FORCE

AUTOMATIC PROGRAMMING SYSTEM USER MANUAL

POINT 4

DATA CORPORATION

POINT 4 DATA CORPORATION
2569 McCabe Way / Irvine, California 92714

FORCE

AUTOMATIC PROGRAMMING SYSTEM USER MANUAL

NOTICE

Every attempt has been made to make this reference manual complete, accurate and up-to-date. However, all information herein is subject to change due to updates. All inquiries concerning this manual should be directed to POINT 4 Data Corporation.

POINT 4 Data Corporation
2569 McCabe Way
Irvine, CA 92714
(714) 754-4114

REVISION RECORD

PUBLICATION NUMBER: AM-140-0023

Revision	Description	Date
A	Preliminary document released as ASP-100-6449-01	03/31/82
B	New Manual Release - this manual corresponds to FORCE Version 1.A which operates under IRIS Rev. 8	05/28/82

LIST OF EFFECTIVE PAGES

Changes, additions, and deletions to information in this manual are indicated by vertical bars in the margins or by a dot near the page number if the entire page is affected. A vertical bar by the page number indicates pagination rather than content has changed.

Page	Rev	Page	Rev	Page	Rev
Cover	-				
Title	-				
ii thru xi	B				
1-1 thru 1-56	B				
2-1 thru 2-19	B				
3-1 thru 3-26	B				
4-1 thru 4-61	B				
5-1 thru 5-46	B				
6-1 thru 6-12	B				
7-1 thru 7-10	B				
8-1 thru 8-33	B				
Appendix Title	-				
A-1 thru A-4	B				
Comment Sheet	B				
Mailer	-				
Back Cover	-				

PREFACE

The FORCE 1.A manual is designed to correspond structurally with the FORCE Automatic Programming System. This design facilitates reference and engenders an effective coordination between FORCE software and its documentation.

Sections 1 through 4 constitute the user portion of the manual. Each section and its subsections are numbered to coincide with the associated FORCE program; the program number is always displayed in the upper right corner of the screen. This enables the user to reference the appropriate manual section from the program name on the CRT.

FORCE program names begin with FS10 (to denote FORCE, Dictionary 1.0) followed by a number which indicates the Level at which the function was implemented. The numbers which follow these first five characters represent the corresponding section in the manual. For example, the program Add Data Elements to a Record Layout shows the name FS103142 on the screen. By covering the first five characters of the name, the user can determine that the function's corresponding manual section is 1.4.2. Similarly, the program FS10645 is explained in Section 4.5, and program FS102323 corresponds to Section 3.2.3 in the manual.

The manual is tabbed on the side of each page to denote the section number and description. Tab positions are staggered for the first four sections. By thumbing through the tabs, the user can quickly reference a desired section. At the beginning of each of the first four sections, a flowchart illustrates the organizational arrangement of related subsections.

Manual text is comprehensive, yet concise; most sections are examined in one page. User sections are divided into three subject areas: PURPOSE, PROCEDURE and NOTES. PURPOSE examines the rationale behind the program. PROCEDURE explains the process involved and indicates any prerequisite functions. If a program has multiple uses, procedures for secondary functions are described following the primary procedure. NOTES offers additional clarifying details, and provides references for further information. A SPECIAL remark is included for procedures which involve several distinct operations.

Section 5 describes the FORCE macros. A subsection is devoted to each macro type, covering command line construction and offering a sample macro expansion. Various charts facilitate the formation of macro commands by outlining required and optional parameters.

The Data Dictionary and FORCE Manager functions are discussed in Section 6. It explores fundamental concepts behind the Dictionary which are essential to an understanding of FORCE, and covers managerial functions which control and optimize Dictionary performance.

Standard operational methods are covered in Section 7 to avoid repeated explanations within each procedure. FORCE prompts, error messages, and interactive techniques are examined. The steps for invoking FORCE are described in this section.

Section 8 covers the methodology and developmental procedures which integrate the individual FORCE functions covered in the user portion of the manual. This section provides insights and techniques for effective application development.

Appendix A includes a glossary which defines various terms as they apply to FORCE.

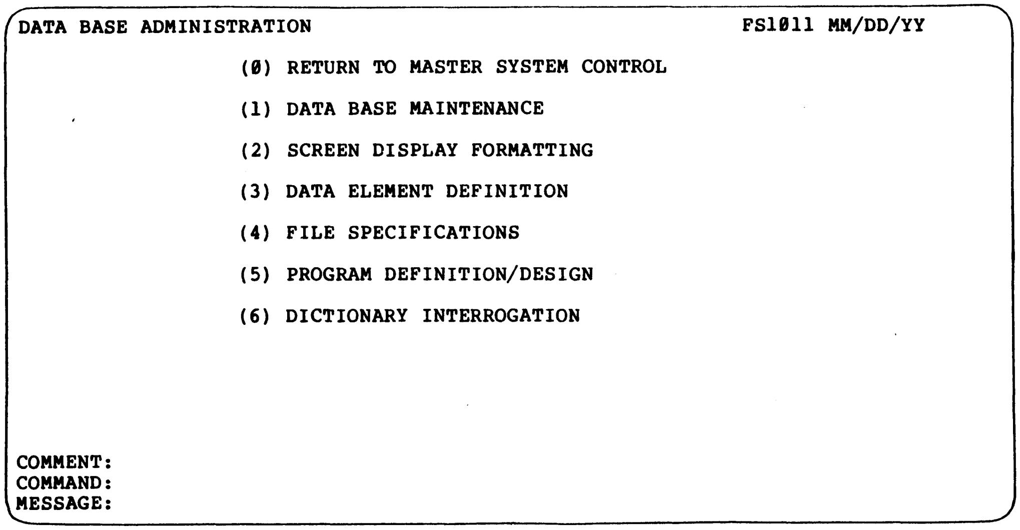
CONTENTS

APPENDICES

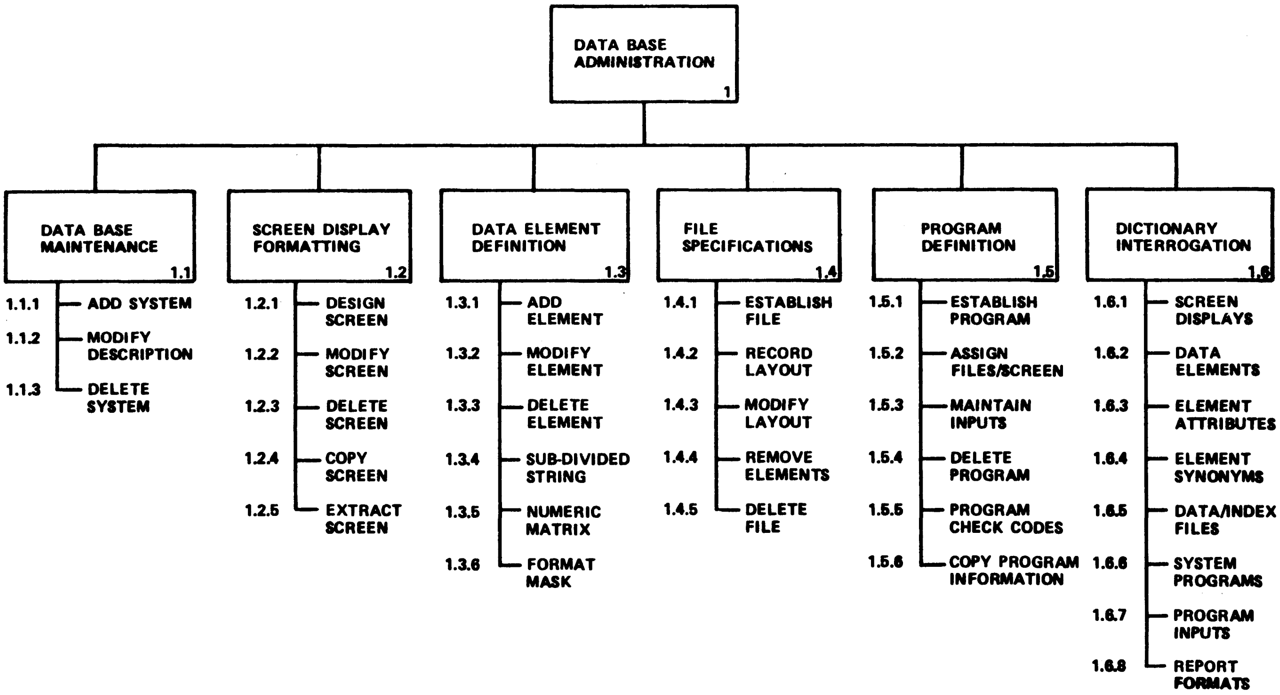
TABLES

Section 1
DATA BASE ADMINISTRATION

PURPOSE: The components of a system must be defined to the Data Base Dictionary for use in source code generation. This information describes the screens, data elements, files and programs which make each system unique. Administration of the Data Base Dictionary involves the specification, management and monitoring of this information.



```
DATA BASE ADMINISTRATION                                         FS1011 MM/DD/YY

                           (0) RETURN TO MASTER SYSTEM CONTROL

                           (1) DATA BASE MAINTENANCE

                           (2) SCREEN DISPLAY FORMATTING

                           (3) DATA ELEMENT DEFINITION

                           (4) FILE SPECIFICATIONS

                           (5) PROGRAM DEFINITION/DESIGN

                           (6) DICTIONARY INTERROGATION




COMMENT:
COMMAND:
MESSAGE:
```



The menu functions within Data Base Administration are arranged in the order that best facilitates a complete system development with FORCE.

1. Data Base Maintenance provides for establishment and maintenance of system names.

2. Screen Display Formatting allows design, maintenance and manipulation of system screens.

3. Data Element Definition specifies characteristics of system data elements. This facility accommodates numerics, strings, subdivided strings and matrices.

4. File Specifications provides functions for establishing and maintaining system files, record layouts and key constructs.

5. Program Definition/Design allows specification of system programs information. This involves coordination of screens, files and data elements within the structure of each program.

6. Dictionary Interrogation allows selective queries of Dictionary information. The output is displayed on the screen.

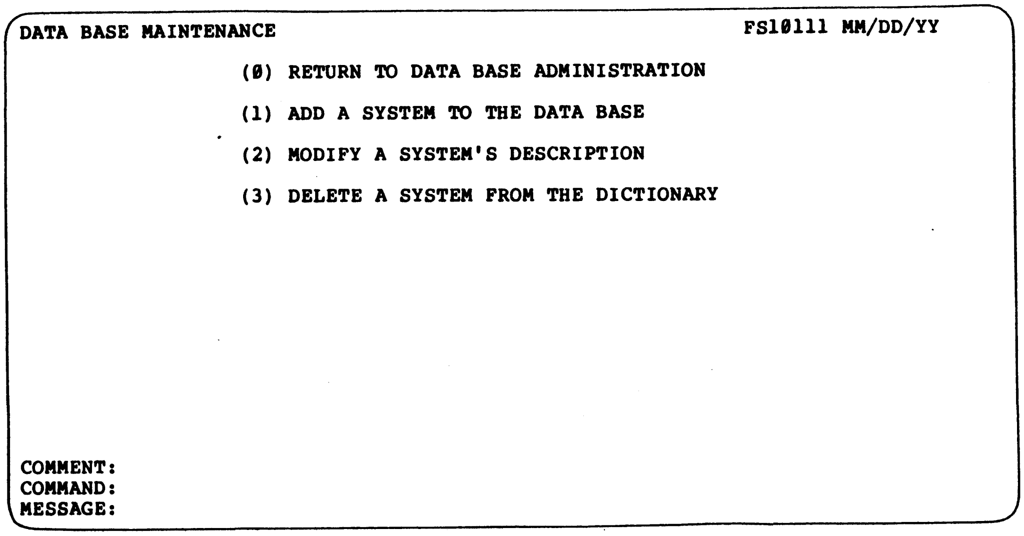
PROCEDURE: Select and enter the number which corresponds to the desired function. Subsequent menu choices are shown in the Data Base Administration flowchart.

NOTES: The FORCE Data Dictionary is built and maintained through the Data Base Administration facility. It allows the user to establish and manage system Dictionary information. System specifications are retrieved from the Dictionary by the Linkage Editor during source code generation.

System data is transferable within the Dictionary for use in different systems.

1.1 DATA BASE MAINTENANCE

__PURPOSE__: This menu provides three functions for high-level system maintenance: Add a System to the Data Base, Modify a System's Description and Delete a System from the Dictionary.



```
DATA BASE MAINTENANCE                                          FS1Ø111 MM/DD/YY
                          (Ø) RETURN TO DATA BASE ADMINISTRATION

                          (1) ADD A SYSTEM TO THE DATA BASE

                          (2) MODIFY A SYSTEM'S DESCRIPTION

                          (3) DELETE A SYSTEM FROM THE DICTIONARY











COMMENT:
COMMAND:
MESSAGE:
```

__PROCEDURE__: Select and enter the number which corresponds to the desired function.

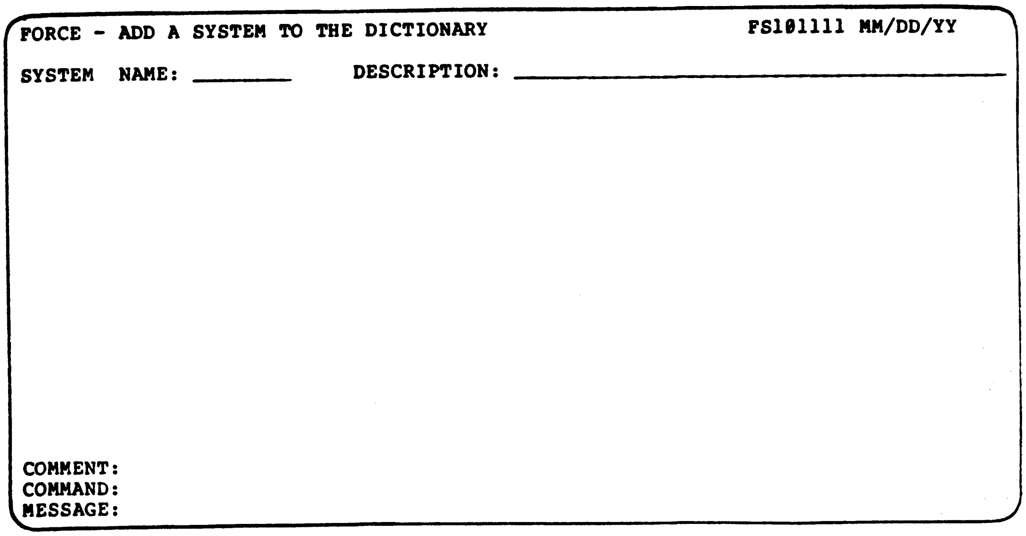
__NOTES__: A FORCE user logged onto the IRIS Manager ID is presented with four additional menu selections. These selections constitute the FORCE manager functions (see 6.2).

Normal applications development with FORCE should __not__ be performed under the Manager ID.

1.1.1 ADD A SYSTEM TO THE DATA BASE

PURPOSE: All system development within the FORCE Dictionary must be preceded by association with a system.

This function establishes a system name within the Dictionary for referencing a complete or partial application/system. Subsequently developed system components (files, programs, etc.) may then be associated with the specified system.



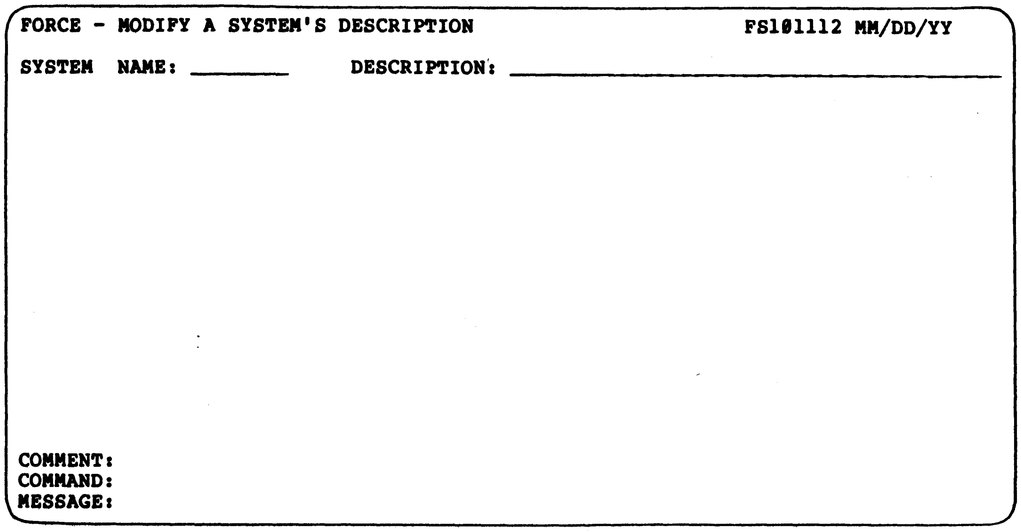
```
FORCE - ADD A SYSTEM TO THE DICTIONARY                    FS101111 MM/DD/YY

SYSTEM  NAME: ________      DESCRIPTION: ________________________________________













COMMENT:
COMMAND:
MESSAGE:
```

PROCEDURE: Enter the system name and its description.

NOTES: A system name in FORCE must begin with a letter and cannot exceed eight characters. The system name and the system description appear on many screens and in most reports. They should be adequately descriptive of the system and meaningful to all potential users.

1.1.2 MODIFY A SYSTEM'S DESCRIPTION

PURPOSE: An established system's description may be modified using this function.



```
FORCE - MODIFY A SYSTEM'S DESCRIPTION                    FS1Ø1112 MM/DD/YY

SYSTEM  NAME: ________     DESCRIPTION: ________________________________________









COMMENT:
COMMAND:
MESSAGE:
```

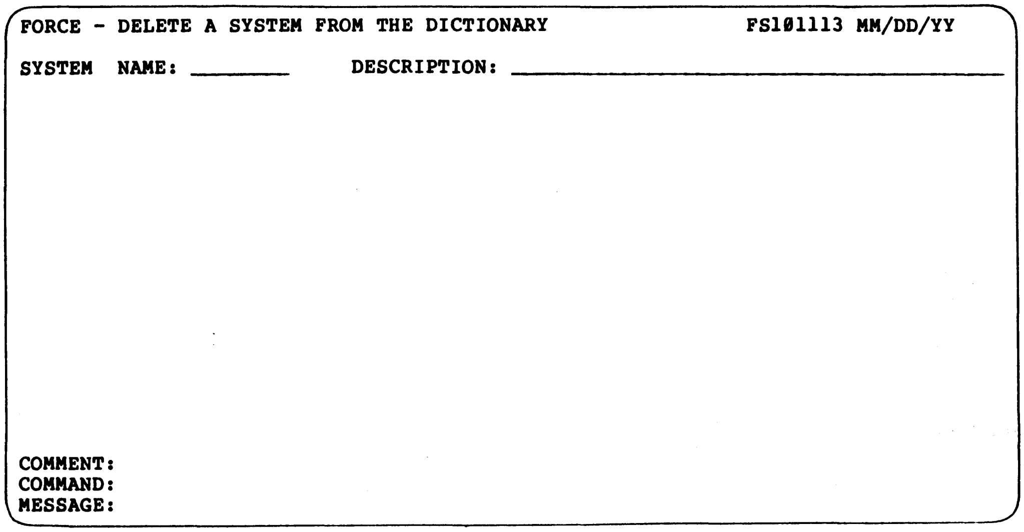
PROCEDURE: Enter the system name. At the Description entry field, enter the modified system description.

DELETE SYSTEM 1.1.3

1.1.3 DELETE A SYSTEM FROM THE DICTIONARY

PURPOSE: A system name that is no longer required may be deleted from the Dictionary.

This function is also used to protect a developed system. A deleted system name provides security against inadvertent Dictionary modifications to a completed system. After the system name is deleted from the Dictionary, system components cannot be referenced for modification or development (see NOTES).



```
FORCE - DELETE A SYSTEM FROM THE DICTIONARY                FS1Ø1113 MM/DD/YY

SYSTEM  NAME: ________      DESCRIPTION: ________________________________________










COMMENT:
COMMAND:
MESSAGE:
```

PROCEDURE: Enter the name of the system to be deleted.

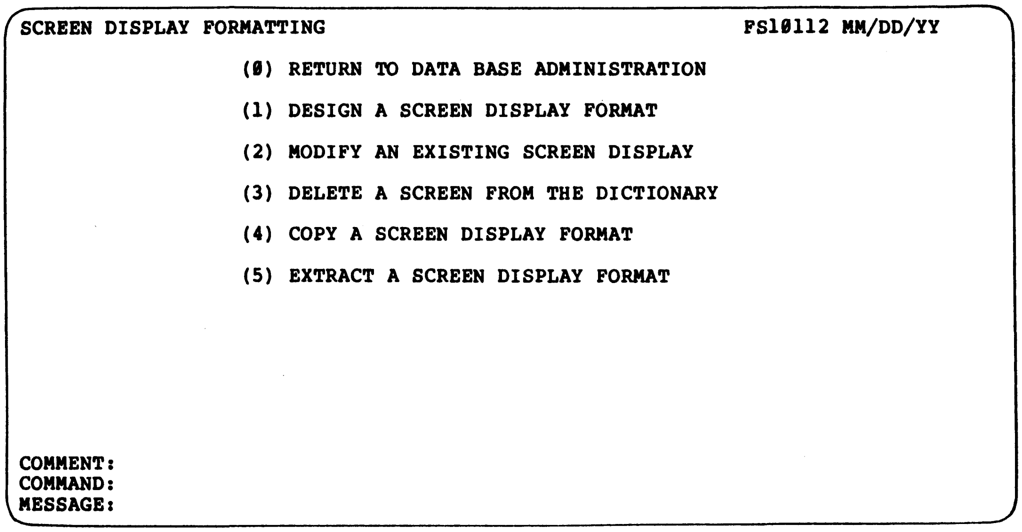
NOTES: This function only deletes the system name from the Dictionary. The structural components of the system remain unaffected in the Dictionary, but can no longer be referenced by the system name.

System name deletion is not irreversible. A deleted system name may be reinstated by the Add a System function (see 1.1.1). By adding the original system name, all references to system components are re-established.

FORCE provides a purge facility under the manager account for the complete removal of all system components from the Dictionary.

1.2 SCREEN DISPLAY FORMATTING

PURPOSE: Screen displays are created by painting the desired screen format on the CRT. The screen format is then stored in the Dictionary and retrieved for code generation or documentation. This menu presents formatting and maintenance functions for screen displays.



```
SCREEN DISPLAY FORMATTING                                    FS10112 MM/DD/YY

                         (0) RETURN TO DATA BASE ADMINISTRATION

                         (1) DESIGN A SCREEN DISPLAY FORMAT

                         (2) MODIFY AN EXISTING SCREEN DISPLAY

                         (3) DELETE A SCREEN FROM THE DICTIONARY

                         (4) COPY A SCREEN DISPLAY FORMAT

                         (5) EXTRACT A SCREEN DISPLAY FORMAT






COMMENT:
COMMAND:
MESSAGE:
```

PROCEDURE: Select and enter the number which corresponds to the desired function.

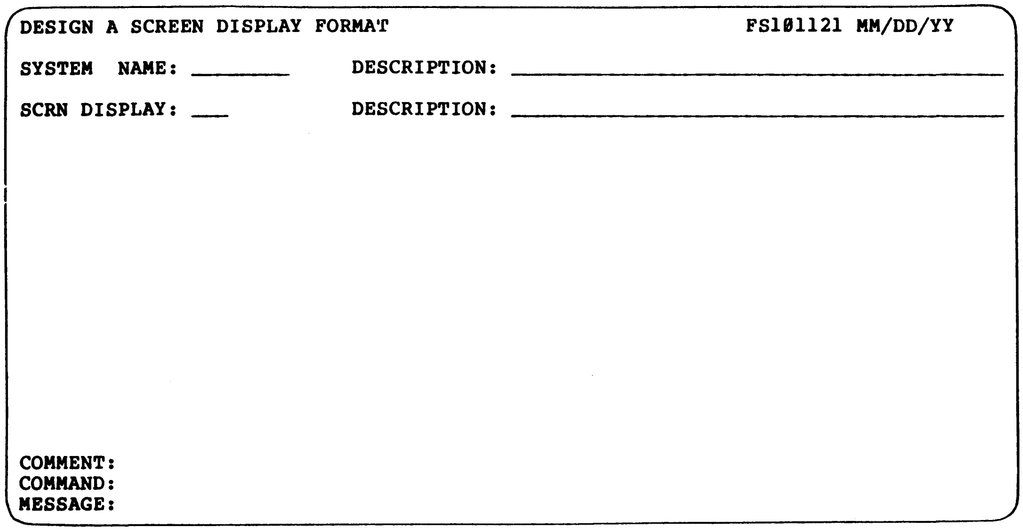
NOTES: Screen displays are referenced in the Dictionary by the associated system and user-assigned screen numbers. Thousands of screen displays may be designed for each system.

Screen displays for menu programs are created automatically by FORCE during menu program generation (see 2.2.1). Menu screens are then extracted from the source code and placed in the Dictionary (see 1.2.5).

1.2.1 DESIGN A SCREEN DISPLAY FORMAT

PURPOSE: Screen displays are designed for use in system programs by directing the cursor control keys and painting the desired screen format on the CRT. Formatted screen displays are assigned a number and stored in the Dictionary with their associated system. FORCE can then generate screen display source code by reterencing the image of a specified screen in the Dictionary.

SPECIAL: Design a Screen Display Format combines two operations. Both operations are documented in this section, each with a PROCEDURE and NOTES entry. The operations are presented as they occur during program execution.



```
DESIGN A SCREEN DISPLAY FORMAT                              FS101121 MM/DD/YY

SYSTEM  NAME: ________     DESCRIPTION: __________________________________

SCRN DISPLAY: ___          DESCRIPTION: __________________________________














COMMENT:
COMMAND:
MESSAGE:
```

PROCEDURE: Enter the name of the system with which the screen display is to be associated. At the Screen Display entry field, specify a three-character screen display number (the first character must be numeric, the second and third are alphanumeric). Then enter a description of the associated program, not the screen itself.

Upon entry of this information, control is transferred to a blank design screen for display formatting.

NOTES: Before entry at the Screen Display field, the user may want to establish a system screen numbering convention (see 8.2.2).

PROCEDURE: After control is transferred from the first screen, a blank formatting screen is presented. The program is in Screen Formatting Mode, as indicated on the Comment Line at the bottom of the screen. While in Screen Formatting Mode, use the cursor control keys to position the cursor at the desired location and enter information exactly as it is to appear on the screen display.

Press RETURN after each completed line entry. The first RETURN updates the cursor position status and enters the previous data string into memory. A second RETURN transfers the cursor to the first position of the next line, then updates the cursor position.

The cursor position is noted on the Command Line in two ways: POSITION-> indicates the horizontal position of the cursor and LINE-> denotes the vertical line position. This information is used in line modification commands. It also informs the user of the exact coordinates for the designed screen display.

An ESCAPE transfers control to Command Input Mode, as indicated on the Comment Line. The cursor positions on the Command Line, where the user may enter commands which facilitate the screen formatting process. The commands are listed below:

Command	Activity
I	Insert characters within a line
D	Delete characters from within a line
E	Erase a line from the screen
M	Move one line to another line
C	Center a line on the screen
X	Extract screen and refresh display format
H	Help module
S	Save the screen in the Dictionary
A	Abort (used only in screen modification)
?	Command line summary information request

Table 1-1 gives procedural information on screen formatting commands.

The ESCAPE key is used to transfer control between Screen Formatting Mode and Command Input Mode. A RETURN also transfers control from Command Input Mode to Screen Formatting Mode.

NOTES: The Save command is the only way to exit this FORCE module.

When formatting data entry screens, it is not necessary to place underscores for input fields. FORCE automatically establishes underscores for the length of the maximum entry during program generation. By retaining underscores in a saved screen format, the user wastes eventual program space.

A " symbol (quotation mark) may not be used when formatting a screen with FORCE. It is recorded as a ' symbol (an apostrophe) in the Dictionary screen image.

In Screen Display Formatting Mode, it is possible to continue typing onto the next line without pressing RETURN, but the I/O buffer size is limited to 80 bytes. If the user fills the buffer without entering a RETURN, the Message Line displays an appropriate message and the cursor positions at the location where the buffer was filled. If this condition occurs, use the X command to refresh the screen.

FORCE automatically places Comment, Command and Message lines at the bottom of every screen. Within generated programs, this capability provides a uniform method of operator/system interaction.

Table 1-1 contains procedures for each of the activities available in Command Input Mode.

TABLE 1-1. SCREEN FORMATTING COMMANDS

Command	Function
A	(A)bort a screen modification Used only when modifying a screen format. To abort modification of a screen display: ● Enter A and RETURN at the Command Line. Control returns to the Screen Display Formatting Menu.
C	(C)enter a line on the screen ● Enter C while in Command Input Mode. FORCE prompts for number of the line to be centered. ● Enter the number of the line to be centered and RETURN. Control returns to Command Input Mode after the line is centered.
D	(D)elete characters from a line ● Position cursor at the desired deletion point. ● Press RETURN to set cursor position coordinates in the status information. ● Press ESCAPE to return to Command Input Mode. ● Enter D at the Command Line, then press RETURN to enter Delete Character Mode. Enter a second D, followed by a RETURN, to delete a character at the specified cursor coordinates. The remaining text on that line is moved back one character position. Additional D entries delete succeeding characters which have been moved into the specified cursor coordinate. ● Enter RETURN at the Command Line to return to Command Input Mode.

