MA–010–001–2H

GEORGE C. MARSHALL SPACE FLIGHT CENTER

(NASA-CR-144093) MIRADS-2 USER'S MANUAL
(Computer Sciences Corp.) 217 p HC $7.75
CSCL 05B

N76-13954

Unclas
G3/82 06067

MARSHALL INFORMATION RETRIEVAL AND DISPLAY SYSTEM

MIRADS-2 USER'S MANUAL

JUNE 1975

NATIONAL AERONAUTICS AND SPACE ADMINISTRATION

MIRADS-2
USER'S MANUAL

Prepared By

COMPUTER SCIENCES CORPORATION
INDUSTRIAL MANAGEMENT SYSTEMS DEPARTMENT

For

MSFC COMPUTATION LABORATORY
DATA CENTER DIVISION

Under

Contract No. NAS8-21805

June 1975

TABLE OF CONTENTS

TABLE OF CONTENTS (Continued)

Section 3 (Continued)

TABLE OF CONTENTS (Continued)

Section 3 (Continued)

LIST OF ILLUSTRATIONS

Figure

LIST OF TABLES

SECTION 1 - INTRODUCTION

1.1 GENERAL

The Marshall Information Retrieval and Display System (MIRADS) is an on-line data storage and retrieval system which allows the user to extract and process information from stored Data Bases. The use of remote terminals to extract and display data from the Data Bases provides a fast and responsive method of obtaining needed information. This on-line processing eliminates the extensive data processing cycle normally required when using an off-line or batch processing mode. The system consists of a general purpose computer program containing several functional modules that provide the overall capabilities of the total system.

MIRADS is a command driven system with on-line editing of user commands which provides error diagnostic messages and recovery procedures that assist in the utilization of the system. Each Data Base must have a Dictionary which describes the structure of the Data Base to MIRADS. It should be noted here that no reprogramming of MIRADS is necessary to add any number of Data Bases to the system. Normally, when a user implements the system or adds a new file, a MIRADS analyst will assist in the preparation of the Dictionary and will instruct the user in the use of the system.

MIRADS utilizes the UNIVAC 1108 computer system operating under the standard EXEC 8 Operating System. Figure 1-1 basically illustrates the usage of the system. Commands or requests are transmitted via communication lines to the computer after being entered by the user through a DCT-500 or Uniscope (Cathode Ray Tube) terminal. The computer, utilizing the MIRADS programs and the previously stored user's data files, performs the necessary functions in response to the user's commands or requests and returns the reply to the user via the DCT-500 or Uniscope terminal.

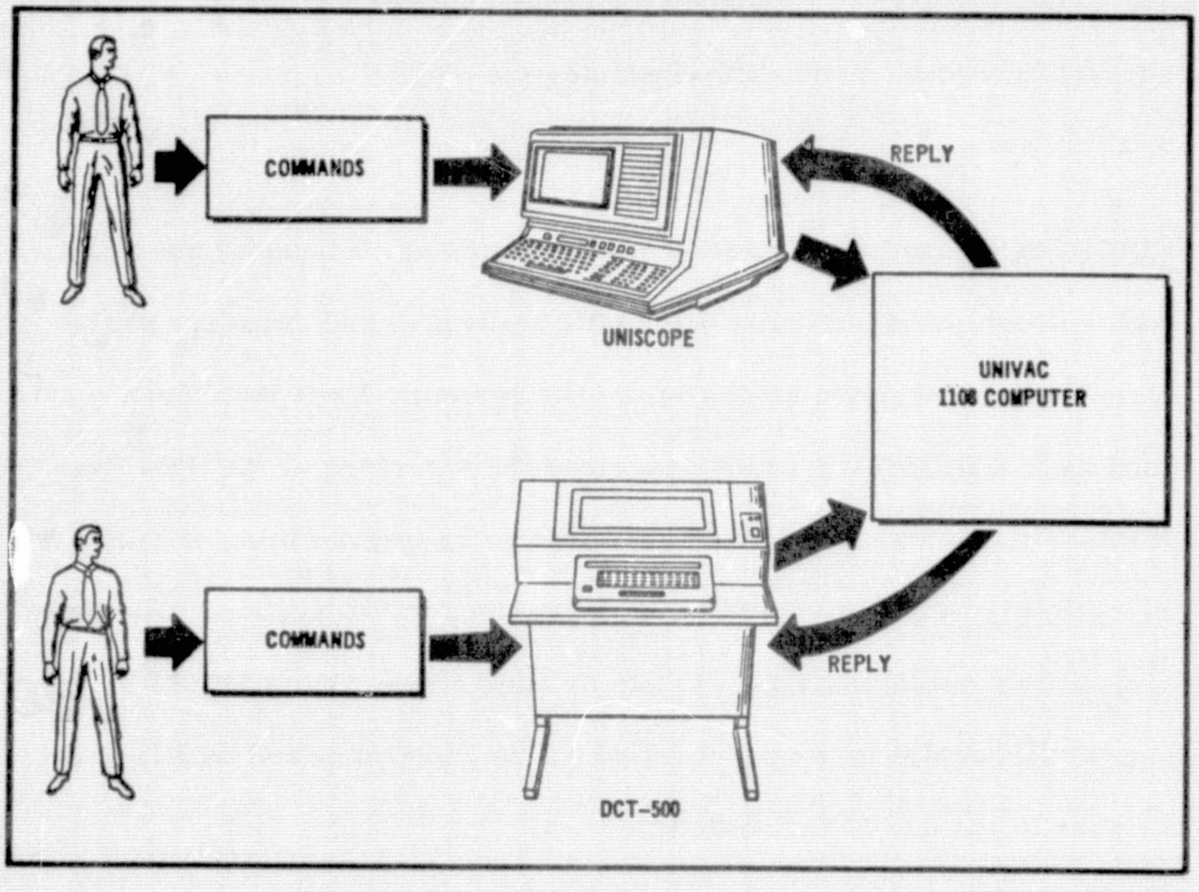


Figure 1-1. MIRADS System Usage

MIRADS provides a highly diversified choice of data processing functions to satisfy user requirements. MIRADS consists of several programs, each performing a series of distinct functions. These programs are referred to as modules. A brief description of MIRADS and its associated modules is given in the following paragraphs. A more detailed discussion is presented in Section 3.

1.2 MIRADS

MIRADS provides the user the capability to select and process variable data without time consuming sequential searches. In response to a user query, the system searches the specified Data Base, sorts the Data Base into a specified order, performs simple or complex computations, prints or displays the results, and updates the Data Base. The modules which are contained in MIRADS are briefly described in the following paragraphs.

1.2.1 Query Module

The Query Module accepts user specified search conditions and retrieves all records meeting the specified conditions from the user Data Base. The retrieved records can then be used by other modules of the system to satisfy user requirements.

1.2.2 Sort Module

The Sort Module provides the capability of reordering output reports into user specified sequences. Major and minor sorts in ascending or descending order may be performed.

1.2.3 Compute Module

The arithmetic functions of addition, subtraction, division, multiplication, exponentiation, counting, and summing are available to the user through the use of the Compute Module. The module can also be used to calculate variable information which is not normally contained in Data Bases.

1.2.4 Print Module

The Print Module allows the user to print output reports in tabular or matrix form. The user has the capability of specifying desired print formats, selecting and printing a report title or heading, supplying actual values for coded data element values which are stored in coded form in the Data Base, and controlling the volume of output received at the terminal.

1.2.5 Format Module

The Format Module allows the user to predefine an output display consisting of from one to three different print formats. The options which are available in the Print Module are also available in the Format Module with an additional feature of being able to format a display covering an entire page for each Data Base record retrieved by the Query Module. The Format Module is generally used in lieu of the Print Module whenever an output report is required which is similar to full page reports generated by dedicated batch processing programs.

1.2.6 Save Module

The Save Module allows the user to retain a Query Set for execution at a later time. This eliminates the necessity of inputting each command within the Query Set for repetitive executions.

1.2.7 Update Module

The Update Module enables the user to perform on-line modification to his Data Base. Functions allowed in update are change, insert, replace, and delete.

SECTION 2 - USER TERMINAL OPERATION PROCEDURES

2.1 INTRODUCTION

MIRADS is designed to be used in an on-line mode of operation through the use of terminal devices. This section describes the detailed operation of the input/output devices which provide the user an interface with the system. At present, the system utilizes DCT-500 and Uniscope terminals. The DCT-500 produces hardcopy output and the Uniscope provides visual displays on a Cathode Ray Tube (CRT). Each type of terminal is controlled and operated from a keyboard which is similar to a standard typewriter keyboard. The remote terminals, conveniently located in the user's area, are interfaced with the computer by communication or telephone lines. Thus, these devices may be classified as remote dial-up devices since they require the user to initiate a telephone call to establish contact with the computer and MIRADS.

Formal training classes will be held prior to a user assuming terminal operation responsibility. These training classes are designed to inform the user of the capabilities of the system and how they may be used to satisfy user requirements. Also, a trained staff of personnel assigned to MIRADS is available to provide assistance in the use of the system.

2.2 DCT-500 OPERATION PROCEDURES

To properly utilize the DCT-500, the user must become familiar with the location and use of various control keys, control characters, indicators, and system messages. In addition to instructions contained in this manual, the user should refer to UNIVAC's Operators Reference Manual (UP-7832) for further information.

The Standard Trouble Reporting Procedure must be followed when problems other than those specifically covered are encountered during the step-by-step operation. This procedure is outlined in Section 2.5 of this manual.

MIRADS is designed so that messages received by the DCT-500 require the user to respond. The responses are in predetermined formats and must be entered by the user exactly as specified by the instructions; otherwise, the system will not operate properly. Directions for correcting normal errors are explained in the step-by-step instructions where feasible, or are covered in the Standard Trouble Reporting Procedure in Section 2.5.

The data which are typed on the DCT-500 must be transmitted to the computer by depressing the RETURN key. The user must remember this to prevent terminal time-outs and to eliminate the redundancy of stating it each time it is required in the step-by-step instructions. Terminal time-outs are explained in Section 2.5.

2.2.1 DCT-500 Control Switches and Indicators

The DCT-500 control switches located on the control panel are used to control the overall operation of the terminal. The control panel is located at the top of the typewriter keyboard. Figure 2-1 illustrates the control switches and indicators which are located on the control panel. All of the control switches found on the control panel are not used by MIRADS. For this reason, only those switches and indicators which are pertinent to MIRADS are identified.

1. BAUD RATE

   This switch is used to control the transfer speed of data between the terminal and computer. The three settings for BAUD RATE are:

110	Sets the DCT-500 transmit and receive data rate at ten characters per second.
150	Sets the DCT-500 transmit and receive data rate at fifteen characters per second.
300	Sets the DCT-500 transmit and receive data rate at thirty characters per second.

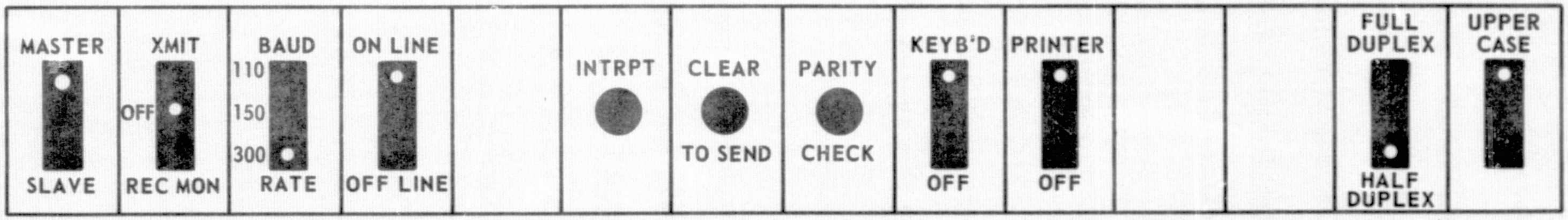


Figure 2-1. Control Panel for DCT-500

The BAUD RATE should be set at 300 before the user attempts to dial-up the DCT-500.

2. ON LINE/OFF LINE

   This switch is used to connect the DCT-500 input and output devices to the terminal-to-computer communication interface (phone line). This switch must be set ON LINE.

3. KEYB'D OFF

   This switch is used to designate the keyboard as the DCT-500 input device. This switch should be in the UP position.

4. PRINTER OFF

   This switch is used to designate the printer as the DCT-500 output device. This switch should be in the UP position.

5. INTRPT

   When this indicator is lit, the interrupt indicator specifies that the computer cannot send or receive data. The DCT-500 will stop transmitting and receiving until an interrupt resume message is transmitted by the computer.

6. CLEAR TO SEND

   When this indicator is lit, this indicates that the communication interface has been established between the DCT-500 and the computer. The DCT-500 is now ready to receive the user's input commands.

7. PARITY CHECK

This indicator lights when the DCT-500 receives a bad data character from the computer. The DCT-500 will continue to receive the message; however, an asterisk (*) will be printed in place of the bad character.

2.2.2 DCT-500 Printer Controls and Indicators

The printer controls the indicators, as shown in Figure 2-2, are located at the right front of the DCT-500 terminal.

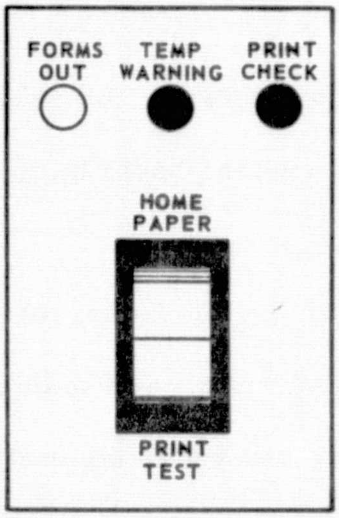


Figure 2-2. Printer Controls and Indicators

1. PRINT TEST

This switch enables the user to check the printer operation. If the OFF LINE switch is set, pressing the PRINT TEST switch causes a line of 132 E's to be printed, carriage return, and line feed.

2. HOME PAPER

   This switch, when depressed, advances the paper to the top of the next form.

3. FORMS OUT

   This light indicates that the printer is out of paper.

4. TEMP WARNING

   This light indicates excessive heat conditions inside the DCT-500 unit. The DCT-500 should be immediately powered down by removing the AC power cord from the 110V wall outlet. The condition should be reported to a MIRADS representative.

5. PRINT CHECK

   This light indicates that an electrical fuse has blown in the printer. The condition should be reported to a MIRADS representative.

2.2.3 Control Characters and Keys

1. END-IMAGE

   Each line typed on the DCT-500, regardless of length, is referred to as an image. An image is ended and transmitted to the computer by depressing the RETURN key. The maximum image length is 132 characters.

2. DELETE

EXEC VIII Level 31 deletions can be accomplished by first suppressing the control (CTL) key and then the 'X' key. This method of correcting error images will delete an entire line of input and the user must reinput the corrected data line. Character correction is explained through the use of the ERASE key.

3. ERASE

EXEC VIII Level 31 character deletions occur when the control (CTL) key is suppressed followed by the 'Z' key. This causes the last character typed to be erased from the image when transmitted to the computer. Any number of continuous characters may be erased with a like number of CTL/'Z' entries. Any desired character(s) can then be entered.

4. INTERRUPT

This character is represented by the key labeled INTRPT. This key should not be used without special instructions from MIRADS system personnel.

5. PROCEED

This key is depressed after the terminal-to-computer communication interface has been established to light the CLEAR TO SEND indicator. The CLEAR TO SEND indicator must be lighted before the user can input any commands into the DCT-500 terminal.

6. RETURN

   This key returns the carriage to home position and transmits the previously typed line to the computer.

7. SHIFT

   This key shifts the keyboard to uppercase.

8. CTL

   This key is used to send control messages to the EXEC VIII system.

2.2.4 DCT-500 System Messages

1. *NO RUN ACTIVE*

   This message is received by the DCT-500 when the correct procedure for initiating a computer run has not been used. A RUN statement must be entered and transmitted to the computer before continuing operations.

2. *WAIT - LAST INPUT IGNCRED*

   This message is received by the DCT-500 when user input is attempted, and the system is not ready to accept the data. The input data must be reentered and transmitted to the computer at the end of the next message received at the DCT-500 or after the message READY is received.

3. READY

This message, when received by the DCT-500, indicates that the system is ready to receive input.

4. *TIMEOUT WARNING*

This message is received by the DCT-500 when no input or output messages have been processed by the DCT-500 for a period of at least five minutes. If additional input or output messages are not received by the DCT-500 in the next five minutes (after the *TIMEOUT WARNING* message is first received), the computer system will automatically terminate the user's RUN and disconnect the terminal from the computer. Users are cautioned that a terminal must be in use without 10 minute delays to prevent *TIMEOUT WARNING* run termination.

5. *TERMINAL INACTIVE*

This message, when received by the DCT-500, indicates that the terminal is no longer interfaced with the computer. A RUN statement must be entered to reestablish the interface.

2.2.5 DCT-500 Initiation Procedure

1. Check the external line plug to ensure it is properly connected to an 110V AC electrical outlet.

2. Depress the circuit breaker pushbutton switch. (See Figure 2-3). This is located at the back of the DCT-500 terminal.

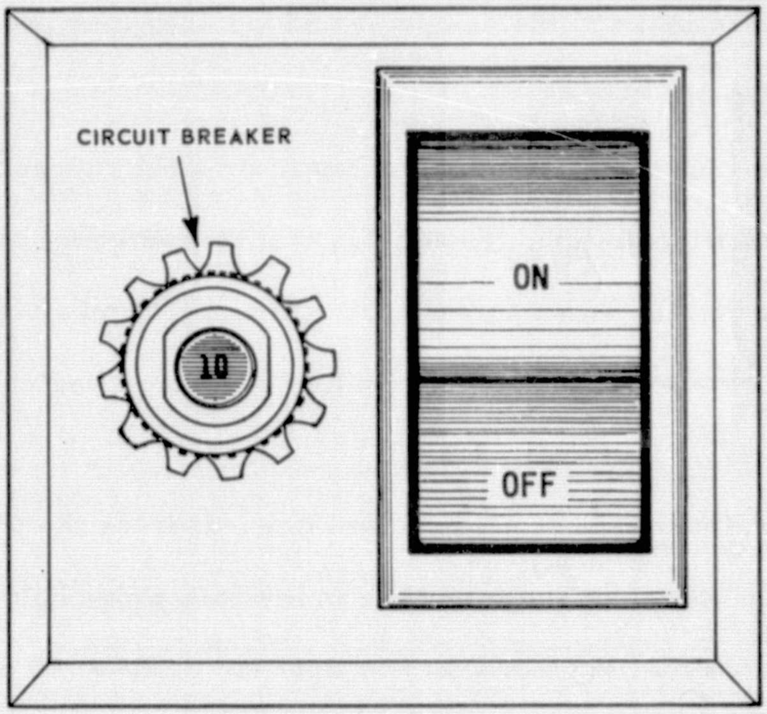


Figure 2-3. DCT-500 Power Panel

3. Set the power ON/OFF switch to the ON position. (See Figure 2-3).

4. Check the control switches on the control panel to ensure that they are properly positioned (see Section 2. 2. 1).

5. Remove the telephone receiver from the data phone and depress the TALK button to receive a dial tone. If no dial tone is heard, hang up the receiver and repeat this step.

6. Upon receiving a dial tone, dial the assigned telephone number to establish the DCT-500-to-computer communication interface. A high pitched continuous tone will signify that an interface has been achieved between

the terminal and the computer and the user may proceed to Step 8.

7. If the high pitched tone is not heard or a busy signal is heard after dialing, it indicates that the computer is inoperable. Wait approximately fifteen (15) minutes and then repeat the DCT-500 initiation procedure beginning with Step 4.

8. After receiving the high pitched tone, depress the DATA button and place the telephone in the data phone holder.

9. Depress the PROCEED key to light the CLEAR TO SEND indicator. If this indicator fails to light, report the problem to a MIRADS representative.

10. The user must now enter a SITE ID to establish the communication interface between the DCT-500 and the computer. The SITE ID is a unique identifier for a given DCT-500 and is assigned by local computer personnel. After entering the SITE ID, the following message should be received at the terminal.

```
ENTER USERID/PASSWORD:
```

After transmitting a blank line, the computer response should be:

```
*DESTROY USERID/PASSWORD ENTRY
*UNIVAC 1100 OPERATING SYSTEM ......
```

If any message is received other than the above, it indicates an abnormal condition and should be reported using the Standard Trouble Reporting Procedure described In Section 2.5.

2.3 UNISCOPE PROCEDURES

The user must become familiar with the location and use of the control keys, editing keys, and message and status indicators to become proficient in the use of the Uniscope.

Problems may occur when using a complex system such as MIRADS. Therefore, the Standard Trouble Reporting Procedure must be followed when problems other than those specifically covered are encountered during the step-by-step operation. This procedure is outlined in Section 2.5 of this manual.

The system is designed so that message requests received by the Uniscope requires the user to respond. The responses required are in predetermined formats and must be entered exactly as specified by the instructions; otherwise the system will not operate properly. Directions for correcting errors are explained in the step-by-step instructions where feasible or are covered in the Standard Trouble Reporting Procedure in Section 2.5.

The user, through the use of the Uniscope keyboard, enters data into the Uniscope and it is simultaneously displayed on the screen. The user should visually verify the data which has been entered to ensure correctness before depressing the TRANSMIT key. This key is depressed one time for each message to transmit the data to the computer. EVERY MESSAGE TRANSMITTED TO THE COMPUTER MUST BEGIN WITH A START-OF-MESSAGE (SOM) CHARACTER. The SOM character is automatically generated by the Executive Operating System.

2.3.1 Cursor

The cursor (˥) is a unique character that is displayed on the Uniscope screen to indicate the location at which the next character entered will be displayed. It also indicates the last position from which data will be transmitted to the computer when the TRANSMIT key is depressed. Whenever the cursor is positioned over a displayed character, the cursor and displayed character will blink alternately. This blinking action makes the cursor easy to see whenever it is positioned over another character. The cursor will advance one step or position for each character that is typed and can also be positioned by the Cursor Control Keys. These keys are non-destructive and therefore do not affect the information stored in the Uniscope memory or shown on the CRT screen. As the cursor moves to within eight positions of the end of any line, an audible alarm may sound and an indicator light which denotes 'End of Line' will be lighted. The indicator will remain lighted as long as the cursor occupies one of the last eight character positions. Additionally, as the cursor enters the bottom line of the screen, the alarm may sound and the Last Line indicator will be lighted. The intensity of the audible alarm may be varied with the LOUDER control key located above the upper right hand corner of the keyboard, if applicable.

2.3.2 Cursor Control Keys

The Cursor Control Keys are used to position the cursor to a specified point on the CRT screen. The user will use the cursor control keys when composing or editing messages.

1. SCAN FORWARD (→)

   This key moves the cursor forward one space at a time when depressed and released. The cursor will be moved ten spaces forward per second when key is held depressed.

2. SCAN BACKWARD (←)

   This key moves the cursor backward one space at a time when depressed and released. The cursor will be moved ten spaces backward per second when key is held depressed.

3. SCAN UP (↑)

   This key moves the cursor up one line at a time when depressed and released. The cursor will be moved up ten lines per second when key is held depressed.

4. SCAN DOWN (↓)

   This key moves the cursor down one line at a time when depressed and released. The cursor will be moved down ten lines per second when key is held depressed.

5. RETURN

This key when depressed positions the cursor at the first position of the next line.

6. The Space Bar moves the cursor forward one space at a time for each depression of the space bar.

7. TAB

This key presently serves no useful purpose.

8. BACK SPACE

This key moves the cursor backwards one space for each BACK SPACE key depression.

2.3.3 Editing Keys

The editing keys are used to correct or change data that has been entered at the keyboard into Uniscope memory and is present on the display screen.

1. CHARACTER ERASE

Depressing the Character Erase key will erase the character at the cursor position and will enter a space at that position on the screen.

2. ERASE TO END OF LINE

The ERASE TO END OF LINE key when depressed will erase all characters forward from the cursor position to the end of the line.

3. ERASE TO END OF DISPLAY

The ERASE TO END OF DISPLAY key when depressed will erase forward to the end of the display all characters in the display starting with the position at the cursor.

4. CURSOR TO HOME

Depressing the CURSOR TO HOME key repositions the cursor to the first position of the first line of the screen.

5. INSERT

When the INSERT key is depressed, all the characters to the right of and including the cursor position will shift one space to the right leaving a space at the cursor position. If the key is depressed in the lower case, only the characters in the line containing the cursor will shift to the right, and the last character on the line will be erased. If the key is depressed in the upper case, all the characters from the cursor position to the end of the display will be shifted to the right, and the last character of the display will be erased.

6. DELETE

Depression of the DELETE key will cause the deletion of the character at the cursor position, and all the characters to the right of the cursor will shift one character to the left. If it is depressed in the lower case, only the characters in the line containing the cursor will move to the left and a space will be inserted at the end of the line. If the keyboard is in the upper case, all the characters from the cursor to the end of the display will shift to the left, and a space will be inserted in the last character in the display.

2.3.4 Control Keys

The Control Keys located in the upper right hand corner of the Uniscope keyboard are used to specify the input data image, adjust the image picture on the Uniscope screen, and to transmit the data from the CRT to the computer.

1. TRANSMIT KEY

   After each line of data is entered, the Transmit key is depressed to transmit the data from the Uniscope to the computer. The data will be transmitted to the computer beginning with the character located at the position of the start-of-message character and continuing from left to right until the cursor is encountered. After transmission, the cursor will return to position one of the next line.

2. MESSAGE-WAITING KEY

The MESSAGE-WAITING key is depressed to interrupt and halt the printing of large volume output. The Transmit Key is depressed to resume printing after examining the print on the Uniscope screen.

3. START-OF-MESSAGE

The START-OF-MESSAGE key specifies the beginning of a message. The Executive Operating System automatically generates the Start-of-Message character upon the transmission of the previous line.

4. POWER SWITCH (Uniscope 100)/ON-OFF SWITCH (Uniscope 300)

The ON-OFF key operated switch is used to apply or remove power to the Uniscope.

5. FOCUS CONTROL (Uniscope 300)

The FOCUS control is used to focus the characters on the screen.

6. LOUDER CONTROL (Uniscope 300)

The LOUDER control is used to vary the volume of the audible alarm. The audible alarm can be completeiy cut off with this control key.

7. INTENSITY CONTROL (Uniscope 100)/BRIGHT CONTROL (Uniscope 300)

   This control varies the intensity of the characters on the screen.

2.3.5 Message and System Status Indicators (Uniscope 300)

This message and system status indicators indicate certain conditions of the Uniscope 300 as described below:

1. END-LINE

   The END-LINE indicator will light when the cursor is positioned anywhere in the last eight character positions of any line. It signals an approaching end of line.

2. LAST-LINE

   The LAST-LINE indicator will light when the cursor is positioned at any location in the last line of a display. It signals an approaching end of display.

