN

Record Format: The record header has the following format.

Data Element	Length in bytes
employee name	(25)
employee SSAN	(9)
work center	(5)

Each node contains up to five of the following entries.

Data Element	Length in bytes
date	(5)
work order number	(4)
job number	(2)
shift code	(1)
actual time	(4)

Filename: Error Suspense File

Description: This file contains all employee time card records which have errors in them.

File Type: Variable length record
Header size: 39 bytes
Node size: 80 bytes

Record Key: Employee SSAN

Record Format: The record header has the following format.

Data Element	Length in bytes
employee name	(25)
employee SSAN	(9)
work center	(5)

Each node contains up to five of the following entries.

Data Element	Length in bytes
date	(5)
work order number	(4)
job number	(2)
shift code	(1)
actual time	(4)

Filename: Error Suspense File Star File

Description: This file contains the screen coordinates of the stars which appear on the displays and printouts of Error Suspense File entries. There is an Error Suspense File Star File record for each Error Suspense File record.

File Type: Variable length record
Header size: 0 bytes
Node size: 60 bytes

Record Key: Employee SSAN

Record Format: Each node contains up to fifteen of the following entries.

Data Element	Length in bytes
line number	(2)
field number	(2)

Filename: Format and Literal File

Description: Each of the three VIMS functional areas has its own Format and Literal File. These files contain all of the large formats and literals which are required by the transactions.

File Type: Variable length record
Header size: 0 bytes
Node size: 80 bytes

Record Key: Transaction abbreviation concatenated with a sequence number.

Record Format: Unformatted.

Filename: High Cost Bench Stock Master File

Description: This file contains information on the parts which are declared to be High Cost Bench Stock.

File Type: Fixed length record
Record size: 45 bytes

Record Key: Item number

Record Format:

Data Element	Length in bytes
federal stock number	(15)
unit price	(6)
EEIC	(3)
charge code	(1)
description	(20)

Filename: Vehicle Historical File

Description: This file contains the maintenance history of all vehicles on the base which are serviced at the maintenance facilities.

File Type: Variable length record
Header size: 0 bytes
Node size: 72 bytes

Record Key: Vehicle registration number

Record Format: Each node contains up to six of the following data entries.

Data Element	Length in bytes
system code	(2)
"S"	(1)
sub-system code	(1)
date	(5)
action taken	(3)

Filename: Vehicle Master File

Description: This file contains all static information on the vehicles being maintained within the VIMS system. Scheduled maintenance data and special maintenance indicators are also included in each vehicle's master record.

File Type: Fixed length record
Record size: 265 bytes

Record Key: Vehicle registration number

Record Format:

Data Element	Length in bytes
management code	(4)
make/type	(7)
one time repair limit	(5)
miles/hours	(6)
scheduled maintenance indicator	(1)
deferred maintenance indicator	(1)
VDP indicator	(1)
work order number	(4)
miles/hours indicator	(1)
R/D code	(1)
mileage exceeded indicator	(1)
age exceeded indicator	(1)

The following data elements are included in the record for each of the four scheduled maintenance categories.

Data Element	Length in bytes
action code	(4)
system code	(3)
charge code	(1)
work center	(3)
job description	(20)
material cost	(3)
standard hours	(4)
due/overdue	(1)
miles/hours	(6)
date	(5)
miles/hours interval	(6)
month interval	(2)

Filename: Warranty File

Description: This file contains warranty information for all guaranteed replacement parts.

File Type: Variable length record
Header size: 0 bytes
Node size: 49 bytes

Record Key: Vehicle registration number

Record Format: Each node of the record has the following format.

Data Element	Length in bytes
part number	(10)
part description	(20)
date installed	(6)
mileage	(5)
warranty period (days)	(3)
warranty period (miles)	(5)

Filename: Work Order File

Description: This file contains all open, suspended, and closed work orders for the maintenance facility.

File Type: Variable length record
Header size: 63 bytes
Node size: 94 bytes

Record Key: Work order number

Record Format: The record header has the following format.

Data Element	Length in bytes
registration	(8)
date received	(5)
time received	(4)
management code	(4)
make/type	(7)
date released	(5)
time released	(4)
R/D code	(1)
mileage exceeded indicator	(1)
age exceeded indicator	(1)
priority	(2)
miles/hours	(6)
user phone	(8)
work order prefix	(1)
work order type	(1)
one time repair limit	(5)

Each node has the following format.

Data Element	Length in bytes
action code	(4)
system code	(3)
charge code	(1)
work center	(3)
job description	(76)
material cost	(3)
standard hours	(4)

Filename: Work Order Scratch File

Description: This file contains employee time card information sorted with respect to work order number and job number. There is a Work Order Scratch File record for every Work Order File record, and that scratch record contains all manhour information for that work order.

File Type: Fixed length record
Record size: 84 bytes

Record Key: Work order number

Record format:

Data Element	Length in bytes
number of jobs open	(2)
number of days to hold/status	(2)
job 1 standard hours	(4)
job 1 actual hours	(4)
job 2 standard hours	(4)
job 2 actual hours	(4)
job 3 standard hours	(4)
job 3 actual hours	(4)
job 4 standard hours	(4)
job 4 actual hours	(4)
job 5 standard hours	(4)
job 5 actual hours	(4)
job 6 standard hours	(4)
job 6 actual hours	(4)
job 7 standard hours	(4)

job 7 actual hours	(4)
job 8 standard hours	(4)
job 8 actual hours	(4)
job 9 standard hours	(4)
job 9 actual hours	(4)
job 10 standard hours	(4)
job 10 actual hours	(4)