TABLE 1-1. SCREEN FORMATTING COMMANDS (Cont)

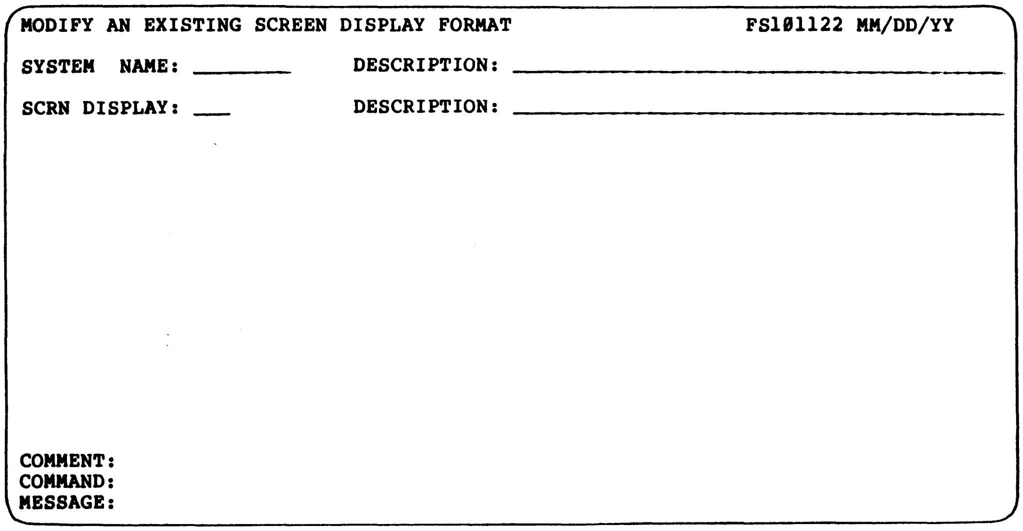
Command	Function
E	(E)rase an entire line from the screen ● Enter E and RETURN while in Command Input Mode. ● FORCE prompts for the line number to be deleted. Entry of a line number plus RETURN deletes the entire line specified and returns control to Command Input Mode. Entry of a RETURN only returns control to Command Input Mode without deleting a line.
H	(H)elp module To display a list of commands with an explanation of each function: ● Enter H and RETURN at the Command Line. The screen clears and a list of commands appears. ● Press RETURN to return to Command Input Mode.
I	(I)nsert characters within a line ● Position cursor at desired insert location. ● Press RETURN to set cursor position and line coordinates in the status information. ● Press ESCAPE to return to Command Input Mode. ● Enter I at the Command Line and press RETURN. Data between the cursor position and the end of the line disappears. ● Enter data to be inserted and press RETURN. The original line is appended to the end of the insertion.
M	(M)ove one line to another line ● Enter M while in Command Input Mode. FORCE prompts for the line to be moved, and the line on the screen where it is to be moved. ● Enter the prompted line numbers and RETURN. Control returns to Command Input Mode after the line is moved.

TABLE 1-1. SCREEN FORMATTING COMMANDS (Cont)

Command	Function
S	(S)ave a formatted screen To store the screen in the Dictionary, associated with the system name and screen number specified when the screen was created: ● Enter S and RETURN at the Command Line. Control returns to the Screen Display Formatting Menu.
X	Extract and refresh screen display To reinstate a screen's original format if it has been disorganized by overrun of screen dimensions when using up-and-down arrows, or by other operator input error: ● Press RETURN before leaving Screen Formatting Mode to save completed work and prevent loss of additions or modifications. ● Use the ESCAPE key to enter Command Input Mode. ● Enter X and RETURN at the Command Line. The screen clears and the correct screen format is displayed. Control returns to Command Input Mode.
?	Command line summary For a quick reference summary of the commands: ● Enter ? and RETURN at the Command Line. The following message appears: THE ONLY VALID ENTRIES ARE I, D, E, C, M, X, H, S OR ? Control returns to Command Input Mode.

1.2.2 MODIFY AN EXISTING SCREEN DISPLAY

PURPOSE: This function allows the user to modify a screen display in the Dictionary. All subsequent use of the screen within FORCE-generated programs reflects the changes.



```
MODIFY AN EXISTING SCREEN DISPLAY FORMAT                FS1Ø1122 MM/DD/YY

SYSTEM  NAME: ________     DESCRIPTION: __________________________________

SCRN DISPLAY: ___          DESCRIPTION: __________________________________










COMMENT:
COMMAND:
MESSAGE:
```

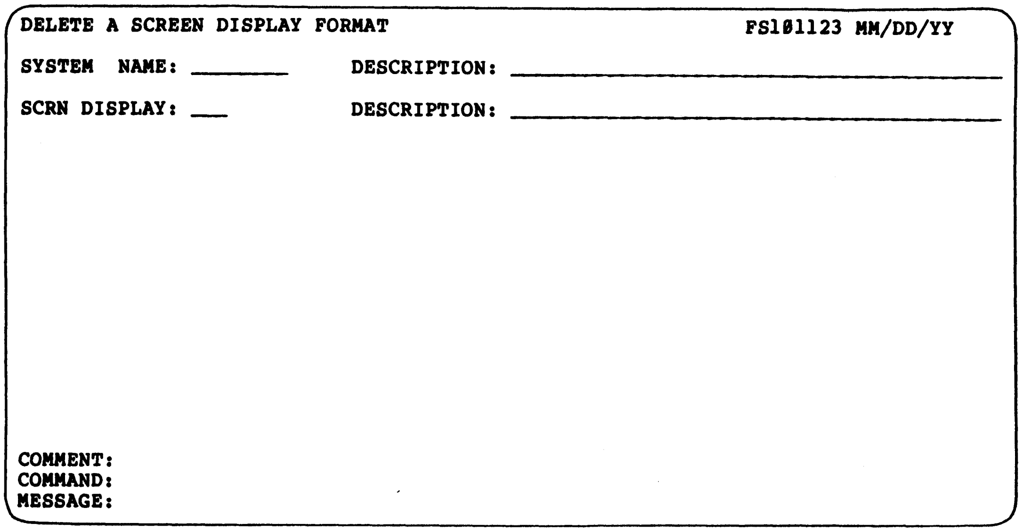
PROCEDURE: Enter the name of the system with which the screen is associated. Then enter the screen display number. Upon completion of these initial inputs, FORCE retrieves the specified screen and displays it on the CRT.

The cursor positions in the upper left hand corner of the screen and the user can make the desired modifications. Any of the FORCE screen formatting functions (described in 1.2.1) may be used to modify the screen. New lines may be added and existing lines may be deleted or changed.

There are two ways to exit this FORCE module. The Save command places the modified screen in the Dictionary, overwriting the original. The Abort command allows the user to abandon the current modification and leave the original screen unchanged in the Dictionary.

1.2.3 DELETE A SCREEN FROM THE DICTIONARY

PURPOSE: A system screen display that is no longer required may be deleted from the Dictionary.

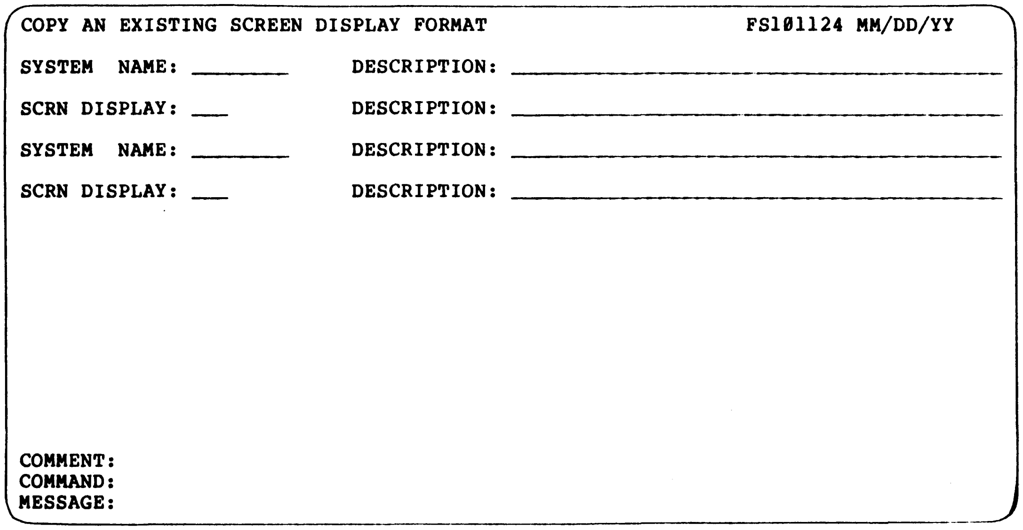


```
DELETE A SCREEN DISPLAY FORMAT                              FS101123 MM/DD/YY

SYSTEM  NAME: _______        DESCRIPTION: ______________________________________

SCRN DISPLAY: ___            DESCRIPTION: ______________________________________










COMMENT:
COMMAND:
MESSAGE:
```

PROCEDURE: Enter the name of the system with which the screen is associated. Then enter the screen display number.

1.2.4 COPY A SCREEN DISPLAY FORMAT

PURPOSE: FORCE allows copying of screen displays within a system and between systems. This saves the time of designing a new screen display for each function. Often a screen that is already in the Dictionary can be copied, modified, and used for another program. This function provides the facility to modify a copied screen.



```
COPY AN EXISTING SCREEN DISPLAY FORMAT                    FS1Ø1124 MM/DD/YY

SYSTEM  NAME: ________      DESCRIPTION: ______________________________

SCRN DISPLAY: ___           DESCRIPTION: ______________________________

SYSTEM  NAME: ________      DESCRIPTION: ______________________________

SCRN DISPLAY: ___           DESCRIPTION: ______________________________









COMMENT:
COMMAND:
MESSAGE:
```

PROCEDURE: At the first System Name field, specify the name of the system from which the screen is to be copied. Then enter the number of the screen display format. These entries constitute the source screen display information.

The next System Name field represents the name of the system to which the screen is to be copied. It may be the same as the source system, or any other system in the Dictionary. At the second Screen Display prompt, assign a screen display number to the new, copied screen. These entries constitute the destination screen display information.

After completion of these initial entries, FORCE clears the screen, then displays the copied screen. The user can make any required modifications to the copied screen, so that it accommodates the new function. All screen design activities (see 1.2.1) may be used in modification of the copied screen.

When modification is complete, enter S in Command Input Mode to save the copied screen. Even if no modifications are required, the copied screen must be saved.

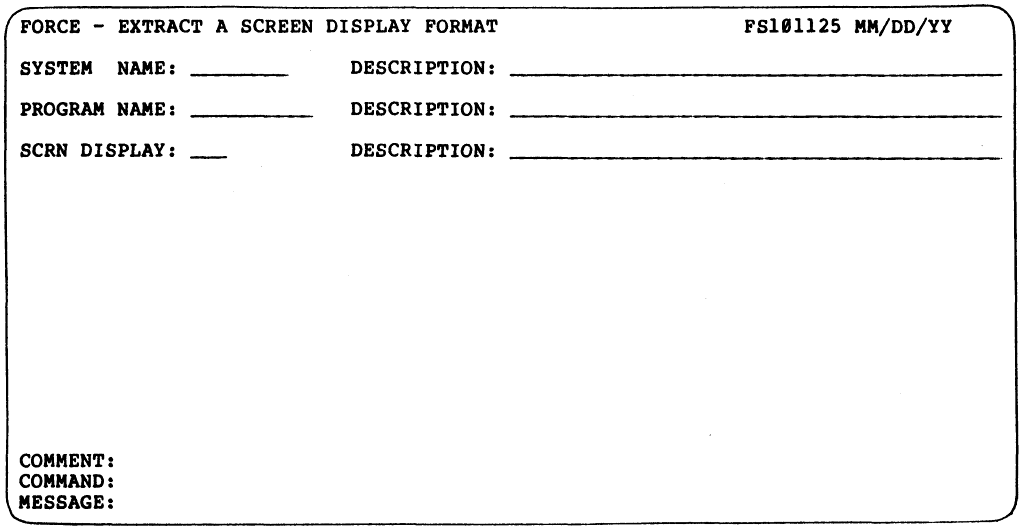
NOTES: The source screen display is not changed by the copying process.

An existing screen display cannot be over-copied.

1.2.5 EXTRACT A SCREEN DISPLAY FORMAT

PURPOSE: With this facility the user directs FORCE to read through a program's source code, extract the image of the screen it uses, and place that screen image in the Dictionary. The screen display can then be modified, copied or printed from the Dictionary.

This function is generally used to retrieve screen displays for FORCE-generated menu programs. Those screens must be extracted because they are created automatically during menu program generation. They are not designed and saved like other FORCE screens (see 1.2.1). Screens from hand-written programs can also be extracted.



```
FORCE - EXTRACT A SCREEN DISPLAY FORMAT                    FS101125 MM/DD/YY

SYSTEM  NAME: ________      DESCRIPTION: ________________________________

PROGRAM NAME: __________    DESCRIPTION: ________________________________

SCRN DISPLAY: ___           DESCRIPTION: ________________________________









COMMENT:
COMMAND:
MESSAGE:
```

PROCEDURE: Prior to extracting a screen display, the associated program must be defined to the Dictionary (see 1.5.1).

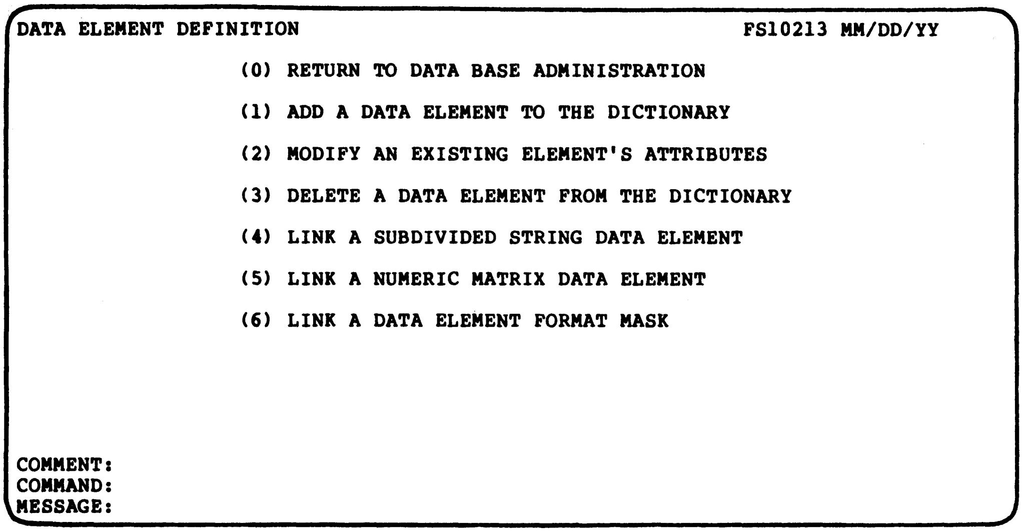
Enter the name of the system with which the program is associated. At the Program Name field, enter the name of the program (as defined to FORCE) that contains the screen. Assign the screen display number at the Screen Display prompt and enter a description.

NOTES: FORCE extracts the image of the specified screen display and places it in the Dictionary. This process involves decoding the screen display source code and storing the screen image - not the code - in the Dictionary.

In order for screens to be extracted from hand-written programs, the source code which creates the screen must follow the structure of similar FORCE-generated code. Specifically, there must be a BASIC "clear-screen" statement (PRINT 'CS') before the initial screen display print statement.

1.3 DATA ELEMENT DEFINITION

PURPOSE: Data Element Definition is the process by which the user specifies system variables and their characteristics. This information is stored in the Dictionary and used during source code generation, or for system documentation. System data element maintenance functions are provided from this menu.



```
DATA ELEMENT DEFINITION                                    FS10213 MM/DD/YY

                          (0) RETURN TO DATA BASE ADMINISTRATION

                          (1) ADD A DATA ELEMENT TO THE DICTIONARY

                          (2) MODIFY AN EXISTING ELEMENT'S ATTRIBUTES

                          (3) DELETE A DATA ELEMENT FROM THE DICTIONARY

                          (4) LINK A SUBDIVIDED STRING DATA ELEMENT

                          (5) LINK A NUMERIC MATRIX DATA ELEMENT

                          (6) LINK A DATA ELEMENT FORMAT MASK




COMMENT:
COMMAND:
MESSAGE:
```

PROCEDURE: Select and enter the number which corresponds to the desired function.

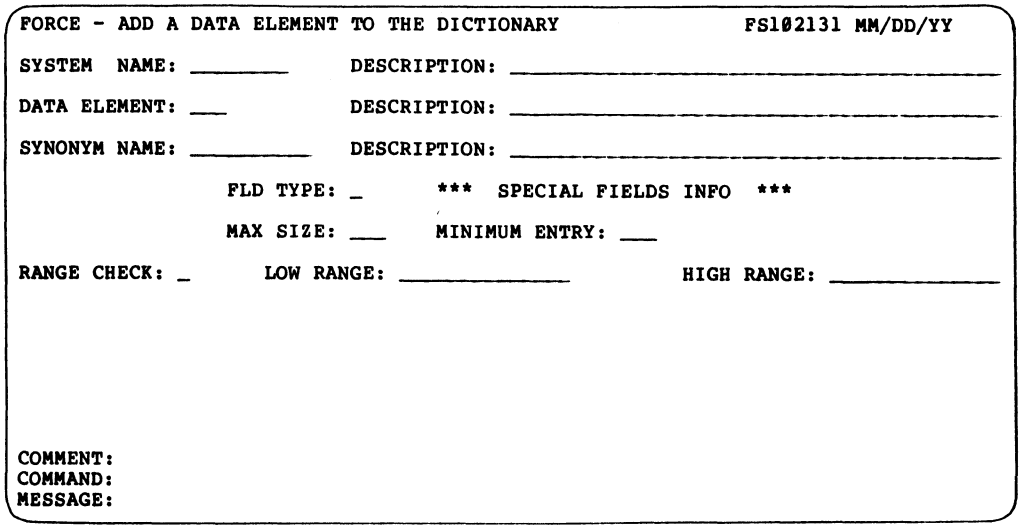
NOTES: Each data element used in FORCE must be a valid IRIS Business BASIC variable.

User-defined data elements may not begin with the letter I or the letter O. These letters, and variations of these letters are used by FORCE as global variables in generated source code.

1.3.1 ADD A DATA ELEMENT TO THE DICTIONARY

PURPOSE: Each system data element and its physical and logical attributes must be defined to the Dictionary. FORCE retrieves this information during source code generation any time the variable is used. Data element specifications allow FORCE to dimension, document and set up edit checks for data elements in the generated source code.

Once in the Dictionary, data elements may be used in different programs and in different systems. Complete data element documentation is also available.



```
FORCE - ADD A DATA ELEMENT TO THE DICTIONARY                 FS1Ø2131 MM/DD/YY

SYSTEM  NAME: ________     DESCRIPTION: ________________________________________

DATA ELEMENT: ___          DESCRIPTION: ________________________________________

SYNONYM NAME: __________   DESCRIPTION: ________________________________________

                  FLD TYPE: _       ***  SPECIAL FIELDS INFO  ***

                  MAX SIZE: ___     MINIMUM ENTRY: ___

RANGE CHECK: _     LOW RANGE: ______________        HIGH RANGE: ______________




COMMENT:
COMMAND:
MESSAGE:
```

PROCEDURE: Enter the name of the system with which the data element is to be associated. Then enter a valid IRIS Business BASıC variable name and a description.

At the Synonym Name field, enter an alphanumeric synonym of ten or fewer characters and a description. The synonym will be related to the data element within the Dictionary. Synonyms allow language-oriented reference to data elements and are used in FORCE Report Preparation.

Only alphanumeric and numeric data elements may be assigned synonyms with this function. String subdivisions and matrix cells must be linked (see 1.3.4 and 1.3.5) before synonym assignment (see 4.5.1).

The Field Type entry registers the data element type within the Dictionary. Valid field types are: (A)lphanumeric, (N)umeric, (M)atrix and (S)ubdivided String.

Input of ? at the Field Type entry presents a field type summary.

(See NOTES for more information on field types).

Entry of (A)lphanumeric or (N)umeric invokes the Default Value entry field. The user may specify a default value for the data element.

A (M)atrix or (S)ubdivided String field type requires entry of Special Fields Information regarding the physical size of the data element.

For a matrix, enter the cell coordinates in the form:

    rrXcc

where rr represents the number of rows and cc represents the number of columns. The X is a delimiter denoting the relationship between coordinates, as in 2X4 (two by four). If the matrix is a single-dimension array, enter only the number of rows.

Entry of a Subdivided String field type requires specification of the number of subdivisions within the data element.

At the Maximum Size entry field, specify the largest number of characters that the data element may contain. For a subdivided string, this entry represents the size of the entire string. For a matrix, it establishes the maximum size of each cell. A RETURN at the Maximum Size field defaults to one.

The Minimum Entry field establishes the minimal number of characters accepted during entry of the data element value. Specify the minimum entry for the data element. A RETURN at the Minimum Entry field defaults to ANY.

At the Range Check prompt, enter (Y)es or (N)o to indicate whether a range check edit should be performed on the data element. If Y is entered, specify the Low Range and the High Range values. A High Range can contain a maximum of 14 alphanumeric characters, and must be equal to or greater than the Low Range parameter. A RETURN at the Range Check field defaults to (N)o.

Range checks for subdivided strings are performed during Subdivided String Linkage (see 1.3.4).

NOTES: Additional information about each data element type is provided below. See Table 1-2 for data element set-up information.

- An Alphanumeric data element is a string variable.

- A Numeric data element holds a number.

- A Matrix is a one or two-dimensional array of data expressed as a sequence of numbers or organized in rows and columns. Matrix cell specifications are established in Link a Numeric Matrix Data Element (see 1.3.5).

- A Subdivided String data element is a physical string variable with logical field subdivisions. Subdivision attributes are specified in Link a Subdivided String Data Element (see 1.3.4).

Data element names and synonym names must be unique within each system. The same data element names or synonym names may be used in different systems.

User-defined data elements may not begin with the letter I or the letter O. These letters, and variations of these letters, are used by FORCE as global variables in generated source code.

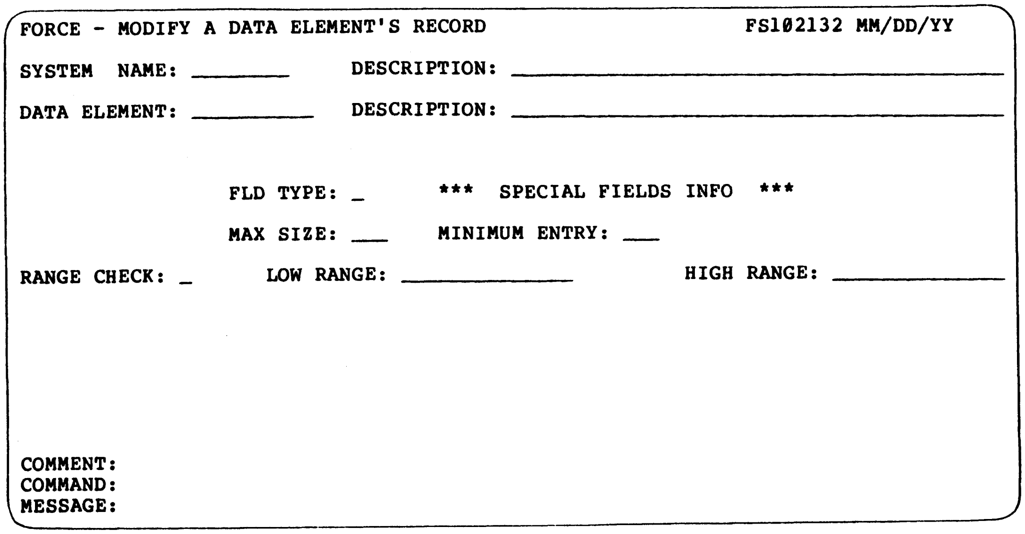
TABLE 1-2. DATA ELEMENT INFORMATION

Entry	Field Type	Variable Type	Size	Synonym Assign-ment
A	Alphanumeric	String	1-999 Bytes*	Yes
N	Numeric	Number	1-14 places (with optional sign & decimal)	Yes
S	Subdivided String	String	1-256 Subdivisions*	No
M	Matrix or Array	Number	Matrix (1-99x1-99 cells) Array (1-1000 cells)	No

*In generated programs, a string or string subdivision may be printed or input with a maximum of 65 characters.

1.3.2 MODIFY AN EXISTING ELEMENT'S ATTRIBUTES

PURPOSE: This function allows modification of the attributes of a data element in the Dictionary. Subsequent use of the data element reflects the modifications.



```
FORCE - MODIFY A DATA ELEMENT'S RECORD                      FS1Ø2132 MM/DD/YY

SYSTEM  NAME: ________    DESCRIPTION: ________________________________________

DATA ELEMENT: __________  DESCRIPTION: ________________________________________


                FLD TYPE: _      ***  SPECIAL FIELDS INFO  ***

                MAX SIZE: ___    MINIMUM ENTRY: ___

RANGE CHECK: _     LOW RANGE: ______________        HIGH RANGE: ______________





COMMENT:
COMMAND:
MESSAGE:
```

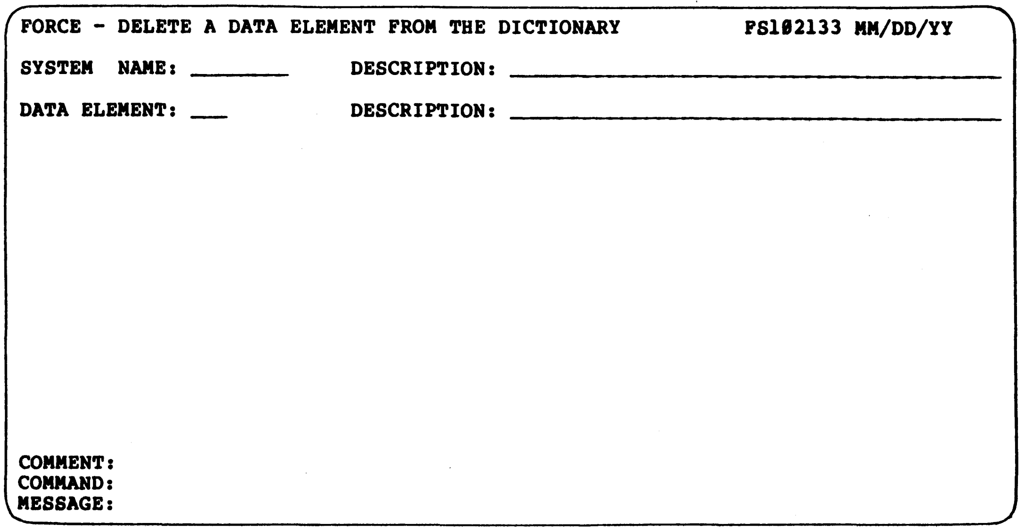
PROCEDURE: Enter the system name with which the data element is associated. Then enter the name of the data element.

FORCE retrieves the data element's attribute information and displays it on the screen. A new description may be entered, or the original description may be retained by pressing RETURN.

Each RETURN progresses to the next data entry field, where current information may be modified. If no modification is required at an entry field, press RETURN to proceed to the next.

1.3.3 DELETE A DATA ELEMENT FROM THE DICTIONARY

PURPOSE: This function allows the user to remove a data element from the Dictionary.



```
FORCE - DELETE A DATA ELEMENT FROM THE DICTIONARY          FS1Ø2133 MM/DD/YY

SYSTEM  NAME: ________     DESCRIPTION: ________________________________________

DATA ELEMENT: ___          DESCRIPTION: ________________________________________










COMMENT:
COMMAND:
MESSAGE:
```

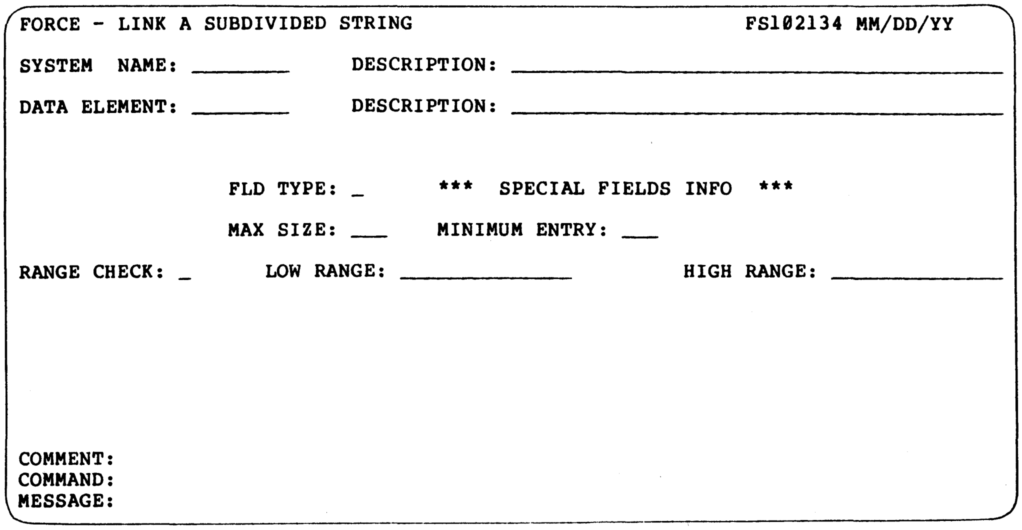
PROCEDURE: Enter the system name with which the data element is associated. Then enter the name of the data element.

NOTES: Execution of this function invalidates all Dictionary references to the specified data element. For example, if the data element were part of a record layout, that record layout is no longer valid.

1.3.4 LINK A SUBDIVIDED STRING DATA ELEMENT

PURPOSE: After a subdivided string data element is defined to the Dictionary (see 1.3.1), the user must specify the attributes of each subdivision.

This function may also be used to modify the attributes of a subdivision.



```
FORCE - LINK A SUBDIVIDED STRING                              FS1Ø2134 MM/DD/YY

SYSTEM  NAME: ________      DESCRIPTION: ________________________________________

DATA ELEMENT: ________      DESCRIPTION: ________________________________________


                FLD TYPE: _      ***  SPECIAL FIELDS INFO  ***

                MAX SIZE: ___    MINIMUM ENTRY: ___

RANGE CHECK: _    LOW RANGE: ______________        HIGH RANGE: ______________






COMMENT:
COMMAND:
MESSAGE:
```

PROCEDURE: Enter the name of the system with which the data element is associated.

At the Data Element entry field, enter the name of the subdivided string's first subdivision. Use the format:

    X$(1)

where X$ is the name of the subdivided string variable and (1) represents the first subdivision. For subsequent subdivisions, a RETURN at the Data Element entry field automatically increments the subdivision by one.

Enter a description of the data element subdivision. This entry should describe the particular data in the subdivision.

An alphanumeric Field Type is displayed, and Maximum Size, Minimum Entry and Range Checks may be specified for the subdivision. This process is repeated for each subdivision.