3. REPLY-INCMPLT

   When the REPLY-INCMPLT indicator lights, it signals that an error has occurred in a transmission of a message from the computer. Correction of this condition will be accomplished by Operations personnel at the computer site. A possible delay will prevent the terminal from further activity until the operations personnel correct the problem.

4. HI-TEMP

The HI-TEMP indicator lights to warn the user that the internal temperature is exceeding normal limits. The user will terminate the computer run and turn the power off to the terminal. The user will notify the MIRADS personnel when this condition occurs.

5. MESSAGE-WAITING

The MESSAGE-WAITING indicator is lighted when a special system message is waiting to be transmitted to the terminal from the computer. The message may be received on the Uniscope screen by depressing the Message-Waiting key located at the top of the keyboard. An audible alarm may also sound intermittently whenever this indicator is lit.

6. WAIT

The WAIT indicator will light during the transmission of messages to or from the computer.

2.3.6 Uniscope Initiation Procedure

The following procedure is used to initialize (power-on) the Uniscope.

1. Check all external line plugs to ensure they are properly connected to an 110V AC electrical outlet.

2. If the terminal is a Uniscope 100, depress the POWER button on the front panel and proceed to Step 4. Remove the cover from the rear of the Uniscope 300 (see Figure 2.4)

and set this white circuit breaker switch to the ON (up) position to activate a cooling fan. After setting the circuit breaker to the ON position, depress the red master clear switch located directly above the circuit breaker.

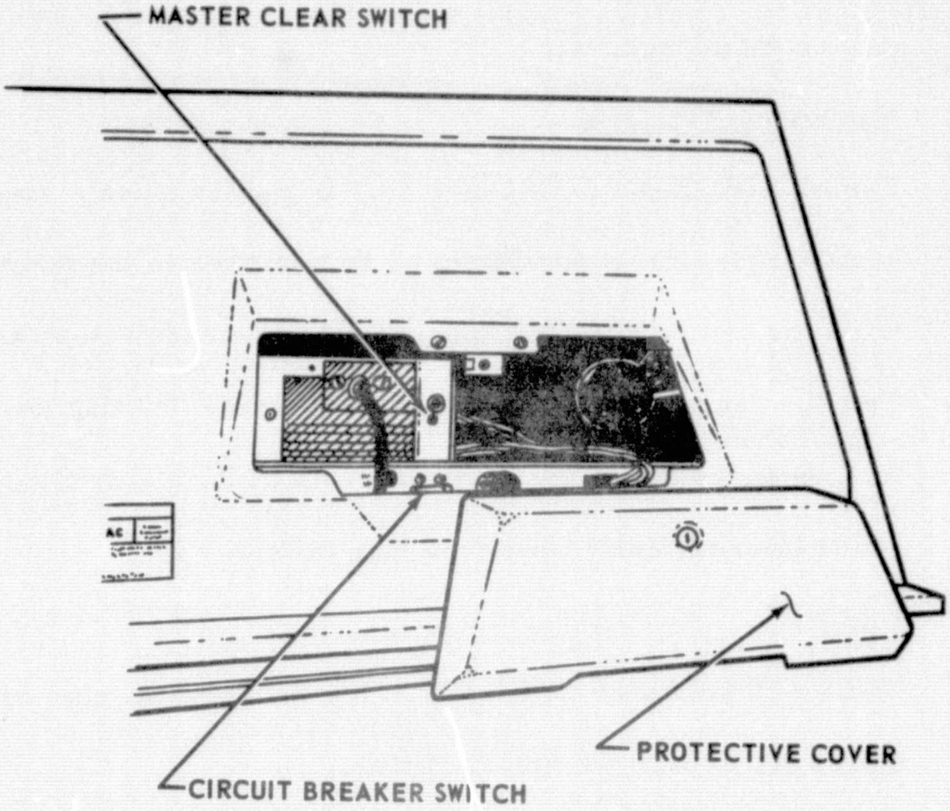


Figure 2-4. Uniscope 300 Adjustment

3. Turn the Keylock Switch, located in the upper right hand portion of the keyboard, clockwise to the ON position.

4. After the above operation is completed, the cursor (⌉) should appear on the face of the screen. If the cursor cannot be seen on the screen, the intensity control should be adjusted until the cursor appears.

5. After the cursor appears on the screen, remove the telephone receiver from the data phone and depress the TALK button to receive a dial tone. If no dial tone is heard, hang up the receiver and repeat this step.

6. Upon receiving a dial tone, dial the assigned telephone number to establish the Uniscope-to-computer communication interface. A high pitched continuous tone will signify that an interface has been achieved between the terminal and the computer and the user may proceed to Step 8.

7. If the high pitched tone is not heard or a busy signal is heard after dialing, it indicates that the computer is inoperable. Wait approximately fifteen (15) minutes and then repeat the Uniscope initiation procedure beginning with Step 6.

8. After receiving the high pitched tone, depress the DATA button and place the telephone in the data phone holder.

9. The user must now enter a SITE ID to establish the communication interface between the Uniscope and the computer. The SITE ID is a unique identifier for a given Uniscope and is assigned by local computer personnel. After entering the SITE ID, the following message should be received at the terminal.

```
ENTER USERID/PASSWORD:
```

10. The user must now enter a USERID/PASSWORD assigned by local computer personnel.

After transmitting a correct entry, the computer response should be:

```
  *DESTROY USERID/PASSWORD ENTRY
*UNIVAC 1100 OPERATING SYSTEM ......
```

If any message is received other than the above, it indicates an invalid entry and the user will have to repeat the procedures beginning with Step 6.

2.4 RUN INITIATION PROCEDURES

Users of MIRADS should be familiar with the EXEC 8 control statements which are necessary to perform inquiries against their Data Base. Each EXEC 8 control statement begins in the first position of a line and is indicated by preceding the actual command with an EXEC 8 control character in position one. Commands entered via a DCT-500 or Uniscope 100 are preceded with the special character '@' while commands which are entered via a Uniscope 300 are preceded with the special character '#'.

NOTE

All examples in this manual are illustrated for the DCT-500 under EXEC VIII Level 31. The computer facility is Slidell Computer Complex.

1. After establishing a communication interface between the terminal and the computer, the user will enter the following RUN statement message and transmit it to the computer:

@RUN ΔIIXXXX, AAAAAAAAAAA, Project-ID, 3, Pages

The RUN statement must be entered in the exact format as shown above. The IIXXXX portion of the RUN statement is used to identify each run that is processed by the computer. The user will enter his first and last initials in the II portion and will enter his telephone exchange in the space where the XXXX appears. The delta (Δ) indicates one space must be left between the N and I. All other data is entered contiguously.

The user will enter an account number in the AAAAAAAAAAA portion of the RUN statement. This number is assigned to the customer when initial use of MIRADS is begun. The account number is used as a charge number in accounting for computer time.

The Project-ID field (Project-ID) is used as a qualifier to uniquely identify the user's application files and must be a one-to-twelve character name.

Pages is used to specifify the maximum number of printed pages the output reports will contain. This is an optional field which does not have to be specified unless the generated output is expected to be over 100 pages. An integer value (1-9999) in this field specifies the page number limit. An inquiry which will generate a large volume report should have a Pages value specified as two times the number of expected pages. If the volume of output is not predictable, it is recommended to specify a large value in the Pages field.

After the user has checked the RUN statement which has been entered to ensure that the data and format are correct, the RETURN key (DCT-500) or TRANSMIT key (Uniscope) is depressed to transmit the RUN statement to the computer.

If an error has been made ir the format or content of the RUN statement, the following message will be received by the terminal:

```
*BAD RUN STATEMENT OR ILL. ACCNT NO.*
```

This message indicates that an error has been made in entering the RUN statement, or an incorrect account number has been entered in the RUN statement.

The entire RUN statement with the error corrected must then be reentered and transmitted to the computer.

2. After the RUN statement has been successfully transmitted to the computer, a message will be received by the Terminal as shown below:

```
DATE: MMDDYY  TIME: HH MMSS
```

The MMDDYY string after DATE will contain the present month, day, and year, and the HHMMSS after TIME will contain the 24 hour clock time in hours, minutes, and seconds to record the date and time when the run was initiated.

After receiving the DATE and TIME message, the user is then ready to begin selection and execution of MIRADS to process requests.

2.4.1 MIRADS Execution

1. The user begins execution by entering the following statement:

```
@XQT Δ MIR*ADS. MIRADS
```

The delta (Δ) between the T and M indicates one space must be left between these characters. Transmit the XQT statement by depressing the Return Key (DCT-500) or Transmit Key (Uniscope).

2. The following message should then be received by the terminal:

```
ENTER QUALIFIER*FILENAME
```

3. If any other message is received by the terminal, it indicates an abnormal condition and should be reported using the Standard Trouble Reporting Procedure in Section 2.5.

After the terminal has received the ENTER QUALIFIER*FILENAME message, the user will enter an authorized file name and transmit it to the computer by depressing the Transmit Key (Uniscope) or Return Key (DCT-500). The system will analyze the status of the control files. A bad status of one of these control files will be indicated by one of the following messages at the terminal:

THE DATA BASE FILE MAY CONTAIN ERRORS - RELOAD TO GUARANTEE ACCURACY

or

FILENAME IS NOT CATALOGED - CALL YOUR MIRADS REPRESENTATIVE

or

FILENAME IS NOT LOADED - CALL YOUR MIRADS REPRESENTATIVE

or

FILENAME IS EXCLUSIVELY ASSIGNED TO ANOTHER RUN - RETRY IN A FEW MINUTES

The first message indicates that the specified file has been disabled due to a computer failure while updates to the Data Base were being performed. The user should reload the files as shown in Paragraph 6.8 of the MIRADS Implementation Manual, or proceed to Paragraph 2.4.2 of this manual for RUN termination procedures and then contact a MIRADS representative for assistance in reloading the files.

The second message indicates the specified file name is invalid or has been deleted from the Master File Directory. The user should check the File Name for validity, that is, if the name has been spelled correctly and input correctly. If the name is valid, the user should reload the files as shown in Paragraph 6.8 of the MIRADS Implementation Manual, or proceed to

Paragraph 2.4.2 of this manual for RUN termination procedures and then contact a MIRADS representative for assistance in reloading the files.

The third message indicates that the specified file has been unloaded to magnetic tape. The user should reload the file as shown in Paragraph 6.8 of the MIRADS Implementation Manual, or proceed to Paragraph 2.4.2 of this manual for RUN termination procedures and then contact a MIRADS representative for assistance in reloading the files.

The fourth message indicates that the requested files have been exclusively assigned to another RUN. The user must wait until the files have been released before he can access them through MIRADS.

4. After the correct Qualifier and File Name have been entered, the user may receive the message:

ENTER PASSWORD

or

READY

The first message indicates the File Name which has been entered has a corresponding Password or Passwords which must now be entered. If the user enters an invalid Password, the terminal receives the message:

```
PASSWORD INCORRECT - PLEASE CORRECT AND REENTER
```

This message means that the Password which has been input does not correspond with any of the Passwords specified in the Dictionary as a valid Password for the file entered previously. The user is allowed two more attempts to enter the Password correctly. The terminal will receive the above message each time the Password is invalid. If the third attempt is in error, program execution (XQT control statement) will be terminated and the following message will be received:

```
THREE INCORRECT PASSWORDS HAVE BEEN ENTERED -
     MIRADS WILL NOW TERMINATE
```

If execution is terminated and the user wishes to continue the use of MIRADS, repeat Paragraph 2.4.1 of this manual; otherwise, proceed to Paragraph 2.4.2 for RUN termination.

The READY message indicates that no Passwords exist for the specified file, or that the correct Password has been entered and the user is ready to begin entering commands to the MIRADS System. The use of the commands available in MIRADS are presented in Section 3.

2. 4. 2 RUN Termination

The user RUN is terminated on the terminal by sending a FIN command to the computer.

```
@FIN
```

After transmitting this command, user accounting information will be displayed on the terminal.

```
RUN ID XXXXXX ACCOUNT:  AAAAAAAAAA PROJECT:  NNNN-Project
TIME:  TOTAL:  HH:MM:SS. SSS
       CPU:    HH:MM:SS. SSS       I/O:    HH:MM:SS. SSS
       CC/ER:  HH:MM:SS.SSS        WAIT:   HH:MM:SS.SSS
START:  HH:MM:SS.   MMM DD, YYYY FIN:    HH:MM:SS   MMM DD, YYYY
*TERMINAL INACTIVE*
```

Once the line inactive message has been received, the user can initiate another RUN or disconnect the terminal.

To disconnect the demand terminal, the user must send a @@TERM command to the computer. This automatically disconnects the demand terminal from the computer.

To power down the DCT-500:

Depress the OFF switch on the power panel located on the back of the unit.

To power down the Uniscope 100:

Depress the POWER button on the front panel.

To power-down the Uniscope 300:

1. Turn keylock switch to OFF position.
2. Depress the red Master Clear pushbutton switch.
3. Move white Circuit Breaker to OFF position.

2.5 STANDARD TROUBLE REPORTING PROCEDURE

Many problems may occur when using a complex system such as MIRADS. Therefore, a Standard Trouble Reporting Procedure must be followed when problems other than those specifically covered are encountered.

2.5.1 DCT-500 Recovery Procedure

An abnormal response on the DCT-500 indicates that a failure has occurred in the computer or DCT-500. This failure may cause the DCT-500 to disconnect itself from the computer. If the CLEAR TO SEND indicator light is not lit, the DCT-500 has lost its communications interface with the computer. By telephone communication, (504) 255-6430, a recorded message will provide the status of the computer at the Slidell Computer Complex and whether or not any delay before reinitialization is necessary. In the event that the computer is functioning properly but the communication interface can not be maintained, the user should contact a MIRADS representative and report the condition.

If the communication interface is not lost, the user should attempt to reinitialize the process which failed. This will normally require the user to input the command:

```
@XQT     MIR*ADS.MIRADS
```

If this fails to give the required response, the user should contact a MIRADS representative and report the condition of the error.

2.5.2 Uniscope Recovery Procedure

An abnormal response on the Uniscope indicates that a failure has occurred in the computer or Uniscope. This failure may cause the Uniscope to disconnect itself from the computer. Immediately below the Uniscope 300 screen is a series of six small lights. The third light from the left will blink every six (6) seconds whenever the Uniscope 300 is properly interfaced with the computer. If this light fails to blink, the Uniscope has lost its communications interface with the computer. This same condition is true if the single light below the Uniscope 100 fails to blink. By telephone communication (504) 255-6430, a recorded message will provide the status of the computer and whether or not any delay before reinitialization is necessary.

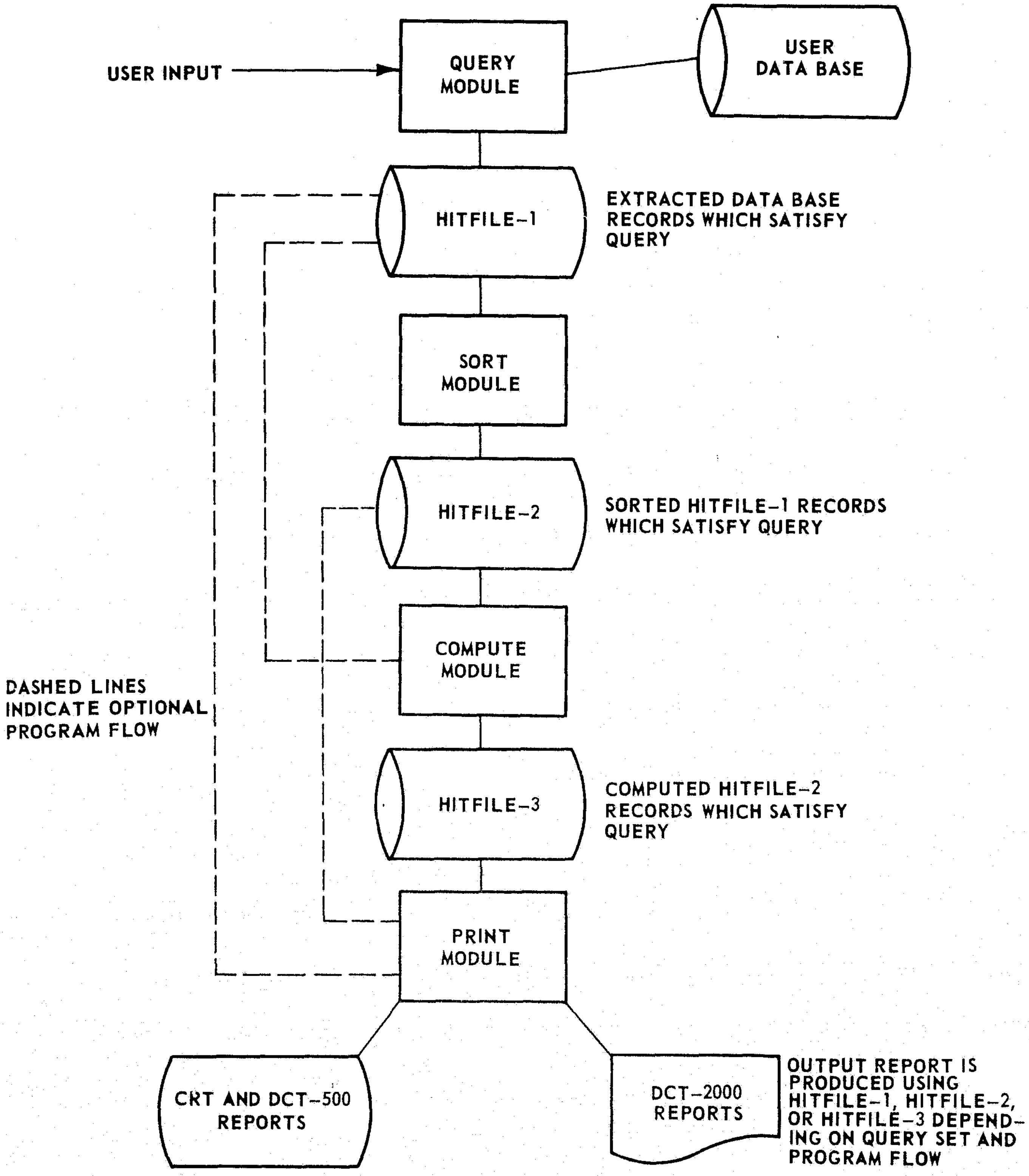
In the event that the computer is functioning properly but the communication interface can not be maintained, the user should contact a MIRADS representative and report the condition.

SECTION 3 - MIRADS COMMANDS

3.1 GENERAL

MIRADS provides the necessary functional capabilities to retrieve, manipulate, format and display data. The Query Set provides a means of questioning which consists of seven major commands, each of which performs a specific function that contributes to the answering of a user's inquiry. The major commands of the Query Set are QUERY, SORT, COMPUTE, PRINT, FORMAT, SAVE, and UPDATE. Minor commands available to accomplish user functions are RUN, TOP, NEW, KEY, and CHECK. Minor commands contained within the save functions are SAVEC, ADD, LIST, DISPLAY, DO, DELETE, and EDIT.

The data flow for a Query Set containing a QUERY, SORT, COMPUTE, PRINT, and RUN command as well as other possible paths of data flow for MIRADS is illustrated in Figure 3-1. The user's QUERY command initiates processing and establishes the criteria for selecting records from the user's Data Base. Addresses of the selected records, which are called Hit records, are extracted and written onto a file named Hitfile-1. The user then has the option to SORT, COMPUTE, or PRINT these records. When specific ordering is required, the SORT command must be contained in the user's input. This command will then resequence the records in Hitfile-1 and write them onto



USER INPUT
QUERY MODULE
USER DATA BASE
HITFILE-1
EXTRACTED DATA BASE RECORDS WHICH SATISFY QUERY
SORT MODULE
HITFILE-2
SORTED HITFILE-1 RECORDS WHICH SATISFY QUERY
COMPUTE MODULE
DASHED LINES INDICATE OPTIONAL PROGRAM FLOW
HITFILE-3
COMPUTED HITFILE-2 RECORDS WHICH SATISFY QUERY
PRINT MODULE
CRT AND DCT-500 REPORTS
DCT-2000 REPORTS
OUTPUT REPORT IS PRODUCED USING HITFILE-1, HITFILE-2, OR HITFILE-3 DEPENDING ON QUERY SET AND PROGRAM FLOW

Figure 3-1. MIRADS Data Flow Diagram

Hitfile-2. Hitfile-2 then becomes the input to the Compute Module or the Print Module depending on the option exercised by the user. If computations are to be made on the Hitfile-2 records, they are made based on the instructions specified by the COMPUTE command and the results are written onto Hitfile-3. Thus the records contained in Hitfile-3 satisfy the QUERY, SORT, and COMPUTE commands and they may be used to print or display the results of the Query Set as specified by the PRINT command. As the Data Flow Diagram in Figure 3-1 is studied, it becomes apparent that the number of Hitfiles produced depends on the Query Set chosen by the user, and that regardless of the number of Hitfiles produced each successive Hitfile still satisfies the Query. That is, each Hitfile generated will relate to the previous Hitfile in that the data contained within is a reordering or calculation dependent upon Hitfile-1 data records. Each command will be presented later in this Section with detailed instructions on its usage.

3.2 SAMPLE QUERY SET

A Sample Query Set is provided merely to illustrate the simplicity of using MIRADS and is not meant to explain the full potential of the commands available to the user.

The first example lists the commands which are required to calculate and print the monthly salary for all employees earning over eight thousand ($8,000.00) dollars annually. The Query Set references the Data Base shown in the second example and produces the output shown in the third example.

ALL EXAMPLES AND ILLUSTRATIONS OF DIALOGUE PROVIDED IN THIS SECTION WILL DIFFERENTIATE THE USER INPUT FROM SYSTEM RESPONSES BY THE USE OF A DELTA (Δ) SYMBOL. EACH USER INPUT IS PRECEDED BY THIS SYMBOL.

EXAMPLE: SAMPLE MIRADS QUERY SET

```
  READY
Δ QUERY, SALARY GT 8000 .
  READY
Δ SORT, SALARY D.
  READY
Δ COMPUTE, ALL, 2, $MONTHLY-SALARY = SALARY / 12.
  READY
Δ PRINT, NAME, DATE-HIRED, $MONTHLY-SALARY.
  READY
Δ RUN
```

EXAMPLE: SAMPLE MIRADS DATA BASE

RECORD NUMBER	NAME	DATE HIRED	SALARY
1	JONES	01-28-60	10,000
2	SMITH	04-26-66	11,000
3	WALLACE	03-17-63	9,000
4	JOHNSON	07-09-69	7,500
5	DAVIS	11-11-61	12,500
6	BROWN	11-06-68	8,500

EXAMPLE: SAMPLE MIRADS REPORT

NAME	DATE HIRED	MONTHLY-SALARY
DAVIS	11-11-61	1,041.67
SMITH	04-26-66	916.67
JONES	01-28-60	833.33
WALLACE	03-17-63	750.00
BROWN	11-06-68	708.33

3.2.1 Messages

While executing MIRADS, the user must be familiar with messages which will be received at the terminal. Normal system responses are:

```
READY

QUERY NOW PROCESSING

ENTER OUTPUT REPORT SITE ID
```

The READY message indicates that the computer program is ready to receive the QUERY, SORT, COMPUTE, PRINT, or other input command. The QUERY NOW PROCESSING message indicates that a valid Query Set has been received by the computer and is being processed. The ENTER OUTPUT REPORT SITE ID message allows the user to direct his output report to a particular output device. If any message is received at the terminal other than user output or one of those described above, an abnormal condition has occurred and the user should consult Section 3.5 which describes MIRADS Diagnostic Messages.

3.3 MIRADS MAJOR COMMANDS

MIRADS is a command driven system. The basic order of the Query Set is the QUERY, SORT, COMPUTE, PRINT, SAVE, and UPDATE commands. The QUERY and PRINT or FORMAT commands are normally required in every Query Set whereas the SORT, COMPUTE, and SAVE commands are optional. An UPDATE to the Data Base may or may not require a QUERY command. The commands are edited and verified for correctness by the computer program as they are transmitted to the computer.

3.3.1 QUERY Command

The QUERY command is the focal point of each Query Set. This command determines which records will be extracted from the user's Data Base and processed by the other major commands. If this command is not properly structured, the records selected may not reflect the true intent of the inquiry.

The QUERY command must follow the system response:

```
READY
```

This message indicates that the system is ready to accept the user's search criteria.

Figure 3-2 illustrates the QUERY command format. Elements within the braces { } are required whereas elements within the brackets [ ] are optional. If more than one entry is given, the choice is optional. The following breakdown refers to the numbered areas in Figure 3-2.

1. $\left\{\begin{matrix}\text{Q}\\ \text{QUERY}\end{matrix}\right\}$

   This element identifies the command as a Query.

2,4,6. [,]

   The comma is the desired means of punctuation. A variable number of spaces can be used in place of the comma.

3. [SEARCH-n]

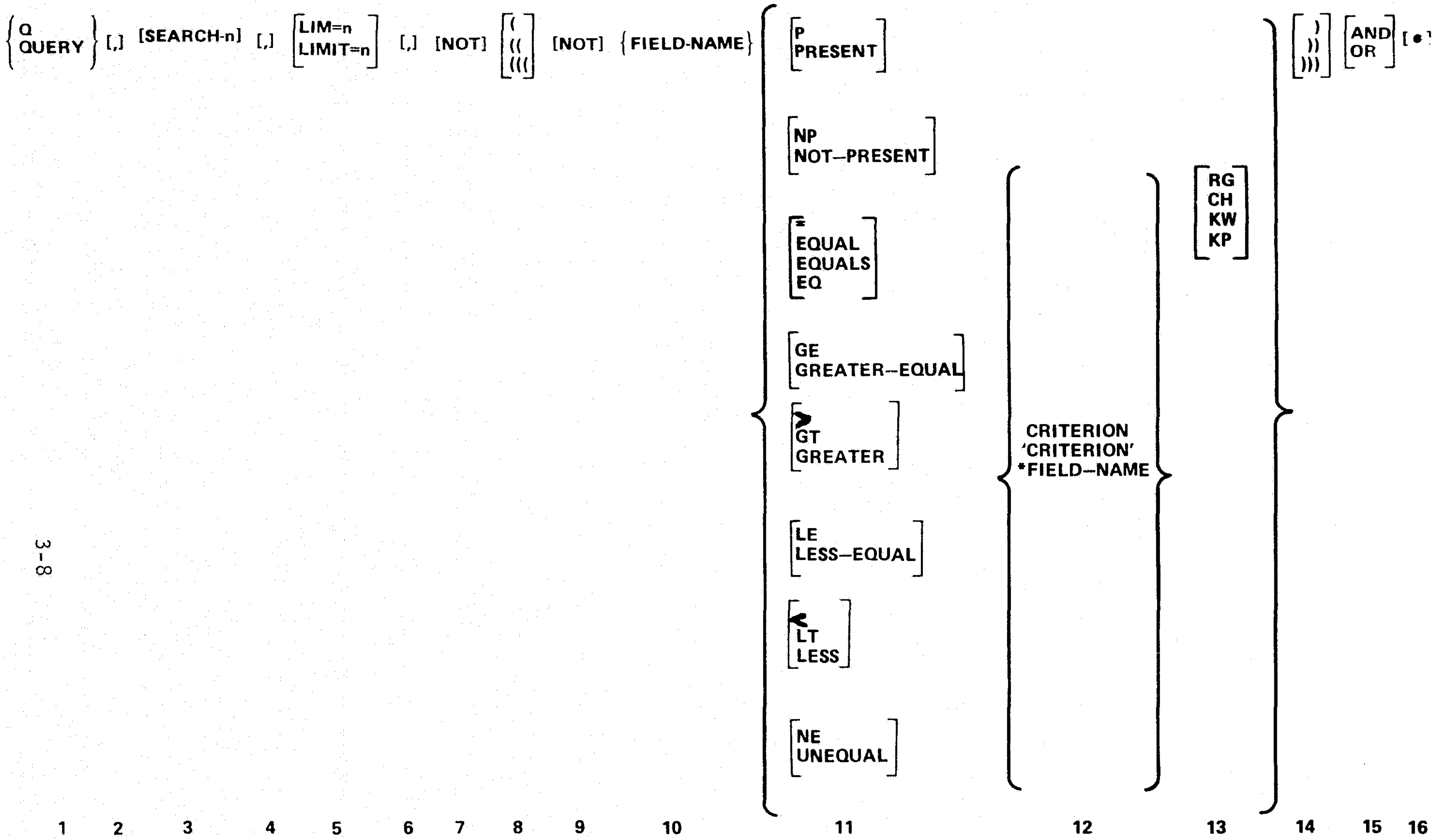
   This element specifies a search limitation for multilevel Data Bases. The value of n (1-8) restricts the number of Data Base levels which are to be extracted by the command.