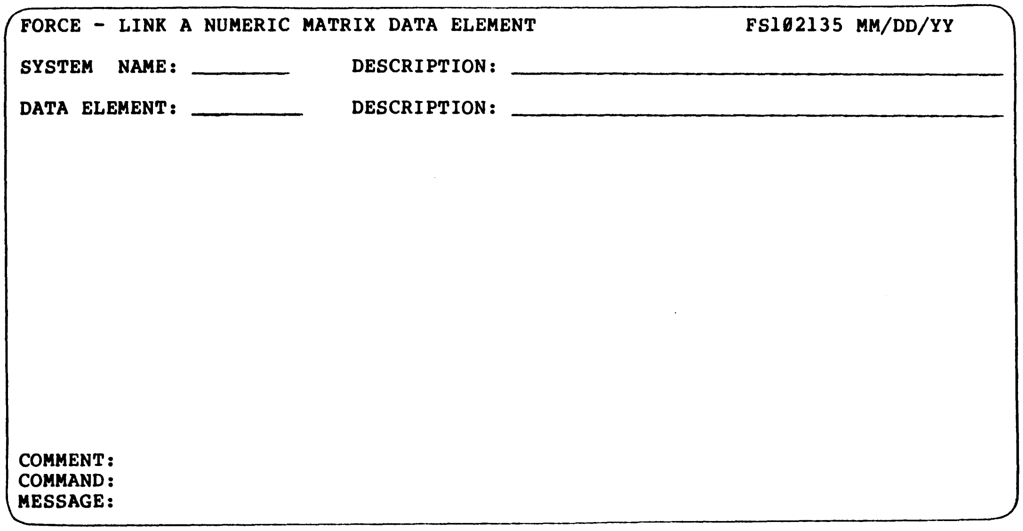
If the maximum size specifed for any subdivision exceeds the total size of the string (when the other subdivisions are included) FORCE generates an error message. The message indicates the number of bytes exceeded by the invalid specification.

To modify the attributes of a linked subdivided string, enter the name of the data element and the subdivision to be modified. Its attributes are displayed on the screen. Each entry field may be modified, or left intact by a RETURN.

1.3.5 LINK A NUMERIC MATRIX DATA ELEMENT

PURPOSE: This function enables the user to specify a description for any cell of a numeric matrix. This information is used in system documentation to denote the use of each matrix cell. It also provides the data necessary to create explicit error messages within the generated source code.

The user may also use this function to modify the description of a matrix cell that was previously linked.



```
FORCE - LINK A NUMERIC MATRIX DATA ELEMENT                FS1Ø2135 MM/DD/YY

SYSTEM  NAME: ________      DESCRIPTION: ______________________________

DATA ELEMENT: _________     DESCRIPTION: ______________________________










COMMENT:
COMMAND:
MESSAGE:
```

PROCEDURE: Enter the name of the system associated with the numeric matrix to be linked.

At the Data Element field, enter the name and coordinates of the matrix cell to be linked. For a standard matrix, use the format:

    M(rrXcc)

where M is the data element name and (rrXcc) represents the row and column coordinates of the matrix cell.

For a single-dimension array, use the format:

    M(nnnn)

where M is the data element name and (nnnn) represents the cell of the array.

Enter a description for the matrix cell. This description provides reterence to the cell and should specify its contents.

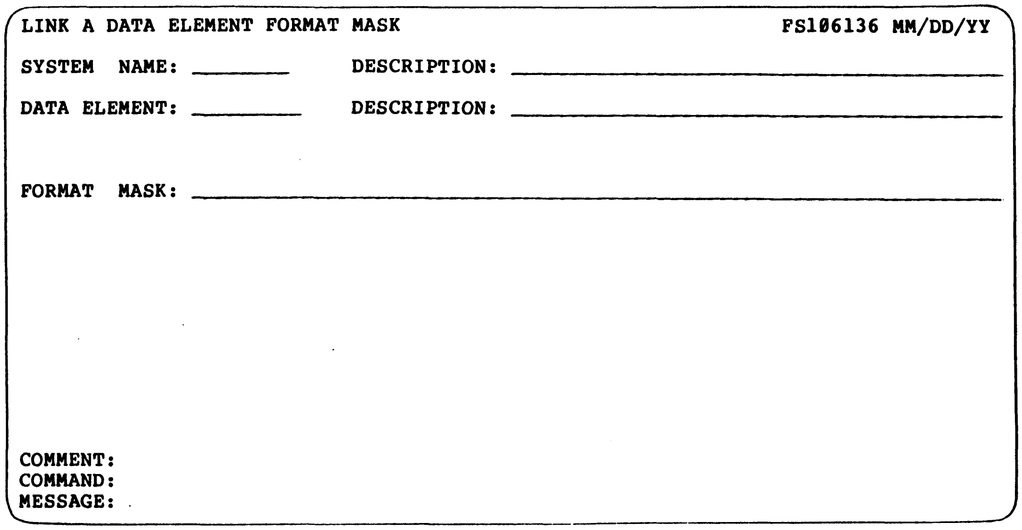
Repeat this process for all cells for which individual reference is desired.

1.3.6 LINK A DATA ELEMENT FORMAT MASK

LINK ELEMENT
FORMAT MASK
1.3.6

PURPOSE: This function establishes a format mask for redisplay of a data element input. Any displayable keyboard characters (except the @ symbol) may be included before, within or after an input. Format masks structure the appearance of data input to enhance interaction.

This function may also be used to modify an established format mask.



```
LINK A DATA ELEMENT FORMAT MASK                              FS1Ø6136 MM/DD/YY

SYSTEM  NAME: _______      DESCRIPTION: ____________________________________

DATA ELEMENT: ________     DESCRIPTION: ____________________________________


FORMAT  MASK: ______________________________________________________________








COMMENT:
COMMAND:
MESSAGE:
```

PROCEDURE: Prior to execution of this function, the affected data element must be defined to the Dictionary. String subdivisions and matrix cells must be linked before masking is performed (see 1.3.4 and 1.3.5).

Enter the name of the system with which the data element is associated. Then enter the data element name.

At the Format Mask entry field, specify the format of the displayed data element value. The @ symbol designates each print position of the represented data. Any displayable keyboard character may be included before, within or following the associated data.

For example, consider a 6-character, numeric data element which stores a date. A format mask could be established using the format:

```
@@/@@/19@@
```

A data entry of 020180 would be automatically re-displayed as:

```
02/01/1980
```

When the format mask is established, FORCE places a | symbol to denote the end of the mask.

Field formatting entry fields are then displayed. Field formatting operations allow further structuring of the data element display. These operations differ for alphanumeric and numeric fields.

For alphanumeric data elements, the user may specify a Justify Mode of (L)eft, (R)ight or (N)one. A RETURN at the Justify Mode entry defaults to (N)one, which redisplays the data exactly as it is input.

At the Filler Character entry field, specify any keyboard character to fill the length of the element when current data occupies only a portion of the specified length. No filler is used if a RETURN is pressed at the Filler Character entry field.

The Float Character entry allows the user to specify a character to be printed immediately before numeric data, regardless of its length. Enter RETURN at the Float Character field if no float character is required.

When modifying a data element format mask, enter the name of the associated system and the data element name. The current masking information is displayed on the screen. At each field, the information may be left intact by a RETURN, or new information may be entered.

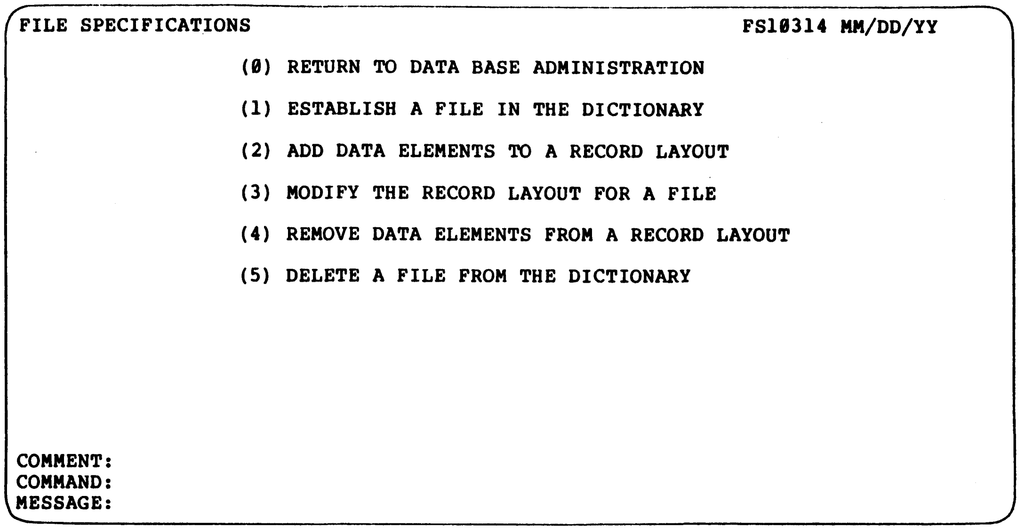
NOTES: The established format mask is used to redisplay the data element value in any data entry program.

Data is written to disk without the mask.

FILE
SPECIFICATIONS
1.4

1.4 FILE SPECIFICATIONS

PURPOSE: System files are established in the Dictionary by assigning file names and descriptions. File record layouts are constructed by designating the data elements that the file is to contain. This menu provides functions to establish and maintain system data files and index files.



```
FILE SPECIFICATIONS                                              FS10314 MM/DD/YY

                       (0) RETURN TO DATA BASE ADMINISTRATION

                       (1) ESTABLISH A FILE IN THE DICTIONARY

                       (2) ADD DATA ELEMENTS TO A RECORD LAYOUT

                       (3) MODIFY THE RECORD LAYOUT FOR A FILE

                       (4) REMOVE DATA ELEMENTS FROM A RECORD LAYOUT

                       (5) DELETE A FILE FROM THE DICTIONARY






COMMENT:
COMMAND:
MESSAGE:
```

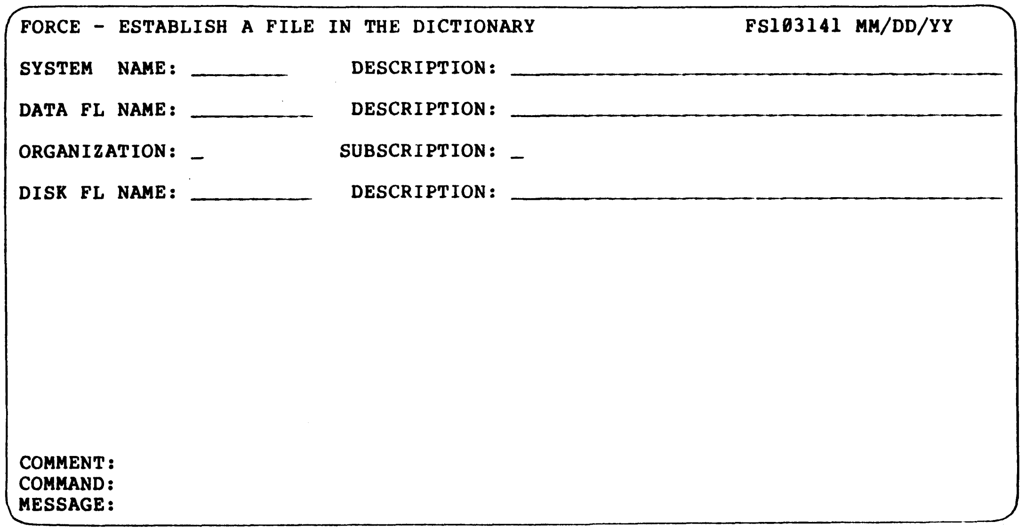
PROCEDURE: Select and enter the number which corresponds to the desired function.

NOTES: Both data files and index files are maintained by these functions. For index files, references to record layouts may be interpreted as key construct information.

1.4.1 ESTABLISH A FILE IN THE DICTIONARY

PURPOSE: This function establishes a file within the Dictionary. The user specifies the file name, description, and the system with which the file is associated. File characteristics are defined, and the associated IRIS disk file name is denoted.

This function may also be used to modify an established file's information.



```
FORCE - ESTABLISH A FILE IN THE DICTIONARY                FS103141 MM/DD/YY

SYSTEM  NAME: ________    DESCRIPTION: ________________________________________

DATA FL NAME: __________  DESCRIPTION: ________________________________________

ORGANIZATION: _          SUBSCRIPTION: _

DISK FL NAME: __________  DESCRIPTION: ________________________________________









COMMENT:
COMMAND:
MESSAGE:
```

PROCEDURE: Enter the name of the system with which the file is to be associated. At the Data File Name field, enter the name and description of the file for reference in the Dictionary (see Section 8.1.2 for index file descriptions). A file name must begin with a letter and may not exceed ten characters.

At the Organization entry field, indicate the organization of the file structure (see Table 1-3). Valid entries are: (C)ontiguous, (F)ormatted, (T)ext, (I)ndexed and (D)isk. A default (RETURN) from this entry field results in a (F)ormatted entry.

Entry of ? at the Organization entry field presents a summary of the valid entries.

If the organization specification is C, F or T, FORCE prompts for a subscription entry. Enter (Y)es or (N)o at the Subscription entry field to indicate whether the file is to have double subscription on string variables for all file I/O statements. A RETURN defaults to (Y)es.

If the organization is (I)ndexed, FORCE prompts for specification of the directory number. Enter the directory number.

Disk File Name designates the actual name of the associated IRIS file on disk. This entry field accommodates disk file structures which contain multiple FORCE files, and specifies the file name to be used in any generated source code.

For example, a disk file may be logically apportioned to contain index directories and data records. But in order to specify the record layouts and key constructs to FORCE, each portion must be defined as a separate file within the Dictionary. The Disk File Name represents the name of the file which actually contains the FORCE files. The Disk File Name is the IRIS file to be accessed in the generated source code.

Enter the associated Disk File Name and its description. A RETURN at this field causes the Data File Name entry to be used. Use RETURN when a multiple file structure is not used.

To modify the information for a previously-established file, enter the associated system and file name. The current file information is displayed. At each field, new information may be entered or current information may be left intact by pressing RETURN.

NOTES: If a multiple FORCE file structure is to be used, the Disk File must be the first file defined to FORCE.

Table 1-3 provides additional information on file organization.

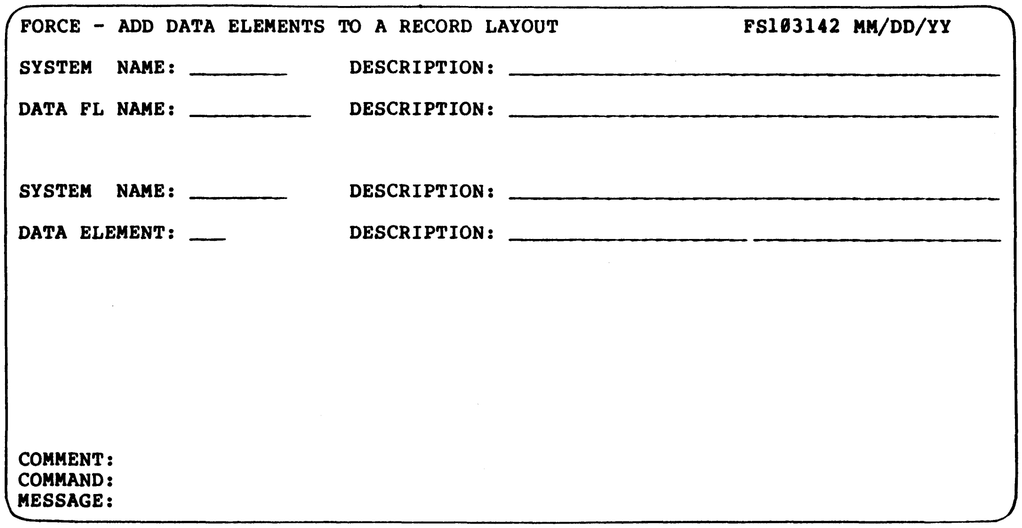
The file organization specified in function 1.4.1 describes the FORCE file structure to be used for subsequent record layout definition. It is not necessarily the organization of the file that will be generated on disk. The organization of the generated IRIS disk file is established during file initialization (see 2.3).

TABLE 1-3. FILE ORGANIZATIONS

Organization	Function
(C)ontiguous	Specify a contiguous organization for a FORCE file which represents the data record portion of an indexed contiguous file, or the data record portion of a contiguous data file.
(F)ormatted	Specify a formatted organization if the file is to contain data records which are structured identically.
(T)ext	Specify a text file if the file record layout is to contain a single string of variable length.
(I)ndexed	Specify an indexed organization when the file contains the key construction for an indexed access. Select an indexed organization even though the actual disk file, which contains the directories, is to be contiguous.
(D)isk	Specify a disk organization if the file being established is the name of the IRIS file to be generated. Use the disk specification regardless of the actual organizational structure of the file. A disk file may contain multiple FORCE files, and its IRIS organization is specified during file initialization (see 2.3).

1.4.2 ADD DATA ELEMENTS TO A RECORD LAYOUT

PURPOSE: This function enables the user to specify the data elements within a file and the order of their appearance. It may also be used to add data elements to established record layouts. The information is stored in the Dictionary and used for file I/O.



```
FORCE - ADD DATA ELEMENTS TO A RECORD LAYOUT                FS1Ø3142 MM/DD/YY

SYSTEM  NAME: ________     DESCRIPTION: _______________________________

DATA FL NAME: __________   DESCRIPTION: _______________________________


SYSTEM  NAME: ________     DESCRIPTION: _______________________________

DATA ELEMENT: ___          DESCRIPTION: _______________________________







COMMENT:
COMMAND:
MESSAGE:
```

PROCEDURE: Prior to executing this function, the affected file must have been established in the Dictionary (see 1.4.1). Additionally, the associated data elements must have been established (see 1.3.1).

Enter the name of the system with which the file is associated. Then enter the name of the data file for which the record layout is to be established.

At the second System Name entry field, enter the name of the system with which the data element is associated. This information is necessary because data elements from different systems may be used.

Data elements are established within a record layout in the order in which they are entered. Consequently, data elements added to an established record layout follow, in sequence, all previously specified elements.

Specify the data element to be established within the record layout.

Repeat this process until all required data elements are added to the record layout. An ESCAPE at the Data Element field indicates that the procedure is completed.

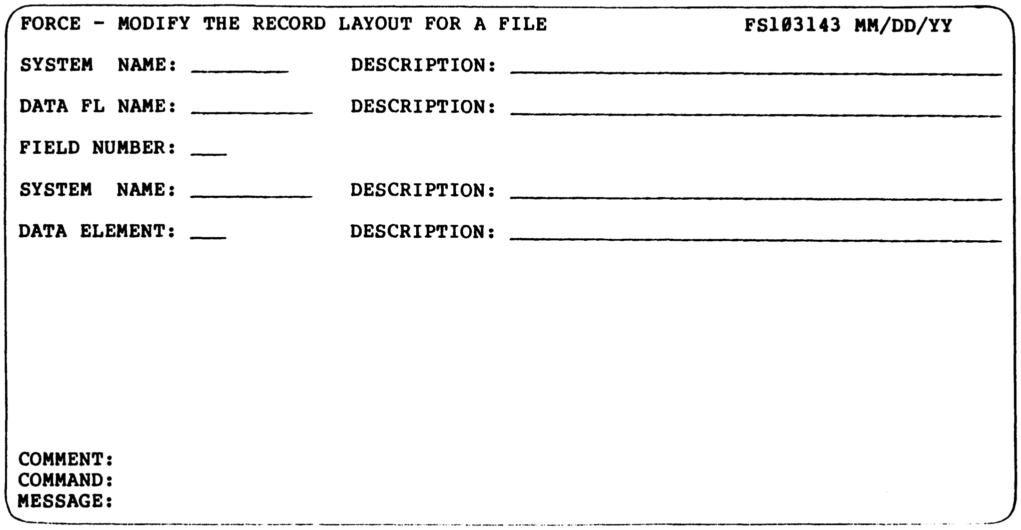
<u>NOTES</u>: It is a two-step process to insert additional data elements in the middle or beginning of a record layout. The elements must be added to the record layout with this function. Then, the record layout must be modified (see 1.4.3) to reflect the desired order.

For the purpose of Dictionary reference, the first data element position within a record layout is field zero. The next position is field one, then field two, etc.

1.4.3 MODIFY THE RECORD LAYOUT FOR A FILE

MODIFY
RECORD LAYOUT
1.4.3

PURPOSE: This function allows the user to modify a file's record layout by substituting different data elements. Data elements can be replaced or rearranged within the record layout.



```
FORCE - MODIFY THE RECORD LAYOUT FOR A FILE                FS1Ø3143 MM/DD/YY

SYSTEM  NAME: _______       DESCRIPTION: ________________________________

DATA FL NAME: _________     DESCRIPTION: ________________________________

FIELD NUMBER: ___

SYSTEM  NAME: _________     DESCRIPTION: ________________________________

DATA ELEMENT: ___           DESCRIPTION: ________________________________








COMMENT:
COMMAND:
MESSAGE:
```

PROCEDURE: Enter the name of the system with which the affected file is associated. Then enter the name of the file.

For the Field Number entry, specify the number which represents the data element's position within the record layout. The field number reflects the sequential order (starting at zero) of the data element in the layout. Data element field information may be obtained through Documentation Services (see 3.2.4).

Completion of these entries causes the current data element field information to be displayed. It may be modified or left intact by pressing RETURN.

The second System Name entry field is for specification of the system associated with a substituted data element.

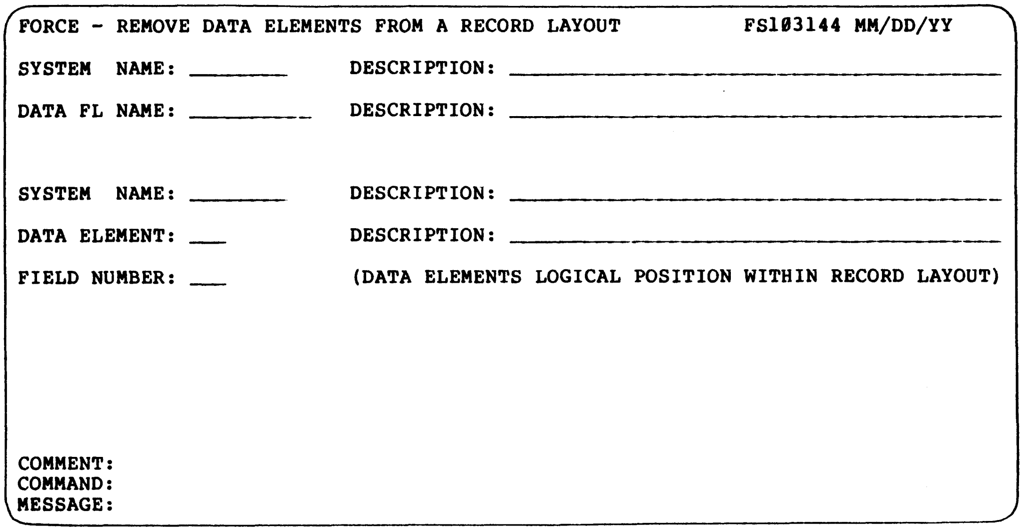
At the Data Element entry field, enter the name of the data element that is to replace the existing data element.

Use the same procedure to rearrange data elements within a file's record layout.

NOTES: To insert a data element at the beginning or in the middle of a record layout, first add the data element (see 1.3.1), then modify the record layout order with this function.

1.4.4 REMOVE DATA ELEMENTS FROM A RECORD LAYOUT

PURPOSE: This function removes data elements from a file's record layout and adjusts the record layout accordingly.



```
FORCE - REMOVE DATA ELEMENTS FROM A RECORD LAYOUT         FS1Ø3144 MM/DD/YY

SYSTEM  NAME: ________     DESCRIPTION: ________________________________________

DATA FL NAME: __________   DESCRIPTION: ________________________________________


SYSTEM  NAME: ________     DESCRIPTION: ________________________________________

DATA ELEMENT: ___          DESCRIPTION: ________________________________________

FIELD NUMBER: ___          (DATA ELEMENTS LOGICAL POSITION WITHIN RECORD LAYOUT)







COMMENT:
COMMAND:
MESSAGE:
```

PROCEDURE: Enter the system with which the affected file is associated. Then enter the name of the file from which data elements are to be removed.

At the second System Name entry field, specify the system associated with the data element to be removed. Then enter the data element.

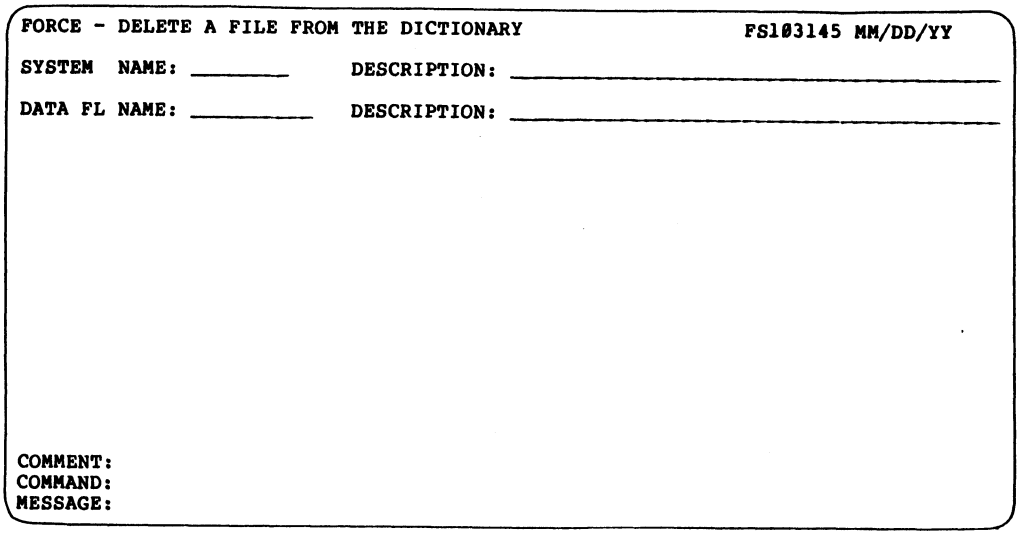
Field Number represents the order of the data element's position in the record layout. This number is relative to zero (the first data element position is zero). Specify number of the data element to be removed.

NOTES: This function compensates for a removed data element by moving up subsequent fields in the record layout. For example, if field two is deleted, the field that follows becomes field two in all subsequent references.

DELETE FILE 1.4.5

1.4.5 DELETE A FILE FROM THE DICTIONARY

PURPOSE: This function allows deletion of a specified file from the Dictionary.



```
FORCE - DELETE A FILE FROM THE DICTIONARY                        FS103145 MM/DD/YY

SYSTEM  NAME: ________      DESCRIPTION: ________________________________________

DATA FL NAME: __________    DESCRIPTION: ________________________________________










COMMENT:
COMMAND:
MESSAGE:
```

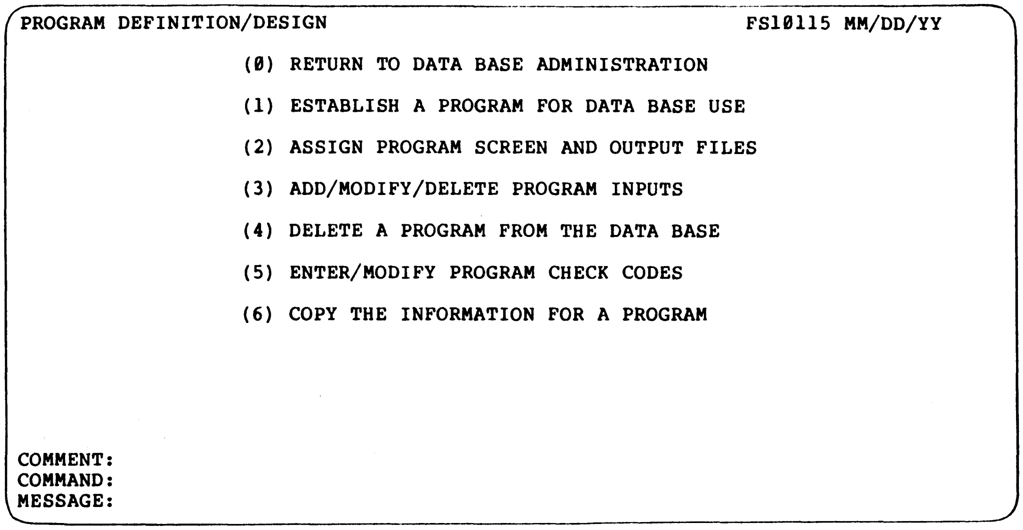
PROCEDURE: Enter the name of the system with which the file is associated. Then specify the name of the file to be deleted.

NOTES: This function invalidates all Dictionary reference to the affected file.

1.5 PROGRAM DEFINITION/DESIGN

PURPOSE: Program Definition/Design involves coordinating established screens, data elements and files as the components of a program structure. This enables FORCE to access the appropriate Dictionary information for program source code generation.

The Program Definition/Design menu provides functions to establish and maintain program information within the Dictionary.



```
PROGRAM DEFINITION/DESIGN                                         FS10115 MM/DD/YY

                         (0) RETURN TO DATA BASE ADMINISTRATION

                         (1) ESTABLISH A PROGRAM FOR DATA BASE USE

                         (2) ASSIGN PROGRAM SCREEN AND OUTPUT FILES

                         (3) ADD/MODIFY/DELETE PROGRAM INPUTS

                         (4) DELETE A PROGRAM FROM THE DATA BASE

                         (5) ENTER/MODIFY PROGRAM CHECK CODES

                         (6) COPY THE INFORMATION FOR A PROGRAM




COMMENT:
COMMAND:
MESSAGE:
```

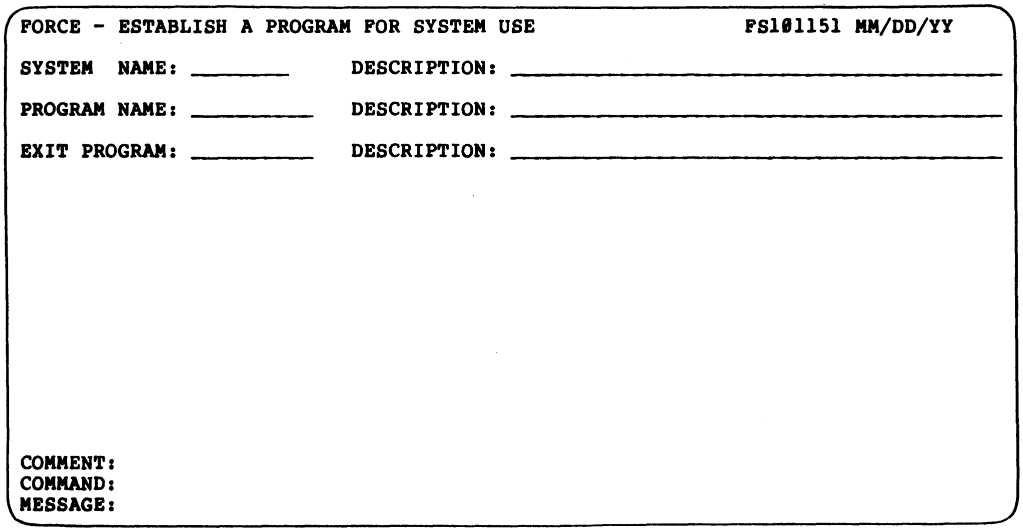
PROCEDURE: Enter the number which corresponds to the desired function.

NOTES: Programs are not generated by any of the functions within Program Definition/Design. The menu does provide the essential operations which complete Dictionary preparation for source code generation.

1.5.1 ESTABLISH A PROGRAM FOR DATA BASE USE

PURPOSE: This function is used to define a program to the Dictionary. The program's exit location (the program to which it chains) is also specified. FORCE requires this information prior to subsequent program development.

Additionally, the exit program or the description for an established program may be modified with this function.



```
FORCE - ESTABLISH A PROGRAM FOR SYSTEM USE                 FS1Ø1151 MM/DD/YY

SYSTEM  NAME: _______     DESCRIPTION: ___________________________________

PROGRAM NAME: _________   DESCRIPTION: ___________________________________

EXIT PROGRAM: _________   DESCRIPTION: ___________________________________











COMMENT:
COMMAND:
MESSAGE:
```

PROCEDURE: Enter the system name with which the program is to be associated. At the Program Name field, enter the program name and its description. Program names within FORCE must conform to IRIS program naming requirements, and may not exceed ten characters.

At the Exit Program entry field, specify an established program (within the same system) to which the new program exits or chains. A default (RETURN) at this entry field establishes an exit to SCOPE.

Programs should be established in order, from the top down, because the Exit Program entry must be a previously defined program. Typically, the first program established is the initial system program, which exits to SCOPE.

To modify an established program's description or exit program, enter the appropriate system name and the name of the program to be modified. The current program information is displayed. New information may be entered, or current information left intact by pressing RETURN.

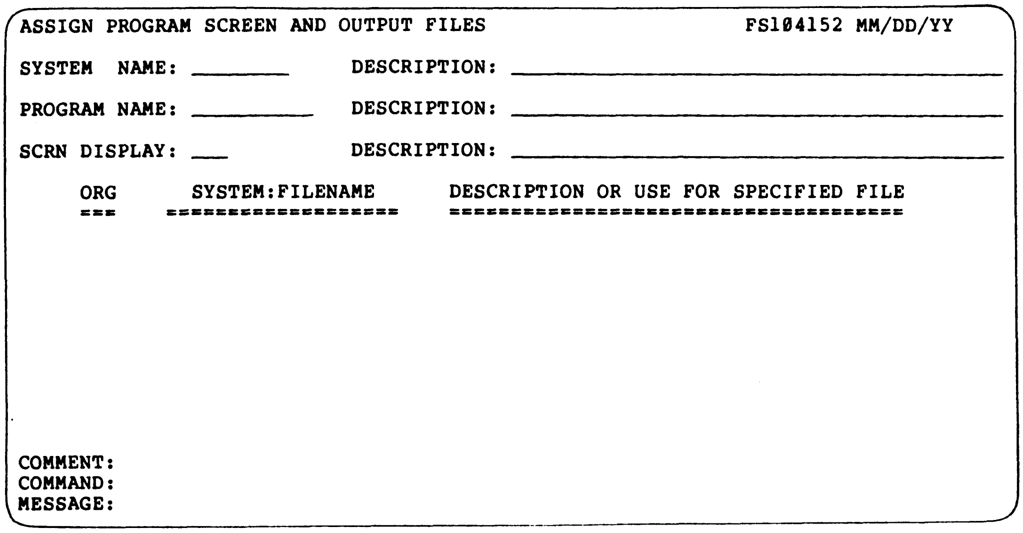
NOTES: Before defining program names to FORCE, the user may want to establish a naming convention for system programs (see 8.2.2).

1.5.2 ASSIGN PROGRAM SCREEN AND OUTPUT FILES

PURPOSE: Specification of the input screen display and output files used by an entry program directs FORCE to use those components in the generated source code. The specified screen is used for data input, record modification, deletion and inquiry.

This function establishes the specified program as an entry program. It is intended for the maintenance of entry programs only. An entry program is used to add, modify, delete and query records within a file.

This function may also be used to change the screen and file assignments for a program.



```
ASSIGN PROGRAM SCREEN AND OUTPUT FILES                         FS104152 MM/DD/YY

SYSTEM  NAME: ________      DESCRIPTION: ________________________________________

PROGRAM NAME: __________    DESCRIPTION: ________________________________________

SCRN DISPLAY: ___           DESCRIPTION: ________________________________________

     ORG      SYSTEM:FILENAME       DESCRIPTION OR USE FOR SPECIFIED FILE
     ===    ===================     ====================================









COMMENT:
COMMAND:
MESSAGE:
```

PROCEDURE: Screens and files specified in this function must be established in the Dictionary (see 1.2.1 and 1.4.1). Specified files must have established record layouts (see 1.4.2). Additionally, the affected program must have been defined (see 1.5.1).

Enter the name of the system with which the program is associated. Then enter the name of the entry program.

At the Screen Display field, enter the screen display number to be used by the generated program. The screen display must be from the same system as the program.

Under the System:Filename column, enter the files to which data is to be written. The filename input should correspond to the filename defined to FORCE, not the associated disk file.

Files from the system associated with the entry program may be input without a system association. To specify a file from a different system, use the format:

    SYSTEM:FILENAME

The first filename entry must be the program's data file. Subsequent entries are for specification of indexed files for key insertion. Up to nine indexed files may be entered.

Upon entry of a valid filename, FORCE displays that file's organization and description, and prompts for another entry.

A RETURN at the filename field indicates a completed entry.

To modify a program's established input screen or output files, specify the appropriate system and the program to be modified. The current program information is displayed. New information may be entered, or current information left intact by pressing RETURN.

Entry of the * symbol over a filename deletes that entry and all succeeding entries.

NOTES: An output data file must have a contiguous or formatted organization. An output index file must have an indexed organization (See Table 1-3).

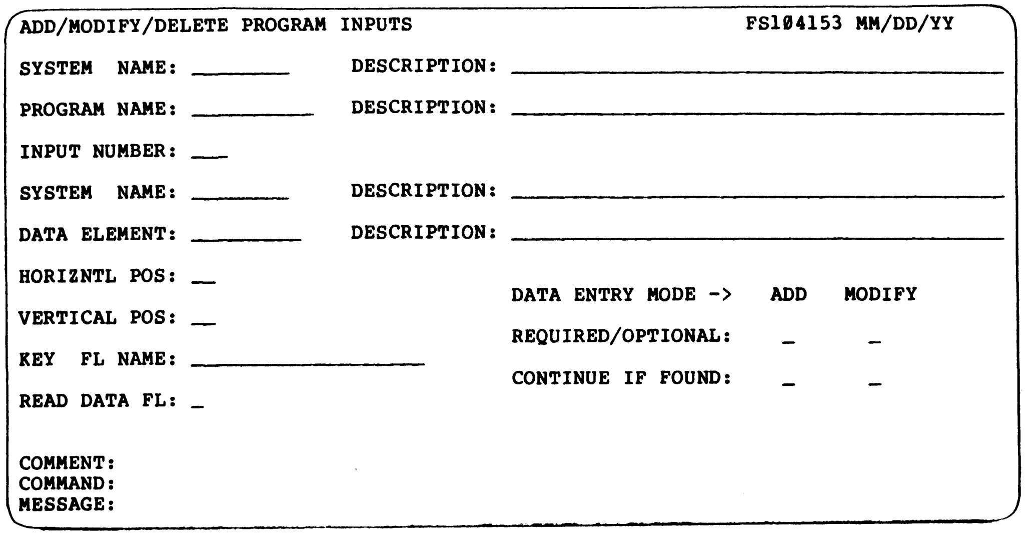
Only one output data file may be specified for an entry program.

PROGRAM INPUTS 1.5.3

1.5.3 ADD/MODIFY/DELETE PROGRAM INPUTS

PURPOSE: This function records an entry program's data input criteria in the Dictionary. This process involves: designating the order of data input; assigning data input to system variables; determining input positions on the screen; and specifying whether an input is required or optional. Indexed file check information is also specified. FORCE can then retrieve this information from the Dictionary and generate program source code accordingly.

This function may also be used to modify or delete established program input information.



```
ADD/MODIFY/DELETE PROGRAM INPUTS                              FS1Ø4153 MM/DD/YY

SYSTEM  NAME: ________     DESCRIPTION: ________________________________________

PROGRAM NAME: __________   DESCRIPTION: ________________________________________

INPUT NUMBER: ___

SYSTEM  NAME: ________     DESCRIPTION: ________________________________________

DATA ELEMENT: _________    DESCRIPTION: ________________________________________

HORIZNTL POS: __
                                        DATA ENTRY MODE ->   ADD    MODIFY
VERTICAL POS: __
                                        REQUIRED/OPTIONAL:    _       _
KEY  FL NAME: ___________________
                                        CONTINUE IF FOUND:    _       _
READ DATA FL: _


COMMENT:
COMMAND:
MESSAGE:
```

PROCEDURE: Execution of this function requires that the referenced program and program specifications be established in the Dictionary (see 1.5.1 and 1.5.2).

Enter the name of the system with which the entry program is associated. Then enter the name of the program.

The first set of procedures describe the process of adding program inputs. Modification and deletion procedures are then described.

Specify the input number to be defined. A RETURN at the Input Number field increments by one the number of the last established input. If no previous inputs have been established, a RETURN defaults to input number one.

The order in which data inputs are added establishes the order in which they are entered during program execution.

At the second System Name entry field, specify the system associated with the data element to receive input. A RETURN at this field defaults to the system name with which the program is associated.

At the Data Element entry field, enter the data element to be assigned the specified input.

For subdivided string data elements, specify the subdivision to receive input. Use the form:

```
D$(nnn)
```

where D$ is the data element name and nnn is the subdivision.

If the input data element is a matrix, indicate which cell is to receive input. Use the form:

```
D(rrXcc)
```

where D is the data element name and rrXcc represents the row and column coordinates of the matrix cell. For a single-dimension array, use the format:

```
D(nnnn)
```

where D is the data element name and nnnn represents the array cell coordinate.

At the Horizontal and Vertical Position entry fields, specify the coordinates where data entry begins on the input screen. This location may be obtained from a reproduction of the screen display (see 3.2.1). Screen display reproductions indicate horizontal coordinates at the top of the screen and vertical coordinates on the sides.

The Required/Optional entry field allows the user to specify whether the input is a (R)equired or (O)ptional field when adding and modifying a record. A default (RETURN) from these entries establishes the input as required when adding a record, and optional when modifying a record.

At the Key File field, specify the index file to be checked after the input is entered. If the file is associated with the same system as the input data element, enter the filename. For files from a different system, use the format:

```
SYSTEM:FILENAME
```

If the input does not require association with a keyfile (it will not initiate an indexed file check), enter RETURN at the Key File Name field. This default completes entry of information for an input.

Upon entry of a Key File Name, specify whether the program's data file is to be read on the retrieved pointer. This READ is used for the modify, delete and inquire operations to retrieve and display the record information.

At the Read Data File field, enter (Y)es or (N)o to indicate if a READ is to be performed on the program's data file based upon the value retrieved from the indexed file check. One READ to a data file is required for execution of the generated file maintenance program.

The Continue If Found entry allows specification of whether data entry should continue (when adding and modifying a record) if the key is found in the indexed file search. FORCE creates the appropriate message in the generated entry program.

The Continue If Found entry is automatically established if the Read Data File entry is (Y)es. The operation will not continue if the program is in Add Mode, and it will continue is the program is in Modify Mode. If the Read Data File entry is (N)o, enter (Y)es or (N)o at each Continue If Found field to specify whether program operation should continue.

Repeat this procedure to add each data input within the program.

To modify or delete a program input, enter the input number. If the entry is a valid input number, FORCE displays the current input information and prompts for deletion of the input.

An entry of (Y)es at the deletion prompt deletes that program input. An entry of (N)o at the deletion prompt invokes the modify function. At each entry field the current information may be left intact by pressing RETURN, or new information may be entered. To delete a Key File input, enter the * symbol over the current entry.

NOTES: This program is for the maintenance of entry programs only.

An entry program cannot use multiple data elements (from different systems) which are assigned the same variable name.

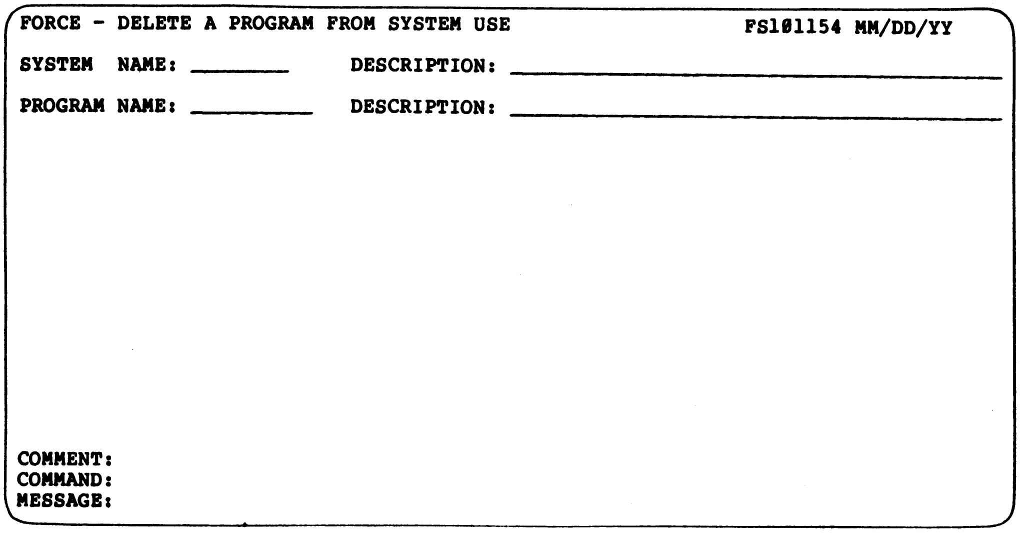
After deletion of an input, subsequent program input numbers are decremented by one. This renumbering of program inputs is performed after the user exits the Add/Modify/Delete Program Inputs function.

Modification or deletion of a program input has no effect on the assigned screen display. Corresponding adjustments to the input screen (if required) must be formatted (see 1.2.1).

In the generated file maintenance program, data may only be input into fields of 65 bytes or less.

1.5.4 DELETE A PROGRAM FROM THE DATA BASE

PURPOSE: The user may delete all Dictionary references to a program if the program is no longer required.



```
FORCE - DELETE A PROGRAM FROM SYSTEM USE                         FS101154 MM/DD/YY
SYSTEM  NAME: ________      DESCRIPTION: ________________________________________
PROGRAM NAME: __________    DESCRIPTION: ________________________________________









COMMENT:
COMMAND:
MESSAGE:
```

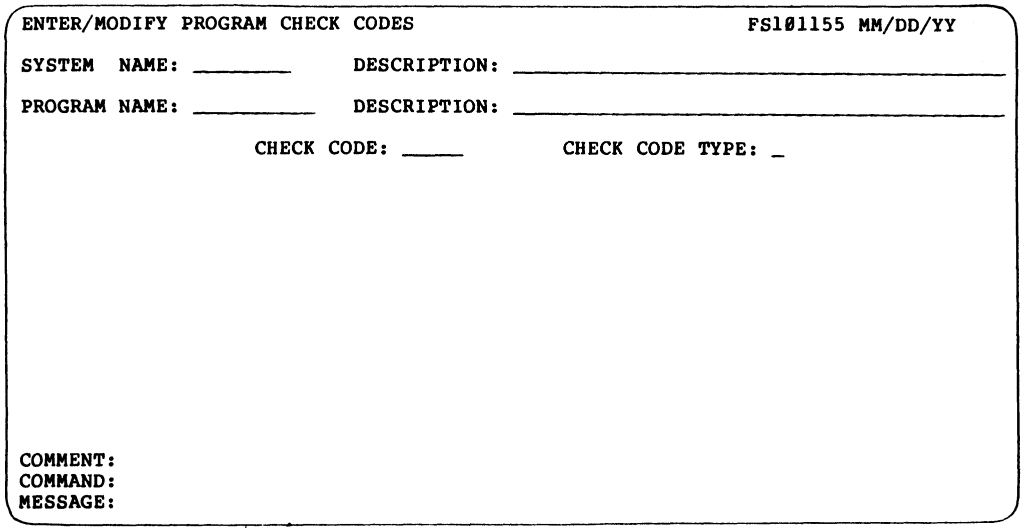
PROCEDURE: Enter the system name with which the program is associated. Then enter the name of the program.

NOTES: This function invalidates the program name within the Dictionary. Consequently, no related program information can be referenced.

PROGRAM
CHECK CODES
1.5.5

1.5.5 ENTER/MODIFY PROGRAM CHECK CODES

PURPOSE: This function allows the user to record and maintain program check codes within the Dictionary. This information then becomes available through system documentation (see 3.3.2).



```
ENTER/MODIFY PROGRAM CHECK CODES                              FS1Ø1155 MM/DD/YY

SYSTEM  NAME: ________     DESCRIPTION: ________________________________________

PROGRAM NAME: __________   DESCRIPTION: ________________________________________

                     CHECK CODE: _____         CHECK CODE TYPE: _










COMMENT:
COMMAND:
MESSAGE:
```

PROCEDURE: Enter the appropriate system and program names.

Specify the program's check code at the Check Code entry field. Then enter the check code type as either (S)ave or (P)rotect.

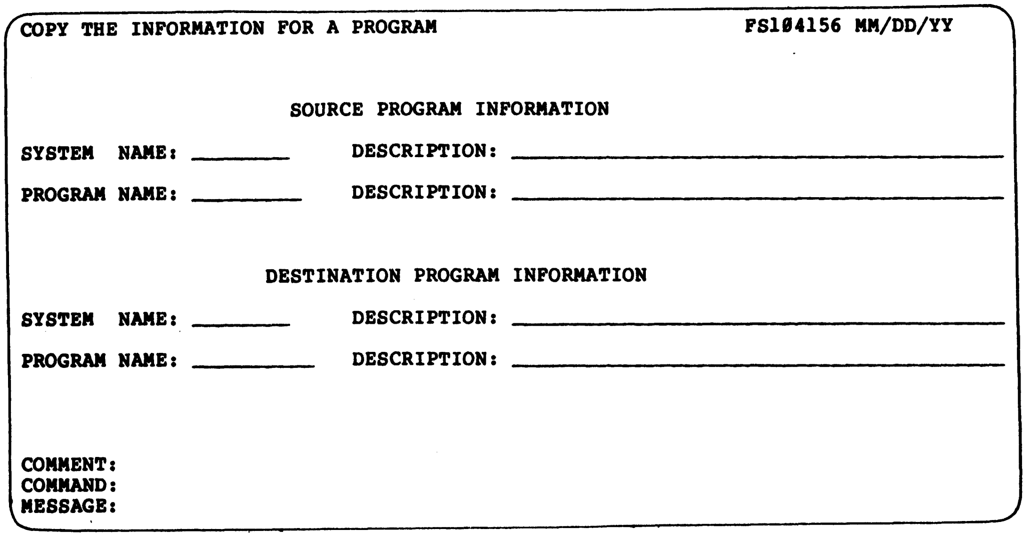
When modifying an established check code, the current Dictionary information is displayed after the program name is entered. At the Check Code and Check Code Type entry fields, new information may be entered, or current information left intact by pressing RETURN.

NOTES: FORCE also provides a facility to print program check code worksheets (see 3.3.1).

1.5.6 COPY THE INFORMATION FOR A PROGRAM

PURPOSE: This function retrieves a complete set of program specifications and uses them for another program which is to perform the same or similar functions. Execution of this function copies pertinent source program information that has been defined to the Dictionary. This information may include the program input screen, output file(s), complete data input specifications and exit location. Program check codes are not copied.

Execution of this function saves the time of entering a new set of program specifications which duplicate those already in the Dictionary. The program's copied information may then be modified, if needed.



```
COPY THE INFORMATION FOR A PROGRAM                          FS104156 MM/DD/YY


                         SOURCE PROGRAM INFORMATION

SYSTEM  NAME: ______       DESCRIPTION: ____________________________________

PROGRAM NAME: _______      DESCRIPTION: ____________________________________


                       DESTINATION PROGRAM INFORMATION

SYSTEM  NAME: ______       DESCRIPTION: ____________________________________

PROGRAM NAME: ________     DESCRIPTION: ____________________________________



COMMENT:
COMMAND:
MESSAGE:
```

PROCEDURE: Source Program Information indicates the program from which information is to be copied. Enter the system name with which the program is associated, then enter the program name.

The Destination Program Information specifies the program to which information is to be copied. Enter the associated system name. Then enter the program name and its description.

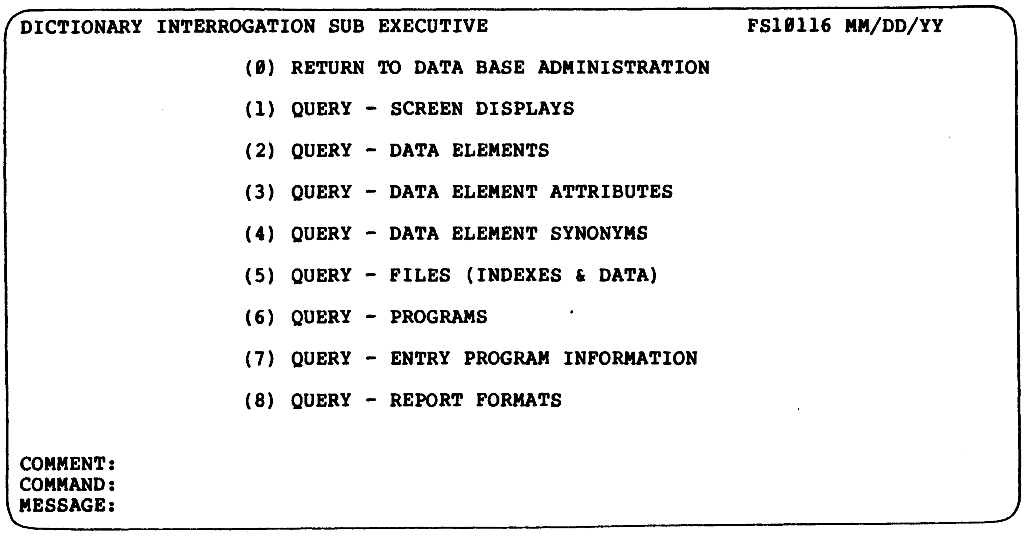
NOTES: Entry of the Destination Program Information establishes the program within the Dictionary, thus eliminating the need for execution of that function (see 1.5.1).

The source program information is not altered by the copying process.

DICTIONARY INTERROGATION 1.6

1.6 DICTIONARY INTERROGATION

PURPOSE: Dictionary Interrogation enables the user to obtain Dictionary information quickly by displaying specified data on the screen. These instant inquiries facilitate system development by providing prompt, updated Dictionary status information.



```
DICTIONARY INTERROGATION SUB EXECUTIVE                         FS10116 MM/DD/YY

                  (0) RETURN TO DATA BASE ADMINISTRATION

                  (1) QUERY - SCREEN DISPLAYS

                  (2) QUERY - DATA ELEMENTS

                  (3) QUERY - DATA ELEMENT ATTRIBUTES

                  (4) QUERY - DATA ELEMENT SYNONYMS

                  (5) QUERY - FILES (INDEXES & DATA)

                  (6) QUERY - PROGRAMS

                  (7) QUERY - ENTRY PROGRAM INFORMATION

                  (8) QUERY - REPORT FORMATS

COMMENT:
COMMAND:
MESSAGE:
```

PROCEDURE: Enter the number that corresponds to the desired function.

1.6.1 QUERY - SCREEN DISPLAYS

PURPOSE: The user may query screen displays for any or all systems within the Dictionary. This information shows screen display numbers, screen descriptions and the associated system.

PROCEDURE: At the Starting Location prompt, specify the query starting point among system names. This entry may be a complete system name or the first unique characters of the system name.

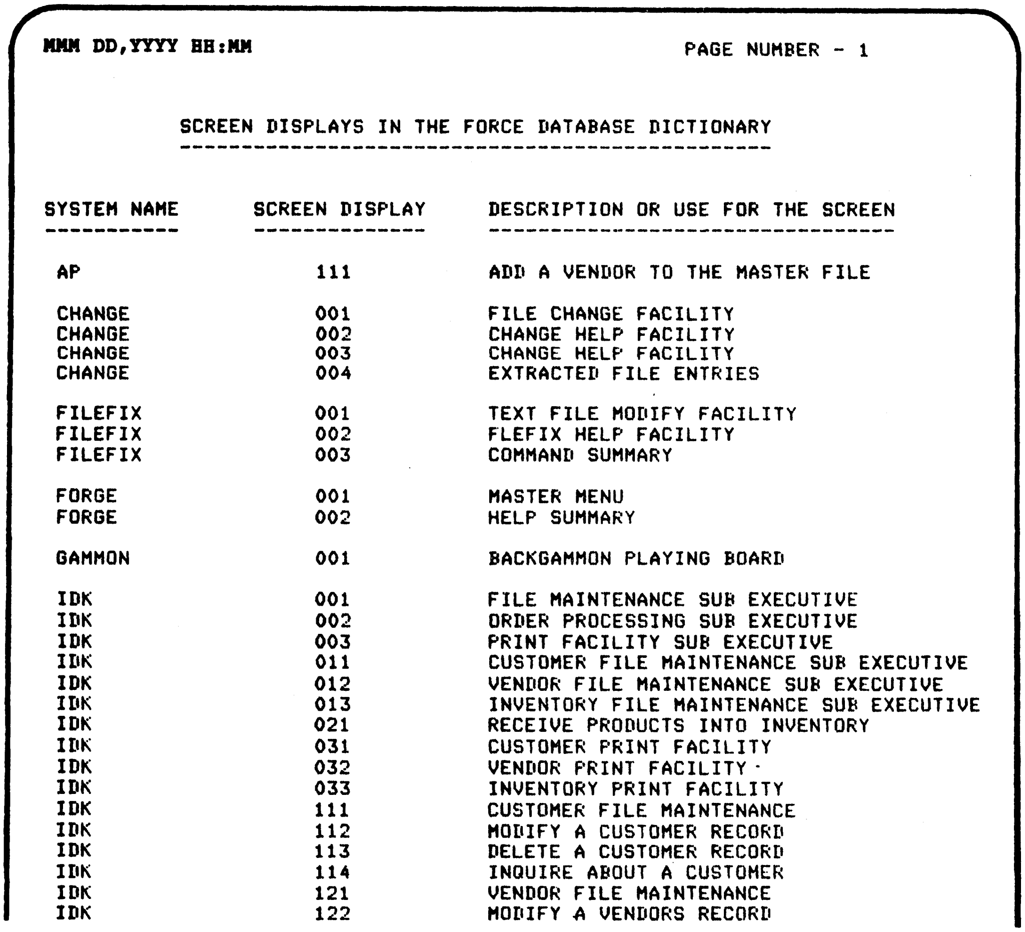
The appropriate Dictionary data is retrieved and displayed beginning at the specified location in the Dictionary.

A RETURN at the Starting Location prompt displays the information beginning alphabetically with the first associated system name.

After the first segment of data is displayed on the screen, each RETURN displays successive screens of information. The final screen segment displays a message indicating that no additional data is available.

An ESCAPE exits to the Query - Screen Displays screen.

NOTES: Below is a sample query screen for the Query - Screen Displays function.



```
MMM DD,YYYY HH:MM                                       PAGE NUMBER - 1


           SCREEN DISPLAYS IN THE FORCE DATABASE DICTIONARY
           ------------------------------------------------


SYSTEM NAME      SCREEN DISPLAY     DESCRIPTION OR USE FOR THE SCREEN
-----------      --------------     ---------------------------------

 AP                   111           ADD A VENDOR TO THE MASTER FILE

 CHANGE               001           FILE CHANGE FACILITY
 CHANGE               002           CHANGE HELP FACILITY
 CHANGE               003           CHANGE HELP FACILITY
 CHANGE               004           EXTRACTED FILE ENTRIES

 FILEFIX              001           TEXT FILE MODIFY FACILITY
 FILEFIX              002           FLEFIX HELP FACILITY
 FILEFIX              003           COMMAND SUMMARY

 FORGE                001           MASTER MENU
 FORGE                002           HELP SUMMARY

 GAMMON               001           BACKGAMMON PLAYING BOARD

 IDK                  001           FILE MAINTENANCE SUB EXECUTIVE
 IDK                  002           ORDER PROCESSING SUB EXECUTIVE
 IDK                  003           PRINT FACILITY SUB EXECUTIVE
 IDK                  011           CUSTOMER FILE MAINTENANCE SUB EXECUTIVE
 IDK                  012           VENDOR FILE MAINTENANCE SUB EXECUTIVE
 IDK                  013           INVENTORY FILE MAINTENANCE SUB EXECUTIVE
 IDK                  021           RECEIVE PRODUCTS INTO INVENTORY
 IDK                  031           CUSTOMER PRINT FACILITY
 IDK                  032           VENDOR PRINT FACILITY
 IDK                  033           INVENTORY PRINT FACILITY
 IDK                  111           CUSTOMER FILE MAINTENANCE
 IDK                  112           MODIFY A CUSTOMER RECORD
 IDK                  113           DELETE A CUSTOMER RECORD
 IDK                  114           INQUIRE ABOUT A CUSTOMER
 IDK                  121           VENDOR FILE MAINTENANCE
 IDK                  122           MODIFY A VENDORS RECORD
```

1.6.2 QUERY - DATA ELEMENTS

PURPOSE: The user may query data elements for any or all systems within the Dictionary. This information shows data element names, descriptions and the associated system.

PROCEDURE: At the Starting Location prompt, specify the query starting point among system names. This entry may be a complete system name or the first unique characters of the system name.