5. $\left[\begin{matrix}\text{LIM = n}\\ \text{LIMIT = n}\end{matrix}\right]$

   This element is used to limit the number of records which are to be extracted by the command. The value for n represents the maximum number of Hits the user requires.

7. [NOT]

   The NOT element (at the phrase level) is used to negate the phrase immediately following it. The entire phrase must be enclosed in parentheses. Example:



```
{Q    } [,] [SEARCH-n] [,] [LIM=n  ] [,] [NOT] [(  ] [NOT] {FIELD-NAME} {[P      ]                                              } [)  ] [AND] [•]
{QUERY}                    [LIMIT=n]             [(( ]                   {[PRESENT]                                              } [)) ] [OR ]
                                                 [(((]                   {                                                       } [)))]
                                                                         {[NP         ]                                          }
                                                                         {[NOT-PRESENT]   {              }   [RG]                }
                                                                         {                {              }   [CH]                }
                                                                         {[=     ]        {              }   [KW]                }
                                                                         {[EQUAL ]        {              }   [KP]                }
                                                                         {[EQUALS]        {              }                       }
                                                                         {[EQ    ]        {              }                       }
                                                                         {                {              }                       }
                                                                         {[GE           ] {              }                       }
                                                                         {[GREATER-EQUAL] {              }                       }
                                                                         {                {              }                       }
                                                                         {[>      ]       { CRITERION    }                       }
                                                                         {[GT     ]       {'CRITERION'   }                       }
                                                                         {[GREATER]       {*FIELD-NAME   }                       }
                                                                         {                {              }                       }
                                                                         {[LE        ]    {              }                       }
                                                                         {[LESS-EQUAL]    {              }                       }
                                                                         {                {              }                       }
                                                                         {[<   ]          {              }                       }
                                                                         {[LT  ]          {              }                       }
                                                                         {[LESS]          {              }                       }
                                                                         {                {              }                       }
                                                                         {[NE     ]       {              }                       }
                                                                         {[UNEQUAL]       {              }                       }

   1     2       3      4      5      6    7     8    9       10                11                    12            13           14     15  16
```

ELEMENTS WITHIN { } ARE REQUIRED.
ELEMENTS WITHIN [ ] ARE OPTIONAL.

Figure 3-2. QUERY Command Format

Q, NOT (AGE EQUAL 40) AND SEX EQUAL 'MALE'

is equivalent to

Q, AGE UNEQUAL 40 AND SEX EQUAL 'MALE'.

8. $\begin{bmatrix} (\\ ((\\ (((\end{bmatrix}$

Complex Queries usually require the use of parentheses to help clarify the intended logic. An example illustrates this point. The Query,

Q, NAME = 'JOHN' AND AGE = 40 OR CITY = 'HUNTSVILLE'.

could be construed to mean everybody who is 40 years old with the name John, or everybody who lives in Huntsville, that is,

Q, ((NAME = 'JOHN' AND AGE = 40) OR CITY = 'HUNTSVILLE').

Likewise, someone else might assume the Query means everybody with the name John, and at the same time they must be 40 years old or they must live in Huntsville, that is,

Q, ((NAME = 'JOHN') AND (AGE = 40 OR CITY = 'HUNTSVILLE')).

Thus, parentheses can be used to help in clearly stating the meaning or intent of the Query. The number of left parentheses must equal the number of right parentheses and the number of open parentheses at any given time must never exceed three. Parentheses are optional and may be omitted except when required to clarify the intent of the search. When parentheses are needed but are omitted, the computer has no way of knowing how the user intended the logical phrases to be grouped.

9. [NOT]

The NOT element is used to negate the first relational that follows. Example,

```
Q, NOT (DEPT EQUAL 1234 ).
```

is equivalent to

```
Q, DEPT UNEQUAL 1234 .
```

Combining NOT's at the phrase and element level, we have

```
Q, NOT (NOT DEPT = 1234 ).
```

which is equivalent to

```
Q, DEPT = 1234 .
```

10. { Field-Name }

This element is the Dictionary name of the data field which is to be examined during the search. It identifies the field which will be tested for Presence or Non-Presence of data or compared with the Criterion or another Field-Name.

11. { Relational }

There are eight different relationals which can be used in the QUERY command.

a. $\begin{bmatrix}\text{P}\\ \text{PRESENT}\end{bmatrix}$

The Field-Name is checked for values which are not equal to spaces. Criterion (Item 12) and Type of Search (Item 13) are not permitted with this relational.

b. $\begin{bmatrix}\text{NP}\\ \text{NOT-PRESENT}\end{bmatrix}$

The Field-Name is checked for values which are equal to spaces. Criterion (Item 12) and Type of Search (Item 13) are not permitted with this relational. The field being checked should be defined as a alphanumeric field in the user's Dictionary for the Data Base.

c. $\begin{bmatrix}=\\ \text{EQUAL}\\ \text{EQUALS}\\ \text{EQ}\end{bmatrix}$

The Field-Name is checked for values which are equal to Criterion or another Field-Name.

d. $\begin{bmatrix}\text{GE}\\ \text{GREATER-EQUAL}\end{bmatrix}$

The Field-Name is checked for values which are greater than or equal to Criterion or another Field-Name.

e. $\begin{bmatrix}>\\ \text{GT}\\ \text{GREATER}\end{bmatrix}$

The Field-Name is checked for values which are greater than Criterion or another Field-Name.

f. $\begin{bmatrix}\text{LE}\\ \text{LESS-EQUAL}\end{bmatrix}$

The Field-Name is checked for values which are less than or equal to Criterion or another Field-Name.

g. $\begin{bmatrix}<\\ \text{LT}\\ \text{LESS}\end{bmatrix}$

The Field-Name is checked for values which are less than Criterion or another Field-Name.

h. $\begin{bmatrix}\text{NE}\\ \text{UNEQUAL}\end{bmatrix}$

The Field-Name is checked for values which are not equal to Criterion or another Field-Name.

12. $\begin{Bmatrix}\text{CRITERION}\\ \text{'CRITERION'}\\ \text{*FIELD-NAME}\end{Bmatrix}$

This element identifies the Criterion or Field-Name with which the Field-Name in Item 10 will be compared. One of these elements is mandatory for all relationals except P, PRESENT, NP, or NOT-PRESENT.

a. [Criterion]

The Criterion element contains the actual numeric or alphanumeric value with which the Field-Name in Item 10 will be compared. The maximum length for the criterion is 48 characters. A criterion element which contains blank characters must be enclosed in quotation marks.

b. [*Field-Name]

The Field-Name (preceded by an asterisk) with which the Field-Name in Item 10 will be compared. This element is used when making field-to-field data comparisons.

13. $\begin{bmatrix} \text{RG} \\ \text{CH} \\ \text{KW} \\ \text{KP} \end{bmatrix}$

This element indicates the Type of Search desired and is used for searching portions or parts of data fields. Type of Search may be used with the equal relational only.

a. RG - Regular Search

The search begins with the leftmost position of the field and compares the entire content of the field against the selection criterion. Example:

Q, VALUE = '13465' RG.

b. CH - Character Search

The first n characters of a field are tested for a value equal to the Criterion. N is equal to the number of Criterion characters enclosed. Example:

Q, NAME EQ JONES CH.

N would be equal to 5, the number of characters in the name JONES. The maximum field size for the Character Search is 132 characters.

c. KW - Keyword Search

The field is searched for a value equal to the Criterion. If matching characters are found at the beginning of a field followed by a space, at the end of a field preceded by a space, or anywhere within the field preceded and followed by a space, the search is satisfied. This type of search is normally used for locating a keyword within the field. The maximum field size for the Keyword Search is 132 characters. Example:

Q, STREET EQUAL MADISON KW.

d. KP - Keyphrase

The field is searched for a value equal to the Criterion. If matching characters appear anywhere within the field, the search is satisfied. This type of search is normally used for searching narrative data for key phrases. The maximum field size for the Keyphrase Search is 132 characters. Example:

```
Q, STREET EQUAL 'MADISON AVE'KP.
```

14. $\begin{bmatrix}) \\ )) \\ ))) \end{bmatrix}$.

Parentheses are used to clearly indicate the intent of a search as explained in Item 3. Right and left parentheses must <u>balance</u>, that is, the number of right parentheses in a query must equal the number of left parentheses.

15. $\begin{bmatrix} \text{AND} \\ \text{OR} \end{bmatrix}$

There are two logical connectors (AND and OR) which can be used with the QUERY command.

a. AND

The AND connector is used for joining together simple queries to form more complex queries. The AND connector is used when being more selective, that is <u>both</u> conditions connected with AND must be true to be acceptable. The use of the AND connector requires that the Field-Name in each element be different. Example:

```
Q, NAME EQ 'JONES P J' AND
  DATE-HIRE EQUAL '01-28-60'.
```

b. OR

The OR connector is also used for joining together simple queries to form more complex queries. The OR connector is used when <u>either</u> of two conditions is acceptable. Example:

```
Q, NAME EQUAL 'JONES P J' OR
  NAME EQUAL 'SMITH R A'.
```

16. [.]

The period is used to terminate the QUERY command. The transmission of a second command will automatically terminate the QUERY command if the period is omitted.

3.3.1.1 Rules for Using QUERY Command

The following rules must be adhered to for the proper use of the QUERY command:

1. Query statements which cannot be contained on one line may be continued on the following line beginning in any position.
2. Elements such as Field-Names, Relationals, Criteria, Negations and Connectors cannot be broken and continued on the following line. They should be entered as a continuous word on the following line.
3. Only one QUERY command is permitted per Query-Set.

4. Query elements must be separated from adjacent elements with a comma when permitted or one or more spaces unless explicity prohibited (see Rule 6).

5. Criteria values must not exceed 48 characters. Criteria containing blanks must be enclosed within quotation marks.

6. Contiguous parentheses must be void of spaces.

7. The maximum number of open parentheses must not exceed three at any time while constructing the QUERY command. One paren eses of a set cannot be contained within another set of parentheses unless its mate is also contained therein.

8. The maximum number of Field-Names on the left side of relationals for a QUERY command is 100.

9. The AND and OR connectors may <u>not</u> be used to join Field-Names on the left side of a relational. For example,

   ```
   Q, AGE AND YEAR-HIRED EQUAL 50 .
   ```

   is an invalid Query whereas,

   ```
   Q, AGE EQUAL 50 AND YEAR-HIRED EQUAL 50 .
   ```

   is acceptable. However, the connectors may be used to join different criteria, or Field-Names on the right side of the relational. For example, Q, AGE = 50 OR 51.

and

Q, AGE = *H OR *B.

are valid QUERY commands.

3.3.1.2 Simple QUERY Commands

The QUERY command consists of elements which are joined together to form phrases. The simple query consists of one phrase which is broken down into the following parts:

1. Field-Name defines the field in the Data Base which is to be used as a basis to determine selection of Hitfile records.
2. Relational defines the condition which must be satisfied between Field-Name and Criterion or another Field-Name.
3. Criterion or Field-Name defines the value or Field-Name to which the first Field-Name is compared. If P, PRESENT, NP, or NOT-PRESENT is used as a relational, this element will be omitted. The following example could be used to locate all individuals within a file whose employee serial number (SERIAL) is greater than 10000. Example:

Q, SERIAL GREATER 10000 .

In this Query, the Field-Name is SERIAL, the Relational is GREATER, and the Criterion is 10000.

The simple query can be used to search a particular data field within a record for a value equal-to, greater-than, less-than, less-than or equal-to, greater-than or equal-to, or not-equal to a given value. Examples:

```
Q, AGE EQUAL 25 .
Q, AGE GREATER 25 .
Q, AGE LESS 25.
Q, AGE LESS-EQUAL 25 .
Q, AGE GREATER-EQUAL 25 .
Q, AGE UNEQUAL 25 .
```

These searches are accomplished by comparing from left to right the data in the Field-Name to the data value of the Criterion or another Field-Name. Comparison against criteria which contain embedded blanks must be enclosed within quotation marks with all blank positions included in the criteria.

There are times when the user is not certain how data is stored in the data records. Nevertheless, MIRADS capability provides a means for locating the data if it exists. This means is generally less efficient than the standard means and should only be used when the data format storage is not known or the answers cannot be obtained otherwise. Section 3.3.1.9 describes other techniques for searching the Data Base.

3.3.1.3 Compound QUERY Commands

Many occasions arise where simple Queries of one phrase cannot answer user questions regarding a Data Base. In this case, a simple Query can be combined with other simple Queries to form a complex Query. These Queries are a composite of simple Queries that can employ any or all of the techniques described for simple or complex Queries. They are combined with the AND and OR connectors. Example:

```
Q, AGE EQUAL 20 AND SALARY GREATER 10000 .
```

In the above example, two simple Queries have been combined to form a complex Query. This example causes the Data Base to be searched to locate all individuals who are twenty (20) years old and whose salary is greater than $10,000. Since the AND connector is used, both conditions must be true before a record is selected to satisfy the Criteria of the Query.

As simple Queries are combined to form complex Queries, the intent of a Query is sometimes lost or vague. Parentheses should be used to ensure that the meaning of a Query is clearly understood as the following example illustrates. Example:

```
Q, SALARY GREATER 20000 OR ((EDUC EQUAL
 'MASTERS' OR 'DOCTORATE') AND (YEAR-HIRED
 GE 60 AND LE 70 )).
```

It is imperative that adjacent parentheses be void of spaces and that the number of left parentheses equal the number of right parentheses.

3.3.1.4 Finding Missing Data

The simplest Query that can be formed is the PRESENCE test for the existence of non-blank data within a particular data field. This condition is considered met only when the data field within the record contains information other than blanks. This type of Query is particularly easy to construct because the user need only specify the Field-Name and Relational since the Criterion is implied by the PRESENCE Relational. Example:

```
Q, ADDRESS PRESENT.
```

In this example every record in the Data Base with a non-blank value for the ADDRESS field will be selected as having met the ADDRESS PRESENCE test.

There are occasions when it is desirable to know when certain data fields are void of data, that is, the data field is blank. The NOT-PRESENT test can be used to determine this condition and it is considered met only when the data field contains all blank information. Example:

```
Q, CITY NOT-PRESENT.
```

Every record in the data base with no information in the data field CITY will be selected as having met the NOT-PRESENT test.

3.3.1.5 Finding Multiple Data Values

The capability to test to see if the contents of a particular data field is equal to one of several different values is often required in data processing as the following example illustrates. Example:

```
Q, EDUCATION EQUAL 10 OR EDUCATION
 EQUAL 11 OR EDUCATION EQUAL 12 .
```

The preceding example could be used to locate all individuals within a data file with 10th, 11th, and 12th grade educations. If the number of particular values being tested is excessive, the repetitious typing of the EDUCATION field gets to be a burden. Alternatively, the Query shown above may be stated as follows. Example:

```
Q, EDUCATION EQUAL 10 OR 11 OR 12 .
```

In this case, the Field-Name (EDUCATION) is identified only once but implied three times.

3.3.1.6 Finding Data within a Range

If the user wishes to test the contents of a particular data field to see if it falls within a given range, the Field-Name does not have to be repeated. Consider the following inquiry:

```
Q, SALARY GE  10000  AND SALARY LE  20000 .
```

The preceding inquiry could be alternatively entered as:

```
Q, SALARY GE  10000 AND LE  20000 .
```

This example is used to find all employees whose salary is within the range of $10,000 to $20,000, inclusive.

3.3.1.7 Finding Low or High Values

The QUERY command may be combined with the SORT command (described in Section 3.3.2) to find the high or low values for a particular data field within a file. The following Query Set could be used to locate an individual with the highest salary. Example:

```
Q, SALARY PRESENT.
S, SALARY D.
P, 1, NAME, SALARY.
```

The QUERY command in this Query Set causes all records within the file (with non-blank SALARY) to be extracted and presented to the Sort Module. The records are then sorted in descending sequence and presented to the Print Module. The Print Module prints only one record (with the P, 1 option) and this record will be the NAME and SALARY for the individual with the highest salary. If more than one person shares a top salary, the PRINT command option for the number of lines to be printed can be raised until the user is assured the top salaried records have been printed.

If low values are desired, the SORT command is changed to an ascending sort and the preceding technique is used.

3.3.1.8 Field-to-Field Comparisons

MIRADS also provides the capability to make field-to-field comparisons within a data record to determine the relations between fields. Example:

```
Q, PLANNED-COST GT *ACTUAL-COST.
```

The above example causes the PLANNED-COST to be compared with the ACTUAL-COST. If the PLANNED-COST is greater than the ACTUAL-COST, the Query is satisfied and the record is written onto the Hitfile. The asterisk (*) for the second field indicates the Criterion is a Field-Name and not a data value. The program retrieves the data in the first Field-Name and uses it for comparison with the data in the other Field-Name.

3.3.1.9 Types of Searches

There are four types of searches which may be used to answer Queries. They are:

1. Regular - RG

2. Character - CH

3. Keyword - KW

4. Keyphrase - KP

The search type specifies whether the entire content of the Data Base field is to be used as the selection criterion or that only a portion of the Data Base field is to be used. This feature allows the user to perform a search technique other than the type specified in the Dictionary which describes his Data Base.

If the user does not specify the type of search to be performed on the QUERY command, the search type will be defined in the user's Dictionary which describes his Data Base.

1. Regular Search

   The Regular search is the primary technique used in MIRADS. The search begins with the leftmost position of the Data Base field and compares the entire content of the field against the selection criterion. To illustrate this type of search, the following Data Base records are used.

	VALUE
Record 1	RADAR STATION
Record 2	RADAR STATION
Record 3	STATION, PDR RADAR
Record 4	RADAR STATION

If the QUERY command is entered

Q, VALUE = 'RADAR STATION 'RG.

Record number 1 will be selected because the Criterion is exactly the same as the Data Base value. Record numbers 2, 3, and 4 are rejected because of the leading and embedded blanks.

2. Character Search

The Character search begins on the left with the first character of a data field and tests it for value equal to the Criterion. The number of characters that are compared is equal to the number of characters in the Criterion. Therefore, the user should make sure that the Criterion is not longer than the data field. Example:

Q, INV-PT-NUMB EQ '422' CH.

If a Character Search was conducted on the following records for an INV-PT-NUMB value of 422,

	INV-PT-NUMB
Record 1	422520053
Record 2	422587931
Record 3	839486422

record-number 1 would meet the selection Criterion. Record number 2 does not qualify because the first three characters are blank and record 3 is omitted because the search is terminated after three characters (the number of characters in 422) have been compared. This test can be used on alpha, alphanumeric, or numeric fields and it is commonly used for searching fields such as name and address when the exact data storage format is not known.

3. Keyword Search

The Keyword search is normally used for searching data fields for keywords. The search begins on the left of a data field by checking for a value equal to the Criterion. If matching characters are found at the beginning of a field followed by a space, at the end of a field preceded by a space, or anywhere within the field preceded and followed by a space, the search is satisfied. Example:

```
Q, NAME EQ 'JIM'KW.
```

This example could be used to locate all persons with a first name of JIM. If a Keyword search was conducted on the following records for NAME equal to 'JIM',

	NAME		
Record 1	JONES	JIMMY	R.
Record 2	SMITH	THOMAS	N.
Record 3	RAINES	JIM,	C.
Record 4	THOMAS	JIM	B.

only record number 4 would meet the selection Criterion. Records 1 and 3 are rejected because of the MY and comma attached to the name JIM.

4. <u>Keyphrase Search</u>

The Keyphrase search is primarily used for searching narrative type data for key phrases. The search technique is a character by character comparison beginning on the left of the data field. If the first comparison fails, the first character of the Criterion is compared with the next n characters of the data. This process continues until the data field is exhausted. If matching characters for the Criterion are found anywhere within the data field, the search is satisfied. This type of searching is less efficient than the Character or Keyword searches and is discouraged when other means can be used for satisfying the Query. Example:

```
Q, TECHN-BRIEF EQUAL 'RADAR SYSTEM'KP.
```

The preceding example could be used to located all TECHN-BRIEFs within a Data Base which contain information regarding RADAR SYSTEMs. If a Keyphrase search was conducted on the following records to locate all TECHN-BRIEFs on RADAR SYSTEMs,

	TECHN-BRIEF
Record 1	GROUND SYSTEMS RADAR REPORTS
Record 2	ADVANCED RADAR SYSTEMS
Record 3	RECOVERY RADAR SYSTEM DATA

record numbers 2 and 3 would be selected because the phrase RADAR SYSTEM is contained in both records. Record number 1 is rejected because RADAR SYSTEM is not equal to SYSTEMS RADAR. The Keyphrase search is not limited to searching for phrases. It may be used for keywords or key strings of data. Nevertheless, the Character search is preferred to the Keyphrase search when it can be used because it operates more efficiently.

3.3.1.10 QUERY Command for Multilevel Data Bases

The flexible structuring capabilities of a Data Base necessitates parameters for the QUERY command when performing a search on a multilevel Data Base. The parameter which is used for searching multilevel Data Bases is shown in the following example:

SEARCH-n

Where n represents a one digit integer value with a range of 1-4.

The function of the SEARCH-n option is to restrict the number of Data Base levels which are to be extracted by the QUERY command. To illustrate this, the following Data Base structure is used.

BASEBALL DATA BASE

```
Level 1          League-1
 Level 2             Team-1-1
    Level 3              Player-1-1-1
                         Player-1-1-2
                         Player-1-1-n
                     Team-1-2
                         Player-1-2-1
                         Player-1-2-2
                         Player-1-2-n
                 League-2
                     Team-2-1
                         Player-2-1-1
                         Player-2-1-2
                         Player-2-1-n
```

The QUERY command extracts information from a multilevel Data Base in sets of related records. If the QUERY command

Q, TEAM = 'data value for TEAM-1-1'.

is executed, the Hit File produced by this command would be generated as record sets in the following order.

```
League-1
   Team-1-1            Set-1
      Player-1-1-1

League-1
   Team-1-1            Set-2
      Player-1-1-2

League-1
   Team-1-1            Set-3
      Player-1-1-n
```

From this Hitfile, any information concerning the requested team and also league and player information is available to the remaining commands of the Query Set. With this type of Hitfile, league and team are duplicate records within each set. If league and/or team are printed by the PRINT command, they will be listed as separate records.

If the QUERY command is specified with the SEARCH-n option

```
Q, SEARCH-2, TEAM = 'data value for TEAM-1-1'.
```

the data sets extracted will be limited to nothing lower than Level-2.

```
League-1
   Team-1-1
```

In this case, only two records will be written onto the Hitfile. From this Hitfile, only the league and team information is available to the remaining commands of the Query Set. With this type of Hitfile, the PRINT command will not duplicate league and team because there are no duplicate records in the Hitfile.

In summation, the SEARCH-n option is used to inhibit the extraction of lower level records which contain unwanted information and eliminates the duplication of requested records.

3.3.1.11 QUERY Command Search Limit

A user who wishes to only sample a Data Base for a given Query Set instead of extracting all records which meet the selection criteria can specify a search limit on the QUERY command. The parameter which is used for limiting the Data Base search is

```
LIMIT=n
  or
LIM=n
```

where n represents the maximum number of Hits the user requires. The entire option can be entered with or without embedded spaces on each side of the equal sign.

```
Q, SEARCH-n, LIM=n,  Field-Name Relational Criterion --.
Q, LIM=n,  Field-Name Relational Criterion ---.
```

If a search limit is specified on the QUERY command and a volume control value is specified on the PRINT command, the smaller limit will be used.

3.3.2 SORT Command

The SORT command provides the capability to sequentially order a set of data records. It operates on the selected records or Hitfile created by the QUERY command. The SORT command can use multi-fields to sort the records of a file in either ascending or descending order. It also provides the capability to sort on partial fields in either ascending or descending order by designating the number of characters within a field to be used as the sort key. Another capability of this command is the selecting of primary and minor sort fields. This is controlled simply by the order in which the fields occur in the SORT command. The first Field-Name in a SORT command is the primary sort field and all additional sort fields are sorted within the sort field to its immediate left. The SORT command uses the standard sorting sequence commonly referred to as the commercial sequence. This sort sequence (ascending) is as follows: @, [,], #, Δ, space, A through Z, special characters)-+<=>&$*(%:? !,\, 0 through 9, quote, ;, /, ., and ¤, ≠.

The user should be aware that sorting is a very time consuming process. Sort time is reduced if the requested sort sequence is as near as possible to the existing file sequence. Specifying fewer characters to be used for the sort key than the complete field size will also save sort time.

The SORT command can be specified following the QUERY command and/or the COMPUTE command.