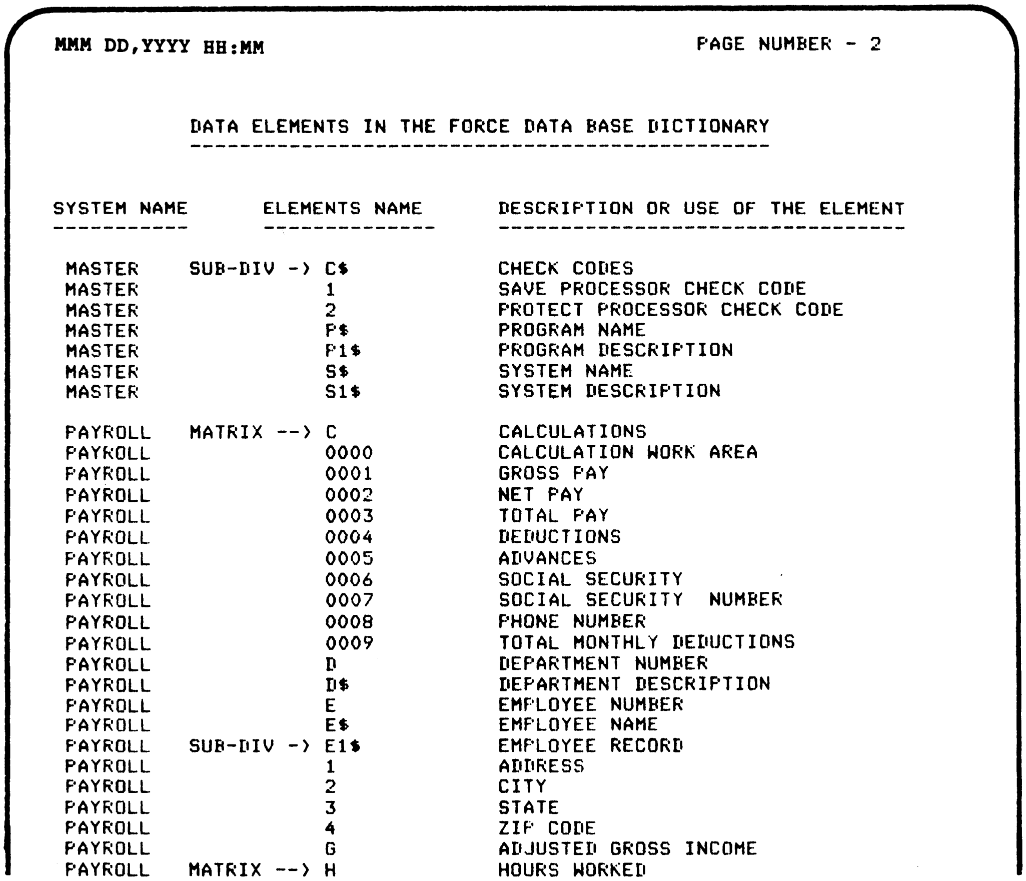
The appropriate Dictionary data is retrieved and displayed beginning at the specified location in the Dictionary.

A RETURN at the Starting Location prompt displays the information, beginning alphabetically with the first associated system name.

After the first segment of data is displayed on the screen, each RETURN displays successive screens of information. The final screen segment displays a message indicating that no additional data is available.

An ESCAPE exits to the Query - Data Elements screen.

NOTES: Below is a sample query screen for the Query - Data Elements function.



```
MMM DD,YYYY HH:MM                                      PAGE NUMBER - 2


          DATA ELEMENTS IN THE FORCE DATA BASE DICTIONARY
          -----------------------------------------------


SYSTEM NAME       ELEMENTS NAME      DESCRIPTION OR USE OF THE ELEMENT
-----------       -------------      ---------------------------------

 MASTER    SUB-DIV -> C$              CHECK CODES
 MASTER                1              SAVE PROCESSOR CHECK CODE
 MASTER                2              PROTECT PROCESSOR CHECK CODE
 MASTER                P$             PROGRAM NAME
 MASTER                P1$            PROGRAM DESCRIPTION
 MASTER                S$             SYSTEM NAME
 MASTER                S1$            SYSTEM DESCRIPTION

 PAYROLL   MATRIX --> C               CALCULATIONS
 PAYROLL               0000           CALCULATION WORK AREA
 PAYROLL               0001           GROSS PAY
 PAYROLL               0002           NET PAY
 PAYROLL               0003           TOTAL PAY
 PAYROLL               0004           DEDUCTIONS
 PAYROLL               0005           ADVANCES
 PAYROLL               0006           SOCIAL SECURITY
 PAYROLL               0007           SOCIAL SECURITY  NUMBER
 PAYROLL               0008           PHONE NUMBER
 PAYROLL               0009           TOTAL MONTHLY DEDUCTIONS
 PAYROLL               D              DEPARTMENT NUMBER
 PAYROLL               D$             DEPARTMENT DESCRIPTION
 PAYROLL               E              EMPLOYEE NUMBER
 PAYROLL               E$             EMPLOYEE NAME
 PAYROLL   SUB-DIV -> E1$             EMPLOYEE RECORD
 PAYROLL               1              ADDRESS
 PAYROLL               2              CITY
 PAYROLL               3              STATE
 PAYROLL               4              ZIP CODE
 PAYROLL               G              ADJUSTED GROSS INCOME
 PAYROLL   MATRIX --> H               HOURS WORKED
```

1.6.3 QUERY - DATA ELEMENT ATTRIBUTES

PURPOSE: The user may query attributes of data elements for any or all systems in the Dictionary. This information displays the data element name, type, maximum and minimum size, and special fields information. It also shows the high range, low range and the associated system.

PROCEDURE: At the Starting Location prompt, specify the query starting point among system names. This entry may be a complete system name or the first unique characters of the system name.

The appropriate Dictionary data is retrieved and displayed, beginning at the specified location in the Dictionary.

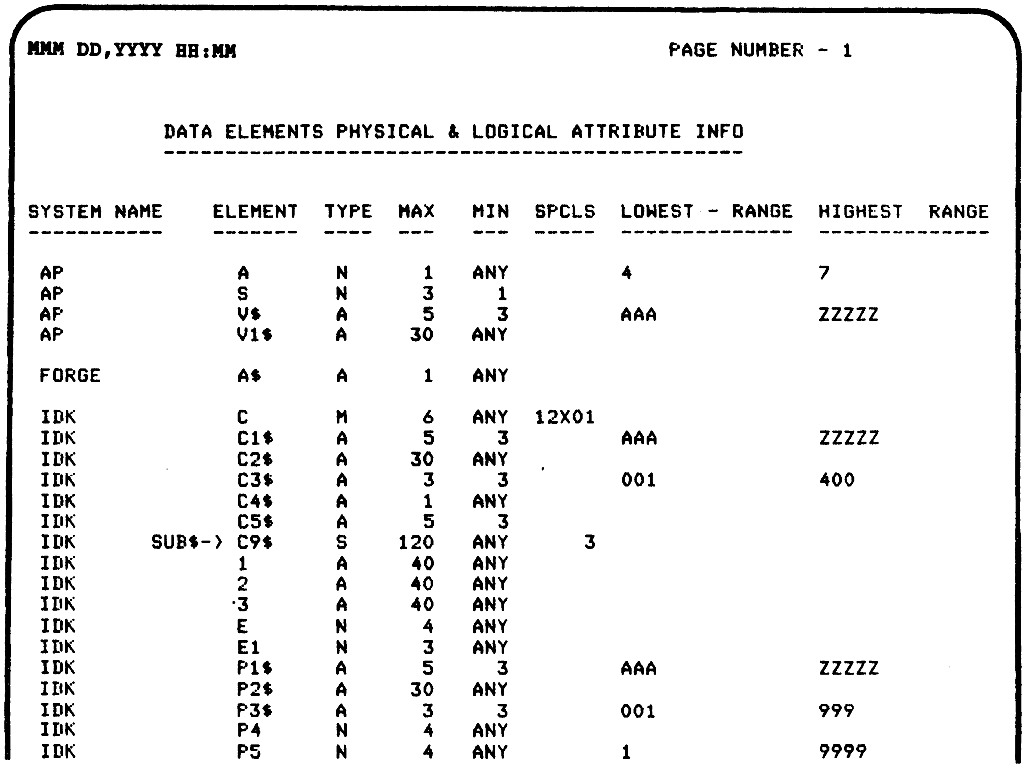
A RETURN at the Starting Location prompt displays the information beginning alphabetically with the first associated system name.

After the first segment of data is displayed on the screen, each RETURN displays successive screens of information. The final screen segment displays a message indicating that no additional data is available.

An ESCAPE exits to the Query - Data Element Attributes screen.

NOTES: Below is a sample query screen for the Query - Data Element Attributes function.

The SPCLS column denotes the number of subdivisions for a subdivided string, and the cell coordinates for a matrix.



```
MMM DD,YYYY HH:MM                                        PAGE NUMBER - 1


              DATA ELEMENTS PHYSICAL & LOGICAL ATTRIBUTE INFO
              -----------------------------------------------

SYSTEM NAME      ELEMENT  TYPE  MAX   MIN  SPCLS  LOWEST - RANGE  HIGHEST  RANGE
-----------      -------  ----  ---   ---  -----  --------------  --------------

 AP                A       N      1   ANY         4               7
 AP                S       N      3     1
 AP                V$      A      5     3         AAA             ZZZZZ
 AP                V1$     A     30   ANY

 FORGE             A$      A      1   ANY

 IDK               C       M      6   ANY  12X01
 IDK               C1$     A      5     3         AAA             ZZZZZ
 IDK               C2$     A     30   ANY
 IDK               C3$     A      3     3         001             400
 IDK               C4$     A      1   ANY
 IDK               C5$     A      5     3
 IDK        SUB$-> C9$     S    120   ANY      3
 IDK               1       A     40   ANY
 IDK               2       A     40   ANY
 IDK               ·3      A     40   ANY
 IDK               E       N      4   ANY
 IDK               E1      N      3   ANY
 IDK               P1$     A      5     3         AAA             ZZZZZ
 IDK               P2$     A     30   ANY
 IDK               P3$     A      3     3         001             999
 IDK               P4      N      4   ANY
 IDK               P5      N      4   ANY         1               9999
```

1.6.4 QUERY - DATA ELEMENT SYNONYMS

PURPOSE: The user may query data element synonyms for any or all systems within the Dictionary. This information shows synonym names, descriptions and the associated system.

PROCEDURE: At the Starting Location prompt, specify the query starting point among system names. This entry may be a complete system name or the first unique characters of the system name.

The appropriate Dictionary data is retrieved and displayed beginning at the specified Dictionary location.

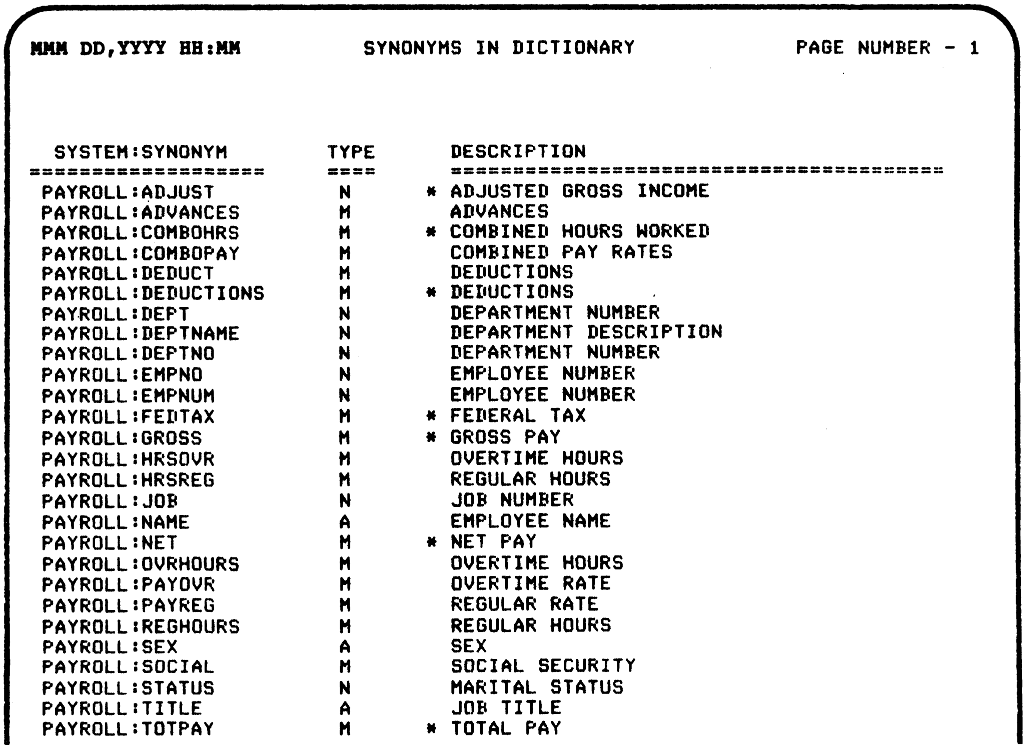
A RETURN at the Starting Location prompt displays the information beginning alphabetically with the first associated system name.

After the first segment of data is displayed on the screen, each RETURN displays successive screens of information. The final screen segment displays a message indicating that no additional data is available.

An ESCAPE exits to the Query - Data Element Synonyms screen.

NOTES: Below is a sample query screen for the Query - Data Element Synonyms function.

Calculated synonyms (see 4.5.3) are denoted by an asterisk.



```
MMM DD,YYYY HH:MM          SYNONYMS IN DICTIONARY           PAGE NUMBER - 1



 SYSTEM:SYNONYM         TYPE      DESCRIPTION
====================    ====      ========================================
PAYROLL:ADJUST           N      * ADJUSTED GROSS INCOME
PAYROLL:ADVANCES         M        ADVANCES
PAYROLL:COMBOHRS         M      * COMBINED HOURS WORKED
PAYROLL:COMBOPAY         M        COMBINED PAY RATES
PAYROLL:DEDUCT           M        DEDUCTIONS
PAYROLL:DEDUCTIONS       M      * DEDUCTIONS
PAYROLL:DEPT             N        DEPARTMENT NUMBER
PAYROLL:DEPTNAME         N        DEPARTMENT DESCRIPTION
PAYROLL:DEPTNO           N        DEPARTMENT NUMBER
PAYROLL:EMPNO            N        EMPLOYEE NUMBER
PAYROLL:EMPNUM           N        EMPLOYEE NUMBER
PAYROLL:FEDTAX           M      * FEDERAL TAX
PAYROLL:GROSS            M      * GROSS PAY
PAYROLL:HRSOVR           M        OVERTIME HOURS
PAYROLL:HRSREG           M        REGULAR HOURS
PAYROLL:JOB              N        JOB NUMBER
PAYROLL:NAME             A        EMPLOYEE NAME
PAYROLL:NET              M      * NET PAY
PAYROLL:OVRHOURS         M        OVERTIME HOURS
PAYROLL:PAYOVR           M        OVERTIME RATE
PAYROLL:PAYREG           M        REGULAR RATE
PAYROLL:REGHOURS         M        REGULAR HOURS
PAYROLL:SEX              A        SEX
PAYROLL:SOCIAL           M        SOCIAL SECURITY
PAYROLL:STATUS           N        MARITAL STATUS
PAYROLL:TITLE            A        JOB TITLE
PAYROLL:TOTPAY           M      * TOTAL PAY
```

1.6.5 QUERY - FILES (INDEXES & DATA)

PURPOSE: The user may query file specifications for any or all systems within the Dictionary. This information includes file name, description, organization, subscription, disk file name and the associated system.

PROCEDURE: At the Starting Location prompt, specify the query starting point among system names. This entry may be a complete system name or the first unique characters of the system name.

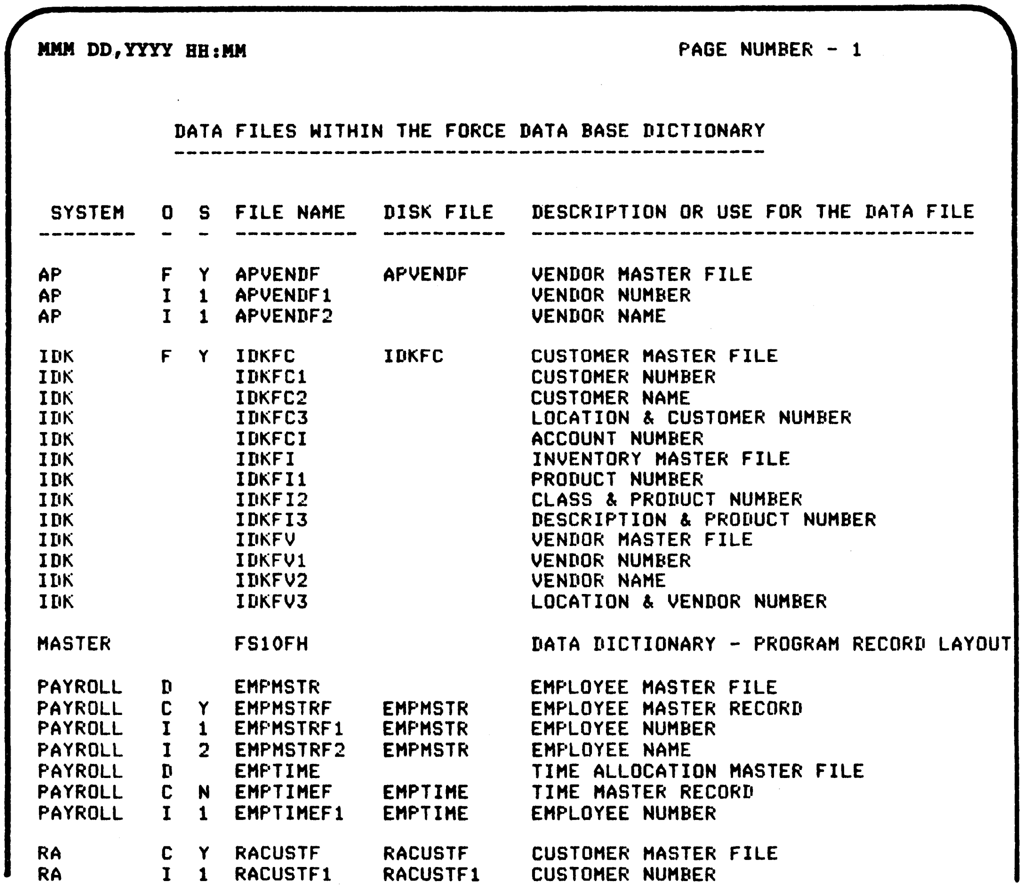
The appropriate Dictionary data is retrieved and displayed, beginning at the specified Dictionary location.

A RETURN at the Starting Location prompt displays the information beginning alphabetically with the first associated system name.

After the first segment of data is displayed on the screen, each RETURN displays successive screens of information. The final screen segment indicates that no additional data is available.

An ESCAPE exits to the Query - Files (Indexes & Data) screen.

NOTES: Below is a sample query screen for the Query - Files (Indexes & Data) function.



```
MMM DD,YYYY HH:MM                                        PAGE NUMBER - 1


               DATA FILES WITHIN THE FORCE DATA BASE DICTIONARY
               ------------------------------------------------


 SYSTEM    O  S  FILE NAME   DISK FILE   DESCRIPTION OR USE FOR THE DATA FILE
 --------  -  -  ----------  ----------  ------------------------------------

AP         F  Y  APVENDF     APVENDF     VENDOR MASTER FILE
AP         I  1  APVENDF1                VENDOR NUMBER
AP         I  1  APVENDF2                VENDOR NAME

IDK        F  Y  IDKFC       IDKFC       CUSTOMER MASTER FILE
IDK              IDKFC1                  CUSTOMER NUMBER
IDK              IDKFC2                  CUSTOMER NAME
IDK              IDKFC3                  LOCATION & CUSTOMER NUMBER
IDK              IDKFCI                  ACCOUNT NUMBER
IDK              IDKFI                   INVENTORY MASTER FILE
IDK              IDKFI1                  PRODUCT NUMBER
IDK              IDKFI2                  CLASS & PRODUCT NUMBER
IDK              IDKFI3                  DESCRIPTION & PRODUCT NUMBER
IDK              IDKFV                   VENDOR MASTER FILE
IDK              IDKFV1                  VENDOR NUMBER
IDK              IDKFV2                  VENDOR NAME
IDK              IDKFV3                  LOCATION & VENDOR NUMBER

MASTER           FS10FH                  DATA DICTIONARY - PROGRAM RECORD LAYOUT

PAYROLL    D     EMPMSTR                 EMPLOYEE MASTER FILE
PAYROLL    C  Y  EMPMSTRF    EMPMSTR     EMPLOYEE MASTER RECORD
PAYROLL    I  1  EMPMSTRF1   EMPMSTR     EMPLOYEE NUMBER
PAYROLL    I  2  EMPMSTRF2   EMPMSTR     EMPLOYEE NAME
PAYROLL    D     EMPTIME                 TIME ALLOCATION MASTER FILE
PAYROLL    C  N  EMPTIMEF    EMPTIME     TIME MASTER RECORD
PAYROLL    I  1  EMPTIMEF1   EMPTIME     EMPLOYEE NUMBER

RA         C  Y  RACUSTF     RACUSTF     CUSTOMER MASTER FILE
RA         I  1  RACUSTF1    RACUSTF1    CUSTOMER NUMBER
```

1.6.6 QUERY - PROGRAMS

PURPOSE: The user may query programs for any or all systems within the Dictionary. This information includes program name, type, description and the associated system.

PROCEDURE: At the Starting Location prompt, specify the query starting point among system names. This entry may be a complete system name or the first unique characters of the system name.

The appropriate Dictionary data is retrieved and displayed, beginning at the specified Dictionary location.

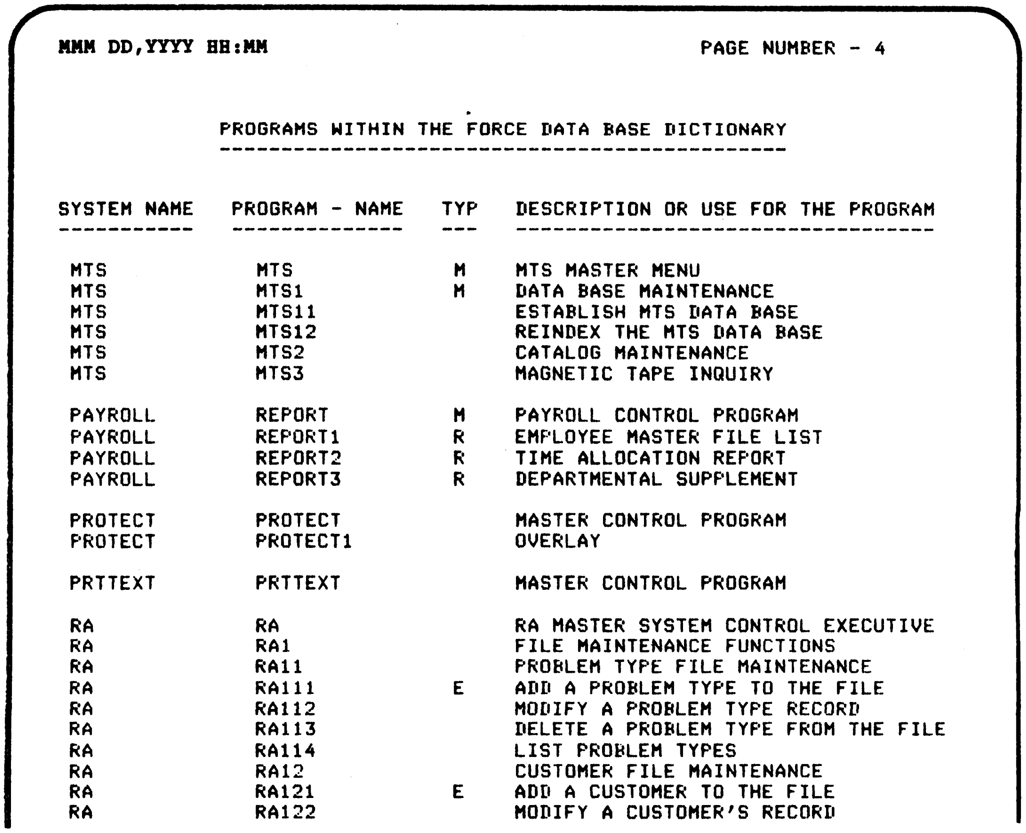
A RETURN at the Starting Location prompt displays the information beginning alphabetically with the first associated system name.

After the first segment of data is displayed on the screen, each RETURN displays successive screens of information. The final screen segment displays a message indicating that no additional data is available.

An ESCAPE exits to the Query - Programs screen.

NOTES: Below is a sample query screen for the Query - Programs function.

Program Type denotes whether the program is an (E)ntry, (M)enu, or (R)eport program.



```
MMM DD,YYYY HH:MM                                  PAGE NUMBER - 4


               PROGRAMS WITHIN THE FORCE DATA BASE DICTIONARY
               ----------------------------------------------

SYSTEM NAME    PROGRAM - NAME   TYP    DESCRIPTION OR USE FOR THE PROGRAM
-----------    --------------   ---    ----------------------------------

 MTS             MTS             M     MTS MASTER MENU
 MTS             MTS1            M     DATA BASE MAINTENANCE
 MTS             MTS11                 ESTABLISH MTS DATA BASE
 MTS             MTS12                 REINDEX THE MTS DATA BASE
 MTS             MTS2                  CATALOG MAINTENANCE
 MTS             MTS3                  MAGNETIC TAPE INQUIRY

 PAYROLL         REPORT          M     PAYROLL CONTROL PROGRAM
 PAYROLL         REPORT1         R     EMPLOYEE MASTER FILE LIST
 PAYROLL         REPORT2         R     TIME ALLOCATION REPORT
 PAYROLL         REPORT3         R     DEPARTMENTAL SUPPLEMENT

 PROTECT         PROTECT               MASTER CONTROL PROGRAM
 PROTECT         PROTECT1              OVERLAY

 PRTTEXT         PRTTEXT               MASTER CONTROL PROGRAM

 RA              RA                    RA MASTER SYSTEM CONTROL EXECUTIVE
 RA              RA1                   FILE MAINTENANCE FUNCTIONS
 RA              RA11                  PROBLEM TYPE FILE MAINTENANCE
 RA              RA111           E     ADD A PROBLEM TYPE TO THE FILE
 RA              RA112                 MODIFY A PROBLEM TYPE RECORD
 RA              RA113                 DELETE A PROBLEM TYPE FROM THE FILE
 RA              RA114                 LIST PROBLEM TYPES
 RA              RA12                  CUSTOMER FILE MAINTENANCE
 RA              RA121           E     ADD A CUSTOMER TO THE FILE
 RA              RA122                 MODIFY A CUSTOMER'S RECORD
```

1.6.7 QUERY - ENTRY PROGRAM INFORMATION

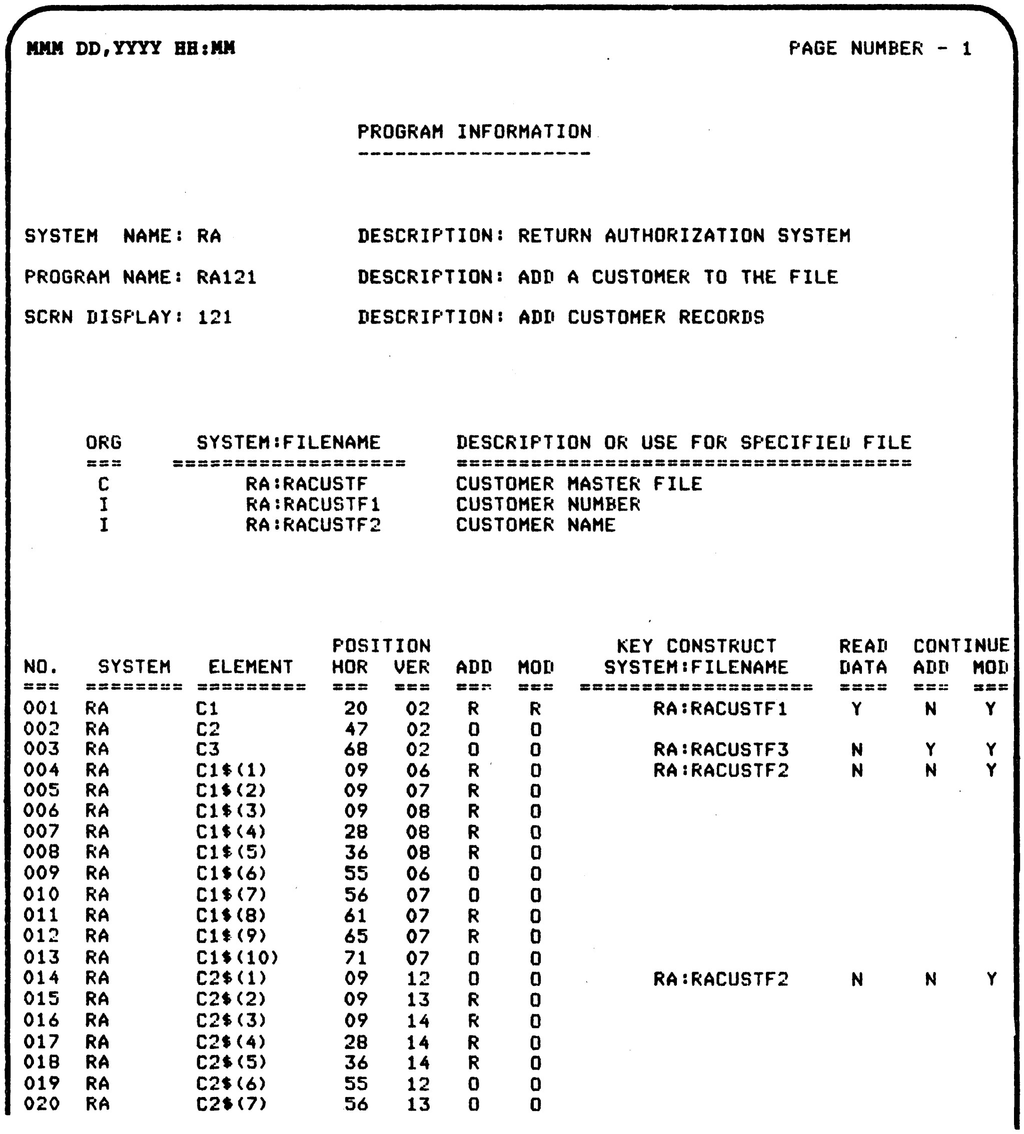
PURPOSE: The user may query files and inputs used by any entry program in the Dictionary. This information includes the program's input screen and output file(s). Also included is a comprehensive set of information about each program input.

PROCEDURE: Enter the name of the system associated with the program to be queried. Then enter the name of the program.

The first query screen displays the name and description of the program, its input screen and output file(s).

After display of the initial query screen, a RETURN invokes display of the program input information.

NOTES: Below is a sample query screen which shows the output for the Query - Entry Program Information function.



```
MMM DD,YYYY HH:MM                                              PAGE NUMBER - 1


                              PROGRAM INFORMATION
                              -------------------


SYSTEM  NAME: RA             DESCRIPTION: RETURN AUTHORIZATION SYSTEM

PROGRAM NAME: RA121          DESCRIPTION: ADD A CUSTOMER TO THE FILE

SCRN DISPLAY: 121            DESCRIPTION: ADD CUSTOMER RECORDS



     ORG      SYSTEM:FILENAME      DESCRIPTION OR USE FOR SPECIFIED FILE
     ===    ==================     =====================================
      C          RA:RACUSTF        CUSTOMER MASTER FILE
      I          RA:RACUSTF1       CUSTOMER NUMBER
      I          RA:RACUSTF2       CUSTOMER NAME



                               POSITION                   KEY CONSTRUCT      READ  CONTINUE
NO.   SYSTEM   ELEMENT        HOR  VER   ADD  MOD     SYSTEM:FILENAME       DATA  ADD  MOD
===  ======== =========       ===  ===   ===  ===   ===================    ====  ===  ===
001  RA        C1              20   02    R    R        RA:RACUSTF1          Y     N    Y
002  RA        C2              47   02    O    O
003  RA        C3              68   02    O    O        RA:RACUSTF3          N     Y    Y
004  RA        C1$(1)          09   06    R    O        RA:RACUSTF2          N     N    Y
005  RA        C1$(2)          09   07    R    O
006  RA        C1$(3)          09   08    R    O
007  RA        C1$(4)          28   08    R    O
008  RA        C1$(5)          36   08    R    O
009  RA        C1$(6)          55   06    O    O
010  RA        C1$(7)          56   07    O    O
011  RA        C1$(8)          61   07    R    O
012  RA        C1$(9)          65   07    R    O
013  RA        C1$(10)         71   07    O    O
014  RA        C2$(1)          09   12    O    O        RA:RACUSTF2          N     N    Y
015  RA        C2$(2)          09   13    R    O
016  RA        C2$(3)          09   14    R    O
017  RA        C2$(4)          28   14    R    O
018  RA        C2$(5)          36   14    R    O
019  RA        C2$(6)          55   12    O    O
020  RA        C2$(7)          56   13    O    O
```

QUERY REPORT FORMATS 1.6.8

1.6.8 QUERY - REPORT FORMATS

PURPOSE: The user may query report formats for any or all systems within the Dictionary. This information includes the format number, type, description and the associated system.

PROCEDURE: At the Starting Location prompt, specify the query starting point among system names. This entry may be a complete system name or the first unique characters of the system name.

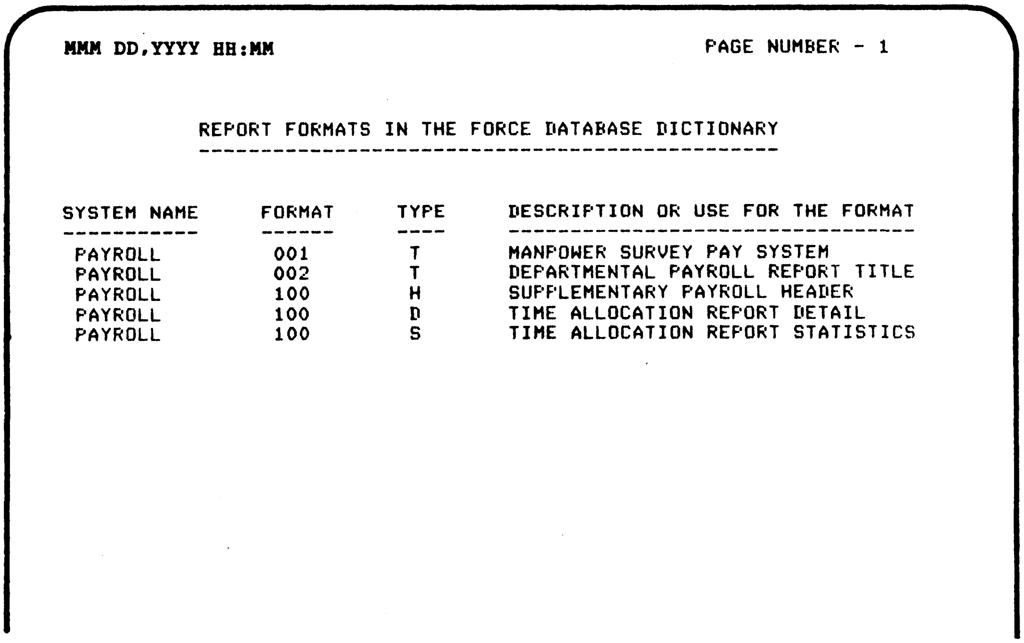
The appropriate Dictionary data is retrieved and displayed, beginning at the specified Dictionary location.

A RETURN at the Starting Location prompt displays the information beginning alphabetically with the first associated system name.

After the first segment of data is displayed on the screen, each RETURN displays successive screens of information. The final screen segment displays a message indicating that no additional data is available.

An ESCAPE exits to the Query - Report Formats screen.

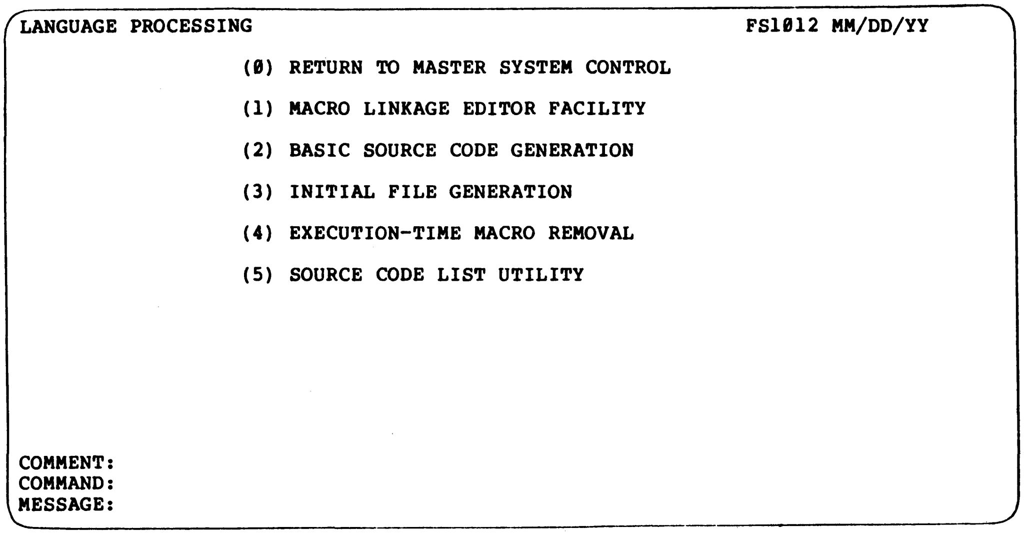
NOTES: Below is a sample query screen for the Query - Report Formats function.



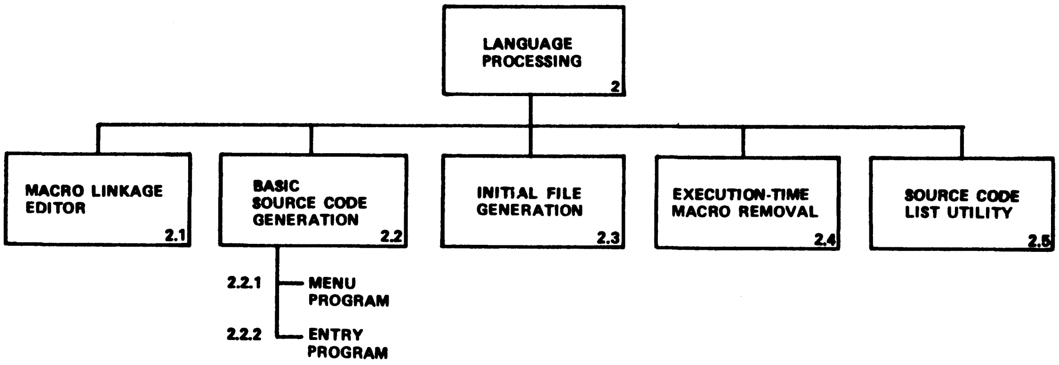
```
MMM DD,YYYY HH:MM                                  PAGE NUMBER - 1


                REPORT FORMATS IN THE FORCE DATABASE DICTIONARY
                -----------------------------------------------


SYSTEM NAME     FORMAT     TYPE     DESCRIPTION OR USE FOR THE FORMAT
-----------     ------     ----     ---------------------------------
 PAYROLL         001        T       MANPOWER SURVEY PAY SYSTEM
 PAYROLL         002        T       DEPARTMENTAL PAYROLL REPORT TITLE
 PAYROLL         100        H       SUPPLEMENTARY PAYROLL HEADER
 PAYROLL         100        D       TIME ALLOCATION REPORT DETAIL
 PAYROLL         100        S       TIME ALLOCATION REPORT STATISTICS
```

Section 2
LANGUAGE PROCESSING

PURPOSE: The FORCE Language Processing menu provides functions for generating IRIS Business BASIC and labeled source code. It allows direct access to the Linkage Editor, which expands source code according to macro command line specifications. This menu also offers functions which facilitate file generation, remove macro statements prior to program execution, and selectively list program text files.



```
LANGUAGE PROCESSING                                      FS1Ø12 MM/DD/YY
                       (Ø) RETURN TO MASTER SYSTEM CONTROL

                       (1) MACRO LINKAGE EDITOR FACILITY

                       (2) BASIC SOURCE CODE GENERATION

                       (3) INITIAL FILE GENERATION

                       (4) EXECUTION-TIME MACRO REMOVAL

                       (5) SOURCE CODE LIST UTILITY







COMMENT:
COMMAND:
MESSAGE:
```


LANGUAGE PROCESSING 2
MACRO LINKAGE EDITOR 2.1
BASIC SOURCE CODE GENERATION 2.2
INITIAL FILE GENERATION 2.3
EXECUTION-TIME MACRO REMOVAL 2.4
SOURCE CODE LIST UTILITY 2.5
2.2.1 MENU PROGRAM
2.2.2 ENTRY PROGRAM

The Language Processing menu selections provide functions for processing previously-specified Dictionary information into program source code.

1. The Macro Linkage Editor Facility generates source code by expanding macro command statements within program text files. It is also used to generate IRIS Business BASIC text files from labeled source text files.

2. BASIC Source Code Generation is used to generate labeled source code for menu and data entry programs.

3. Initial File Generation facilitates the creation of disk files by combining the BUILDXF and FORMAT functions. It is performed by accessing the Dictionary for pertinent file information.

4. Execution-Time Macro Removal is used to save program space by renumbering programs so that REMACRO, REMINFO and REMSTOP statements are overstored at execution time.

5. The Source Code List Utility selectively prints program listings for programs specified by the user.

PROCEDURE: Enter the number which corresponds to the desired function. Subsequent menu choices are shown in the Language Processing flowchart.

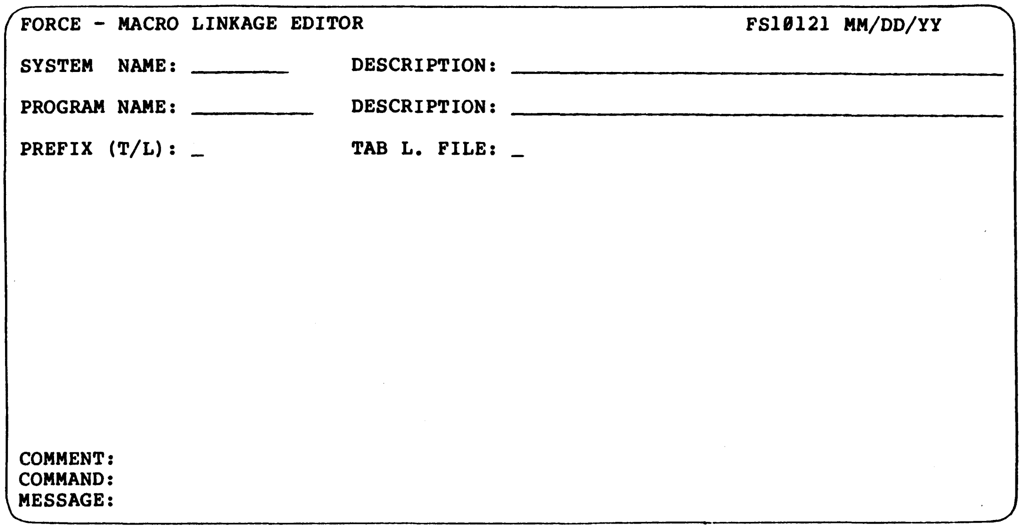
NOTES: For more information on labeled source code, see Section 8.1.

2.1 MACRO LINKAGE EDITOR FACILITY

PURPOSE: The Linkage Editor generates source code by expanding FORCE macro command statements based upon Dictionary information. It is used to process program source code that contains macro statements.

A program may be altered by changing its Dictionary specifications and processing it through the Linkage Editor. The resultant source code reflects the Dictionary modifications. A program may also be modified by editing its macro command statements and re-expanding the source code through the Linkage Editor. This function establishes the processed program in the Dictionary with the associated system.

The Linkage Editor is also used to process and expand labeled source files into the corresponding IRIS Business BASIC source files.



```
FORCE - MACRO LINKAGE EDITOR                              FS10121 MM/DD/YY

SYSTEM  NAME: _______      DESCRIPTION: _________________________________

PROGRAM NAME: _________    DESCRIPTION: _________________________________

PREFIX (T/L): _            TAB L. FILE: _










COMMENT:
COMMAND:
MESSAGE:
```

PROCEDURE: Enter the name of the system with which the program is associated. Then enter the name of the program that is to be expanded. The program name specified should not contain the L. or T. prefix.

At the Prefix entry field, specify the type of program text file to be expanded (see NOTES). Valid entries are: (T)ext or (L)abeled.

Enter L if a labeled program text file is to be converted to Business BASIC and expanded. Enter T if the source code to be expanded is Business BASIC. A RETURN at this field defaults to an L entry.

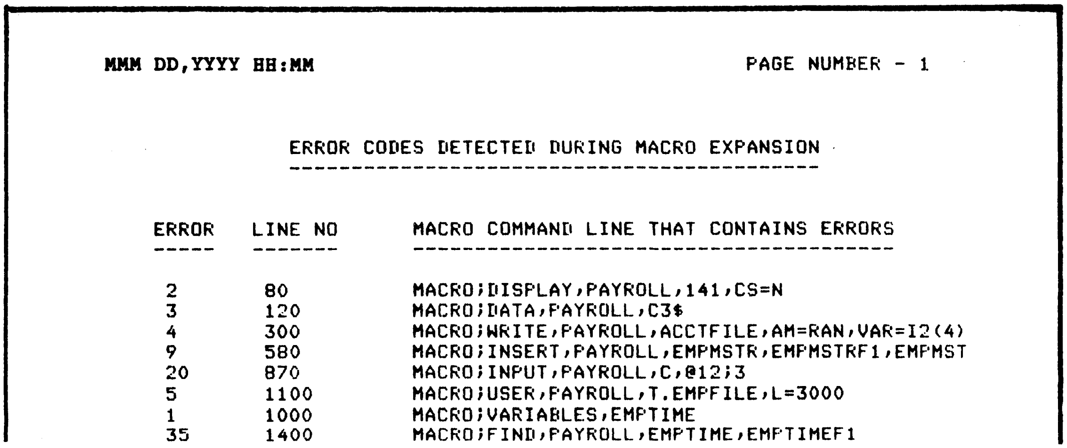
The Tab L. File entry directs FORCE to align a labeled text file as it is processed. Program statements in the labeled text file are indented to enhance the readability of the source code. If truncation occurs because of indentation, a message is displayed on the CRT and comments are inserted in the expanded source code to indicate each truncated line.

Enter (Y)es or (N)o at the Tab L. File field. A RETURN defaults to an entry of (N)o. This entry applies only when processing an L. file.

If the Linkage Editor encounters errors in the source file's macro commands, the expansion process is terminated. The original text file is not altered; instead, a summary of the command line error(s) is displayed on the screen. The summary notes the error code number (see Table 2-1), the line containing the invalid macro, and the macro statement.

If the Linkage Editor encounters undefined or duplicate labels when processing a labeled text file, the affected lines are displayed on the CRT with the appropriate message.

The user may print the error report by entering P at the Command Line. A sample macro error report is shown below.



```
MMM DD,YYYY HH:MM                                  PAGE NUMBER - 1


                  ERROR CODES DETECTED DURING MACRO EXPANSION
                  -------------------------------------------


      ERROR    LINE NO      MACRO COMMAND LINE THAT CONTAINS ERRORS
      -----    -------      ---------------------------------------

       2        80          MACRO;DISPLAY,PAYROLL,141,CS=N
       3        120         MACRO;DATA,PAYROLL,C3$
       4        300         MACRO;WRITE,PAYROLL,ACCTFILE,AM=RAN,VAR=I2(4)
       9        580         MACRO;INSERT,PAYROLL,EMPMSTR,EMPMSTRF1,EMPMST
       20       870         MACRO;INPUT,PAYROLL,C,@12;3
       5        1100        MACRO;USER,PAYROLL,T.EMPFILE,L=3000
       1        1000        MACRO;VARIABLES,EMPTIME
       35       1400        MACRO;FIND,PAYROLL,EMPTIME,EMPTIMEF1
```

<u>NOTES</u>: All FORCE-generated program text files use a standard naming convention. The file has the same name as the program, with an L. prefix to denote that it is a labeled text file. For example, for a program defined as APM53 to the Dictionary, L.APM53 is generated on disk as the labeled program source file.

Subsequently, the L. program is processed by the Linkage Editor, which creates a corresponding IRIS Business BASIC text file, and expands its macros. A prefix of T. replaces L. in the name of the expanded BASIC text file. Using the previous example, the L.APM53 text file would be expanded as T.APM53 into a BASIC text file.

An expanded BASIC program text file does not overwrite its labeled program text file counterpart. Both programs remain on disk (in an L. and a T. file), and either may be expanded at any time. However, each time a program is expanded, it overwrites the current T. text file.

The Linkage Editor does not affect manual modifications to a program during re-expansion. It only re-expands the program's macros.

Macros may be expanded from any Business BASIC or labeled text file with a T. or L. prefix.

For more information on labeled source code, see 8.1.

Table 2-1 summarizes the macro error codes.

MACRO ERROR CODES TABLE 2-1

TABLE 2-1. MACRO ERROR CODES

Error No.	Description
01	INVALID SYSTEM NAME System name specified within the macro command line was not located in the FORCE Dictionary files.
02	INVALID SCREEN DISPLAY NUMBER Screen display number specified within the macro command line is not valid for the specified system.
03	INVALID DATA ELEMENT Data element name specified within the macro command line is not valid for the specified system.
04	INVALID DATA FILE NAME Data file name specified within the macro command line is not valid for the specified system.
05	INVALID PROGRAM NAME Program name specified within the macro command line is not valid for the specified system.
06	INVALID USER FILE NAME User file name specified within the macro command line is not valid for the specified system.
07	INVALID MACRO TYPE Macro type specified within the macro command line is not valid for the specified system.
08	RECORD LAYOUT NOT YET ESTABLISHED Data file specified does not yet have any data elements assigned.
09	CHANNEL NUMBER MISSING The macro command type specified requires that a channel number be specified using the format #=n.
10	ACCESS VARIABLE MISSING The macro command type specified requires that a variable to access the file be given in the format VAR=element.
11	ACCESS METHOD MISSING The macro command type specified requires that the access method desired be given in the format AM=type (type=RAN,SEQ,DIR,RND).
12	INCORRECT ACCESS METHOD The access method specified within the macro command line is invalid.

TABLE 2-1. MACRO ERROR CODES (Cont)

Error No.	Description
13	END OF FILE LOCATED When searching for the corresponding ending macro command line, the end of the file was located.
14	INVALID ENDING MACRO LOCATED REMSTOP line does not match the remacro command line format.
15	NESTED MACRO COMMANDS A second macro command line was encountered while searching for the REMSTOP line.
16	INDEX NUMBER MISSING No index number specified within the macro command line.
17	MAT READ/WRITE INVALID Due to the record layout of the data file, a MAT READ/WRITE is not possible.
18	INVALID DATA ELEMENT IN RECORD LAYOUT Data element specified within the record layout in the macro command line is not valid for the corresponding system.
19	INVALID LINE NUMBER GOSUB or line number specified is out of the physical range allowed.
20	CURSOR POSITION MISSING Relative address for horizontal and vertical cursor positioning is missing, or the format is invalid.
21	AUTHORIZATION LEVEL TOO LOW Macro type specified in command line is a higher level than the user is authorized to access.
22	NO MAXIMUM SIZE DEFINED FOR A SUBDIVIDED STRING No size (maximum) defined for a subdivided string.
23	INVALID SUBDIVISION NUMBER OR INCORRECT FORMAT Subdivision number specified is invalid for data element specified, or format is incorrect. Var$(n).
24	INSTALLED FORCE LEVEL DOES NOT ALLOW ACCESS TO MACRO FORCE level installed does not allow access to macro type specified.

TABLE 2-1. MACRO ERROR CODES (Cont)

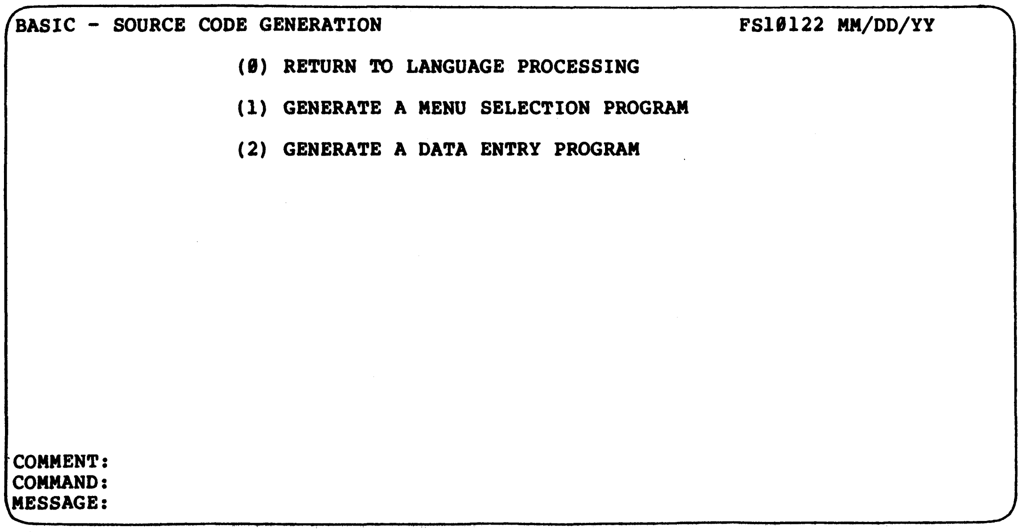
Error No.	Description
25	SPECIFIED MACRO NOT ENABLED FOR INSTALLED VERSION FORCE level installed does not have specified macro type enabled.
26	NO DESCRIPTION SPECIFIED FOR SUBDIVIDED ELEMENT No description specified for subdivided data element that was specified within the macro command line.
27	UNAUTHORIZED USE OF SUBDIVIDED STRING ENCOUNTERED Unauthorized use of subdivided string.
28	SUBDIVISION SIZE NOT YET ESTABLISHED Subdivision size not yet established for portion of subdivided string.
29	FILE FORMAT TYPE MISSING OR INVALID
30	ORGANIZATION TYPE MISSING OR INVALID The file organization within the macro command line is missing or invalid.
31	INVALID "KEY=" PARAMETER The "KEY=" parameter is not valid for the specified search.
32	INVALID RECORD LAYOUT FOR A TEXT FILE The record layout specified is not valid for a text file. It must be a single string.
33	DIRECTORY NUMBER MISSING OR INVALID The directory number is not specified, or is invalid for the specified search.
34	INVALID DATA ELEMENT TYPE WITHIN KEY CONSTRUCT The data element type is not valid for the key construct. These may only be constructed of numeric or alphanumeric elements.
35	REMINFO LINE MISSING OR INVALID The REMINFO command line was not entered, or is invalid for the specified macro.
36	INVALID "FOUND=" REMINFO PARAMETER The "FOUND=" parameter is invalid for the specified search macro.

TABLE 2-1. MACRO ERROR CODES (Cont)

Error No.	Description
37	CONFLICT IN MACROINFO LINES - "FOUND=" AND "G=" An incongruity was encountered between the "FOUND=" and the "G=" parameters. A callable routine specified by a "G=" parameter may not be assigned another exit location by a "FOUND=" parameter.
38	INVALID "ERR=" REMINFO PARAMETER The "ERR=" parameter in the reminfo command line is invalid.
39	INVALID USER MACRO PARAMETER The user macro parameter specified is invalid.
40	UNDEFINED USER MACRO PARAMETER No value provided for a substituted user macro parameter.
41	INVALID REPORT FORMAT SPECIFIED Output format number specified in the macro command line is not valid for the specified system.
42	INVALID 'SKIP=NN' PARAMETER SPECIFIED Invalid parameter to skip vertical lines.
43	INVALID SYNONYM NAME The synonym name specified is invalid for the report level.
44	SYSTEM ERROR OCCURRED DURING INDEX CONSTRUCTION Insufficient allocation for report index file construction.
45	TOO MANY ELEMENT/SYNONYMS/CALCS/STATISTICS IN PROGRAM The number of program variables exceeds the maximum allowable.
46	PROGRAM SPECIFIED NOT A (R)EPORT OR (E)NTRY PROGRAM The program specified in the macro command line is not established in the Dictionary as a (R)eport or an (E)ntry program.
47	INVALID DATA ELEMENT TYPE FOR SPECIFIED MACRO Matrices and subdivided strings are invalid for an Input Macro.

2.2 BASIC SOURCE CODE GENERATION

PURPOSE: Source code for menu and data entry programs is generated automatically by executing functions available from this facility.



```
BASIC - SOURCE CODE GENERATION                                FS10122 MM/DD/YY

                     (0) RETURN TO LANGUAGE PROCESSING

                     (1) GENERATE A MENU SELECTION PROGRAM

                     (2) GENERATE A DATA ENTRY PROGRAM




COMMENT:
COMMAND:
MESSAGE:
```

BASIC SOURCE
CODE GENERATION
2.2

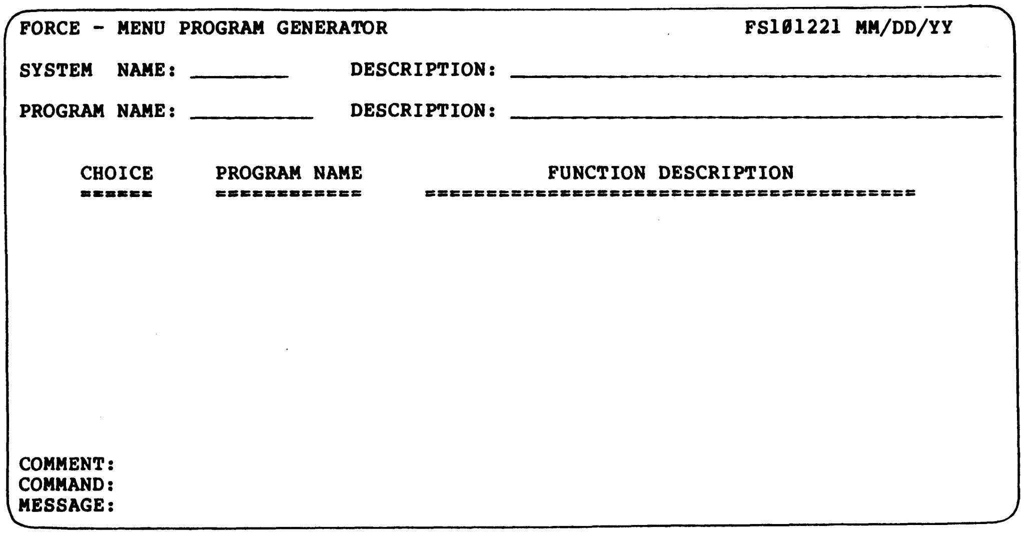
PROCEDURE: Select and enter the number which corresponds to the desired function.

NOTES: Functions under this menu generate labeled source code (see 8.1).

2.2.1 GENERATE A MENU SELECTION PROGRAM

PURPOSE: This function generates menu selection program source code automatically, after the user specifies the programs to which the menu program chains. Generated menu programs are standardized in structure and interaction. Menu selections are centered horizontally and vertically on each generated menu screen.

Menu programs are generated in labeled source code.



```
FORCE - MENU PROGRAM GENERATOR                          FS101221 MM/DD/YY

SYSTEM  NAME: ________      DESCRIPTION: ________________________________________

PROGRAM NAME: __________    DESCRIPTION: ________________________________________

     CHOICE     PROGRAM NAME          FUNCTION DESCRIPTION
     ======     ============     ========================================













COMMENT:
COMMAND:
MESSAGE:
```

PROCEDURE: Prior to executing this function, the menu program name must have been established within the Dictionary (see 1.5.1).

Enter the system with which the menu program is associated. Then enter the menu program name.

The first menu selection (CHOICE 0) appears automatically. It reflects the Exit Program specified when the program was established (see 1.5.1). A RETURN designates the displayed information as valid. If a different exit location is required, enter the appropriate program name.

In the FUNCTION DESCRIPTION column, enter the corresponding program description exactly as it is to appear in the generated menu.

Programs specified as menu choices need not be established in the dictionary.

If a program specified as a menu choice has been established in the Dictionary, the program's description is displayed in the FUNCTION DESCRIPTION column. That description may be retained by pressing RETURN, or a new description may be entered.

Specify a program name and a description for each required menu choice.

To indicate that all required menu selections have been specified, enter RETURN at the PROGRAM NAME column.