After the previous command has been transmitted to the computer, a system message

READY

indicates that the SORT command can now be entered and transmitted to the computer. Figure 3-3 illustrates the SORT command format.

$$\begin{array}{cccccc}
\left\{\begin{matrix}\text{S}\\ \text{SORT}\end{matrix}\right\} & [\,'\,] & \left\{\begin{matrix}\text{Field-Name}\\ \text{\$New-Variable}\end{matrix}\right\} & \left[\begin{matrix}\text{A}\\ \text{ASCENDING}\\ \text{D}\\ \text{DESCENDING}\end{matrix}\right] & [\text{n}] & \left[\begin{matrix},\\ .\end{matrix}\right] \\
1 & 2 & 3 & 4 & 5 & 6
\end{array}$$

ELEMENTS WITHIN { } ARE REQUIRED

ELEMENTS WITHIN [ ] ARE OPTIONAL

Figure 3-3. SORT Command Format

The following numbers correspond with the element numbers in Figure 3-3. Elements within braces { } are required while elements within brackets [ ] are optional.

1. $\left\{\begin{array}{l}\text{S}\\\text{SORT}\end{array}\right\}$

   This element identifies the command as a SORT command.

2. [,]

   The comma is the desired means of punctuation. A variable number of spaces can be used in place of the comma.

3. $\left\{\begin{array}{l}\text{Field-Name}\\\text{\$New-Variable}\end{array}\right\}$

   This element represents the Dictionary name or the New-Variable name which is to be used as a key to sort the records.

4. $\left[\begin{array}{l}\text{A}\\\text{Ascending}\\\text{D}\\\text{Descending}\end{array}\right]$

   This element indicates the order of the sort (ascending or descending). This element will default to an A (ascending) if no option is specified.

   a. $\left[\begin{array}{l}\text{A}\\\text{Ascending}\end{array}\right]$

      This option causes the field to be sorted in ascending order or from low to high.

b. $\begin{bmatrix} \text{D} \\ \text{Descending} \end{bmatrix}$

This option causes the field to be sorted in descending order or from high to low.

5. [n]

This numerical element indicates the number of leftmost characters of the designated Field-Name which are to be used as the sort key. If this option is omitted, the entire data field will be used for sort purposes. The maximum value for n is 48, however n must never have a value which is greater than the field size. This option should not be used with a New-Variable.

6. $\begin{bmatrix} , \\ . \end{bmatrix}$

Sort field separator and terminator.

a. [,]

The comma option provides a minor sort capability and allows the user to specify additional fields to be used in sorting. A maximum of ten fields or 78 characters may be so designated. A space can be used in place of the comma.

b. [.]

The period option terminates the SORT command. The transmission of the next command will automatically terminate the SORT command if the period is omitted.

3.3.2.1 Rules for Using SORT Command

The following rules must be adhered to for the proper use of the SORT command:

1. Sort statements which cannot be contained on one line may be continued on the following line beginning in any position.
2. Elements such as Field-Name cannot be broken and continued on the following line. They should be entered as a continuous word on the following line.
3. The SORT key can contain a maximum of 10 sort fields, or a combined total of 78 characters.
4. The SORT command is optional and may be omitted; however, if it is used it should follow the QUERY command and/or COMPUTE command.
5. Data Base fields which contain more than 48 characters must be limited to no more than 48 characters when used as a sort key.
6. The SORT command must follow the COMPUTE command if a New-Variable Sort is to be performed.

3.3.2.2 Single Sort Fields

The simplest sort that can be formed is the request to sort the records on one Field-Name, either in ascending or descending order. Example:

```
S, NAME.
```

In this example, the SORT command would sort the file on the field called NAME in ascending order (default option) using the entire field length as the sort key.

Original Hitfile Sequence		New Hitfile Sequence	
Rec No.	Name	Rec No.	Name
1	JONES P J	5	DAVIS D D
2	SMITH R A	4	JOHNSON P A
3	WALLACE I N	1	JONES P J
4	JOHNSON P A	2	SMITH R A
5	DAVIS D D	3	WALLACE I N

Example:

```
S, NAME DESCENDING.
```

In this example, the SORT command would sort the file on the field called NAME in descending order using the entire field length as the sort key.

Original Hitfile Sequence		New Hitfile Sequence	
Rec No.	Name	Rec No.	Name
1	JONES P J	3	WALLACE I N
2	SMITH R A	2	SMITH R A
3	WALLACE I N	1	JONES P J
4	JOHNSON P A	4	JOHNSON P A
5	DAVIS D D	5	DAVIS D D

Example:

```
S, CITY A 3.
```

In this example, the file is sorted in ascending order using only the first three characters of the field called CITY.

Original Hitfile Sequence		New Hitfile Sequence	
Rec No.	City	Rec No.	City
1	HUNTSVILLE	2	HOUSTON
2	HOUSTON	1	HUNTSVILLE
3	WASHINGTON	4	NEW YORK
4	NEW YORK	5	ST. LOUIS
5	ST. LOUIS	3	WASHINGTON

3.3.2.3 Multiple Sorts Fields

Example:

```
S, SEX A, SERIAL D.
```

In this example, records are being sorted on two data fields. The first or primary sort field is SEX which is used to sort the file in ascending order. The second or minor sort field is SERIAL which is used to sort the records in descending order within SEX.

Original Hitfile Sequence			New Hitfile Sequence		
Rec No.	Sex	Serial	Rec No.	Sex	Serial
1	M	100000	4	F	100004
2	M	100005	3	F	100003
3	F	100003	2	M	100005
4	F	100004	5	M	100002
5	M	100002	1	M	100000

Example:

```
S, SALARY A 2, SERIAL D.
```

In this example, records are sorted on two fields. The first or primary sort field is the first two characters of SALARY and is used to sort the file in ascending order. The second or minor sort field is SERIAL which is used to sort records in descending order within the primary sort field.

Original Hitfile Sequence			New Hitfile Sequence		
Rec No.	Salary	Serial	Rec No.	Salary	Serial
1	010000	100000	4	008500	100004
2	011000	100005	3	009000	100003
3	009000	100003	2	011000	100005
4	008500	100004	5	012500	100002
5	012500	100002	1	010000	100000

3.3.3 COMPUTE Command

The COMPUTE command provides the capability of performing calculations on the data records extracted from the user's Data Base and written onto the Hitfile by the Query Module. Calculation capabilities include addition, subtraction, multiplication, division, exponentiation, counting, and summing. The COMPUTE command is entered after the system has responded with the message:

READY

Figure 3-4 illustrates the COMPUTE command format. A choice of elements within braces { } is required while elements within brackets [ ] may be omitted. The following breakdown refers to the numbered items in Figure 3-4.

1. $\left\{\begin{array}{l}\text{C}\\\text{COMPUTE}\end{array}\right\}$

   This element identifies the command as a COMPUTE command.

2, 4, 6. [,]

   A comma is the desired means of punctuation. A variable number of spaces can be used in place of the comma.

3. $\left\{\begin{array}{l}\text{ALL}\\\text{NONE}\\\text{Break-Field-Name}\end{array}\right\}$

   This element is referred to as the Break-Field-Element and it determines when the New-Variable is to be printed and its computed value reset to zero.

$$\begin{array}{cccccccccc}
\begin{Bmatrix}\text{C}\\ \text{COMPUTE}\end{Bmatrix} & [,] & \begin{Bmatrix}\text{ALL}\\ \text{NONE}\\ \text{BREAK-FIELD-NAME}\end{Bmatrix} & [,] & \begin{bmatrix}\text{n}\end{bmatrix} & [,] & \begin{Bmatrix}\text{\$NEW-VARIABLE}\end{Bmatrix} & \begin{Bmatrix}=\\ \text{EQ}\\ \text{EQUAL}\\ \text{EQUALS}\end{Bmatrix} & \begin{Bmatrix}\text{COUNT FIELD-NAME}\\ \text{SUM FIELD-NAME}\\ \text{ALGEBRAIC EXPRESSION}\end{Bmatrix} & [.] \\
1 & 2 & 3 & 4 & 5 & 6 & 7 & 8 & 9 & 10
\end{array}$$

ELEMENTS WITHIN { } ARE REQUIRED
ELEMENTS WITHIN [ ] ARE OPTIONAL

Figure 3-4. COMPUTE Command Format

a. { ALL }

The ALL element directs the Compute Module to report the specified calculation, sum, or count for each record in the Hitfile.

b. { NONE }

The NONE element directs the Compute Module to report the specified calculation, sum, or count after all records in the Hitfile have been processed.

c. { Break-Field-Name }

The Break-Field-Name directs the Compute Module to report the specified calculation, sum, or count (and reset it to zero) when the Data Base value of this field changes. If the value never changes, the calculation, sum, or count is reported when all records in the Hitfile have been processed.

5. [n]

This element controls the accuracy of the computed New-Variable and is the number of decimal places after the decimal point in the New-Variable. The maximum value for n is 8. If this element is omitted, a value of zero (0) will be assumed.

7. { $New-Variable }

This element establishes the title or name by which the result of the computation, sum, or count is identified. A dollar sign ($) is required as the first character of the New-Variable-Name with any combination of alphanumeric characters up to 35. There must be at least one character in addition to the dollar sign. Imbedded blanks are not allowed. Hyphens (-) may be used to separate words if desired.

8. {
=
EQ
EQUAL
EQUALS
}

The New-Variable must be separated from the SUM, COUNT, or Algebraic Expression by an equal sign (=) or any of the equivalents shown.

9. {
SUM Field-Name
COUNT Field-Name
Algebraic Expression
}

This element indicates the type of computation desired.

a. { SUM Field-Name }

The SUM element generates a running total of the sum of data values for the specified Field-Name. This total can be printed when the contents of a specified field changes by using the Break-Field-Element described earlier. After a break, the SUM is reinitialized to zero and summarization continues until the next break occurs.

b. $\{$COUNT Field-Name$\}$

The COUNT element counts the number of occurrences of a given Field-Name provided the value contained in the Field-Name is not blank. The Break-Field-Element establishes when the COUNT is reinitiated.

c. $\{$Algebraic Expression$\}$

The Algebraic element is an algebraic expression of the computation that is to be performed. The Break-Field-Element establishes when the calculation is performed.

(1) Only the operations of addition (+), subtraction (-), multiplication (*), division (/), or exponentiation (**) can be performed using the designated symbols.

(2) Operations within parentheses are performed first. If there is more than one level of parentheses, the information within the lowest (internal) level of parentheses is performed first.

(3) Within parentheses or in an expression without parentheses, the calculation is performed with the following hierarchy for operations--exponentiation,

multiplication or division, and addition or subtraction. Arithmetic functions are performed from right to left within each priority level. For example, the expression

SALARY * PERCENT - BONUS / MONTHS

would be performed in three steps

(a) BONUS / MONTHS = Value A

(b) SALARY * PERCENT = Value B

(c) Value B - Value A = Result

Because of the hierarchical order of processing algebraic expressions, it is better to use parentheses to ensure that mathematical operations are performed in the intended order. A maximum of three levels of parentheses (at any given point in the development of the expression) is the only restriction.

10. [.]

A period terminates the COMPUTE command. Transmission of the next command will automatically terminate the COMPUTE command if the period is omitted.

3.3.3.1 Rules for Using COMPUTE Command

The following rules must be adhered to for proper use of the COMPUTE command:

1. A period (decimal point) may be used within the command but must not be immediately followed by a space or it will be interpreted as the end of the statement. Example: 12.0 is acceptable whereas 12. is not.

2. COMPUTE commands which cannot be contained on one line may be continued on the following line starting in any position.

3. Compute elements cannot be broken and continued on the following line. They should be entered as a continuous word on the following line.

4. A space must precede and follow the minus sign (-). The remaining Compute elements may or may not be separated from adjacent elements with one or more spaces.

5. Constants must not be enclosed in quotation marks.

6. A maximum of 20 COMPUTE commands or New-Variables is permitted for each Query Set.

7. The New-Variable field must begin with a dollar ($) sign, may contain any combination of alphanumeric characters up to 36 including the dollar sign, and must contain at least one character other than the dollar sign. Hyphens (-) may be used to separate words when desired.

8. A maximum of three levels of parentheses is allowed for an Algebraic Expression within the COMPUTE command

and the number of left parentheses must be equal to the number of right parentheses.

9. New-Variables may not be used as a Break-Field-Name.

10. Each COMPUTE command may contain up to 40 elements with the New-Variable, equal sign, Field-Names, parentheses and operators (addition, subtraction, etc.) counting as one each.

11. Each COMPUTE command may contain a maximum of 10 constants.

12. Each COMPUTE command may contain a maximum of 14 Field-Names.

13. New-Variables used in the Algebraic Expression portion of a COMPUTE command must have been previously defined in a prior COMPUTE command. Only one New-Variable can be created in a COMPUTE statement.

14. Exponentiation restrictions are:

    (a) Integers can be raised to integer power

    (b) Real numbers can be raised to an integer power

    (c) Positive real numbers can be raised to a real power

    Any exception to these restrictions will result in an error.

3.3.3.2 Simple COMPUTE Commands

The COMPUTE command can be used to sum, count or make algebraic calculations. The results of these calculations vary greatly depending on which Break-Field-Element (ALL, NONE, Break-Field-Name) is used with the COMPUTE command. The following examples will illustrate how the COMPUTE command may be used to obtain the desired results.

3.3.3.3 SUM Using the COMPUTE Command

The SUM function causes the Compute Module to add the successive contents of a particular data field forming a sub-total. This subtotal can be broken when the contents of a specified data field changes by using the Break-Field-Element option described later. After a break, the SUM is reinitialized to zero and the formation of another sub-total continues until the break occurs. Consider the following examples operating on the HITFILE shown.

HITFILE

Employee-Name	Annual-Salary
ALLEN	6000.00
BLAIR	6000.00
DAVIS	12000.00
FARR	12000.00
GATES	12000.00
SCOTT	18000.00
TURNEY	18000.00

Examples:

```
1. C,ANNUAL-SAL, 2, $TOTAL-ANNUAL-SALARY-PER-CLASS =
     SUM ANNUAL-SAL.
2. C,NONE, 2, $TOTAL-ANNUAL-SALARY = SUM ANNUAL-SAL.
```

Sample Printout:

```
EMPLOYEE-                          TOTAL-ANNUAL-          TOTAL-ANNUAL-
NAME        ANNUAL-SALARY      SALARY-PER-CLASS       SALARY

ALLEN            6000.00
BLAIR            6000.00            12000.00
DAVIS           12000.00
FARR            12000.00
GATES           12000.00            36000.00
SCOTT           18000.00
TURNEY          18000.00            36000.00               84000.00
```

In the first example, the total annual salary for each salary classification (6000.00, 12000.00 and 18000.00) is to be calculated to two decimal places. The COMPUTE command directs the Compute Module to begin summing the ANNUAL-SAL field until the Break-Field-Element ANNUAL-SAL changes value. The sub-total (12000.00) is then recorded and its value reinitialized to zero. The summation then begins for the next salary class and continues until all values are derived as shown in the column headed by $TOTAL-ANNUAL-SALARY-PER-CLASS above.

In the second example, the total annual salary for all salary classifications is to be computed to two decimal places. The break element NONE is used in this example to create a sub-total or sum as each record is processed. The NONE Break-Field-Element indicates to the Compute Module that the sub-total is to be continued until the end-of-file is reached. At that point the total annual salary (84000.00) is recorded in the column headed by $TOTAL-ANNUAL-SALARY. It should be emphasized that the Hitfile would have to be in sequence by ANNUAL-SAL to produce the above results. Note: If the Hitfile were not in sequence by ANNUAL-SAL, a SORT command would be necessary.

Thus, the two examples are functionally equal except for the Break-Field-Element. This option causes the Compute Module to produce entirely different results. The ALL Break-Field-Element has no real value when using the SUM or COUNT options since the sub-total or COUNT would be reported and reinitialized for each record processed.

3.3.3.4 COUNT Using the COMPUTE Command

The COUNT function counts the number of occurrences of a given non-blank data field. The data field used for the COUNT option is limited to a maximum of 48 characters. The Break-Field-Element establishes when the count is reported and reinitiated just as it does with the SUM function. The following examples illustrate the use of the COUNT function for the Hitfile shown in the previous example.

Examples:

1. C, ANNUAL-SAL, 0, $COUNT-EMPLOYEES-PER-SALARY-CLASS = COUNT ANNUAL-SAL.
2. C, NONE, 0, $COUNT-ALL-EMPLOYEES = COUNT ANNUAL-SAL.

Sample Printout:

```
EMPLOYEE-   ANNUAL-    COUNT-EMPLOYEES-     COUNT-ALL-
NAME        SALARY     PER-SALARY-CLASS     EMPLOYEES

ALLEN        6000.00
BLAIR        6000.00          2
DAVIS       12000.00
FARR        12000.00
GATES       12000.00          3
SCOTT       18000.00
TURNEY      18000.00          2                  7
```

For the examples above, there is a one-to-one correspondence between ANNUAL-SAL and EMP-NAME; therefore, a count of the ANNUAL-SAL will result in a count of the employees.

In the first example, an employee count (to 0 decimal places) is to be taken by salary classifications. Every time a record is processed a count sub-total is incremented by 1 provided the Break-Field-Element ANNUAL-SAL does not contain all spaces. This sub-total continued until the value for ANNUAL-SAL changes. The sub-total is then reported under the COUNT-EMPLOYEES-PER-SALARY-CLASS column, reinitialized to zero, and the process continued until the end-of-file is reached.

In the second example, an employee count (to 0 decimal places) is to be calculated for the entire file. The Break-Field-Element NONE directs the Compute Module to maintain the count sub-total of the

ANNUAL-SAL field until the end-of-file is reached. The total is then reported under the COUNT-ALL-EMPLOYEES column.

NOTE

> If the Hitfile was not in sequence by ANNUAL-SAL, a SORT command would be necessary.

3.3.3.5 Algebraic Expression Using the COMPUTE Command

This capability of the COMPUTE command may be used for solving algebraic equations involving addition, subtraction, multiplication, division and exponentiation. The Break-Field-Element determines when the formula calculations are to be made just as with the SUM and COUNT functions. The following example illustrates the multiple use of COMPUTE commands for a given Query Set as well as a formula calculation for the Hitfile previously shown.

Examples:

```
1.  C, NONE, 2, $TOTAL-SALARIES = SUM ANNUAL-SAL.
2.  C, NONE, 0, $TOTAL-COUNT = COUNT ANNUAL-SAL.
3.  C, NONE, 2, $AVG-MONTHLY-SALARY =
      ($TOTAL-SALARIES / $TOTAL-COUNT) / 12.
```

Sample Printout:

```
EMPLOYEE-   ANNUAL-    TOTAL-     TOTAL-   AVG-MONTHLY-
NAME        SALARY     SALARIES   COUNT    SALARY

ALLEN        6000.00
BLAIR        6000.00
DAVIS       12000.00
FARR        12000.00
GATES       12000.00
SCOTT       18000.00
TURNEY      18000.00   84000.00     7         1000.00
```

This example illustrates how a formula may be used in the COMPUTE command to calculate the average monthly salary for a group of employees given only their annual salaries. Three COMPUTE commands are required to solve this particular problem. The first command computes the $TOTAL-SALARIES for the group and the second command is used to tabulate the $TOTAL-COUNT or the number of employees in the file. The New-Variables $TOTAL-SALARIES and $TOTAL-COUNT are then used in the third COMPUTE command to calculate the New-Variable $AVG-MONTHLY-SALARY. It is important to note that the New-Variables referenced in the Algebraic Expression portion of the last COMPUTE command must have been defined in a prior command. The average monthly salary is computed by first dividing the $TOTAL-SALARIES by the $TOTAL-COUNT and then subsequently dividing that result by 12. Parentheses are used in the Expression to clearly indicate the order of operations.

Since the Break-Field-Element NONE is used in the last COMPUTE command, the New-Variable $AVG-MONTHLY-SALARY is not computed until the entire file has been processed.

When multiple COMPUTE commands are specified with each using the same Break-Field-Element, they are executed in the order in which they are transmitted to the computer.

Complex algebraic equations may be solved using the formula capability of the COMPUTE command. The following example might be used to compute a new monthly salary assuming that each employee in the given file is to receive a 12% increase in salary.

Example:

```
C,ALL, 2,$NEW-MONTHLY-SALARY = (OLD-SALARY +
  (OLD-SALARY * .12)) / 12.
```

Sample Printout:

```
NAME          OLD-SALARY      NEW-MONTHLY-SALARY

ALLEN           6000.00               560.00
BLAIR           6000.00               560.00
DAVIS          12000.00              1120.00
FARR           12000.00              1120.00
GATES          12000.00              1120.00
SCOTT          18000.00              1680.00
TURNEY         18000.00              1680.00
```

The computation is performed by first multiplying the OLD-SALARY by .12 clearing the innermost set of parentheses. The result of this step is then added to the OLD-SALARY thus clearing the outer set of parentheses. This total is divided by 12 giving the final answer which is reported in the New-Variable $NEW-MONTHLY-SALARY, correct to two decimal places. The entire calculation described above occurs for each record since the ALL Break-Field-Element is used in the COMPUTE command.

3.3.4 PRINT Command

The PRINT command permits the user to display data from the Hitfile in a user specified format which allows data suppression, table lookup, columnar positioning and line skipping. The PRINT command allows up to five lines of Field-Titles or a total of fifty Field-Titles; therefore, the capability of printing five lines of titles and five lines of data for each record in the final HITFILE. The data suppression capability suppresses printing of successive major and/or minor fields until the value of the specified field changes. The table lookup feature decodes a value within a field and prints out a stored literal in place of the field value. This feature is useful for displaying data in more meaningful terminology in the event that the data is recorded in a coded or condensed format. The columnar positioning capability facilitates printing data beginning in any column on the print page instead of being restricted to the next available column. The line skipping capability permits the user to space out the print format to skip lines before or after printing each report line.

After the user transmits the previous command, the computer will indicate its readiness to receive the PRINT command by sending the message

READY

to the terminal. At this time, the user can input the PRINT command under the following format.

Figure 3-5 shows the PRINT command format. Elements shown within braces { } are required while elements shown within brackets [ ] are optional. The following breakdown refers to the numbered areas in Figure 3-5.

1. $\begin{Bmatrix} \text{P} \\ \text{PRINT} \end{Bmatrix}$

   This element identifies the command as a PRINT command.

2,4,6. [,]

   The comma is the desired means of punctuation. A variable number of spaces can be used in place of the comma.

```
{ P     }         [ n   ]         [ BRK-CNT     ]   [ SPACE n ]   [PPOS n]   [ BRK n   ]
{ PRINT }  [,]    [ SM  ]   [,]   [ BREAK-COUNT ]   [ SP n    ]              [ BREAK n ]
                  [ SUM ]
    1       2        3       4           5                6            7           8


{ Field-Name    }   [ BRK n   ]   [ LOOKUP ]   [ GROUP n ]   [*n]   [ SPACE n ]   [ ; ]
{ $New-Variable }   [ BREAK n ]   [ TLU    ]   [ G n     ]          [ SP n    ]   [ , ]
                                  [ L      ]                                      [ . ]
        9               10            11            12         13        14         15
```

ELEMENTS WITHIN { } ARE REQUIRED

ELEMENTS WITHIN [ ] ARE OPTIONAL

Figure 3-5. PRINT Command Format

3. $\begin{bmatrix} \text{n} \\ \text{SM} \\ \text{SUM} \end{bmatrix}$

This element indicates the number of lines of printed output desired. The element can also have a direct effect on the QUERY command. Unnecessary Data Base searching can be restricted through the user of the 'n' option if the Query Set does not contain a SORT command. In this case, the Query Module will only search until the number of Hits is equal to the output limit specified in the PRINT command. If this element is omitted, the system will default to printing of the complete Hitfile.

a. [n]

The n indicates the number of lines of output to be printed and must be a value from 1-999999 if specified.

b. $\begin{bmatrix} \text{SM} \\ \text{SUM} \end{bmatrix}$

The SUM option for print control is used in conjunction with the COMPUTE command. It will suppress the printing of all Hit records except for the record at the time a New-Variable is to be printed.

5. [BREAK-COUNT
BRK-CNT]

This option operates in conjunction with the BREAK element. The number of times each Break-Field causes a line advancement will be printed at the end of the generated report when this option is specified.

7. [SPACE n
SP n]

This element permits the skipping of n blank lines before the printing of a report line. The n represents the number (1-8) of blank lines to be skipped. A value of 9 for n will cause the line printer to do a page-ejection.

8. [PPOS n]

This element provides for the placement of the data for a given Field-Name or New-Variable at a specified print position on the report line. Otherwise, the print position of each data field is calculated by the Print program from information supplied in the Dictionary. The n represents the print position (1-128) of the first data character from the identifying Field-Name or New-Variable.

9,11. [BREAK n
BRK n]

This option permits the skipping of n blank lines after printing a report line when the value contained in the Data Base field specified as a Break-Field changes. Specifying the option before or after the Field-Name causes the line to be printed and then skip n number of lines. The n represents the number (1-8) of blank lines to be skipped. A value of 9 for n will cause the line printer to do a page-ejection.

10. [Field-Name
$New-Variable]

This element indicates which Field-Names or New-Variables are to be printed in the output. A minimum of one Field-Name or New-Variable is mandatory. These names are printed in the order in which they appear in the PRINT command. Each name must be separated from the next name with a comma, semicolon, or space.

12. [LOOKUP
TLU
L]

This element is used to print special messages or titles as indicated by special code values contained in a field. The Field-Name must have been designated as a Table-Lookup field in the Dictionary. Example: A one digit field could contain the coded names of cities such as Chicago, Detroit, and New York City. When this field is printed, the user may wish the full name of each city to be printed rather than the coded value. The Table Lookup capability may be employed to accomplish this objective.

13. $\begin{bmatrix}\text{GROUP n}\\ \text{G n}\end{bmatrix}$

This element is used to indicate the print suppression of repeating data values for a particular data field. The associated Field-Name value will be printed only when the field value changes. The value of n indicates a field suppression. It can be used with the * n element below in which case the G n would represent a major field suppression and the * n would be a minor suppression dependent on G. N may be any value from 1 to 9 and must be preceeded by a space.

14. [* n]

This element is used in conjunction with the GROUP n suppression element to indicate minor fields which are to be suppressed when the major Field-Name as specified by G n is suppressed. The value of n must correspond with some value of n for a prior GROUP n operation. There must be a space between the * and the number n.

15. [SPACE n
SP n]

This element permits line skipping of n blank lines (1-8) after the printing of a report line. A value of 9 for n will cause the line printer to do a page-ejection. The print positioning on the report line will not be reset to one unless this element is followed by a semicolon.

16. [,
.
;]

This element is used as punctuation of the PRINT command.

a. [,]

The comma is used to separate Field-Names and New Variables within the PRINT command. In most cases, the comma is optional for punctuation, however, the comma will be required when multiple options are being used on a specified field. The comma will guarantee that the options are performed on the proper field.

b. [;]

The semicolon also separates Field-Names and New-Variables within the PRINT command with the difference being that the Field-Name or New-Variable following the semicolon will be printed on a new line of the report.

c. [.]

The period terminates the PRINT command. Transmission of the next command will automatically terminate the PRINT command if the period is omitted.