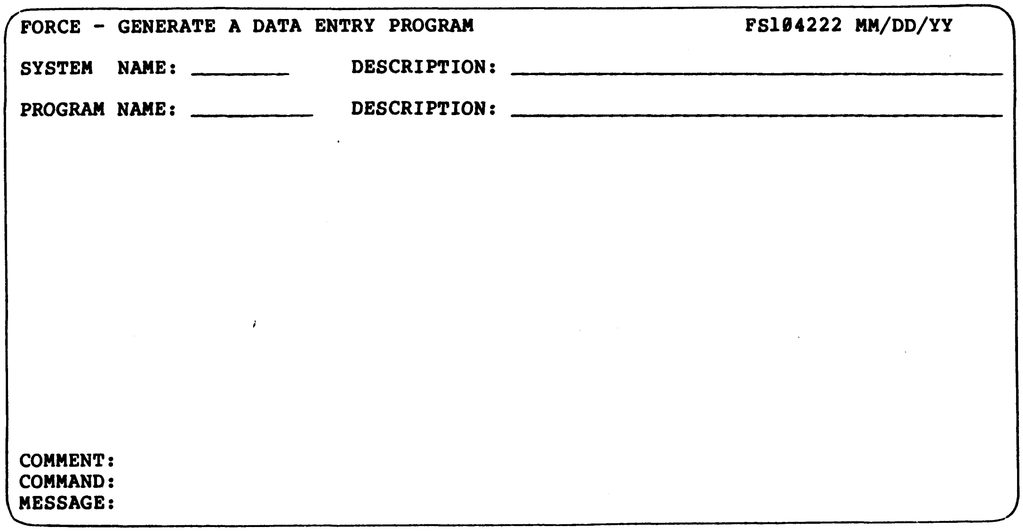
NOTES: Menu programs are generated using the standard FORCE naming convention for text files. The preface L. is added to each program name. For example, if SWI25 is the program name defined to FORCE, the text file name of the generated program is L.SWI25 on disk.

A regenerated menu program overwrites the current source file on disk.

2.2.2 GENERATE A DATA ENTRY PROGRAM

PURPOSE: This function generates data entry program source code automatically, after the user specifies the program's required functional components. A generated entry program may be used to add, modify, delete or query data records. The entry program functions according to specifications defined under Program Definition. Generated entry programs are standardized in structure and interaction.

Data entry programs are generated in labeled source code.



```
FORCE - GENERATE A DATA ENTRY PROGRAM                    FS104222 MM/DD/YY

SYSTEM  NAME: _______      DESCRIPTION: ______________________________

PROGRAM NAME: _________    DESCRIPTION: ______________________________












COMMENT:
COMMAND:
MESSAGE:
```

PROCEDURE: Prior to executing this function, the required components and procedural logic of the data entry program must be established within the Dictionary (see 1.5).

Enter the name of the system with which the entry program is associated. Then enter the name of the data entry program to be generated.

NOTES: Data entry source code is generated on disk using the standard FORCE naming conventions for text files. The preface L. is added to each program name. For example, if TWI26 is the program name defined to FORCE, the text file name of the generated program name is L.TWI26 on disk.

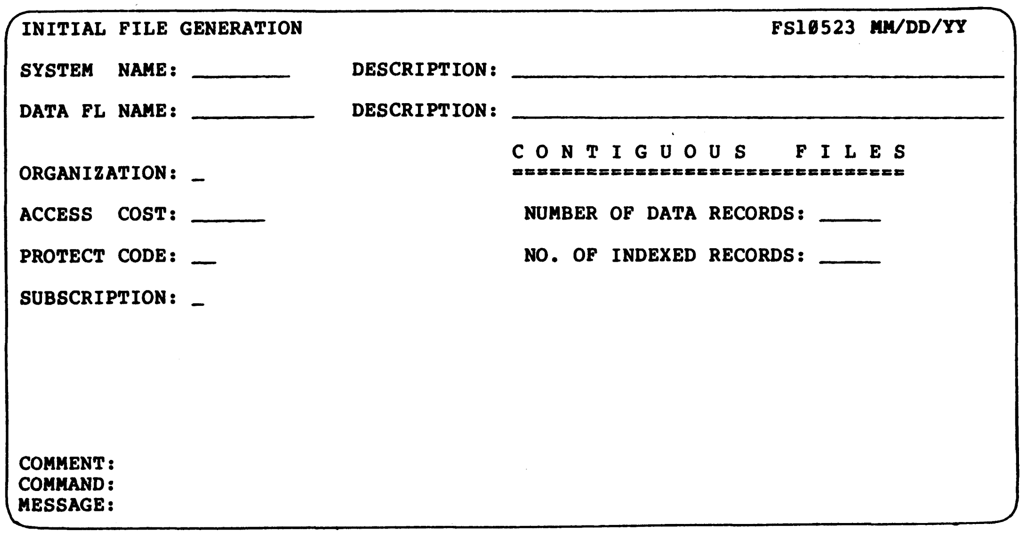
A regenerated data entry program overwrites the current source file on disk.

2.3 INITIAL FILE GENERATION

PURPOSE: This function facilitates the creation of any type of disc file that is accessible by an IRIS Business BASIC program. This operation combines the BUILDXF and FORMAT functions, and is performed by accessing the Dictionary for pertinent file information.

The file created by this function is the Disk File specified when establishing a file in the Dictionary (see 1.4.1).

Initial File Generation may also be used to re-initialize files to their original empty state.



```
INITIAL FILE GENERATION                                  FS10523 MM/DD/YY

SYSTEM  NAME: ________    DESCRIPTION: ______________________________________

DATA FL NAME: __________  DESCRIPTION: ______________________________________

                                       CONTIGUOUS   FILES
ORGANIZATION: _                        ==============================

ACCESS  COST: ______                    NUMBER OF DATA RECORDS: _____

PROTECT CODE: __                        NO. OF INDEXED RECORDS: _____

SUBSCRIPTION: _





COMMENT:
COMMAND:
MESSAGE:
```

PROCEDURE: Prior to initializing a file with this function, the file and its record layout must be defined to the Dictionary (see 1.4.1 and 1.4.2).

Enter the name of the system which contains the file to be generated.

The name entered at the Data File entry depends on the structure of the associated disk file specified in 1.4.1. For disk files that contain multiple files (e.g., data files and indexes), the generated file must employ a record layout that accommodates the required file structure. The data file specified in this entry directs FORCE to use that file's record layout in generation of the file.

If the disk file contains data records and indexed records, specify the name of the data file (not its associated disk file) so the size of that file's record layout may be computed.

If the data portion of a contiguous disk file contains multiple data record layouts, specify the file with the longest record layout. This assures sufficient record length allocation for all data records.

For disk files with multiple indexes, any of the FORCE file names may be used.

After entry of the appropriate file name, FORCE displays the related disk file name. This is the file that is actually generated on disk.

At the Organization entry field, specify the IRIS file organization. The valid entries are: (C)ontiguous, (F)ormatted and (T)ext.

Enter the Protect Code and Access Cost for the file to be generated.

If the organization is contiguous or formatted, FORCE prompts for entry of the file subscription parameter. This data is used to determine appropriate field and record lengths, using the established record layout. Valid entries are: (Y)es, which establishes double subscription on all string variables, and (N)o, which disallows double subscription on string variables.

For contiguous files, FORCE requires entry of the Number of Data Records and Indexed Records. Enter the number of data records.

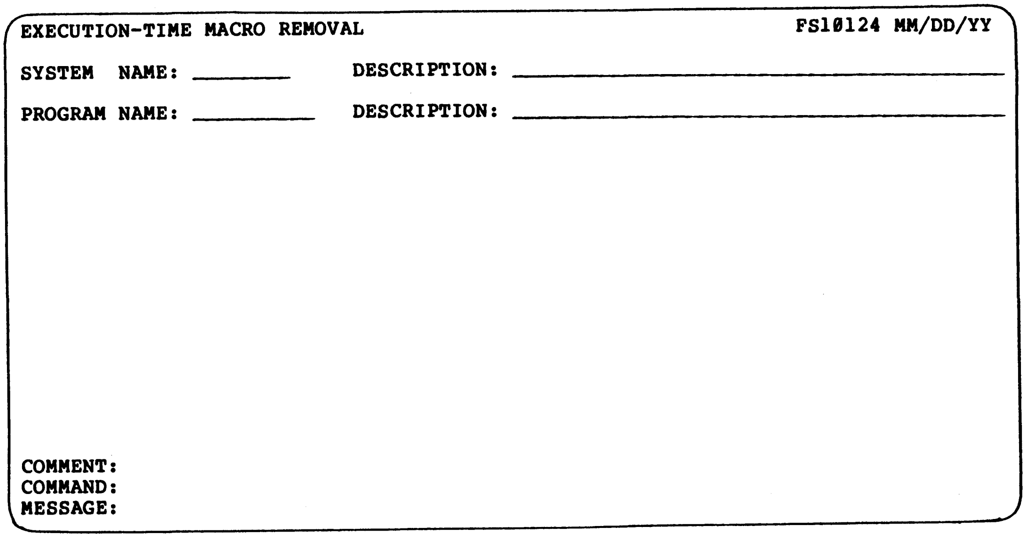
If space is to be reserved in this file for one or more directories, the maximum number of indexed records must be specified. Upon entry of a positive number of indexed records, underscores appear prompting for the key size (in bytes) of the first directory. This entry field is repeated for up to 15 directories. To signal that there are no more directories, press RETURN without entering any data.

NOTES: For all file types, the appropriate access cost and protection code attributes are associated with the generated file.

If the specified organization entry is formatted, a formatted file is created with record zero corresponding to the file layout referenced in the Dictionary. For a contiguous file organization, the record length and the file size are established, and any required index directories are created and linked to the Free Chain List.

2.4 EXECUTION-TIME MACRO REMOVAL

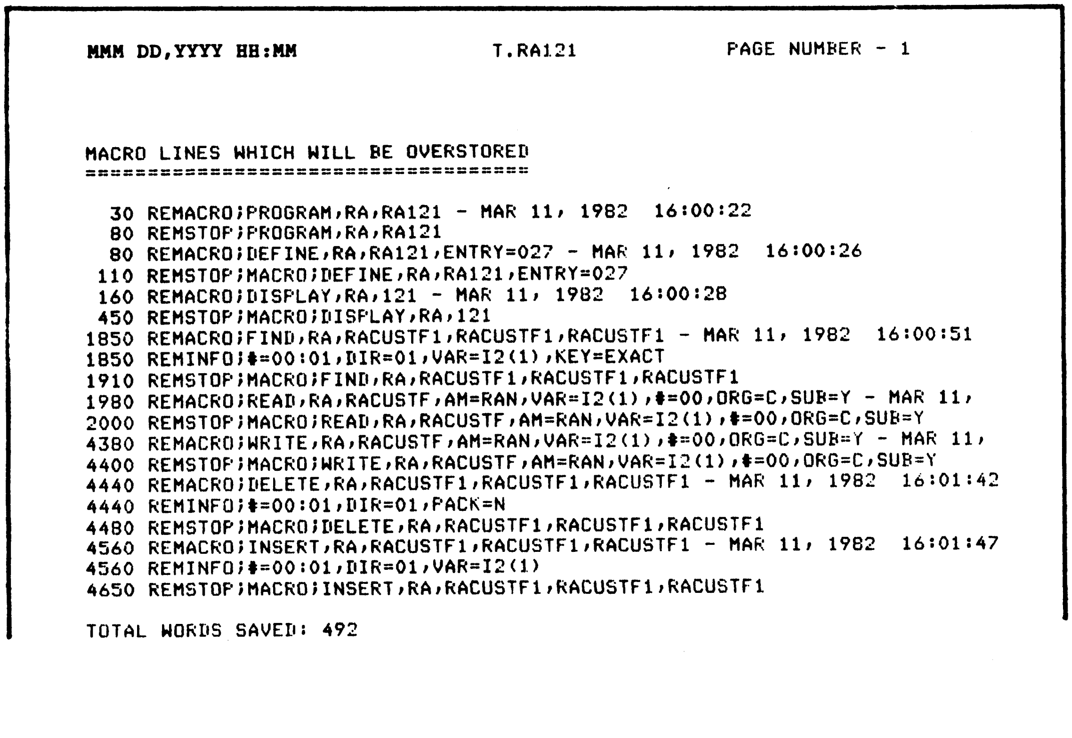
PURPOSE: This function is used to save program space by overstoring (at execution time) REMACRO, REMINFO and REMSTOP statements. The size of expanded programs may be reduced by Execution-Time Macro Removal.



```
EXECUTION-TIME MACRO REMOVAL                                FS10124 MM/DD/YY

SYSTEM  NAME: ________     DESCRIPTION: ______________________________________

PROGRAM NAME: _________    DESCRIPTION: ______________________________________





COMMENT:
COMMAND:
MESSAGE:
```

PROCEDURE: Enter the name of the system with which the program to be processed is associated. Then enter the name of the program.

When Execution-Time Macro Removal is complete, a report is printed, indicating the lines to be overstored and the number of words saved. Below is a sample macro removal report.



```
MMM DD,YYYY HH:MM              T.RA121             PAGE NUMBER - 1



MACRO LINES WHICH WILL BE OVERSTORED
====================================

  30 REMACRO;PROGRAM,RA,RA121 - MAR 11, 1982  16:00:22
  80 REMSTOP;PROGRAM,RA,RA121
  80 REMACRO;DEFINE,RA,RA121,ENTRY=027 - MAR 11, 1982  16:00:26
 110 REMSTOP;MACRO;DEFINE,RA,RA121,ENTRY=027
 160 REMACRO;DISPLAY,RA,121 - MAR 11, 1982  16:00:28
 450 REMSTOP;MACRO;DISPLAY,RA,121
1850 REMACRO;FIND,RA,RACUSTF1,RACUSTF1,RACUSTF1 - MAR 11, 1982  16:00:51
1850 REMINFO;#=00:01,DIR=01,VAR=I2(1),KEY=EXACT
1910 REMSTOP;MACRO;FIND,RA,RACUSTF1,RACUSTF1,RACUSTF1
1980 REMACRO;READ,RA,RACUSTF,AM=RAN,VAR=I2(1),#=00,ORG=C,SUB=Y - MAR 11,
2000 REMSTOP;MACRO;READ,RA,RACUSTF,AM=RAN,VAR=I2(1),#=00,ORG=C,SUB=Y
4380 REMACRO;WRITE,RA,RACUSTF,AM=RAN,VAR=I2(1),#=00,ORG=C,SUB=Y - MAR 11,
4400 REMSTOP;MACRO;WRITE,RA,RACUSTF,AM=RAN,VAR=I2(1),#=00,ORG=C,SUB=Y
4440 REMACRO;DELETE,RA,RACUSTF1,RACUSTF1,RACUSTF1 - MAR 11, 1982  16:01:42
4440 REMINFO;#=00:01,DIR=01,PACK=N
4480 REMSTOP;MACRO;DELETE,RA,RACUSTF1,RACUSTF1,RACUSTF1
4560 REMACRO;INSERT,RA,RACUSTF1,RACUSTF1,RACUSTF1 - MAR 11, 1982  16:01:47
4560 REMINFO;#=00:01,DIR=01,VAR=I2(1)
4650 REMSTOP;MACRO;INSERT,RA,RACUSTF1,RACUSTF1,RACUSTF1

TOTAL WORDS SAVED: 492
```

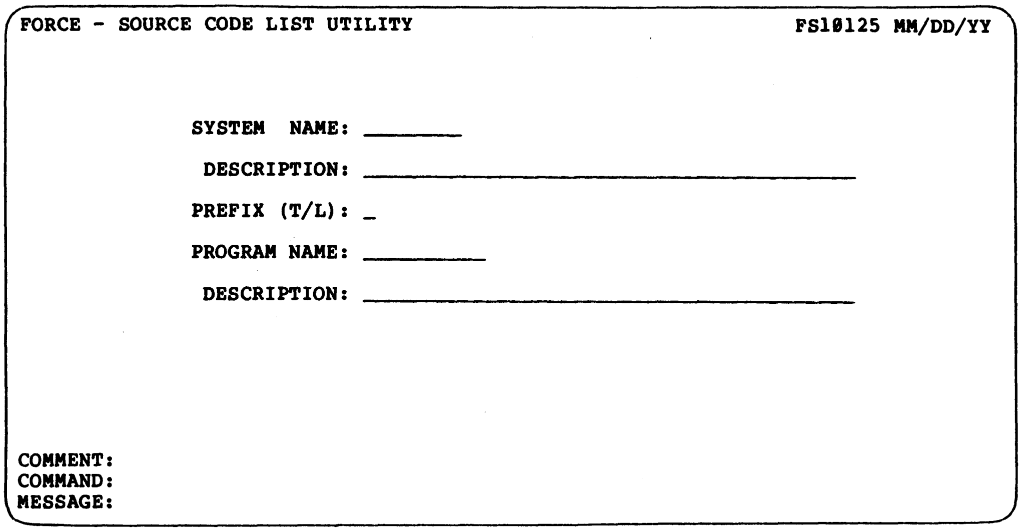
NOTES: Execution-Time Macro Removal is a process by which each REMACRO, REMINFO and REMSTOP statement is assigned the same line number as the succeeding line. These lines are then overstored when the source code is loaded for execution.

REMACRO lines are directives to the Linkage Editor to expand the appropriate source code. They are unexecutable REM statements in the generated source code and therefore are expendable.

To recover any lines overstored with this function, re-expand the program through the Linkage Editor.

2.5 SOURCE CODE UTILITY

PURPOSE: This facility enables the user to selectively print listings of program text files. BASIC text files (prefixed with T.) or labeled source code (prefixed with L.) may be listed for any or all programs within a system.



```
FORCE - SOURCE CODE LIST UTILITY                              FS10125 MM/DD/YY


                     SYSTEM  NAME: ________

                      DESCRIPTION: ________________________________________

                     PREFIX (T/L): _

                     PROGRAM NAME: __________

                      DESCRIPTION: ________________________________________






COMMENT:
COMMAND:
MESSAGE:
```

PROCEDURE: Enter the name of the system with which the program text files are associated.

Entry of ? at any of the selection fields invokes a help module which provides information about the list utility.

At the Prefix field, specify whether the text files to be listed are in (T)ext or (L)abeled source code. A RETURN at this field defaults to an entry of (L)abeled.

If only one program is to be listed, enter the name of the program (as defined to FORCE) at the Program Name field. The specified program is then listed to the printer according to FORCE printer output assignments.

To list a number of programs, enter RETURN at the Program Name field. A sorted inventory of all programs from the specified system is then accumulated and displayed on the CRT in groups of 36. This inventory may then be modified by removing any programs for which a listing is not required.

After the initial list of programs has been displayed, a second help module may be invoked.

Enter ? for operator assistance. This help screen aids in a final determination of program source files to be listed. Commands described in the help module are used to modify the list of programs prior to executing the list function.

Valid screen modification commands are:

A All - erase all program names from the screen
E Execute - initiate listing of the selected programs
P Page - store the current screen and display the next list
R Restart - restart the review of the current page
W Wrap - store all reviewed programs and restart selection
nn nn - erase file(s) nn from the screen

Each page (group of 36 programs) should be examined before entering the (E)xecute command. This facility only prints listings of those programs that have been reviewed.

NOTES: The Source Code List Facility incorporates labeled source code symbols which can be used to format a printed text file. These symbols correspond closely to ANSI standards as print control characters.

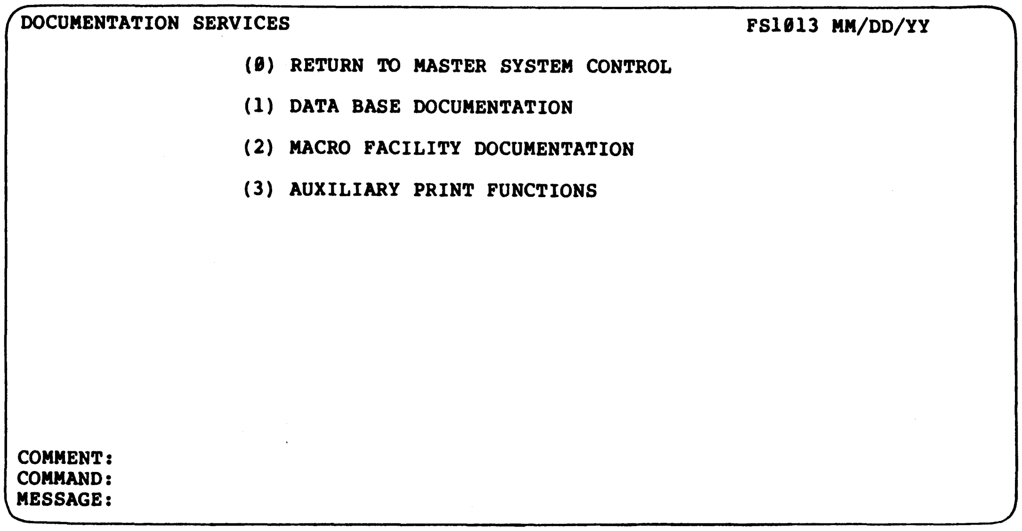
The symbols: + (plus sign), * (asterisk) and . (period) indicate a commented line when placed in the first position of a line of labeled source code. When listing a program containing these symbols, a + symbol initiates a form feed (new page). The * symbol forces a top of form when encountered within the last ten lines of the page (to ensure that a section of code is not split between pages).

FORCE checks the program names specified for listing to ensure that they have been defined to the Dictionary. Program source code can be listed for any program that has been defined to the Dictionary. If a program specified for listing has not been generated or written, or if it cannot be located on disk with the indicated L. or T. prefix, FORCE prints a page containing the appropriate message.

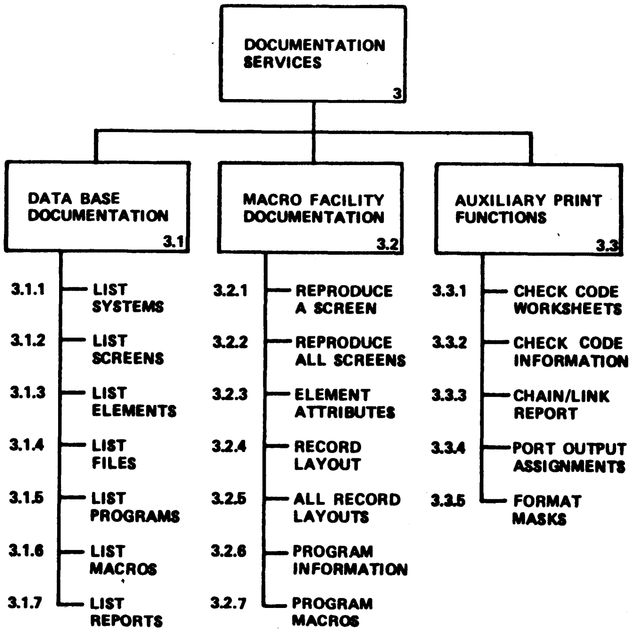
When listing multiple programs with this facility, the first page of each program listing is begun on an even-numbered page. This enables the user to begin each program listing with an "up" page, by adjusting the printer paper so that the first fold of the initial listing creates an "up" page.

Section 3
DOCUMENTATION SERVICES

PURPOSE: Documentation is available on virtually all information within the Dictionary. Supporting system documentation enables the FORCE user to plan and monitor application development with hard-copy reports. Reports are also used to provide status updates to customers and management. All documentation is retrieved from the Dictionary, and reflects the current data base.



```
DOCUMENTATION SERVICES                                          FS1Ø13 MM/DD/YY
                         (Ø) RETURN TO MASTER SYSTEM CONTROL

                         (1) DATA BASE DOCUMENTATION

                         (2) MACRO FACILITY DOCUMENTATION

                         (3) AUXILIARY PRINT FUNCTIONS










COMMENT:
COMMAND:
MESSAGE:
```



The three functional areas of Documentation Services are Data Base Documentation, Macro Facility Documentation and Auxiliary Print Functions.

1. Data Base Documentation provides selected lists of: systems, screens, data elements, files, programs, program macros and report formats.

2. Macro Facility Documentation gives detailed reports on system components: screen reproductions, data element attributes, record layouts, program information and system macros.

3. Reports on port output assignments, data element masks, check codes and program chain linkage are available from Auxiliary Print Functions.

PROCEDURE: Select and enter the number which corresponds to the desired function. Subsequent menu choices are shown in the Documentation Services flowchart.

NOTES: Reports obtained under Documentation Services document system information that has been established within the Dictionary. These reports should not be confused with the sophisticated report generation facility under Report Preparation.

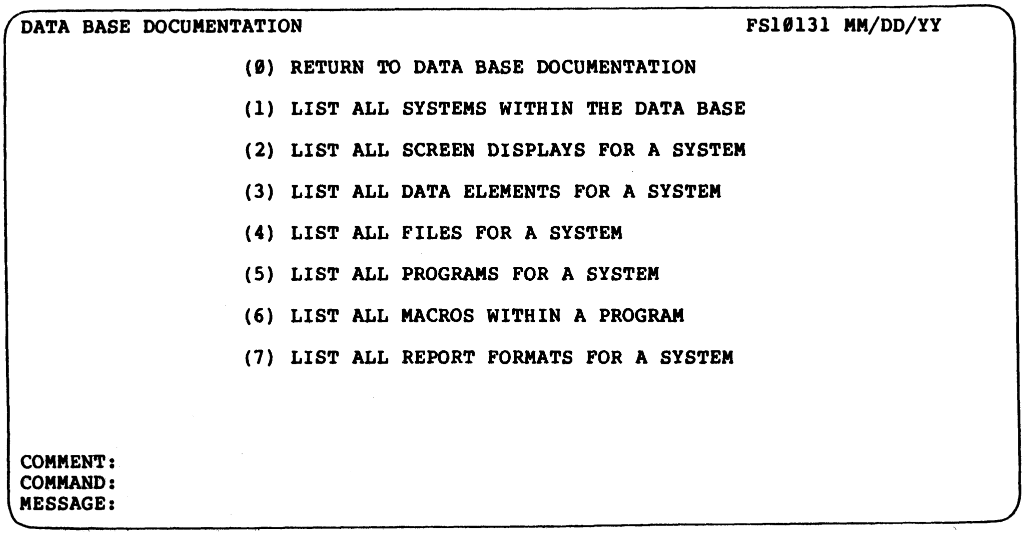
Documentation Services provides reports formatted for paper that is at least 8 1/2 x 11 inches.

If the assigned printer is busy, FORCE displays an appropriate message and maintains the specified job until the printer becomes available.

Printer output assignments are maintained as a FORCE manager function (see Section 6.2.4).

3.1 DATA BASE DOCUMENTATION

PURPOSE: Data Base Documentation provides selective reports of Dictionary information listing systems, screens, data elements, files, programs, macros and report formats. This menu allows selection of the appropriate function.



```
DATA BASE DOCUMENTATION                                    FS1Ø131 MM/DD/YY

                      (Ø) RETURN TO DATA BASE DOCUMENTATION

                      (1) LIST ALL SYSTEMS WITHIN THE DATA BASE

                      (2) LIST ALL SCREEN DISPLAYS FOR A SYSTEM

                      (3) LIST ALL DATA ELEMENTS FOR A SYSTEM

                      (4) LIST ALL FILES FOR A SYSTEM

                      (5) LIST ALL PROGRAMS FOR A SYSTEM

                      (6) LIST ALL MACROS WITHIN A PROGRAM

                      (7) LIST ALL REPORT FORMATS FOR A SYSTEM



COMMENT:
COMMAND:
MESSAGE:
```

PROCEDURE: Enter the number which corresponds to the desired function.

3.1.1 LIST ALL SYSTEMS WITHIN THE DATA BASE

PURPOSE: Utilizing this function, an alphabetic list of all system names within the Dictionary may be directed to a printer or CRT screen. The system description is also shown in the report.

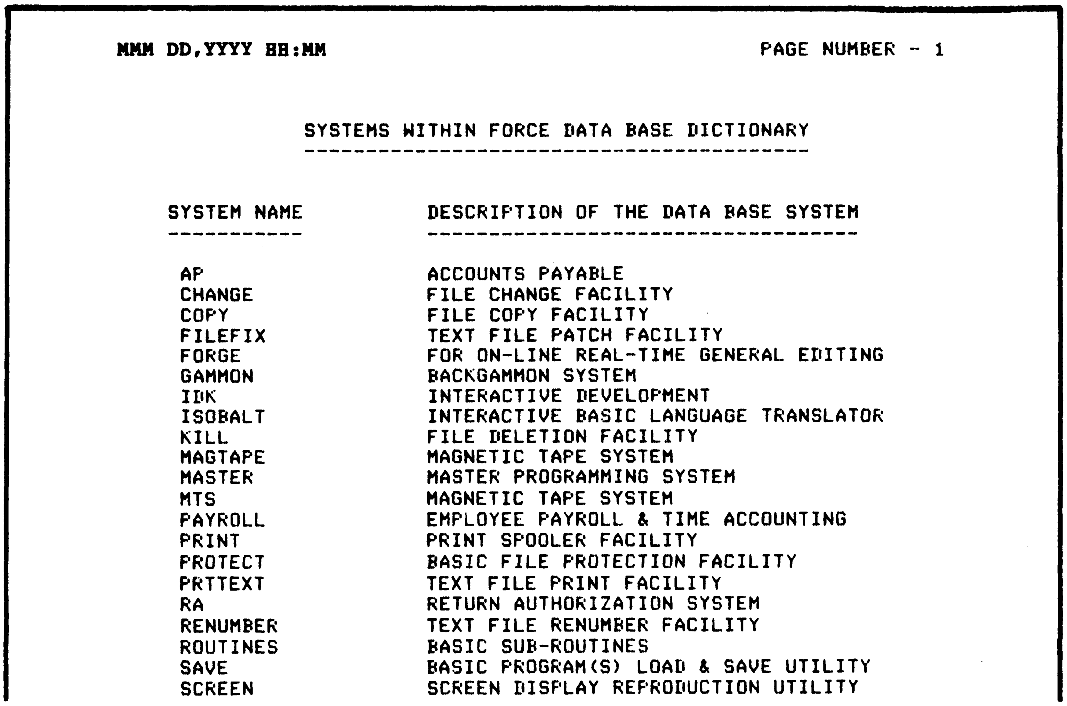
PROCEDURE: The Comment Line prompts the user to direct report output to the CRT screen (S) or the printer (P). Enter S or P to initiate the report.

If S is specified, systems are listed on the screen. Each RETURN displays the next portion of the list. The final screen segment indicates that no more information is available.

If P is specified, FORCE asks if there is 8 1/2 x 11 paper in the printer. A (Y)es initiates printing of the report and a (N)o returns to the Data Base Documentation menu.