3.3.4.1 Rules for Using the PRINT Command

The following rules must be adhered to in using the PRINT command. Any deviation from these rules will result in an error.

1. More than 1 line of input is permitted for the PRINT command. Continuation lines can begin in any position.

2. The n option used to indicate the number of lines of output desired must be from 1 to 999999.

3. In using the SPACE n option, the value of n must be between 1 and 9 inclusive for the space before print. Space after print can be 0 to 9.

4. The value of n must be between 1 and 128 inclusive when using the PPOS option. When positioning fields within a line, the value of n must allow two spaces beyond the last position that any previous Field-Name or New-Variable occupies.

5. Field-Names must be entered exactly as they are described in the Dictionary.

6. A maximum of 50 Field-Names can be included in a PRINT command.

7. The value of n in the GROUP n option and the * n option must not be greater than 9.

8. The * n option cannot be used to indicate minor field suppression unless there is a prior GROUP n major field suppression having the same value of n.

9. In using the semicolon (;) the maximum number of title or heading lines allowed is 5. This will permit up to five lines of titles at the beginning of the report and five corresponding data lines for each record of the Hitfile.

10. All options of SPACE n, BRK n, PPOS n, GROUP n, and * n must have a space between the element and its numeric value.

3.3.4.2 Simple PRINT Commands

T! e PRINT command is used to format the data values which will appear on each line of the output report. The PRINT functions are divided into two types: Line related and Field related.

1. Line Related Functions

The Line related functions are the Line Number (n, SM, SUM), SPACE, and semicolon (;) options. The

Line Number option must be followed by a comma and indicate how many lines of output are to be printed in a report. The n specifies the number of lines between 1 and 999999 inclusive that are to be printed in the report. If no option is specified, the system maximum of 999999 is the default.

The following example illustrates the form for printing the Field-Names NAME and SALARY. Example:

P, NAME, SALARY.

The omission of the Line Number option will cause the entire report (system maximum) to be printed.

The user may limit the number of lines to be printed by using the nn option on the QUERY command. The nn option is used in this manner to prevent users from getting great volumes of output data which obviously cannot be printed on slow output devices such as the DCT-500. Nevertheless, if the complete output is required, it can be obtained by ommitting the option. The SUM or SM option is used to prevent the printing of any record or print line that does not contain a value for a New-Variable of a previous COMPUTE command. To illustrate the usage of the SUM or SM option, two examples (on the following pages) are given of the same Query Set. The only difference between the two Query Sets is the Line Number option. The first example uses the system maximum so that each Hit record is printed on the output report. In the second example, the SM option is used so that only two report lines are printed. The main function of this Query Set is to calculate the average salary for the two organizations 89 and 93.

The SPACE n option is related to the output line and permits variable spacing before and after printing of the report lines. The function, when used with the first field of a line, must come before the first Field-Name or New-Variable and means to skip n blank lines before printing the designated report line. The function, when used after a Field-Name or New Variable, means to skip n blank lines after printing the designated report line. The SPACE n function must not be separated from the related Field-Name or New-Variable by any punctuation (only spaces). The maximum value for n is 9. Values of one to eight specify line spacing of a like number of lines and a value of nine specifies ejection to a new page. A value of zero will print the field on the next line without resetting

Example 1: Standard Control for PRINT

```
  READY
Δ Q, ORG = 89 OR 93.
  READY
Δ S, ORG D, DEPT A.
  READY
Δ C, ORG, 2, $TOTAL-SALARY = SUM PAY.
  READY
Δ C, ORG, 0, $TOTAL-PERS = COUNT PAY.
  READY
Δ C, ORG, 2, $AVG-SALARY = $TOTAL-SALARY / $TOTAL-PERS.
  READY
Δ P, ORG, DEPT, PAY, $AVG-SALARY.
  READY
Δ RUN
  QUERY NOW PROCESSING
  FILE CONTAINS 562 RECORDS
  QUERY SELECTED 7 RECORDS
  ENTER OUTPUT REPORT SITE ID
Δ (Blank Line)
  Q, ORG = 89 OR 93.
  S, ORG D, DEPT A.
  C, ORG, 2, $TO'1AL-SALARY = SUM PAY.
  C, ORG, 0, $TOTAL-PERS = COUNT PAY.
  C, ORG, 2, $AVG-SALARY = $TOTAL-SALARY / $TOTAL-PERS.
  P, ORG, DEPT, PAY, $AVG-SALARY.


ORGANIZATION    DEPARTMENT     SALARY       AVG-SALARY
     93             101        6000.00
     93             102        8500.00
     93             103       10000.00
     93             104        7500.00         8000.00
     89             100       10000.00
     89             102       11000.00
     89             104       12000.00        11000.00
READY
```

Example 2: SUM Control for PRINT

```
  READY
Δ Q, ORG = 89 OR 93.
  READY
Δ S, ORG D, DEPT A.
  READY
Δ C, ORG, 2, $TOTAL-SALARY = SUM PAY.
  READY
Δ C, ORG, 0, $TOTAL-PERS = COUNT PAY.
  READY
Δ C, ORG, 2, $AVG-SALARY = $TOTAL-SALARY / $TOTAL-PERS.
  READY
Δ P, SM, ORG, DEPT, PAY, $AVG-SALARY.
  READY
Δ RUN
  QUERY NOW PROCESSING
  FILE CONTAINS 562 RECORDS
  QUERY SELECTED 7 RECORDS
  ENTER OUTPUT REPORT SITE ID
Δ (Blank Line)
  Q, ORG = 89 OR 93.
  S, ORG D, DEPT A.
  C, ORG, 2, $TOTAL-SALARY = SUM PAY.
  C, ORG, 0, $TOTAL-PERS = COUNT PAY.
  C, ORG, 2, $AVG-SALARY = $TOTAL-SALARY / $TOTAL-PERS.
  P, SM, ORG, DEPT, PAY, $AVG-SALARY.


ORGANIZATION     DEPARTMENT     SALARY        AVG-SALARY
     93              104         7500.00        8000.00
     89              104        12000.00       11000.00
READY
```

the Print position to one. The following examples illustrate the proper use of the SPACE n option.

Example 1: Space n Option

```
PRINT,SPACE 1 FLD-A, FLD-B.
```

The report generated by this PRINT command will print the title line and then space one blank line before printing each data line for FLD-A and FLD-B. The output for this command will be as follows:

```
FIELD-A        FIELD-B

ALPHA            1000
BETA             2000
GAMMA            3000
DELTA            4000
```

Example 2: Space n Option

```
PRINT, FLD-A SPACE 0,FLD-B.
```

The Space option used in this manner will cause the output report to be printed in the following manner:

```
FIELD-A
                FIELD-B
ALPHA
                  1000
BETA
                  2000
GAMMA
                  3000
DELTA
                  4000
```

The semicolon (;) is used to separate multiple occurrences of Field-Names or New-Variables when a new output line is to begin with the next Field-Name or New-Variable. The following example illustrates the use of the semicolon. Example:

```
PRINT,NAME, STREET, CITY;SERIAL, SALARY.
```

In this example, the fields NAME, STREET and CITY will be printed on one report line followed by SERIAL and SALARY on the next line. The title lines printed will also be specified on two lines in the same manner as the report lines.

Example 1 on the following page illustrates the use of the BRK and BRK-CNT options. Each time a new COG is encountered, the print format will advance to a new page. Within each page, a change in STP will cause the printer to advance one blank line. Page 1 and 2 illustrates how the

Example 1: BRK n and BRK-CNT Options

```
    READY
△  Q,(GRD = 14 OR 15) AND (COG = 1 OR 2).
    READY
△  S,COG.
    READY
△  P, COG L BRK 9 BRK-CNT G 1, GRD, STP BRK 1.
    READY
△  RUN
    QUERY NOW PROCESSING
      FILE CONTAINS        5301   RECORDS
    QUERY SELECTED            9   RECORDS
    ENTER OUTPUT REPORT SITE-ID
△  (Blank Line)
    Q,(GRD = 14 OR 15) AND (COG = 1 OR 2).
    S,COG.
    P,  COG L BRK 9 BRK-CNT G 1, GRD, STP BRK 1.

                                                          PAGE 1
               COG                      GRD       STP

    OFFICE OF DIRECTOR,  MSFC           15         8
                                        15         8

                                        15         6
                                        15         6

                                                          PAGE 2
               COG                      GRD       STP

    CHIEF COUNSEL                       15         7

                                        14        10

                                        14         8

                                        15         6

                                        14         9

               1  BREAKS FOR COG
               6 BREAKS FOR STP
```

change in STP caused one blank line to be inserted. Notice that the Criterion value for COG has been changed into a more meaningful title through the lookup capability. The BRK-CNT option generated the output at the bottom of the report specifying the number of breaks which occurred for each Break-Field.

2. Field Related Functions

The Field related functions are Field-Name, New-Variable,. PPOS, LOOKUP, GROUP, and Asterisk (*). The Field-Name and New-Variable function is required and describes the Field-Name and/or New-Variable to be printed.

The function must be separated from all other functions by a space or punctuation. A maximum of 50 Field-Names and/or New-Variables may be used, provided they can be printed within the 5 line maximum for each record.

The PPOS n function provides for the horizontal placement of a field on a print line. The value n is the print position where the first character of the data field is to be displayed or printed. The numeric value may be one, two, or three digits. It must not conflict with a print position already used by another field and each successive n must always progress in value. It must be separated from all other functions by a space or punctuation. The maximum value for n is 128.

The LOOKUP, TLU, or L function indicates that the data value for the associated Field-Name must be converted to the data value or literal in the lookup table and printed on this report. The function may be used only once for each Field-Name. The function must be separated from all other functions by a space or punctuation.

The GROUP or G function will permit the suppression of repetitive fields by controlling on the data content of the field. The n value from 1-9 allows up to nine controlling levels. A change in G 1 will cause the fields controlled by G 2 to G 9 to be printed even if they themselves have not changed. However, a change in G 4 will cause the fields controlled by G 5 to G 9 to be printed but will not affect G 1 to G 3. The function may be used only once for each field title. The function must be separated from all other functions by a space or punctuation. The maximum value for n is 9.

The * n function is associated with the GROUP function. The field designated by the * n is the controlled field and the field designated by a G n is the control field. The * n function is associated with the G n function of the same numeric value such as G 3 and * 3. Thus the *3 data field is suppressed whenever the G 3 field is suppressed. The G n and * n functions must not be used for the same Field-Name. They may be used only once for each Field-Name. The function must be separated from all other functions by a space or punctuation. The maximum value for n is 9.

The following example illustrates the use of the Field related functions. Example:

P, PPOS 5 NAME, CITY G 1, STATE * 1, LOC TLU.

The data field NAME will begin printing in print position 5 (PPOS 5) on the output line followed by CITY, STATE, and LOC. The fields CITY and STATE will start printing on the first line of output and will not be printed again on subsequent lines until the data value for CITY changes. The data value found in the Field-Name LOC will be used to search a Lookup Table in the Dictionary for a corresponding literal. The literal will be printed for the LOC field rather than some coded value.

Example 1: Group Suppression

```
   READY
Δ Q, ORG = 'ACCT'.
   READY
Δ S,GRD D,STP D.
   READY
Δ P, SIR, GRD G 1,  STP G 2.
   READY
Δ RUN
   QUERY NOW PROCESSING
   FILE CONTAINS 896 RECORDS
   QUERY SELECTED 10 RECORDS
   ENTER OUTPUT REPORT SITE ID
Δ (Blank Line)
   Q,ORG = 'ACCT'.
   S,GRD D,STP D.
   P,SIR,GRD G 1,  STP G 2.

            LAST-NAME          GRD    STP
      JONES                     15     08
      BROWN                            07
      SMITH                            06
      THOMAS                           05
      BLACK                     14     10
      DOE                              07
      JOHNSON
      SHANNON                          06
      LAMOTTE                   13     08
      BURNS                            07
   READY
```

In this example, two fields use the GROUP n function. Field suppression will be performed with the fields GRD and STP. The suppression of STP is controlled by a change in the content of GRD. Each time a different data value is found in GRD, STP is printed. Equal values for STP will then be suppressed until the value of GRD changes.

Example 2: Group Suppression

```
   READY
 Δ Q, ORG = 'ACCT'.
   READY
 Δ S, GRD D, STP D.
   READY
 Δ P, SIR, GRD G 1, STP * 1.
   READY
 Δ RUN
   QUERY NOW PROCESSING
   FILE CONTAINS 896 RECORDS
   QUERY SELECTED 10 RECORDS
   ENTER OUTPUT REPORT SITE ID
 Δ (Blank Line)
   Q, ORG = 'ACCT'.
   S, GRD D, STP D.
   P, SIR, GRD G 1, STP * 1.

          LAST-NAME           GRD     STP
      JONES                    15      08
      BROWN
      SMITH
      THOMAS
      BLACK                    14      10
      DOE
      JOHNSON
      SHANNON
      LAMOTTE                  13      08
      BURNS
   READY
```

Example 2 illustrates the GROUP n function and the associated * n function. In this example, STP printing is controlled by the * 1 option. The * 1 option is dependent upon the G 1 option. STP data prints only when GRD data is printed. GRD is suppressed until its data content changes. By comparing example one and two, the user can easily see the difference in operation of the GROUP n and * n functions.

3.3.5 Formatted Print

The user has the capability to predefine an output display consisting of one to three different print formats for displaying data from the Hitfile merely by using the FORMAT command and the assigned names of the desired formats. To display retrieved data in this fashion, the user must have previously defined the designated formats. The advantage over the PRINT command is that, once defined, the formats can be specified over and over in numerous Query Sets without the necessity of keying in a lengthly command for each set. Formatted Print is generally used to produce output reports which are similar to full page reports generated by dedicated batch processing programs. The commands associated with Formatted Print are:

1. DEFINE

   This command is used to name and create the format for Formatted Print.

2. FORMAT

   This command is used to select previously defined print formats which have been saved.

3.3.5.1 DEFINE Command

The DEFINE command is used to structure a desired format for displaying data from the Hitfile.

Figure 3-6 illustrates the structure of the DEFINE command. Elements contained within braces { } are required while elements shown with brackets [ ] are optional. The following breakdown refers to the numbered areas in Figure 3-6.

1. $\left\{\begin{matrix}\text{DEF}\\\text{DEFINE}\end{matrix}\right\}$

   This element identifies the command as a DEFINE command.

2. $\{,\}$

   The comma is the means of punctuation following the command designator.

3. $\left\{\text{Name}\right\}$

   This element specifies a unique one-to-twelve character name which is used to identify the print format requested on the FORMAT commands. The allowable character set for composing the Print-Format-Name is A through Z, 0 to 9, and hyphen (-).

4. $\{,\}$

   The comma is a means of punctuation following the Print-Format-Name.

5. $\left[\begin{matrix}\text{SPACE n}\\\text{SP n}\end{matrix}\right]$

   This element permits the skipping of n blank lines before printing the specified Field-Name. The value for n has a range of 1 to 54.

```
                                      {   *FIELD-NAME       }
                                      {  **FIELD-NAME       }
                                      {   *FIELD-NAME n     }
                                      {    FIELD-NAME (n)   }
{DEF   } {,}  {NAME}  {,}  [SPACE n]  {    FIELD-NAME       }  [F w.d]  [SPACE n]  [PPOS n]  [LOOKUP]  [RJ]  [RJT]  [PAGENR n]  {;}
{DEFINE} { }  {    }  { }  [SP n   ]  {   $NEW-VARIABLE     }           [SP n   ]            [TLU   ]  [LJ]  [LJT]              {,}
                                      {  *$NEW-VARIABLE     }                                [L     ]                            {•}
                                      { **$NEW-VARIABLE     }
                                      {  *$NEW-VARIABLE  n  }
                                      {   'LITERAL'         }
                                      {   n                 }

    1     2     3      4       5                 6                 7         8         9         10      11     12      13       14
```

ELEMENTS WITH { } ARE REQUIRED

ELEMENTS WITHIN [ ] ARE OPTIONAL

Figure 3-6. DEFINE Command Format

6.
```
{ *Field-Name          }
{ **Field-Name         }
{ *Field-Name n        }
{ Field-Name (n)       }
{ Field-Name           }
{ $New-Variable        }
{ *$New-Variable       }
{ **$New-Variable      }
{ *$New-Variable n     }
{ 'Literal'            }
{ n                    }
```

This element identifies which Field-Names, New-Variables, and/or literals are to be printed in the output report.

a. [*Field-Name]

An asterisk preceeding a Field-Name specifies to print only the Dictionary Report Title. If the user wishes to print only the data value for the specified field, he must specify the Field-Name without the asterisk.

b. [**Field-Name]

The data value for the field will be printed immediately following the Dictionary Report Title on the same line with one space between the title and value. For example, **FLD-A will print as FIELD-A 12345.

c. [*Field-Name n]

The Dictionary Report Title will be printed for the field specified. The data value for this field will be equated to a numeric identifier, called a synonym, specified by n which has a range of 1-200. To print the data value, the numeric identifier n must be positioned. (See option 6-k.) For example, *FLD-A 2 will print the title FIELD-A and equate the data value for FLD-A to 2 so that it can be positioned at a later time. The value 2 can now be used for printing the data value of FLD-A in the same manner as specifying FLD-A (see option 6-k).

d. [Field-Name (n)]

This option will print only the data value for the subscripted Field-Name. The subscript (n) must be an integer value with a range of 1 up to the maximum number of elements contained within the table or array. By using the subscript, the user can identify any given element within the table or array to be printed.

e. [Field-Name]

This element identifies the Data Base field which is to be printed on the output report. Specifying only Field-Name indicates that only the value of the field is to be printed.

f. [$New-Variable]

This element identifies values which have been generated by the COMPUTE command. Only the computed value will be printed. New-Variables must be preceeded by the dollar ($) sign.

g. [*$New-Variable]

This element identifies the New-Variable-Title which has been specified by the COMPUTE command. Only the title for the New-Variable will be printed.

h. [**$New-Variable]

This element identifies values and title which have been generated by the COMPUTE command. The New-Variable value will be printed immediately following the New-Variable-Title on the same line with one space separating the value and title.

i. [*$New-Variable n]

This element identifies the New-Variable-Title which is to be printed and assigns an integer value, called a synonym, to be equated to the New-Variable value. The value for n must be unique and has a range of 1-20. (See option 6-k for usage of n.)

j. ['Literal']

This element identifies literal information which can be used as display titles and comments. Literals can contain up to 128 characters and must be enclosed with quote marks.

k. [n]

This numeric identifier specifies a data value for a previously defined Field-Name or New-Variable.

This identifier, or synonym, specifies the printing of the data value centered below the title of the element used in option 6-c or 6-i.

7. [Fw.d]

This element can be used following the specification of a New-Variable to indicate the size of the New-Variable. If this option is used it must be input immediately following the New -Variable. The w represents the total width of the field and d represents the number of decimal positions. For example, a value 124. 73 would be represented as F6. 2.

8. $\begin{bmatrix} \text{SPACE n} \\ \text{SP n} \end{bmatrix}$

This element permits the skipping of n blank lines after the specified Field-Name or New-Variable has been printed. The value for n has a range of 1 to 54. The print positioning of the new report line will be reset to the first position of the line.

9. [PPOS n]

This element provides for the placement of the data for a given Field-Name, New-Variable, or literal at a specified print position on the report line. The value of n (1-128) represents the print position of the first data character from the identifying Field-Name, New-Variable or literal. If this option is not specified, the print positioning will be automatically calculated by the DEFINE program.

10. [LOOKUP
TLU
L]

This element is used to print special messages or titles as indicated by a coded value contained in a field. The Field-Name must have been designated as a Table-Lookup field in the Dictionary. Example: A one digit field in the Data Base could contain coded values for cities such as Atlanta, New Orleans, or Houston. When this field is printed, the user may wish to print the full name of each city rather than the coded value. The Table-Lookup capability can be employed to accomplish this objective.

11. [RJ
LJ]

This element is used to specify that a data value is to be printed right or left justified within the print field.

a. [RJ]

This element specifies right justification of the data value within a print field. For example, assume the Dictionary defines the size of a field as seven characters in size and the Data Base actually contains three.

Normal Printing

		A	B	C		

Right Justification

				A	B	C

b. [LJ]

This element specifies left justification of the data value within a print field. For example, assume the Dictionary defines the size of a field as seven characters in size and the Data Base actually contains three.

Normal Printing

		A	B	C		

Left Justification

A	B	C				

12. $\begin{bmatrix}\text{RJT}\\\text{LJT}\end{bmatrix}$

This element is used to specify that a Field-Name title is to be printed right or left justified within the print field. The title will be right or left justified over the data value in the same manner as the examples in option 11.

a. [RJT]

This element specifies that the title will be right justifield over the data value.

b. [LJT]

This element specifies that the title will be left justified over the data value.

13. [PAGENR n]

This element provides the capability to sequentially number each display. If this option is requested, it must be positioned on the display via the SPACE and/or PPOS options. The value for n represents the maximum number of digits which are allowed for the display number. For example, if a value of 2 is specified for n, the maximum number of displays will be 99.

14. $\left\{ \begin{array}{c} ; \\ , \\ . \end{array} \right\}$

This element is used for punctuating the DEFINE command.

a. [;]

The semicolon is used to separate Field-Names, New-Variables, and/or literals within the DEFINE command when the user wishes to specify the beginning of a new line.

b. [,]

The comma is used to separate Field-Names, New-Variables, and/or literals within the DEFINE command. A comma cannot be used between a Field-Name, New-Variable, or literal and its options.

c. [.]

The period is used to terminate the DEFINE command.

3.3.5.2 FORMAT Command

The FORMAT command permits the user to display data from the Hitfile and literals in a user specified format as previously defined in the DEFINE command. The FORMAT command allows a maximum of three (3) format names per FORMAT command. Each format may have up to 54 lines and 128 character positions per line. A format may be printed for every record in the Hitfile or printing of a format may be suppressed until the end of the report for summary reporting. Fields from the data base may also be used to control the printing of a format. Format spacing will determine where printing of a format begins. Page numbering of the format is an optional feature which must be defined in the DEFINE command.

Figure 3-7 illustrates the structure of the FORMAT command. Elements contained within braces { } are required while elements within brackets [ ] are optional. The following breakdown refers to the numbered areas in Figure 3-7.

1. $\left\{\begin{matrix}\text{F}\\\text{FORMAT}\end{matrix}\right\}$

   This element identifies the command as a FORMAT command.

2. $\{\,,\,\}$

   The comma is the means of punctuation following the command designator.

3. $\left[\begin{matrix}\text{SP n}\\\text{SPACE n}\end{matrix}\right]$

   The SPACE option permits spacing of a specific number of lines or skipping to a new page before the printing of the

$$\begin{array}{cccccccc}
\left\{\begin{matrix}\text{F}\\ \text{FORMAT}\end{matrix}\right\} & \left\{\,,\,\right\} & \left[\begin{matrix}\text{SPACE n}\\ \text{SP n}\end{matrix}\right] & \left[\begin{matrix}\text{ALL}\\ \text{NONE}\\ \text{BREAK-FIELD}\end{matrix}\right] & \left[\text{SPLIT}\right] & \left\{\text{NAME}\right\} & \left[\begin{matrix}\text{SPACE n}\\ \text{SP n}\end{matrix}\right] & \left\{\begin{matrix},\\ .\end{matrix}\right\} \\
1 & 2 & 3 & 4 & 5 & 6 & 7 & 8
\end{array}$$

ELEMENTS WITHIN { } ARE REQUIRED.

ELEMENTS WITHIN [ ] ARE OPTIONAL.

Figure 3-7. FORMAT Command Format

Format Pattern. The value for n greater than 53 causes printing to begin on a new page.

4. $\left[\begin{array}{l}\text{ALL}\\ \text{NONE}\\ \text{BREAK-FIELD}\end{array}\right]$

This element is referred to as the break element and determines when a display is to be used. This element may be input before or after the Pattern-Name. If no option is specified, ALL is assumed.

a. [ALL]

The ALL option indicates that the FORMAT name will be printed for every record in the Hitfile.

b. [NONE]

The NONE option indicates the FORMAT name will be printed only one time at the end of the report. This type report is considered a summary report.

c. [BREAK-FIELD]

The BREAK-FIELD option allows the user to specify a Data Base Field-Name to be used as the control for printing the Format-Name. The contents of the Data Base field is tested for value changes. Whenever the value changes, the data set will be displayed under the named format.

5. [SPLIT]

The SPLIT option allows the user to separate the output of a Format-Name. Without this option, if the entire pattern cannot be printed on the current page, the printer will eject to a new page before starting to print the Format-Name. If the SPLIT option is specified, printing of the Format-Name will begin on the current page and will be completed on the following page.

6. {FORMAT-NAME}

This element specifies the unique one-to-twelve character name of a print display which has been previously defined by the DEFINE command. A maximum of three different print displays can be used for the same Query set.

7. $\begin{bmatrix}\text{SP n}\\ \text{SPACE n}\end{bmatrix}$

The SPACE option specifies the amount of spacing to be done after the printing of the Format-Name. The value for n greater than 53 causes printing to begin on a new page.

8. $\left\{\begin{matrix} , \\ . \end{matrix}\right\}$

This element is used for punctuating the FORMAT command.

a. { , }

The comma is used to separate the display name and the options associated with that name from other display names.

b. [.]

The period is used to terminate the FORMAT command. If the period is omitted the command will be terminated by the introduction of the next command.

If the user inputs an improper Format-Name, the system responds with the message:

```
FORMAT-NAME IS NOT A PRINT FCRMAT NAME
FORMAT COMMAND DISCARDED-PLEASE CHECK AND RE-ENTER

READY
```

At this time, the user can enter a Print Definition Format for this particular Query Set.

3.3.5.3 Formatted Print Constraints

1. Element groups consisting of Field-Name and its associated options or New-Variable and its associated options cannot have embedded punctuation but must be separated by other element groups by a comma or semicolon.
2. PPOS n and RJT or LJT options cannot be used within the same element group.
3. Spaces are required between each element of an element group.
4. The maximum number of lines for a display is 54.
5. The total number of characters to be displayed on any line must not exceed 128.

6. The total number of fields and/or literals contained within a display must not exceed 600.

7. A maximum of three different Format-Names can be used to display the output of a Query-Set.

8. The SPLIT option of the FORMAT command can only be used once in association with any Format-Name.

9. More than one line of input is permitted for the DEFINE and FORMAT commands with the continuation lines beginning in any position.

10. The same Format-Name must never be used more than once in the FORMAT command.

3.3.5.4 Formatted Print Example

The following example illustrates the capability of printing an output report under two separate formats. The print formats are defined as PATTERN1 and PATTERN2. PATTERN1 represents a detailed report while PATTERN2 represents a summary report. PATTERN2 will only be printed when the value of FLD-B changes and the New-Variables are computed as shown in the Query Set.

3.3.6 Saved QUERY Commands

3.3.6.1 SAVE Command

The SAVE command permits the user to specify a Query Set to be saved, under a unique name, for execution at a later time. Initially the user must type in his Query Set. If this Query Set is to be

PATTERN1 DEFINITION

```
DEFINE, PATTERN1,SPACE 2 'SPECIAL DEMONSTRATION REPORT - FORMAT COMMAND'

PPOS 45, 'FUNCTION' PPOS 91;;

'(THIS IS A SAMPLE OF THE OPTIONS AVAILABLE IN FORMATTED PRINT)' PPOS 42

SPACE 2;'DETAIL REPORT' PPOS 63; '(3RD)' PPOS 121;*FLD-A1;'   ';*FLD-B 2, *FLD-C 3 PPOS 120;

1, 2, 3;;;;'E-FIELD' PPOS 66 ;; FLD-E PPOS 66, SPACE 9 *NAR, NAR PPOS 11,

SPACE 5 *FLD-F PPOS 66 ;; FLD-F PPOS 66 ;;; PPOS 64 *NAR;NAR;SPACE 3 *TLU-A 4,

TABLE FUNCTION';

4, TLU-A TLU SP 1; *$DIFFERENCE ;; $DIFFERENCE ;; **FLD-A, **FLD-B PPOS 30,

**$SAMENR PPOS 60;

'PAGE' PPOS 114, PAGENR 3.
```

PATTERN2 DEFINITION

```
DEFINE, PATTERN2, SPACE 4 PPOS 49 'SUMMARY (DEMONSTRATION) REPORT';;
PPOS 14 'THIS REPORT PATTERN WILL BE PRINTED IN CONJUNCTION WITH THE SPECIAL DEM',
'ONSTRATION REPORT';;;*FLD-A 1, *FLD-B 2, *FLD-C 3, *FLD-D 4, *FLD-E 5, *FLD-F 6;
1, 2, 3, 4, 5, 6;;;;
*TLU-A, *FLU-B PPOS 20, 'TAB-F' PPOS 29; TLU-A, TLU-B TLU PPOS 20, TLU-A TLU PPOS 29
SPACE 4;
*$TOTAL-A, *$TOTAL-B PPOS 20, *$TOTAL-C PPOS 40, *$TOTAL-D PPOS 60,
*$TOTAL-E PPOS 80, *$TOTAL-F, *$TOTAL-G PPOS 120;$TOTAL-A, $TOTAL-B PPOS 20, $TOTAL-C
PPOS 40, $TOTAL-D PPOS 60, $TOTAL-E PPOS 80, $TOTAL-F, $TOTAL-G PPOS 120 SPACE 7;
*NAR;NAR SPACE 10,
'THIS SUMMARY REPORT IS PRODUCED FOR THE UNIQUE OCCURRENCES OF FIELD-B VALUES-',
'IT WILL BE INTERLACED WITH THE DETAIL REPORT', SPACE 10 'PAGE' PPOS 120,
PAGENR 3.
```

QUERY SET USING PATTERN1 AND PATTERN2

```
Q, FLD-A PRESENT.

S, FLD-B ASCENDING.

C, FLD-B, 1, $TOTAL-A = COUNT FLD-A.

C, FLD-B, 1, $TOTAL-B = COUNT FLD-B.

C, FLD-B, 1, $TOTAL-C = COUNT FLD-C.

C, FLD-B, 1, $TOTAL-D = SUM FLD-D.

C, FLD-B, 1, $TOTAL-E = SUM FLD-E.

C, FLD-B, 1, $TOTAL-F = SUM FLD-F.

C, FLD-B, 4, $TOTAL-G = FLD-E + FLD-F.

C, ALL, 1, $DIFFERENCE = FLD-D - FLD-E.

C, ALL, 1, $SAMENR = FLD-D + FLD-E

FORMAT, SPACE 55 PATTERN1, SPACE 55 FLD-B PATTERN2.

RUN
```

```
                                            SPECIAL DEMONSTRATION REPORT - FORMAT COM

                                          (THIS IS A SAMPLE OF THE OPTIONS AVAILABLE I



                                                              DETAIL REPORT

FIELD-A  FIELD-B
1023 .4    321 .6



                                                                  E-FIELD

                                                                  10021 .6









NARRATIVE *** ***** *** THIS EXAMPLE ILLUSTRATES HOW A DEFINED FORMAT CAN BE USED TO STR





                                                                  FIELD-F

                                                                  23010, 9


                                                                NARRATIVE
*SINGLE HIT RECORD EXTRACTED BY THE QUERY COMMAND - UP TO 600 LITERALS AND FIELD NAME



TABLE-A  TABLE FUNCTION
    2      GAMMA

DIFFERENCE

701 .8

FIELD-A  1023 .4              FIELD-B 321 .6                SAMENR 1345 .0
```

FOLDOUT FRAME

```
DEMONSTRATION REPORT - FORMAT COMMAND FUNCTION

SAMPLE OF THE OPTIONS AVAILABLE IN FORMATTED PRINT)


          DETAIL REPORT
                                                                    (3RD)
                                                                   FIELD-C
                                                                   60061.1



             E-FIELD

             10021.6





DEFINED FORMAT CAN BE USED TO STRUCTURE AN ENTIRE PAGE FOR ONE *************



              FIELD-F

              23010.9

            NARRATIVE
UP TO 600 LITERALS AND FIELD NAMES CAN BE PRINTED USING FORMATTED PRINT ****







        SAMENR 1345.0
                                                             PAGE    1
```

FOLDOUT FRAME 2

```




                                                SUMMARY (DEMONSTRATION) REPORT

             THIS REPORT PATTERN WILL BE PRINTED IN CONJUNCTION WITH THE SPECIAL DEMONST



FIELD-A  FIELD-B   FIELD-C      FIELD-D       FIELD-E       FIELD-F
7216.5    6004.2   3969.0       6105.5        3322.6        901.8



TABLE-A            TABLE-B  TAB F
  1                 ALPHA   BETA




TOTAL-A            TOTAL-B             TOTAL-C              TOTAL-D            TOTAL-E T
 98890.4           63021.8             36540.0              72312.1            89995.4 6







                                                          NARRATIVE
**** EACH TIME THE VALUE OF FIELD-B CHANGES, THE TOTAL VALUES ARE COMPUTED AND THIS SUMM










THIS SUMMARY REPORT IS PRODUCED FOR THE UNIQUE OCCURRENCES OF FIELD-B VALUES - IT WILL
```

FOLDOUT FRAME 3

```
56 57 58 59 60 61 62 63 64 65 66 67 68 69 70 71 72 73 74 75 76 77 78 79 80 81 82 83 84 85 86 87 88 89 90 91 92 93 94 95 96 97 98 99 100 101 102 103 104 105 106 107 108 109 110 111 112 113 114 115 116 117 118 119 120 121 122 123 124 125 126 127 128
```

```
 (DEMONSTRATION) REPORT

JUNCTION WITH THE SPECIAL DEMONSTRATION REPORT



     FIELD-F
     901.8







     TOTAL-D            TOTAL-E TOTAL-F                          TOTAL-G
     72312.1            89995.4 62789.3                          68.4751







   NARRATIVE
VALUES ARE COMPUTED AND THIS SUMMARY REPORT IS GENERATED ****








NCES OF FIELD-B VALUES - IT WILL BE INTERLACED WITH THE DETAIL REPORT








                                                                  PAGE  2
```

saved, enter the SAVE command followed by a unique one to twelve character Query-Set-Name chosen by the user. If the Query Set contains no edit errors, it will be saved for later use, or until the user deletes it. The Query Set may or may not be executed at the time it is saved.

After receiving the system response,

```
READY
```

the user can enter the SAVE command under the format given below.

The format of the SAVE command is:

```
SAVE, Query-Set-Name.
```

The Query-Set-Name represents a unique name of one to twelve characters specified by the user. An example to illustrate the usage of the SAVE command is shown.

The Saved Query STDAGE will be stored in a cataloged file. The SAVE command must be entered within the Query Set after the PRINT or FORMAT command. Example:

```
Δ Q, AGE P.
  READY
Δ P, NAME, AGE, SEX.
  READY
Δ SAVE,STDAGE.
  READY
```

3.3.6.2 Minor Commands Under SAVE

Associated with the SAVE command is a set of minor commands which the user may utilize to list all Queries that have been saved and to execute, delete, and/or display any Saved Query. The commands used for these functions are:

```
LIST
DO
DELETE
DISPLAY
```

3.3.6.3 LIST Command

The LIST command allows a user to obtain a list of all Saved Query-Set-Names for a particular application. This is accomplished simply by entering the command LIST immediately following the system statement READY. The example below illustrates the use of the LIST command. Example:

LIST Command for Saved Queries

```
  READY
Δ LIST
  TRYIT     STDAGE     STANDARD
  READY
```

In the above example, the two Query Sets TRYIT and STDAGE have been previously saved by a SAVE command. After the

Query-Set-Names have been printed on the terminal, the system will return the statement:

READY

3.3.6.4 DO Command

The DO command is used to execute a Saved Query. The format of the DO command is:

DO, Query-Set-Name.

where Query-Set-Name represents the unique name of one to twelve characters. In this case, the Query-Set-Name specifies the Query Set to be executed. This command will retrieve the saved query from the control table and enter it into the control stream so that execution will follow. The DO command is entered immediately following the system request:

READY

When the Saved Query goes into execution, the system will reply:

QUERY NOW PROCESSING

In order to execute the Saved Query STDAGE, the user should make the following entry:

```
  READY
Δ DO,STDAGE.
  QUERY NOW PROCESSING
```

3.3.6.5 DELETE Command

The DELETE command is used to delete a Saved Query. The format of the DELETE command is:

```
DELETE,Query-Set-Name.

         OR

DEL,Query-Set-Name.
```

where Query-Set-Name is the unique one to twelve character name of the Query Set to be deleted.

To delete the STDAGE Query Set, enter after the READY message:

```
DELETE, STDAGE.
```

3.3.6.6 DISPLAY Command

The DISPLAY command is used to generate a line-by-line listing (no execution) of the commands contained within a Saved Query Set. The format of the DISPLAY command is:

```
DISPLAY,Query-Set-Name.

         OR

DIS,Query-Set-Name.
```

where Query-Set-Name is the unique name of the Query Set to be displayed. An example of the DISPLAY command is:

```
  READY
Δ DISPLAY, STDAGE.
  Q, AGE P.
  P, NAME, AGE, SEX.
  READY
```

3.3.6.7 Saved Query Examples

1. This example illustrates a Saved Query and its usage.

   Example 1:

```
Δ  @RUN,SJ1205,AAAAAAAAAA,1000-PEOPLE,3
   DATE: 093174         TIME: 122120
Δ  @XQT MIR*ADS.MIRADS
   ENTER QUALIFIER*FILENAME
Δ  1000-PEOPLE*PEOPLE
   READY
Δ  Q, SEX = 'MALE'.
   READY
Δ  S, AGE D, NAME A.
   READY
Δ  C,NONE,0,$TOT-MR = COUNT SEX.
   READY
Δ  C,NONE,0,$TOT-AGE = SUM AGE.
   READY
Δ  C,NONE, 2,$AVG-AGE = $TOT-AGE / $TOT-MR.
   READY
Δ  P, SP 1 NAME,AGE G 1, $AVG-AGE.
   READY
Δ  SAVE,AVGAGE.
   READY
```

Example: (Continued)

```
Δ RUN
  QUERY NOW PROCESSING
    FILE CONTAINS 35 RECORDS
    QUERY SELECTED 6  RECORDS
    ENTER OUTPUT REPORT SITE ID
Δ     (Blank Line)
        Q, SEX = 'MALE'.
        S, AGE D, NAME A.
        C, NONE, 0, $TOT-MR = COUNT SEX.
        C, NONE, 0, $TOT-AGE = SUM AGE.
        C, NONE, 2, $AVG-AGE = $TOT-AGE / $TOT-MR.
        P,SP 1 NAME, AGE G 1, $AVG-AGE.


            NAME                  AGE           AVG-AGE

  SMITH         DOUG D             52

  ADAMSON       JOHN R             43

  JOHNSON       FRANK S            29

  JONES         BART H             27

  MORGAN        TOM J

  ADAMS         JOHN D             26                34.00
  READY
Δ LIST
    AVGAGE      STANDARD
  READY
```

Example 1: (Continued)

```
△  DO, AVGAGE.
   QUERY NOW PROCESSING
     FILE CONTAINS 35 RECORDS
     QUERY SELECTED 6 RECORDS
     ENTER OUTPUT REPORT SITE ID
△      (Blank Line)
         Q, SEX = 'MALE'.
         S, AGE D, NAME A.
         C, NONE, 0, $TOT-MR = COUNT SEX.
         C, NONE, 0, $TOT-AGE = SUM AGE.
         C, NONE, 2, $AVG-AGE = $TOT-AGE / $TOT-MR.
         P, SP 1 NAME, AGE G 1, $AVG-AGE.


             NAME                AGE          AVG-AGE

   SMITH          DOUG B          52

   ADAMSON        JOHN R          43

   JOHNSON        FRANK S         29

   JONES          BART H          27

   MORGAN         TOM J

   ADAMS          JOHN D          26             34.00
   READY
△  STOP
   MIRADS EXECUTION NORMALLY TERMINATED
△  @FIN
```

In example 1, a Query Set is generated and saved under the name of AVGAGE. This query searches the user's Data Base for all male employees to calculate the average age. After the execution of the Query Set, the user requested a list of all Saved Queries and the response shows that AVGAGE is the only Saved Query in this user's Data Base. After listing the name of the Saved Query, the user executes AVGAGE with the DO command and then terminates MIRADS.

2. Example 2 illustrates the usage of the DISPLAY and DELETE commands for a Saved Query. The Saved Query is displayed so that the user can determine if it should be kept for later use. The Saved Query was no longer needed so the user elected to delete it from the system. After the DELETE command, the Query AVGAGE is no longer available for use.

   Example 2:

```
Δ @RUN, SJ1205, AAAAAAAAAA, 1000-PEOPLE, 3
  DATE:  100274     TIME:  122620
Δ @XQT MIR*ADS. MIRADS
  ENTER QUALIFIER*FILENAME
Δ 1000-PEOPLE*PEOPLE
  READY
Δ DISPLAY,  AVGAGE.
  Q, SEX = 'MALE'.
  S, AGE D, NAME A.
  C, NONE, 0, $TOT-MR = COUNT SEX.
  C, NONE, 0, $TOT-AGE = SUM AGE.
  C, NONE, 2,  $AVG-AGE = $TOT-AGE/$TOT-MR.
  P,  SP  1 NAME, AGE G 1, $AVG-AGE.
  READY
Δ DELETE, AVGAGE.
  READY
```

3.3.6.8 SAVEC Command

The SAVEC command operates in a similar manner to the SAVE command. The differences between the two commands is that SAVEC can save a single command or several commands which compose a partial Query-Set. The SAVE command retains an entire Query-Set for later execution while SAVEC retains command for insertion into a Query-Set at a later time.

After receiving the system message,

READY

the user can enter the SAVEC command under the format

SAVEC, Save-Command-Name.

where Save-Command-Name represents a unique one-to-twelve character name which identifies the saved command or commands. The allowable character set for the Save-Command-Name is A through Z, 0 to 9, and hyphen (-).

3.3.6.9 ADD Command

The ADD command enables the user to insert saved commands into a Query-Set. The user enters his commands in the normal manner of generating a Query-Set; however, at the point when the saved command is required, the user inserts the command(s) by entering the ADD command.

ADD, Save-Command-Name.

Save-Command-Name represents the one-to-twelve character name which uniquely identifies the command(s) to be inserted. Example of SAVEC and ADD.

```
Δ @RUN
  DATE: 100274      TIME: 132620
Δ @XQT MIR*ADS. MIRADS
  ENTER QUALIFIER*FILENAME
Δ 1000-PEOPLE*PEOPLE
  READY
Δ C, NONE, 0, $TOT-SEX = COUNT SEX.
  READY
Δ C, NONE, 0, $TOT-AGE = SUM AGE.
  READY
Δ C, NONE, 2, $AVG-AGE = $TOT-AGE/$TOT-SEX.
  READY
Δ SAVEC, COMPAGE.
  READY
Δ @FIN
```

This example saves three compute commands under the name of COMPAGE. These commands can be inserted into any Query-Set at a later time.

```
  READY
Δ Q,SEX = MALE.
  READY
Δ ADD, COMPAGE.
  READY
Δ P, SEX, $TOT-SEX, $TOT-AGE, $AVG-AGE.
```

With this QUERY command the computations will be performed for all males. If the same information is required for females, the COMPUTE commands could be inserted following a QUERY command.

```
Q,SEX = FEMALE.
```

3.3.7 EDIT Command

The EDIT command permits the user to modify the contents of a Saved-Query-Set or Saved-Commands without the necessity of creating a new Saved-Query-Set. Edit functions can perform such tasks as deleting and inserting lines, changing the contents of a specified line, and renaming a Saved-Query-Set with a user specified name.

After receiving the system response,

```
READY
```

the user can enter the EDIT command under the format

```
ED, Name.
   or
EDIT, Name.
```

Name represents the one-to-twelve character name of a previously saved Query-Set, Print-Format, or Saved-Commands. Punctuation (comma and period) is optional on the command. A variable number of spaces can be used in place of the comma and the period can be omitted.

After transmitting the EDIT command, the system will respond with the message,

ENTER EDIT COMMAND

At this time the user should specify the edit functions which will satisfy the desired modifications. The edit functions are divided into three groups: (1) Positioning, (2) Editing, and (3) Exit.

3.3.7.1 Positioning EDIT Commands

The edit commands for specifying a unique line within a saved element are Top, Next Line, Locate, and Line n.

1. Top Position

   If the user wishes to position the editor at the beginning of the save element, he should enter and transmit the letter

   T

   The editor will be positioned above the first command line of the saved element. From this point, the user must transmit the command N to position the editor at the first command line.

2. Next Line

   To position the editor at a line other than the one presently being accessed, the user enters the command

N
or
N n
or
N - n
or
n

a. N

This option will position the editor at the line immediately below the one where the editor was previously positioned.

b. N n

This option specifies that the editor will advance n lines below the presently positioned line. For example, if the editor is positioned at line number 3 of the save element and a command N 2 is entered, the editor will be positioned at line number 5.

c. N - n

This option specifies that the editor will be positioned n lines above the presently positioned line. For example, if the editor is positioned at line number 4 of the save element and a command N - 1 is entered, the editor will be positioned at line number 3.

d. n

This option specifies that the editor will be positioned at line number n. For example, if a value of n is specified as 2, the editor will be positioned at line number 2.

3. Locate

If the user wishes to position the editor on a line which contains a specified value, he enters the command

```
L Value
```

where 'Value' can be any alphanumeric string of characters from 1 to 82 in length. The editor will search each line in the save element for a matching 'Value' beginning with the line at which the editor is presently located. Lines above the editor position will not be searched. The editor will be positioned at the first line which contains a value equal to that specified on the Locate command.

3.3.7.2 Editing Commands

The edit commands for modifying the contents of a save element are Change, Insert, and Delete.

1. CHANGE Command

The CHANGE command enables the user to modify the content of any specified line or group of lines within a save element. The format of the CHANGE command is:

```
C  /Old-Value/New-Value/
              or
C  /Old-Value/New-Value/G
              or
C  /Old-Value/New-Value/ALL
              or
C  /Old-Value/New-Value/n
```

a. C /Old-Value/New-Value/

This option will replace the Old-Value with the New-Value for the first occurrence of the Old-Value on the line indicated.

b. C /Old-Value/New-Value/G

This option will replace the Old-Value with the New-Value for each occurrence of the Old-Value on the specified line.

c. C /Old-Value/New-Value/ALL

This option will replace the Old-Value with the New-Value for each occurrence of the Old-Value on the specified line and every line to the end of the save element.

d. C /Old-Value/New-Value/n

This option will replace the Old-Value with the New-Value for the first occurrence of the Old-Value on the next n lines.

2. INSERT Command

The INSERT command is used to insert a new line into a saved element at a user specified position. The user selects the insert position by specifying one of the positioning commands of the editor. The new line will be inserted immediately after the line at which the editor is positioned. The user specifies the insert line by entering the command

```
I Insert-Line
```

The Insert-Line can contain from 1 to 82 characters and must be separated from the I by one space.

3. DELETE Command

The DELETE command enables the user to delete one line from a saved element. The editor must be positioned by one of the positioning commands to the line to be deleted. The user enters

```
D
```

to delete that line. The editor remains positioned at the deleted line and must be positioned with one of the positioning commands to go to a new line.

4. PRINT Command

The PRINT command enables a user to list lines contained within a saved element. This option is especially useful when modifications have been made to a saved element. The user can print the saved element to ensure that the changes which he has specified are in agreement with what he intended. The user must position the editor by one of the positioning commands to the line with which he wished to initiate the listing. The command is entered

```
P
or
P n
```

where n represents the number of lines to be printed. If the n option is omitted, only one line will be printed.

3.3.7.3 EDIT EXIT Commands

The EXIT commands enable the user to terminate the editor and specifies whether or not the modifications should be implemented.

1. EXIT Command

The EXIT command terminates the editor and rewrites the saved element with all changes which were specified. The saved element will be retained under its original saved name. The user enters the command as one word.

EXIT

2. NAME Command

The NAME command terminates the editor and rewrites the saved element with all changes which were specified. The saved element will be retained under a new name which must be specified by the user. The user enters the command

NAME, Save-Name

where Save-Name represents a unique one-to-twelve character name which will now be used to access this new saved element. The original save element will be retained under its old name.

3. OMIT Command

The OMIT command terminates the editor but does not make any modifications to the original saved element. The command is entered as

OMIT

4. EXEC 8 Control Statement

The editor will be terminated if the user transmits any EXEC 8 Control Statement. Terminating the editor in this manner will delete any changes which may have been specified by the user. EXEC 8 Control Statements on the DCT-500 or Uniscope 100 terminal are those which contain the @ in column one while on the Uniscope 300 terminal the # is in column one.

3.3.7.4 Edit Example

```
  READY
Δ ED QS1
  ENTER EDIT COMMAND
Δ P 5
  Q,FLD-A P
  OR FLD-B = ABC
  OR FLD-C = 123.
  P,FLD-A,
  FLD-B,
  005:
Δ N-5
  000:
Δ N
  Q ,FLD-A P
  001:
Δ C  /FLD-A/FLD-B/
  Q,FLD-B P
  001:
Δ T
  000:
Δ P 7
  Q,FLD-B P
  OR FLD-B = ABC
  OR FLD-C = 123.
  F, FLD-A,
  FLD-B,
  FLD-C.
  END OF FILE  .
  000:
Δ NAME TESTB
  READY
```

This example illustrates the capability of changing the contents of a previously Saved-Query-Set named QS1 while retaining the original QS1 and generating a new Query-Set named TESTB.

3.3.8 UPDATE Command

The UPDATE command allows the user to make modifications to his Data Base. The available functions to the command are change, delete, insert, and replace. Before specifying an update function of change, delete, or replace, a QUERY command must be specified to select the Data Base records which are to be updated. When the insert function is not preceded by a QUERY command, a new record will be inserted at the end of the Data Base.

Figure 3-8 shows the UPDATE command format. Elements shown within braces { } are required while elements shown within brackets [ ] are optional. The following breakdown refers to the numbered areas in Figure 3-8.

1. $\left\{\begin{matrix}\text{U}\\ \text{UPDATE}\end{matrix}\right\}$

   This element identifies the command as an UPDATE command.

2. [,]

   The command is an optional means of punctuation. If the comma is omitted, a variable number of spaces can be used.

$$\begin{array}{cccccccc}
\left\{\begin{matrix}\text{U}\\ \text{UPDATE}\end{matrix}\right\} & [,] & \left\{\begin{matrix}\text{C}\\ \text{D}\\ \text{I}\\ \text{R}\end{matrix}\right\} & [,] & \left\{\begin{matrix}\text{FIELD–NAME}\\ \\ \text{RECORD–TYPE}\end{matrix}\right\} & \left\{\begin{matrix}=\\ \text{EQ}\\ \text{EQUAL}\\ \text{EQUALS}\end{matrix}\right\} & \left\{\text{'NEW–VALUE'}\right\} & \left[\begin{matrix}'\\ .\end{matrix}\right] \\
\\
1 & 2 & 3 & 4 & 5 & 6 & 7 & 8
\end{array}$$

ELEMENTS WITHIN { } ARE REQUIRED
ELEMENTS WITHIN [ ] ARE OPTIONAL

Figure 3-8. UPDATE Command Format

3. $\left\{\begin{matrix}\mathrm{C}\\\mathrm{D}\\\mathrm{I}\\\mathrm{R}\end{matrix}\right\}$

This element specifies the update function which is to be performed.

a. [C]

This option specifies a change function. Fields contained within records selected by a QUERY command will receive new values as specified by the UPDATE command.

b. [D]

This option specifies a delete function. The delete function can be used to space out a specified field within records selected by a QUERY command or to delete records of a given type under a specified level.

c. [I]

This option specifies an insert function. A new record will be inserted into the Data Base immediately following the first record selected by a QUERY command. If the insert function is performed without a previous QUERY the new record will be inserted at the end of the Data Base.

d. [R]

This option specifies a replace function. The replace function is used to replace a Data Base record of a specified type.

4. $\left\{\begin{array}{l}\text{Field-Name}\\ \text{Record-Type}\end{array}\right\}$

This element indicates which fields within the Data Base record or which record type will be updated.

5. $\left\{\begin{array}{l}=\\ \text{EQ}\\ \text{EQUAL}\\ \text{EQUALS}\end{array}\right\}$

The Field-Name must be separated from the New-Value by an equal sign (=) or any of the equivalents shown.

6. $\left\{\text{'New-Value'}\right\}$

The New-Value element specifies the update value to be placed in the Data Base field specified by the Field-Name element.

7. $\left[\begin{array}{c},\\ .\end{array}\right]$

This element is used as punctuation of the UPDATE command.

a. [,]

The comma is used to separate update sets (Field-Name = 'New-Value') when more than one field is to be updated by a single command.

b. [.]

The period terminates the UPDATE command. Transmission of the next command will automatically terminate the UPDATE command if the period is omitted.

3.3.8.1 Rules for Using the UPDATE Command

The following rules must be adhered to in using the UPDATE command. Any deviation from these rules will result in an error and termination of the inquiry.

1. More than 1 line of input is permitted for the UPDATE command. Continuation lines can begin in any position.
2. Field-Names must be entered exactly as they are described in the Dictionary.
3. A maximum of 200 Field-Names can be included in one UPDATE command.
4. Change, delete, and replace functions require a previous QUERY command to select records which will be updated.
5. Fields which overlap indexed fields cannot be updated.
6. Fields from different record levels cannot be specified on the same UPDATE command.

7. The update New-Value cannot exceed 48 characters and must be enclosed within quote marks.

3.3.8.2 Update Change Function

The Change Function enables the user to perform on-line updates to fields contained within records selected by a QUERY command.

ORIGINAL DATA BASE

EMP-NO	GRD	PAY
00100	09	11000
00101	10	12800
00102	09	11500
00103	11	13000

If a user wishes to make a change to the Data Base shown above, he must first select the records to be changed.

```
Q,GRD = 9.
```

This QUERY command will select records for employees '00100' and '00102'.

```
U,C,GRD = '10', PAY = '12500'.
```

This UPDATE command will change the Data Base content for fields GRD and PAY for the two employees selected by the QUERY command.

UPDATED DATA BASE

EMP-NO	GRD	PAY
00100	10	12500
00101	10	12800
00102	10	12500
00130	11	13000

3.3.8.3 Update Delete Function

The delete function enables the user to perform on-line deletions to his Data Base. Before a delete function can be performed, the user must input a QUERY command to select the records which are to be updated. Once the records have been selected, the user can select specific fields within the record to be spaced out or he may specify an entire record to be deleted from the Data Base.

ORIGINAL DATA BASE

EMP-NO	GRD	PAY
00100	10	12500
00101	10	12800
00102	10	12500
00103	11	13000

```
Q, EMP-NO = 101.
```

This QUERY command will select the record for employee '00101'.

If the UPDATE command

```
U,D, GRD, PAY.
```

is entered, the fields of GRD and PAY will be spaced out.

UPDATED DATA BASE

EMP-NO	GRD	PAY
00100	10	12500
00101		
00102	10	12500
00103	11	13000

To delete a specified record type, the user must enter a QUERY command to select the records which are to be affected.

ORIGINAL DATA BASE

RECORD TYPE	EMP-NO	GRD	PAY
101	00100	10	12500
101	00101	10	12500
101	00102	10	12500
101	00103	11	13000

```
Q,EMP-NO = 101.
U,D,101.
```

The QUERY command will select the record for employee '00101' which is specified as a 101 record type. The UPDATE command will delete all records of type 101 which selected by the QUERY. In this case, only one record is deleted from the Data Base.

UPDATED DATA BASE

EMP-NO	GRD	PAY
00100	10	12500
00102	10	12500
00103	11	13000

3.3.8.4 Update Insert Function

The insert function enables the user to make on-line insertions into the Data Base. This function may be performed with or without a previous QUERY command. If a QUERY command is specified, the new record will be inserted following the first Hit Record selected by the QUERY. If a QUERY command is not specified, the new record will be inserted at the end of the Data Base.

ORIGINAL DATA BASE

EMP-NO	GRD	PAY
00100	10	12500
00102	10	12500
00103	11	13000

```
Q, EMP-NO = 100.
U, I, EMP-NO = '00101', GRD = '12', PAY = '15000'.
```

This example will insert a new record following the record for employee '00100'.

UPDATED DATA BASE

EMP-NO	GRD	PAY
00100	10	12500
00101	12	15000
00102	10	12500
00103	11	13000

If you refer to the original Data Base and perform the UPDATE command without the QUERY command, the updated Data Base would be:

EMP-NO	GRD	PAY
00100	10	12500
00102	10	12500
00103	11	13000
00101	12	15000

3.3.8.5 Update Replace Function

The replace function enables the user to perform on-line replacements of records within his Data Base. The user must enter a QUERY command to select the records which are to be replaced. After selecting the records, the UPDATE command specifies the New-Values that are to be placed into the Data Base fields. This function differs from the change function in that any field which is not specified on the UPDATE command will be spaced out.

ORIGINAL DATA BASE

EMP-NO	GRD	PAY	NAME
00100	09	11000	SMITH
00101	10	12000	BROWN
00102	10	12500	MILLER
00103	11	13000	JONES

```
Q, EMP-NO = 101.
U, R, EMP-NO = '00104', PAY = '12500', NAME = 'WHITE'.
```

The QUERY command selects the record for employee '00101' to be replaced by the UPDATE command. Referring to the updated Data Base, you will note that the field GRD was spaced out because a New-Value was not specified on the UPDATE command.

UPDATED DATA BASE

EMP-NO	GRD	PAY	NAME
00100	09	11000	SMITH
00104		12500	WHITE
00102	10	12500	MILLER
00103	11	13000	JONES

3.3.8.6 UPDATE Data Validation

The execution of an UPDATE command requires the specification of a PRINT command. This PRINT command should specify the fields which are being updated. Following execution, the output report will reflect the updated fields. The user is given the opportunity to examine the new data for accuracy. Following the generated report, the system responds with the message:

IS THIS UPDATE PERMANENT (YES OR NO)

At this time, the user must reply 'YES' if the data is correct and he wishes to permanently update the Data Base. If a user responds 'NO', the Data Base will be reinitialized to its original contents.

The following example illustrates the method of permanently updating the Data Base.

```
  READY
Δ Q, TEAM = 25.
  READY
Δ U,C, TEAM = '30'.
  READY
Δ P, TEAM.
  READY
Δ RUN
  QUERY NOW PROCESSING
  FILE CONTAINS 239 RECORDS
  QUERY SELECTED 1 RECORDS
  ENTER OUTPUT REPORT SITE ID
Δ (Blank Line)
  Q, TEAM = 25.
  U, C, TEAM = '30'.
  P, TEAM.

  TEAM
   30
  IS THIS UPDATE PERMANENT (YES OR NO)
Δ YES
  READY
```

3.4 MIRADS MINOR COMMANDS

There are five minor commands associated with MIRADS. RUN specifies that pro‿essing of the Query Set is to begin and directs the user requested output to either a high or low-volume terminal. The TOP command is used to delete a Query Set from going into execution. The NEW command is used to access a new Data Base. The KEY command generates a listing of all indexed fields contained in the Data Base. The CHECK command produces a listing of the previous commands which have been entered for a Query-Set.

3.4.1 RUN Command

The RUN command signifies the end of a Query Set and that processing is to begin. The data which satisfies the Query Set will be displayed on the terminal specified by the user.

The RUN command has only one field, the command field, and does not need to be terminated with a period.

```
RUN
```

Upon receiving the system message READY the user can transmit the RUN command to the computer. After the user has transmitted the RUN command, the system should respond with the message

```
QUERY NOW PROCESSING
```

If the Query Set requires a sequential search, the message following the RUN command will be

```
WARNING - QUERY FORCES SEQUENTIAL SEARCH
DO YOU WISH TO CONTINUE WITH THIS FORCED
  SEQUENTIAL SEARCH (YES OR NO)
```

If one of the above messages is not received, the user should refer to Section 3.5 for MIRADS Diagnostic Messages. If the user wishes to execute the sequential search, he must enter and transmit YES, otherwise, he must transmit NO.

After the requested records have been retrieved from the Data Base and the specified report has been formatted, the system will generate a copy of the user's Query Set, specify the file size of the user's Data Base, and specify the number of records which satisfied the criterion of the QUERY command.

```
FILE CONTAINS n RECORDS
QUERY SELECTED n RECORDS
ENTER OUTPUT REPORT SITE ID
```

At this time, the user should enter the SITE ID which identifies the device which will be used to receive the output report. This allows the user to select a high speed printer for high volumes of printout or a low speed printer for low volumes of printout, or a cataloged file which will retain the data for later use.

The SITE ID will be specified to individual users on a need-to-know basis by a MIRADS representative.

Example: Query Set using the RUN Command

```
Δ Q,SALARY  GT  1200.
  READY
Δ S,SALARY  D.
  READY
Δ P,NAME,SALARY.
  READY
Δ RUN
  QUERY NOW PROCESSING
  FILE CONTAINS 500 RECORDS
  QUERY SELECTED 3 RECORDS
  ENTER OUTPUT REPORT SITE ID
Δ (Blank Line)
  Q,  SALARY GT  1200.
  S,  SALARY D.
  P,  NAME, SALARY.

NAME                           SALARY
    DOE       JOHN D            16000
    SMITH     FRANK F           15000
    JONES     TOM J            125000
```

The user has the capability to specify the output print device, restrict the number of records to be printed, and specify a box header page for his printed output. The user must enter a response under the format shown in Figure 3-9.

The following numbers correspond with the element numbers in Figure 3-9. Elements within braces { } are required while elements within brackets [ ] are optional.

$$\left\{\begin{array}{l}\text{SITE-ID}\\ \text{DRUM}\\ \text{NONE}\\ \text{PRINT = n}\\ \text{BLANK LINE}\end{array}\right\} \qquad \left[\begin{array}{l}\text{PRINT = n}\\ \text{MESSAGE}\end{array}\right] \qquad \left[\text{MESSAGE}\right]$$

1 2 3

ELEMENTS WITHIN { } ARE REQUIRED
ELEMENTS WITHIN [ ] ARE OPTIONAL

Figure 3-9. Site-ID Command Format

1. $$\left\{\begin{array}{l}\text{Site-ID}\\ \text{DRUM}\\ \text{NONE}\\ \text{PRINT = n}\\ \text{Blank line}\end{array}\right\}$$

This element specifies the hardware address of the output device. One of the five options must be used.

a. [Site-ID]

Specifies the hardware address of the output device. This hardware identifier varies from one computer center to another.

b. [DRUM]

Allows the user to direct the Query Set output to a user cataloged file on FASTRAND or to a magnetic tape. MIRADS automatically generates a file which is temporarily used by MIRADS to contain the Query Set output. This file is identified by MIRADShhmmss where hhmmss is the hour, minute, and second that the file was generated by MIRADS.

c. [NONE]

This option is used when no output from the Query Set is required. The output report will be deleted.

d. [PRINT = n]

Specifies the restriction of printing only the first n records of the Hit list. A value of 1-999999 can be specified for n. The output report will be printed on the initiating device.

e. [Blank line]

Allows the user to direct his Query Set output report to the initiating terminal by transmitting a blank line as the Site-ID. This format, when used while executing from the DCT-500 or Uniscope terminal, should be used only when the volume of output is not too excessive.

2. $\begin{bmatrix} \text{PRINT} = \text{n} \\ \text{Message} \end{bmatrix}$

This element allows to limit the number of records from the Hit list to be printed on the device specified in element one or to insert a message into the report.

a. [PRINT = n]

Limits the amount of printout to n number of records This option must not be specified in element one if used in element two. A value of 1 to 999999 may be specified for n.

b. [Message]

Specifies a 1 to 24 character box letter message that is used by computer operations personnel to properly direct the Query Set output to the requestor.

3. [Message]

This element specifies a 1 to 24 character box letter message that is used by computer operations personnel to properly direct The Query Set to the requestor. This element cannot be used if 'Message' was specified in element two.

3.4.1.1 Examples of Output Device Selection

```
DCTXX , PRINT = 10 , MAIL TO BIN 211   HVILLE.
```

This example will cause the first ten Hit records selected by the QUERY command to be printed on the output device addressed as 'DCTXX'. The message 'MAIL TO BIN 211 HVILLE' will be printed in Box Letters on the first page of the output report.

```
Δ @RUN
Δ @XQT 1000-MIRADS*MA. AMIRADS

  ENTER OUTPUT REPORT SITE ID
Δ DRUM, OUTPUT OF MIRADS QUERY.
  DRUM FILE NAME IS MIRADS093524
  READY
```

This example assigns and catalogs a FASTRAND file MIRADS093524. Upon completion of MIRADS, the user can copy the contents of his Query output (MIRADS093524) onto a tape file or the output can be directed to a medium or high speed printer by a @SYM control statement.

```
@SYM, U      MIRADS093524, , SITE-ID
```

```
Δ @RUN
Δ @XQT 1000-MIRADS*MA. AMIRADS

  ENTER OUTPUT REPORT SITE ID
Δ DRUM, OUTPUT OF MIRADS QUERY.
  DRUM FILE NAME IS MIRADS102913
  READY

Δ @ASG, T MYTAPE., 8C, SAVE05 . TAPE FOR QUERY OUTPUT
Δ @COPY, F MIRADS102913., MYTAPE.
Δ @FIN
```

Within this example, MYTAPE represents a 1-12 character unique name assigned by the user to the output tape. Once all Query Sets have been executed, the user directs the output to the tape by the @COPY, F statement. The information recorded on the tape can be used as input to some other computer program, computer site, or it can be used as a @SYM tape to be printed on special forms provided by the user. The user must record the save tape number which is provided in the accounting information printed on the output of the initiating terminal at the end of his RUN.

Any user of MIRADS should have a complete understanding of the DRUM option for directing Query output before attempting its use. A MIRADS representative is available to provide any instructional aid required.

3.4.1.2 Error Diagnostic Messages for the RUN Command

The improper use of the Site-ID options for the RUN command will result in an error condition. This condition will be indicated by one of the following messages being received at the initiating terminal.

```
31  XXXXXX IS NOT A VALID PRINT LIMITING VALUE
```

This message indicates that the user has entered an invalid print limit, improper punctuation, or an invalid character. The user must reenter the entire command making sure that it is valid before transmitting it to the computer.

```
30  XXXXXX IS NOT A VALID SITE ID
```

This message indicates that the user has specified an invalid Site-ID on an otherwise valid command. The user must reenter only the Site-ID ensuring that it is valid before transmitting it to the computer.

```
3 INVALID SITE ID MESSAGES HAVE BEEN ENTERED
OUTPUT REPORT WILL BE STORED ON DRUM
```

The user is allowed three chances to transmit a valid Site-ID. If the third attempt is in error, the system will automatically store the generated report on a cataloged drum file. The system will then output the file name. If the requested information is needed, the user must obtain a valid Site-ID before executing MIRADS. The user can then reenter the execute control statement @XQT MIR*ADS. MIRADS and reenter the Query Set. If the output is not required, the user should terminate the RUN by transmitting @FIN.

Example: Query Set using the RUN command

```
     READY
  Δ  Q, SALARY GT '12000'.
     READY
  Δ  S,SALARY D.
     READY
  Δ  P, NAME, SALARY.
     READY
  Δ  RUN
       .QUERY NOW PROCESSING
          FILE CONTAINS 500 RECORDS
          QUERY SELECTED 3 RECORDS
          ENTER OUTPUT REPORT SITE ID
  Δ          (Blank Line)
           Q, SALARY GT '12000'.
           S, SALARY D.
           P, NAME, SALARY.


        NAME              SALARY

     DOE       JOHN D       16000
     SMITH     FRANK F      15000
     JONES     TOM J        12500
     READY
```

3.4.2 TOP Command

If a Query Set or command has been entered in error, the user can inhibit the execution of the Query Set by entering the command:

```
TOP
```

Following the TOP command, the system will respond with the message:

READY

At this time, the user can enter a new Query Set.

3.4.2.1 TOP Example

```
ΔQ, DEPT P.
 READY
ΔP,DEPT,PB72,AB72.
 READY
ΔTOP
 READY
```

In the above example, the user decided that this inquiry would not satisfy his requirements, therefore, the command TOP was entered so that a new Query Set could be entered.

3.4.3 NEW Command

The NEW command allows the user to access a different Data Base from the one which was previously queried.

The NEW command has only one field, the command field, and does not need to be terminated with a period.

When a user desires to access a new Data Base, he enters the command

```
NEW
```

following the system message

```
READY
```

At this time the system will respond with the message

```
ENTER QUALIFIER*FILENAME
```

3.4.4 KEY Command

The KEY command is used to produce a listing of all Data Base Field Names which are indexed for rapid retrieval of information. The level number, field size, and data type for each Field Name listed is shown.

Following the message

```
READY
```

the user can enter and transmit the command

```
KEY
```

The system responds with the output listing in the following format.

```
LEVEL   FIELD NAME   FIELD SIZE   DATATYPE
  n       FLD-1          n         Type-1
  n       FLD-2          n           .
  n       FLD-3          n           .
  .         .            .           .
  .         .            .           .
  .         .            .           .
  n       FLD-n          n         Type-n
READY
```

The n under LEVEL represents a three digit integer value which specifies the Data Base record level for the Field Name. The Field Names which are listed are indexed fields. The number of characters contained in the field are given under FIELD SIZE and the type of data contained in the field is given under DATATYPE.

3.4.5 CHECK Command

The CHECK command allows the user to generate a listing of the commands which have been previously specified in the Query-Set. presently being entered. This command is oriented for users of a CRT type terminal where images roll off the screen.

Following the message

```
READY
```

the user can enter the command

CHECK

The system responds by listing all commands entered up to this time in the Query-Set. Following the list, the system responds with the message

READY

at which time, the user can complete the Query-Set.

3.5 MIRADS DIAGNOSTIC MESSAGES

MIRADS is designed to provide edit checking and validation of user input commands. All errors detected by the MIRADS edit programs result in an error message being displayed describing the reason for the error. The user can generally recover from the error condition by retransmitting the command correctly. Terminal messages fall into four categories--error messages, warning messages, processing messages, and termination messages. Error messages are more severe than warning messages and they require action by the user. Warning messages caution the user about a potential problem but require no action by the user. Processing messages are displayed when normal execution of MIRADS occurs and they are not answerable. Termination messages are displayed by MIRADS. If the user does not understand the meaning of a message or if MIRADS is not performing or responding as expected, the MIRADS

Representative should be contacted for assistance. If the user received a message or error condition which is not explained in this section, he should refer to the Standard Trouble Reporting Procedure as given in Section 2.4.

3.5.1 MIRADS Error Messages

The following example indicates how the MIRADS Error Message Table is to be used:

```
  READY
Δ QUERY, NAM EQ 'JONES'.
  112  NO DICTIONARY ENTRY FOR FIELD NAME NAM
  01   QUERY COMMAND DISCARDED - PLEASE CHECK AND REENTER
  READY
Δ QUERY, NAME EQ 'JONES'.
  READY
Δ PRINT, NAME, ADDRESS.
  READY
  RUN
```

NOTE

Underlines would not appear on the terminal and are used here for emphasis only.

The error message NO DICTIONARY ENTRY FOR FIELD-NAME indicates the user keyed-in an invalid Field-Name (NAM), that is, one which is not defined in the Dictionary for the Data Base. The user recovered by entering the command correctly. Following the error message, the system will transmit the message READY. After analyzing the error, the user can then reenter the command.

MIRADS ERROR MESSAGE TABLE

MIRADS Driver

Number Error Message

01 _____COMMAND DISCARDED - PLEASE CHECK AND REENTER
Meaning: The specified command was discarded from the Query-Set due to the inability to interpret the request. The command should be reentered under the rules given in a section of this manual pertaining to the respective command.

02 _____COMMAND CONTAINED AN ERROR - ALL CONTINUATIONS WILL BE DISCARDED
Meaning: Occasionally a remote terminal operates with an unusually slow response to user input. At this time, some users input several lines of a command before any response is received for the first line. If an error is detected in the command on the first or any continuation line, the entire command will have to be reentered under this condition.

03 Element-Name IS NOT A QUERY SET OR PRINT FORMAT NAME
Meaning: The Element-Name specified on a SAVE function has not been generated for the requested Data Base. No Saved-Query-Set or Print-Format-Definition has been generated under the specified name

04 _____COMMAND HAS ALREADY BEEN ENTERED - THIS ENTRY DISCARDED
Meaning: The specified command type has already been entered for this Query-Set. Query-Sets can contain multiple commands for SORT and COMPUTE only.

05 X IS NOT PERMITTED IN A NAME
Meaning: The character specified in position X is an invalid character.

06 Element-Name IS ALREADY A SAVED NAME
Meaning: The name you have chosen for a Saved-Query-Set, Print-Format-Definition, or Saved-Command has already been used. A unique name must be specified on the command. To obtain a list of all names which have been used, enter the LIST command.

MIRADS Driver (Continued)

Number Error Message

07 Element-Name IS NOT A QUERY SET NAME
Meaning: The Saved-Query-Set name specified on a save function does not exist. Enter the LIST command to obtain a list of valid names.

08 Element-Name IS NOT A PRINT FORMAT NAME
Meaning: The name requesting a formatted print definition does not exist. Enter the LIST command to obtain a valid list of print definitions.

09 ILLOGICAL INPUT - PLEASE CHECK AND REENTER
Meaning: MIRADS is not capable of interpreting this command probably due to a mispelled word or missing space or comma.

10 NO NAME GIVEN IN FORMAT COMMAND - PLEASE CHECK AND REENTER
Meaning: Your request for a formatted print definition does not identify the formats which you wish to use.

11 NO NAME GIVEN IN DELETE COMMAND - PLEASE CHECK AND REENTER
Meaning: The Saved-Query-Set was not identified by name on the DELETE command.

12 NO NAME GIVEN IN DISPLAY COMMAND - PLEASE CHECK AND REENTER
Meaning: The Saved-Query-Set was not identified by name on the DISPLAY command.

13 NO NAME GIVEN IN DO COMMAND-PLEASE CHECK AND REENTER
Meaning: The Saved-Query-Set was not identified by name on the DO command.

14 NO NAME GIVEN IN DEFINE COMMAND - PLEASE CHECK AND REENTER
Meaning: The print definition specified has no name by which it can be identified or referenced.

MIRADS Driver (Continued)

Number Error Message

15 DUPLICATE SAVE COMMANDS ILLOGICAL - THIS COMMAND DISCARDED
Meaning: The second SAVE command which was entered for this Query-Set has been discarded. The Query Set will be saved under the name specified on the first SAVE command.

16 INCOMPLETE QUERY SET - NO QUERY COMMAND PRESENT
Meaning: An attempt was made to execute a Query-Set without a QUERY command.

17 INCOMPLETE QUERY SET - NO PRINT COMMAND PRESENT
Meaning: An attempt was made to execute a Query-Set without a PRINT command.

18 NO NAME GIVEN IN SAVE COMMAND - PLEASE CHECK AND REENTER
Meaning: A SAVE command was specified but no name was given to uniquely identify the Query-Set.

19 CURRENT PASSWORD DOES NOT PERMIT FILE UPDATING
Meaning: The password you have selected is not a valid password for Data Base updates. Query-Sets which do not have an UPDATE command can be executed.

20 UPDATE NOT ALLOWED WITH SORT OR COMPUTE COMMANDS
Meaning: Sort and compute functions are not allowed while doing an update.

21 SORT COMMAND NOT PERMITTED WITH UPDATE
Meaning: A SORT command was entered into a Query-Set which contains an UPDATE command.

22 ALL COMPUTE COMMANDS MUST BE ENTERED CONTIGUOUSLY - THIS COMMAND DISCARDED
Meaning: The COMPUTE command last entered was discarded because a command other than COMPUTE was entered into the Query-Set. Multiple COMPUTE commands must be entered continuously.

MIRADS Driver (Continued)

Number — Error Message

23 COMPUTE NOT PERMITTED WITH UPDATE
Meaning: Computations cannot be performed in a Query-Set which contains an update request. A separate Query-Set should be executed to perform the computations.

24 ILLOGICAL RUN COMMAND - NO QUERY OR UPDATE COMMAND PRESENT
Meaning: An attempt was made to execute an incomplete Query-Set. The Query-Set must have a QUERY or UPDATE command.

25 ILLOGICAL RUN COMMAND - NO PRINT OR FORMAT COMMAND PRESENT
Meaning: An attempt was made to execute an incomplete Query-Set. The Query-Set must have a PRINT or FORMAT command.

26 PREVIOUS COMMAND COMPLETE - CONTINUATION DISCARDED
Meaning: An attempt was made to enter a continuation line for the previous command which was terminated with a period. The continuation line was deleted.

27 PREVIOUS COMMAND INCOMPLETE - IT HAS BEEN DISCARDED
Meaning: A new command was introduced to the Query-Set before the previous command was complete. A complete command should be entered for the discarded command.

28 THREE INCORRECT REPLIES HAVE BEEN RECEIVED - FORCED SEARCH WILL NOT BE PERFORMED
Meaning: A yes or no answer was not given to specify the execution of a sequential search of the Data Base.

29 PLEASE ANSWER EITHER YES OR NO
Meaning: An improper response was given concerning the continuance of a sequential search.

30 New-Variable-Name IS AN UNDEFINED NEW VARIABLE
Meaning: The New-Variable-Name specified has not been defined by a COMPUTE statement before a RUN was attempted.

QUERY Command

Number Error Message

100 LIMIT NUMBER CANNOT BE GREATER THAN 6 DIGITS
Meaning: The maximum value for the search limit option (LIMIT = n) of the QUERY command is 999999. The value specified on your QUERY command exceeds that value.

101 LIMIT NUMBER CANNOT BE LESS THAN 1 DIGIT
Meaning: The minimum value for the search limit option (LIMIT = n) of the QUERY command is 1. The value specified on your QUERY command is less than 1.

102 LIMIT NUMBER IS NOT NUMERIC
Meaning: The value specified for the search limit option (LIMIT = n) is not numeric.

103 SEARCH LEVEL NUMBER MUST BE BETWEEN 1 AND 8
Meaning: The value specified for the search level option (SEARCH-n) is not within the range of 1 to 8.

104 PARENTHESES EXCEEDS 3 LEVELS
Meaning: More than three levels of parentheses were used in establishing the logical order of the QUERY command. A maximum of three levels is allowed.

105 RIGHT PARENTHESES DO NOT EQUAL LEFT PARENTHESES
Meaning: The QUERY command was terminated with an imbalance in parentheses. There is no way to determine the logical intent of the command.

111 FIELD NAME HAS BEEN OMITTED
Meaning: A relational and search criterion was specified without identifying the Data Base field to be tested.

112 NO DICTIONARY ENTRY FOR FIELD NAME
Meaning: The specified Field-Name is not defined in the Dictionary. Please check your Dictionary listing for the proper Field-Names.

QUERY Command (Continued)

Number Error Message

113 NO PREVIOUS FIELD NAME FOR RELATIONAL
Meaning: A relational was encountered without a Field-Name specified.

114 NO PREVIOUS FIELD NAME AND RELATIONAL FOR DATA
Meaning: A search criterion was encountered but no Field-Name and relational were specified for that criterion.

115 RELATIONAL WAS NOT ENCOUNTERED
Meaning: A Field-Name and search criterion were specified without an intervening relational.

116 DATA FIELD EXCEEDS 48 CHARACTERS
Meaning: The search criterion has a maximum of 48 characters. The criterion specified exceeds the maximum number of characters. If possible, perform a Character, Keyword, on Keyphrase search.

117 NUMERIC DATA CONTAINS BLANK OR NON-NUMERIC CHARACTERS OR FIELD NAME IS INVALID
Meaning: A numeric search criterion contains non-numeric character or next Field-Name is not defined in the Dictionary.

118 DATA FIELD EXCEEDS DICTIONARY LENGTH - FIELD NAME IS ______
Meaning: The search criterion specified for the ______ Field-Name contains more characters than the field can possibly have in it.

119 DECIMAL FIELD EXCEEDS ALLOWABLE SIZE - FIELD NAME IS ______
Meaning: The search criterion specified for the ______ Field-Name contains more decimal positions than are defined for the field in the Dictionary.

120 UNSIGNED DATA CONTAINS ILLEGAL SIGN - FIELD NAME IS ______
Meaning: A plus or minus sign was specified as part of the search criterion for a field that is defined as unsigned by the Dictionary.

QUERY Command (Continued)

Number Error Message

121 DATA WAS NOT ENCOUNTERED
Meaning: The search criterion was not specified for a Field-Name and relational. It is possible that input similar to FLD-A GT FLD-B should have been FLD-A GT *FLD-B.

122 SEARCH NOT ALLOWED ON FIELD GREATER THAN 48 CHARACTERS FIELD NAME IS _____
Meaning: The Field-Name specified for the search contains more than 48 characters and can only be specified on a PRINT or DEFINE command.

123 CONNECTOR WAS NOT ENCOUNTERED
Meaning: Multiple Field-Names or search criteria were specified without the logical connectors of AND or OR.

124 FOLLOWING DATA UNRECOGNIZABLE _____
Meaning: The data specified does not meet any of the formats of the available requirements or options on the QUERY command. The data will be ignored.

125 AND/OR CONDITIONS NEED CLARIFICATION---REWRITE OR REINPUT WITH MORE PARENTHESES
Meaning: The requested logic for multiple AND/OR conditions cannot be determined as input. Reenter the command using parentheses for clarification.

126 TOTAL NUMBER OF ELEMENTS EXCEEDS 100
Meaning: The total number of Field-Names to the left of a relational exceeds the maximum limit of 100.

127 DATA FIELD CONTAINS MORE THAN 1 SIGN
Meaning: The search criterion contains more than one plus (+) or minus (-) sign.

128 KP, KW SEARCH NOT ALLOWED ON RELATIONAL OTHER THAN EQUAL
Meaning: Keyphrase and Keyword searches are allowed only with the equal relational.

QUERY Command (Continued)

Number Error Message

129 TOO MANY DIGITS LEFT OF DECIMAL POINT
Meaning: The integer portion of the search criterion value exceeds the maximum values which can be contained in the Data Base field.

130 EDITING IS DISCONTINUED
Meaning: Editing of the QUERY command was terminated because too many errors were encountered.

131 FIELD CANNOT BE REFERENCED WITH USERS PRESENT SECURITY KEY
Meaning: The requested field cannot be accessed under the specified security key. To access this field, the proper security key must be used.

132 BINARY INTEGER FIELD HAS A MAXIMUM OF TEN NUMERIC DIGITS
Meaning: The decimal equivalent of the Binary field cannot contain more than ten numeric digits.

133 INVALID CHARACTER IN BINARY INTEGER FIELD
Meaning: The Binary field contains a non-numeric character other than + or -.

134 DECIMAL DIGITS NOT ALLOWED IN BINARY INTEGER FIELD
Meaning: The Binary field must contain only integer values.

135 INVALID CHARACTER IN FLOATING POINT ENTRY
Meaning: The entry should only contain numeric, E, D, +, -, or . (decimal point); some character other than these was input.

136 ONLY 9 SIGNIFICANT DIGITS ARE ALLOWED IN SINGLE PRECISION ENTRY
Meaning: Only 9 numeric digits can be input for the mantissa in a single precision entry.

137 ONLY 18 SIGNIFICANT DIGITS ARE ALLOWED IN DOUBLE PRECISION ENTRY
Meaning: Only 18 numeric digits can be input for the mantissa in a double precision entry.

QUERY Command (Continued)

Number Error Message

138 INVALID CHARACTER IN SINGLE PRECISION ENTRY
Meaning: The entry contains some character other than the numeric, +, -, E, or decimal point allowed for single precision entries.

139 INVALID CHARACTER IN DOUBLE PRECISION ENTRY
Meaning: The entry contains some character other than the numeric, +, -, D, or decimal point allowed for double precision entries.

140 ONLY 2 EXPONENTIAL DIGITS ARE ALLOWED IN SINGLE PRECISION ENTRY
Meaning: The entry expressing the powers of ten after the mantissa exceeded the allowable two digits.

141 ONLY 3 EXPONENTIAL DIGITS ARE ALLOWED IN DOUBLE PRECISION ENTRY
Meaning: The entry expressing the powers of ten after the mantissa exceeded the allowable three digits.

142 SINGLE PRECISION EXPONENT MUST BE BETWEEN -38 AND +38
Meaning: The single precision entry must be between the approximately limits 10^{-38} and 10^{38}.

143 DOUBLE PRECISION EXPONENT MUST BE BETWEEN -308 AND +308
Meaning: The double precision entry must be between the approximately limits 10^{-308} and 10^{308}.

SORT Command

Number — Error Message

200 SPACE OR COMMA MISSING FOLLOWING SORT DESIGNATION
Meaning: The command disignator S or SORT was not followed by a space or comma before the first sort key (Field-Name) was entered.

201 SORT COMMAND MUST CONTAIN ONE FIELD NAME
Meaning: The command was terminated with a period before a sort key (Field-Name) was specified.

202 NO DICTIONARY NAME OR NEW VARIABLE NAME FOR FIELD NAME
Meaning: The sort key (Field-Name) which has been entered cannot be found in the Data Base Dictionary.

203 SPECIFIED SORT LIMIT IS GREATER THAN FIELD SIZE
Meaning: The number of characters specified for the sort key (Field-Name) is greater than the actual size of the field.

204 SORT NOT ALLOWED ON FIELD GREATER THAN 48 CHARACTERS
Meaning: The sort key (Field-Name) is larger than 48 characters for a single field. If this Field-Name is required, the Limit option must be used.

205 SORT NOT ALLOWED ON FIELD NAMES TOTALING MORE THAN 78 CHARACTERS
Meaning: The number of characters for all sort keys (Field-Names) total more than 78 characters.

206 EDITING IS DISCONTINUED
Meaning: More than two errors were encountered and editing was terminated.

207 FIELD CANNOT BE REFERENCED WITH USERS PRESENT SECURITY KEY
Meaning: The requested field is not available for the security key which has been specified. To access this field, a proper security key must be used.

SORT Command (Continued)

Number	Error Message
208	SORT BEFORE COMPUTE CANNOT BE USED TO SORT NEW VARIABLES Meaning: New Variables must be computed before an attempt to sort a New Variable is made.

COMPUTE Command

Number Error Message

300 BREAK FIELD IS A NEW VARIABLE
Meaning: Break-Field is not a Data Base Field-Name, New-Variable names cannot be used as a Break-Field.

301 BREAK FIELD GREATER THAN 12 CHARACTERS
Meaning: Field-Name in break-field exceeds the maximum limit of 12 characters.

302 DECIMAL DIGIT IS NOT NUMERIC OR DELIMITER MISSING
Meaning: Digit entered for the number of decimal positions required for the New-Variable is not numeric or was omitted.

303 DECIMAL DIGIT GREATER THAN 9 - TOO LARGE
Meaning: Value entered for decimal digit is greater than its maximum value of 9.

304 NO EQUAL SIGN
Meaning: No equal sign was encountered following the New-Variable name.

305 TOO MANY EQUAL SIGNS IN THIS COMMAND
Meaning: Command contains more than 1 equal sign. Only one equal sign is allowed.

306 NO NEW VARIABLE LEFT OF EQUAL SIGN
Meaning: Receiving field for computed value was not specified.

307 COUNT FIELD WAS NOT REQUESTED
Meaning: Data Base field to be counted was not specified.

308 COUNT FIELD HAS MORE THAN 12 CHARACTERS
Meaning: An invalid Field-Name was entered for the field to be counted.

309 FIELD NAME GREATER THAN 12 CHARACTERS
Meaning: The specified Field-Name contains more than the maximum of 12 characters.

COMPUTE Command (Continued)

Number	Error Message
310	ONE NEW VARIABLE HAS MORE THAN 36 CHARACTERS Meaning: New-Variable name contains more than the maximum of 36 characters.
311	COMPUTE ON NON-NUMERIC FIELD NOT ALLOWED Meaning: Attempt to compute an algebraic expression using non-numeric data.
312	OPERATION NOT ALLOWED ON FIELD GREATER THAN 48 CHARACTERS Meaning: Data Base field is larger than 48 characters. The field can be used only on PRINT or DEFINE commands.
313	FORMULA CONTAINS LESS THAN 3 ELEMENTS Meaning: Compute command is missing New-Variable, relational, or Field-Name.
314	MORE THAN 3 LEFT PARENTHESES IN THIS COMMAND Meaning: More than 3 open left parentheses were used in this command. Open parentheses are restricted to 3 levels.
315	MORE THAN 3 RIGHT PARENTHESES IN THIS COMMAND Meaning: More than 3 open right parentheses were used in this command. Open parentheses are restricted to 3 levels.
316	THE RIGHT PARENTHESES DO NOT EQUAL THE LEFT PARENTHESES Meaning: The parentheses counter is out of balance. There must be an equal number of right and left parentheses.
317	TOO MANY OR TOO FEW DELIMITERS IN COMMAND Meaning: A space or comma is missing to separate the elements specified in the command.
318	TOO MANY DIGITS IN ONE OF THESE CONSTANTS______ Meaning: The constant exceeds its maximum value of 999999999.999999999.

COMPUTE Command (Continued)

Number Error Message

319 NO OPERATOR AFTER ONE OF THE NEW VARIABLES
Meaning: No operator was found for a New-Variable in the command.

320 MORE THAN 20 UNIQUE NEW VARIABLES FOR THIS COMMAND
Meaning: The maximum number (20) of New-Variables was exceeded for this Query-Set.

321 MORE THAN 20 ELEMENTS FOR THIS FORMULA
Meaning: The formula for a single COMPUTE command cannot contain more than 20 Field-Names, arithmetic operators, and constants.

322 MORE THAN 120 FIELD NAMES FOR THESE COMMANDS
Meaning: More than 120 Field-Names were used in all COMPUTE commands in this Query-Set.

323 MORE THAN 100 ELEMENTS FOR THIS COMMAND
Meaning: More than 100 arithmetic operators were specified in all COMPUTE commands contained in the Query-Set.

324 EDITING IS DISCONTINUED
Meaning: More than two errors were encountered and editing was terminated.

325 NO DICTIONARY OR NEW VARIABLE NAME FOR:______
Meaning: The Field-Name which has been specified is not defined in the Dictionary or the New-Variable specified was not computed.

326 FIELD NAME CARRIED OVER FROM 1ST STATEMENT TO NEXT OR OPERATOR IS MISSING
Meaning: An improper continuation line has been entered or an operator was omitted.

327 FIELD CANNOT BE REFERENCED WITH USERS PRESENT SECURITY KEY
Meaning: The requested field cannot be accessed under the specified security key. To access this field, the proper security key must be used.

COMPUTE Command (Continued)

Number Error Message

328 INVALID CHARACTER IN FLOATING POINT ENTRY
Meaning: The entry should only contain numeric, E, D, +, -, or . (decimal point), some character other than these was input.

329 ONLY 9 SIGNIFICANT DIGITS ARE ALLOWED IN SINGLE PRECISION ENTRY
Meaning: Only 9 numeric digits can be input for the mantissa in a single precision entry.

330 ONLY 18 SIGNIFICANT DIGITS ARE ALLOWED IN DOUBLE PRECISION ENTRY
Meaning: Only 18 numeric digits can be input for the mantissa in a double precision entry.

331 INVALID CHARACTER IN SINGLE PRECISION ENTRY
Meaning: The entry contains some character other than the numeric, +, -, E, or decimal point allowed for single precision entries.

332 INVALID CHARACTER IN DOUBLE PRECISION ENTRY
Meaning: The entry contains some character other than the numeric, +, -, D, or decimal point allowed for double precision entries.

333 ONLY 2 EXPONENTIAL DIGITS ARE ALLOWED IN SINGLE PRECISION ENTRY
Meaning: The entry expressing the powers of ten after the mantissa exceeds the allowable two digits.

334 ONLY 3 EXPONENTIAL DIGITS ARE ALLOWED IN DOUBLE PRECISION ENTRY
Meaning: The entry expressing the powers of ten after the mantissa exceeds the allowable three digits.

335 SINGLE PRECISION EXPONENT MUST BE BETWEEN -38 AND +38
Meaning: The single precision entry must be between the approximate limits 10^{-38} and 10^{38}.

336 DOUBLE PRECISION EXPONENT MUST BE BETWEEN -308 AND +308
Meaning: The double precision entry must be between the approximate limits 10^{-308} and 10^{308}.

PRINT Command

Number	Error Message
400	BREAK ALREADY INPUT FOR FIELD-NAME Meaning: An attempt was made to specify the BREAK option more than once for the same Field-Name.
401	MORE THAN 1 CHARACTER INPUT FOR NUMBER OF SPACES TO BE SKIPPED Meaning: Value specified for number of spaces to be skipped is greater than 9.
402	NUMBER OF SPACES TO BE SKIPPED IS NOT NUMERIC Meaning: Value specified for number of spaces to be skipped is not numeric within the range of 0 to 9.
403	NO STARTING PRINT POSITION STATED FOR PPOS OPTION Meaning: No print position was specified for column positioning of a field.
404	PPOS OPTION NOT IN RANGE 1 THRU 128 Meaning: The value specified for column positioning of a field does not conform to the size of a print line.
405	MORE THAN ONE CHARACTER INPUT FOR GROUP OR ASSOCIATED GROUP LEVEL Meaning: Suppression option is not in the range of 1 to 9.
406	GROUP OR ASSOCIATED GROUP LEVEL IS NOT NUMERIC Meaning: An illegal value was specified for group suppression.
407	NUMBER OF DATA PRINT LINES EXCEEDS 5 Meaning: Request to print more than 5 lines per hit is illegal. If required, use the FORMAT command.
408	MORE THAN 50 FIELD NAMES AND/OR NEW VARIABLES Meaning: Request to print more than 50 values per hit is illegal. If required, use the FORMAT command.
409	NO DICTIONARY OR NEW VARIABLE NAME FOR FIELD NAME Meaning: The specified Field-Name does not correspond to any name contained in the Dictionary or the New-Variable was not computed in this Query-Set.

PRINT Command (Continued)

Number — Error Message

410 NO FIELD NAME OR NEW VARIABLE SUBMITTED
Meaning: No Field-Name or New-Variable was specified to be printed. Command must have at least one.

411 SPACE AFTER PRINT ALREADY INPUT FOR FIELD-NAME
Meaning: Attempt to specify the SPACE n option more than once at same time.

412 SPACE BEFORE PRINT ALREADY INPUT FOR FIELD-NAME
Meaning: Attempt to specify the SPACE n option more than once at same time.

413 PPOS ALREADY INPUT FOR FIELD NAME
Meaning: PPOS option has already been specified for the given field.

414 STARTING PRINT POSITION FOR PPOS OPTION IS NOT NUMERIC
Meaning: Starting print position does not conform to numeric range of 1 to 128.

415 LOOKUP ALREADY INPUT FOR FIELD-NAME
Meaning: Table lookup option has already been specified for the given field.

416 GROUP ALREADY INPUT FOR FIELD-NAME
Meaning: Group suppression option has already been specified for given field.

417 INVALID NEW VARIABLE NAME SUBMITTED
Meaning: The New-Variable specified was not computed in this Query-Set.

418 EDITING IS DISCONTINUED
Meaning: More than two errors were encountered and editing was terminated.

Number Error Message

419 GROUP FIELD NOT SPECIFIED FOR ASSOCIATED LEVEL GROUPING-IGNORED
Meaning: Group Suppression specified without giving a Field-Name to be suppressed.

420 NO LOOKUP TABLE SPECIFIED FOR THIS FIELD
Meaning: Table Lookup was specified for a Field-Name which has no lookup table.

421 SPECIFIED PRINT POSITION FOR _____ IS LESS THAN FIRST AVAILABLE PRINT POSITION
Meaning: The specified field will overlay a previous field if printed in this position.

422 THE PRINTING OF _____ STARTING IN PRINT POSITION _____ WILL EXTEND BEYOND POSITION 128
Meaning: PPOS position does not allow for sufficient columnar positions of the print line for the indicated Field-Name.

423 MORE THAN 1 CHARACTER INPUT FOR BREAK INDICATOR
Meaning: Only one character can be input to specify page advancement on the BREAK option. This character must be an integer value in the range of 1 to 9.

424 BREAK INDICATOR NOT NUMERIC
Meaning: The value for n on the BREAK n option is not numeric within the range of 1 to 9.

425 FIELD CANNOT BE REFERENCED WITH USERS PRESENT SECURITY KEY
Meaning: The requested field is not available on any command for the security key which has been specified. To access this field, a proper security key will have to be used.

DEFINE Command

Number | Error Message

500 PATTERN NAME CONTAINS MORE THAN 12 CHARACTERS
Meaning: The maximum of 12 characters for a pattern name has been exceeded.

501 DEFINE COMMAND IS INCOMPLETE
Meaning: The elements required for a complete command were not processed before a period was found.

502 _____INVALID, FIELD ALREADY ENTERED FOR ELEMENT GROUP
Meaning: A second field name was encountered within an element group. Punctuation is required to separate element groups and only one field is allowed per element group.

503 MORE THAN 200 FIELDS ENTERED
Meaning: The maximum number of 200 fields for one format has been exceeded.

504 _____IS AN INVALID SUBSCRIPTED FIELD
Meaning: A subscript was used with a field which was not defined as a subscripted field in the dictionary.

505 INVALID CHARACTER IN FIELD
Meaning: A character not allowed in field names has been used. Valid characters for field names are A-Z, 0-9, and -.

506 LITERAL ALREADY INPUT IN ELEM_____THE ABOVE LITERAL IS ALREADY IN THE ELEMENT GROUP
Meaning: There should be only one literal in an element set. A second literal is invalid at this time.

507 LITERAL MUST PROCEED FIELD-NAME
Meaning: Element set contains literal and field name. The literal must be input first.

508 LITERAL CONTAINS MORE THAN 130 CHARACTERS
Meaning: Literal exceeds the limit of 130 characters for a print line.

509 ASTERISK FIELD-NAME NOT INPUT
Meaning: Field name should follow the asterisk, none was input.

DEFINE Command (Continued)

Number — Error Message

510 NEW VARIABLE WAS NOT DEFINED IN THE COMPUTE COMMANDS
Meaning: The new variable in this command was not defined in the COMPUTE command.

511 MORE THAN 20 NEW VARIABLES INPUT FOR PATTERN
Meaning: The maximum of 20 new variables per pattern has been exceeded.

512 _____IS AN INVALID NEW VARIABLE NAME
Meaning: This new variable name was not defined in the COMPUTE command.

513 _____OPTION IS ILLEGAL IN THIS ELEMENT GROUP
Meaning: This option conflicts with an option which has been input previously for this element group.

514 SYNONYM CONTAINS MORE THAN 3 CHARACTERS
Meaning: A synonym must contain from one to three numeric digits. More than three digits have been entered for this synonym.

515 NO SYNONYM INPUT FOR FIELD-NAME
Meaning: *Field-Name must be followed by a synonym, none was found.

516 SYNONYM NOT FOUND IN TABLE
Meaning: The table of numeric identifiers for field names does not contain this numeric value.

517 _____IS AN INVALID FIELD-ENTRY
Meaning: This option is invalid when used at this point in the element set.

518 PPOS FIELD CONTAINS MORE THAN 3 CHARACTERS
Meaning: The value of the PPOS field exceeds its limit of three characters.

519 PPOS FIELD GREATER THAN 132 CHARACTERS
Meaning: Print positioning has exceeded the limit of 132 characters for this print line.

DEFINE Command (Continued)

Number Error Message

520 2 PPOS FIELDS IN SAME ELEMENT GROUP
Meaning: Two different print positions have been specified for this element group. Only one PPOS is allowed for an element set.

521 PPOS NOT NUMERIC
Meaning: Print position contains a non-numeric value. The value should be between 1 and 128.

522 JUSTIFICATION ENTERED TWICE FOR FIELD
Meaning: Two data justification entries have been made for this field. Only one justification is allowed for a field.

523 NUMBER OF PAGES OMITTED
Meaning: A numeric value was not given for page number when indicated by the PAGENR option.

524 NUMBER OF PAGES NOT NUMERIC
Meaning: A non-numeric value has been specified for page numbering.

525 NON-NUMERIC SPACING ATTEMPTED
Meaning: The value given for spacing is not numeric.

526 MORE THAN 54 LINES ATTEMPTED
Meaning: The limit of 54 lines per pattern has been exceeded.

527 NO DECIMAL INDICATED FOR NEW VARIABLE FIELD
Meaning: New variable field does not indicate where the decimal should be located.

528 NEW VARIABLE DECIMAL NOT NUMERIC
Meaning: New variable format Fw. d contains some character other than 0 through 9.

529 ILLEGAL CHARACTER IN DECIMAL FIELD
Meaning: New variable format Fw. d was improperly constructed.

530 MORE THAN 54 LINES INPUT-CLOSED
Meaning: The limit of 54 lines has been exceeded. No more input will be accepted.

DEFINE Command (Continued)

Number Error Message

531 EDITING DISCONTINUED
Meaning: More than two errors were encountered and editing was terminated.

532 _____OPTION INVALID IN ELEMENT SET
Meaning: This option cannot be used in this element set.

533 _____IS A PRINT POSITION PREVIOUSLY REFERENCED
Meaning: An attempt was made to reference a print position which has already been filled.

534 PPOS GREATER THAN 132 REFERENCED
Meaning: This print positioning will exceed the maximum value of 132 allowed for a print line.

535 NOT ALL SYNONYMS USED IN DEFINITION OF FORMAT
Meaning: Numeric identifiers for field names in this pattern were not all referenced.

UPDATE Command

Number	Error Message
600	INCORRECT ACTION CODE Meaning: Action code must be 'I', 'R', 'C', or 'D'.
601	QUOTE MARK MISSING ON DATA Meaning: The New-Value specified must be enclosed within quote marks.
602	FIELD NAME GREATER THAN 12 CHARACTERS Meaning: Field-Name cannot be greater than 12 characters.
603	NO DICTIONARY NAME FOR FIELD NAME Meaning: Specified Field-Name is not contained in the Dictionary.
604	UPDATE CHANGE NOT ALLOWED ON KEY FIELD Meaning: Data Base Key Field cannot be updated.
605	DICTIONARY SPECIFIES THIS FIELD AS NON-UPDATE Meaning: An attempt to update a field which is defined by the Dictionary as non-updatable.
606	CRITERION WAS NOT ENCOUNTERED Meaning: New value was not specified for the Field-Name.
607	RELATIONAL WAS NOT ENCOUNTERED Meaning: Relational absent between Field-Name and New-Value.
608	DATA SPLIT BETWEEN LINES - ERROR Meaning: Improper continuation line. New-Values cannot be continued from one line to another.
609	MISPLACED QUOTE MARK Meaning: A quote mark was encountered which is not being used to enclose a New-Value.
610	ALPHA DATA TOO LONG OR QUOTE MARK IS MISSING Meaning: New-Value contains more characters than the specified field will contain. The New-Value may be missing a terminating quote mark.

UPDATE Command (Continued)

Number Error Message

611 NUMERIC DATA TOO LONG OR QUOTE MARK IS MISSING
Meaning: New-Value is greater than the maximum value which can be stored in the Data Base field.

612 MIXED LEVELS NOT PERMITTED IN A SINGLE UPDATE STATEMENT
Meaning: Mixed level records cannot be updated by one command.

613 UPDATE FIELD COUNT EXCEEDS 200
Meaning: More than 200 elements contained in the UPDATE command.

614 NON-NUMERIC DATA IN NUMERIC FIELD
Meaning: Attempt to place non-numeric data into a field described as numeric.

615 RECORD MISSING OR IMPROPER CODE FOR RECORD DELETION
Meaning: Records specified to be deleted are not contained in the Data Base.

616 INCORRECT LEVEL CODE IN UPDATE COMMAND
Meaning: Specified record level is not defined in the Dictionary.

617 KEY FIELD MISSING FOR INSERT ADD OR REPLACE ACTION
Meaning: Insert, add, or replace must specify a value for the Key field.

618 EDITING IS DISCONTINUED
Meaning: More than two errors were encountered. Editing is terminated.

619 UPDATE OF INDEXED FIELDS IN SEQUENTIAL SEARCH NOT PERMITTED
Meaning: Indexed fields cannot be updated after a sequential search.

UPDATE Command (Continued)

Number	Error Message
620	FIELD CANNOT BE REFERENCED WITH USERS PRESENT SECURITY KEY Meaning: The requested field cannot be accessed under the specified security key. To access this field, the proper security key must be used.

FORMAT Command

Number **Error Message**

700 SPACE AFTER INPUT FOR PATTERN NAME
Meaning: Attempt to specify the SPACE n option more than once at the same time.

701 SPACE BEFORE INPUT FOR PATTERN NAME
Meaning: Attempt to specify the SPACE n option more than once at the same time.

702 SPACE INDICATOR NUMBER INVALID
Meaning: The value input for skipping n blank lines before or after printing is not valid.

703 PATTERN BREAK OPTION ALREADY INPUT, ALL INVALID
Meaning: The Break-Field option ALL was input for a display pattern which already contains a Break-Field.

704 PATTERN BREAK OPTION ALREADY INPUT, NONE INVALID
Meaning: The Break-Field option NONE was input for a display pattern which already contains a Break-Field.

705 PATTERN BRK-OPTION INPUT, BREAK-FIELD-NAME INVALID
Meaning: The option specified for the Break-Field-Name as a display pattern is not a Dictionary name, ALL, or NONE.

706 SPLIT ALREADY INPUT FOR PATTERN NAME
Meaning: The SPLIT option was input twice for the same display pattern.

707 INVALID PUNCTUATION IN COLUMN ______
Meaning: The specified column position contains an invalid punctuation character.

708 PATTERN OR FIELD NAME INVALID
Meaning: The Pattern-Name specified is not unique or the Field-Name is not in the Dictionary.

709 EDITING IS DISCONTINUED
Meaning: More than two erros were encountered so editing of this command was terminated.

FORMAT Command (Continued)

Number	Error Message
710	DUPLICATE PATTERN NAMES HAVE BEEN INPUT Meaning: The requested Pattern-Name has already been added to this Query Set and cannot be added again.

3.5.2 Other Diagnostic Messages

Warning Message Table

Number	Warning Message
1	WARNING-QUERY FORCES A SEQUENTIAL SEARCH Meaning: This message indicates that the Data Base is being searched one record at a time and that response time for the Query Set may be unduly long depending on the size of the file.

Processing Message Table

Number	Processing Message
1	READY Meaning: The previous message has been properly received and the system is ready to receive the next user command.
2	QUERY NOW PROCESSING Meaning: Query validation has been completed for the Query Set and data base processing or searching has begun.

Termination Message Table

Number	Termination Message
1	MIRADS EXECUTION NORMALLY TERMINATED Meaning: Successful execution has been concluded and the system has been terminated normally.