NOTES: The report output for this function contains the same data, whether it is directed to the screen or the printer.

Below is a sample report.



```
MMM DD,YYYY HH:MM                                  PAGE NUMBER - 1


                 SYSTEMS WITHIN FORCE DATA BASE DICTIONARY
                 -----------------------------------------


    SYSTEM NAME          DESCRIPTION OF THE DATA BASE SYSTEM
    -----------          -----------------------------------

     AP                  ACCOUNTS PAYABLE
     CHANGE              FILE CHANGE FACILITY
     COPY                FILE COPY FACILITY
     FILEFIX             TEXT FILE PATCH FACILITY
     FORGE               FOR ON-LINE REAL-TIME GENERAL EDITING
     GAMMON              BACKGAMMON SYSTEM
     IDK                 INTERACTIVE DEVELOPMENT
     ISOBALT             INTERACTIVE BASIC LANGUAGE TRANSLATOR
     KILL                FILE DELETION FACILITY
     MAGTAPE             MAGNETIC TAPE SYSTEM
     MASTER              MASTER PROGRAMMING SYSTEM
     MTS                 MAGNETIC TAPE SYSTEM
     PAYROLL             EMPLOYEE PAYROLL & TIME ACCOUNTING
     PRINT               PRINT SPOOLER FACILITY
     PROTECT             BASIC FILE PROTECTION FACILITY
     PRTTEXT             TEXT FILE PRINT FACILITY
     RA                  RETURN AUTHORIZATION SYSTEM
     RENUMBER            TEXT FILE RENUMBER FACILITY
     ROUTINES            BASIC SUB-ROUTINES
     SAVE                BASIC PROGRAM(S) LOAD & SAVE UTILITY
     SCREEN              SCREEN DISPLAY REPRODUCTION UTILITY
```

3.1.2 LIST ALL SCREEN DISPLAYS FOR A SYSTEM

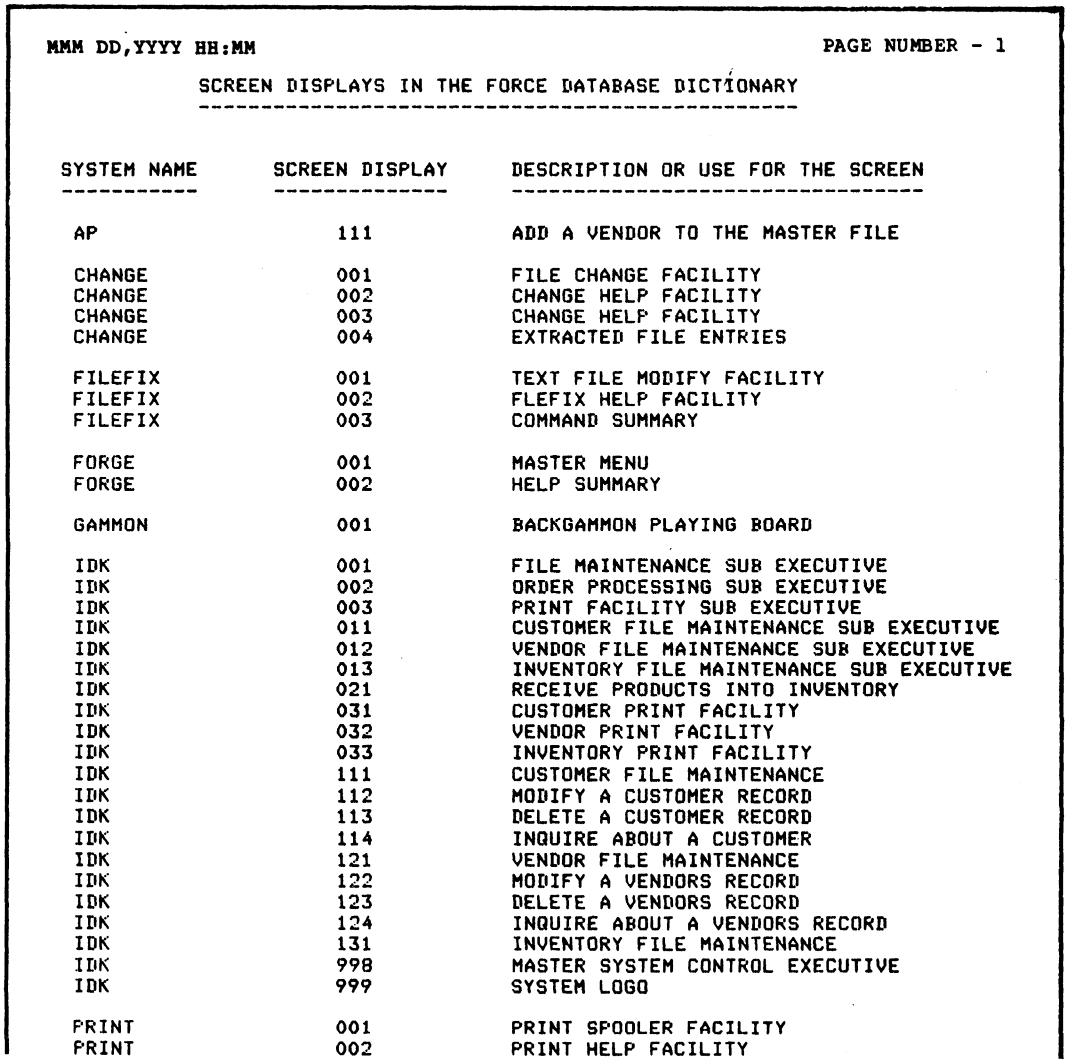
PURPOSE: This function prints a selected list of screen display numbers and their descriptions. Screens may be listed for a specific system, or all screens within the Dictionary may be listed. The list of system screens is arranged in ascending order by screen numbers.

PROCEDURE: Enter the name of the system for which screen displays are to be listed.

Enter RETURN to print a list of all screen displays, beginning alphabetically with the first associated system name.

After specification of the report output, FORCE asks if the printer uses 8 1/2 x 11 paper. A (Y)es initiates printing of the report, and a (N)o returns to the Data Base Documentation menu.

NOTES: Below is a sample listing of screen displays.



```
MMM DD,YYYY HH:MM                                              PAGE NUMBER - 1
                 SCREEN DISPLAYS IN THE FORCE DATABASE DICTIONARY
                 ------------------------------------------------

  SYSTEM NAME       SCREEN DISPLAY      DESCRIPTION OR USE FOR THE SCREEN
  -----------       --------------      ---------------------------------

   AP                    111            ADD A VENDOR TO THE MASTER FILE

   CHANGE                001            FILE CHANGE FACILITY
   CHANGE                002            CHANGE HELP FACILITY
   CHANGE                003            CHANGE HELP FACILITY
   CHANGE                004            EXTRACTED FILE ENTRIES

   FILEFIX               001            TEXT FILE MODIFY FACILITY
   FILEFIX               002            FLEFIX HELP FACILITY
   FILEFIX               003            COMMAND SUMMARY

   FORGE                 001            MASTER MENU
   FORGE                 002            HELP SUMMARY

   GAMMON                001            BACKGAMMON PLAYING BOARD

   IDK                   001            FILE MAINTENANCE SUB EXECUTIVE
   IDK                   002            ORDER PROCESSING SUB EXECUTIVE
   IDK                   003            PRINT FACILITY SUB EXECUTIVE
   IDK                   011            CUSTOMER FILE MAINTENANCE SUB EXECUTIVE
   IDK                   012            VENDOR FILE MAINTENANCE SUB EXECUTIVE
   IDK                   013            INVENTORY FILE MAINTENANCE SUB EXECUTIVE
   IDK                   021            RECEIVE PRODUCTS INTO INVENTORY
   IDK                   031            CUSTOMER PRINT FACILITY
   IDK                   032            VENDOR PRINT FACILITY
   IDK                   033            INVENTORY PRINT FACILITY
   IDK                   111            CUSTOMER FILE MAINTENANCE
   IDK                   112            MODIFY A CUSTOMER RECORD
   IDK                   113            DELETE A CUSTOMER RECORD
   IDK                   114            INQUIRE ABOUT A CUSTOMER
   IDK                   121            VENDOR FILE MAINTENANCE
   IDK                   122            MODIFY A VENDORS RECORD
   IDK                   123            DELETE A VENDORS RECORD
   IDK                   124            INQUIRE ABOUT A VENDORS RECORD
   IDK                   131            INVENTORY FILE MAINTENANCE
   IDK                   998            MASTER SYSTEM CONTROL EXECUTIVE
   IDK                   999            SYSTEM LOGO

   PRINT                 001            PRINT SPOOLER FACILITY
   PRINT                 002            PRINT HELP FACILITY
```

3.1.3 LIST ALL DATA ELEMENTS FOR A SYSTEM

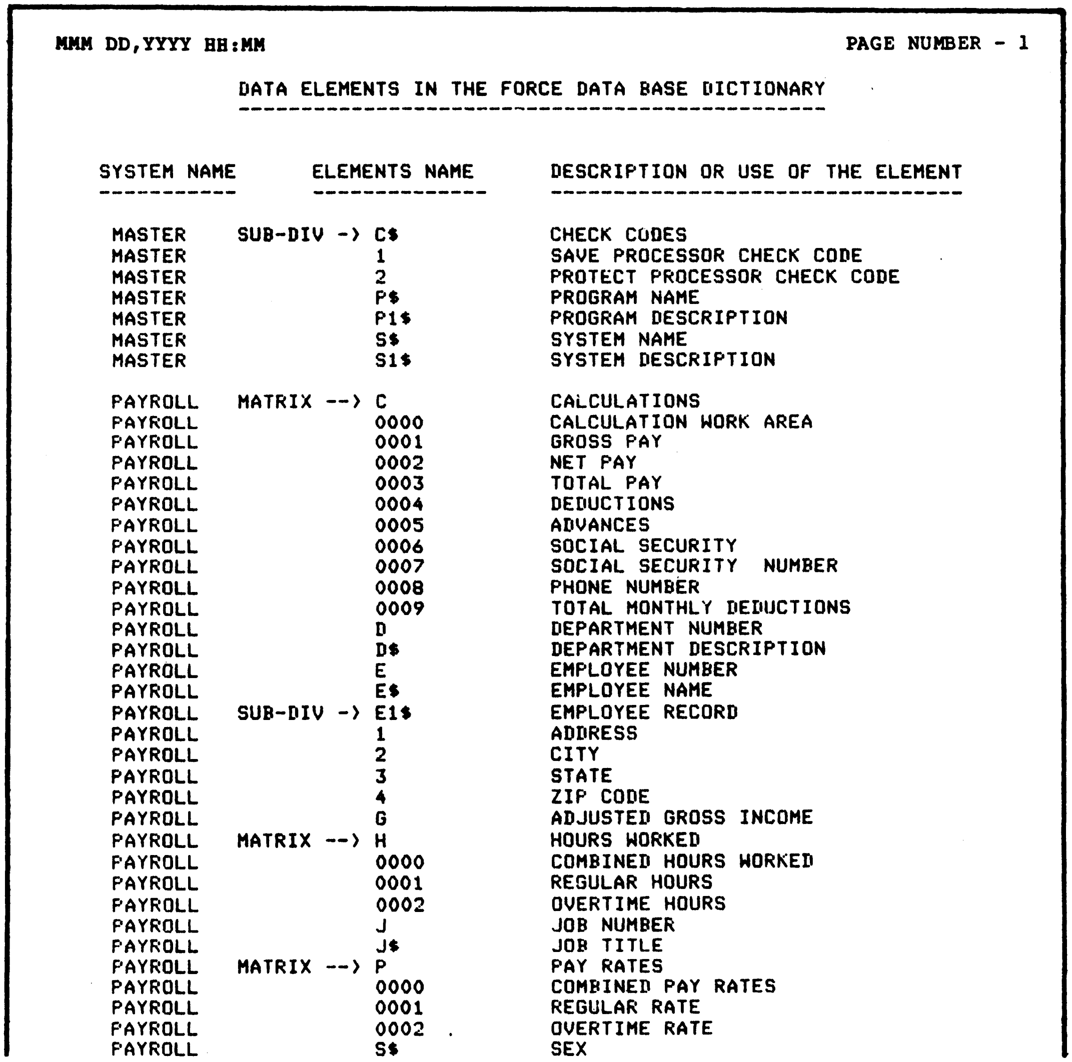
PURPOSE: This function prints a selected list of data elements and their descriptions. Data elements may be listed for a specified system, or all data elements within the Dictionary may be listed. The list of data elements is arranged alphabetically by the associated system name.

PROCEDURE: Enter the name of the system for which data elements are to be listed.

Enter RETURN to print a list of all data elements, beginning alphabetically with the first associated system name.

After specification of the report output, FORCE asks if the printer uses 8 1/2 x 11 paper. A (Y)es initiates printing of the report, and a (N)o returns to the Data Base Documentation menu.

NOTES: Below is a sample listing of data elements.



```
MMM DD,YYYY HH:MM                                      PAGE NUMBER - 1

                 DATA ELEMENTS IN THE FORCE DATA BASE DICTIONARY
                 -----------------------------------------------

   SYSTEM NAME     ELEMENTS NAME        DESCRIPTION OR USE OF THE ELEMENT
   -----------     --------------       ---------------------------------

    MASTER    SUB-DIV -> C$             CHECK CODES
    MASTER                1             SAVE PROCESSOR CHECK CODE
    MASTER                2             PROTECT PROCESSOR CHECK CODE
    MASTER                P$            PROGRAM NAME
    MASTER                P1$           PROGRAM DESCRIPTION
    MASTER                S$            SYSTEM NAME
    MASTER                S1$           SYSTEM DESCRIPTION

    PAYROLL   MATRIX --> C              CALCULATIONS
    PAYROLL               0000          CALCULATION WORK AREA
    PAYROLL               0001          GROSS PAY
    PAYROLL               0002          NET PAY
    PAYROLL               0003          TOTAL PAY
    PAYROLL               0004          DEDUCTIONS
    PAYROLL               0005          ADVANCES
    PAYROLL               0006          SOCIAL SECURITY
    PAYROLL               0007          SOCIAL SECURITY  NUMBER
    PAYROLL               0008          PHONE NUMBER
    PAYROLL               0009          TOTAL MONTHLY DEDUCTIONS
    PAYROLL               D             DEPARTMENT NUMBER
    PAYROLL               D$            DEPARTMENT DESCRIPTION
    PAYROLL               E             EMPLOYEE NUMBER
    PAYROLL               E$            EMPLOYEE NAME
    PAYROLL   SUB-DIV -> E1$            EMPLOYEE RECORD
    PAYROLL               1             ADDRESS
    PAYROLL               2             CITY
    PAYROLL               3             STATE
    PAYROLL               4             ZIP CODE
    PAYROLL               G             ADJUSTED GROSS INCOME
    PAYROLL   MATRIX --> H              HOURS WORKED
    PAYROLL               0000          COMBINED HOURS WORKED
    PAYROLL               0001          REGULAR HOURS
    PAYROLL               0002          OVERTIME HOURS
    PAYROLL               J             JOB NUMBER
    PAYROLL               J$            JOB TITLE
    PAYROLL   MATRIX --> P              PAY RATES
    PAYROLL               0000          COMBINED PAY RATES
    PAYROLL               0001          REGULAR RATE
    PAYROLL               0002          OVERTIME RATE
    PAYROLL               S$            SEX
```

3.1.4 LIST ALL FILES FOR A SYSTEM

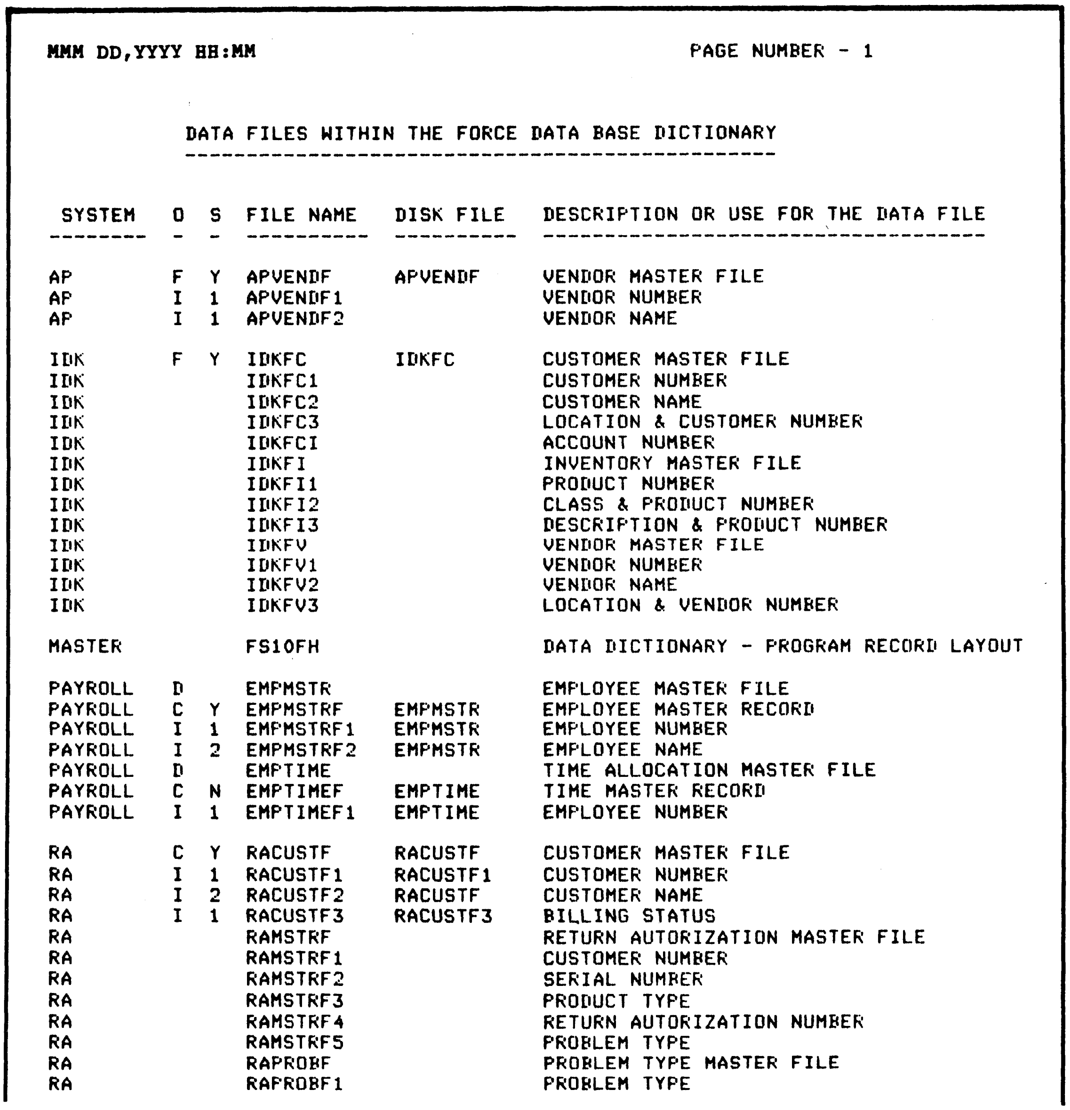
PURPOSE: This function prints a selected list of files and their characteristics. Files may be listed for a specified system, or all files within the Dictionary may be listed. The list of files is arranged alphabetically within each system.

PROCEDURE: Enter the name of the system for which files are to be listed.

Enter RETURN to print a list of all files, beginning alphabetically with the first associated system name.

After specification of the report output, FORCE asks if the printer uses 8 1/2 x 11 paper. A (Y)es initiates printing of the report, and a (N)o returns to the Data Base Documentation menu.

NOTES: Below is a sample listing of files.



```
MMM DD,YYYY HH:MM                          PAGE NUMBER - 1


         DATA FILES WITHIN THE FORCE DATA BASE DICTIONARY
         -----------------------------------------------

 SYSTEM  O S FILE NAME DISK FILE DESCRIPTION OR USE FOR THE DATA FILE
-------- - - --------- --------- ----------------------------------

AP       F Y APVENDF   APVENDF   VENDOR MASTER FILE
AP       I 1 APVENDF1            VENDOR NUMBER
AP       I 1 APVENDF2            VENDOR NAME

IDK      F Y IDKFC     IDKFC     CUSTOMER MASTER FILE
IDK          IDKFC1              CUSTOMER NUMBER
IDK          IDKFC2              CUSTOMER NAME
IDK          IDKFC3              LOCATION & CUSTOMER NUMBER
IDK          IDKFCI              ACCOUNT NUMBER
IDK          IDKFI               INVENTORY MASTER FILE
IDK          IDKFI1              PRODUCT NUMBER
IDK          IDKFI2              CLASS & PRODUCT NUMBER
IDK          IDKFI3              DESCRIPTION & PRODUCT NUMBER
IDK          IDKFV               VENDOR MASTER FILE
IDK          IDKFV1              VENDOR NUMBER
IDK          IDKFV2              VENDOR NAME
IDK          IDKFV3              LOCATION & VENDOR NUMBER

MASTER       FS10FH              DATA DICTIONARY - PROGRAM RECORD LAYOUT

PAYROLL  D   EMPMSTR             EMPLOYEE MASTER FILE
PAYROLL  C Y EMPMSTRF  EMPMSTR   EMPLOYEE MASTER RECORD
PAYROLL  I 1 EMPMSTRF1 EMPMSTR   EMPLOYEE NUMBER
PAYROLL  I 2 EMPMSTRF2 EMPMSTR   EMPLOYEE NAME
PAYROLL  D   EMPTIME             TIME ALLOCATION MASTER FILE
PAYROLL  C N EMPTIMEF  EMPTIME   TIME MASTER RECORD
PAYROLL  I 1 EMPTIMEF1 EMPTIME   EMPLOYEE NUMBER

RA       C Y RACUSTF   RACUSTF   CUSTOMER MASTER FILE
RA       I 1 RACUSTF1  RACUSTF1  CUSTOMER NUMBER
RA       I 2 RACUSTF2  RACUSTF   CUSTOMER NAME
RA       I 1 RACUSTF3  RACUSTF3  BILLING STATUS
RA           RAMSTRF             RETURN AUTORIZATION MASTER FILE
RA           RAMSTRF1            CUSTOMER NUMBER
RA           RAMSTRF2            SERIAL NUMBER
RA           RAMSTRF3            PRODUCT TYPE
RA           RAMSTRF4            RETURN AUTORIZATION NUMBER
RA           RAMSTRF5            PROBLEM TYPE
RA           RAPROBF             PROBLEM TYPE MASTER FILE
RA           RAPROBF1            PROBLEM TYPE
```

3.1.5 LIST ALL PROGRAMS FOR A SYSTEM

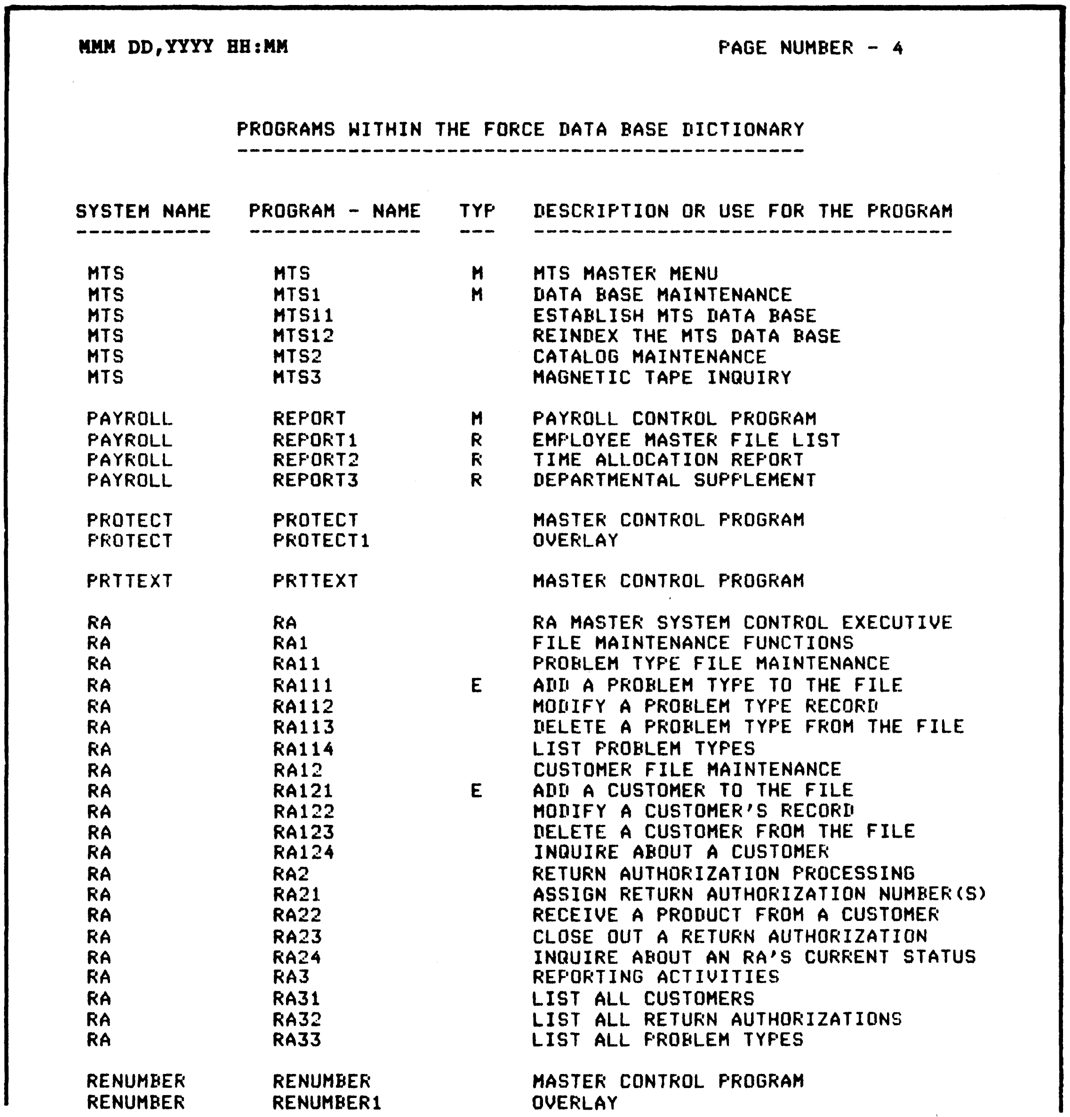
PURPOSE: This function prints a selected list of programs and their descriptions. Programs may be listed for a specified system, or all programs within the Dictionary may be listed. The list of programs is arranged alphabetically within each system.

PROCEDURE: Enter the name of the system for which programs are to be listed.

Enter RETURN to print a list of all programs, beginning alphabetically with the first associated system name.

After specification of the report output, FORCE asks if the printer uses 8 1/2 x 11 paper. A (Y)es initiates printing of the report, and a (N)o returns to the Data Base Documentation menu.

NOTES: Below is a sample listing of programs.



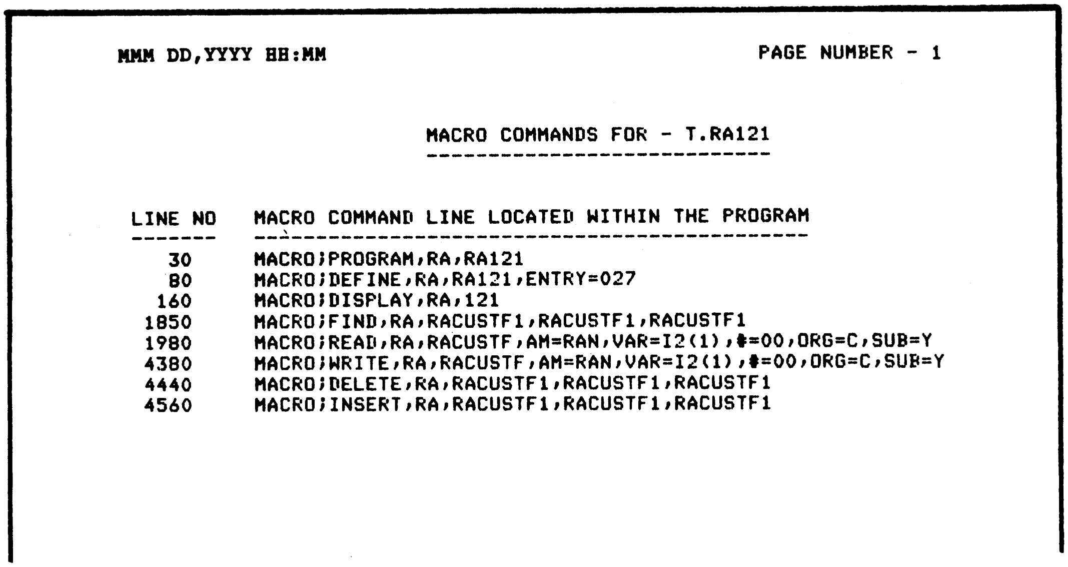
```
MMM DD,YYYY HH:MM                                   PAGE NUMBER - 4


                PROGRAMS WITHIN THE FORCE DATA BASE DICTIONARY
                ----------------------------------------------

SYSTEM NAME   PROGRAM - NAME   TYP   DESCRIPTION OR USE FOR THE PROGRAM
-----------   --------------   ---   ----------------------------------

MTS             MTS             M    MTS MASTER MENU
MTS             MTS1            M    DATA BASE MAINTENANCE
MTS             MTS11                ESTABLISH MTS DATA BASE
MTS             MTS12                REINDEX THE MTS DATA BASE
MTS             MTS2                 CATALOG MAINTENANCE
MTS             MTS3                 MAGNETIC TAPE INQUIRY

PAYROLL         REPORT          M    PAYROLL CONTROL PROGRAM
PAYROLL         REPORT1         R    EMPLOYEE MASTER FILE LIST
PAYROLL         REPORT2         R    TIME ALLOCATION REPORT
PAYROLL         REPORT3         R    DEPARTMENTAL SUPPLEMENT

PROTECT         PROTECT              MASTER CONTROL PROGRAM
PROTECT         PROTECT1             OVERLAY

PRTTEXT         PRTTEXT              MASTER CONTROL PROGRAM

RA              RA                   RA MASTER SYSTEM CONTROL EXECUTIVE
RA              RA1                  FILE MAINTENANCE FUNCTIONS
RA              RA11                 PROBLEM TYPE FILE MAINTENANCE
RA              RA111           E    ADD A PROBLEM TYPE TO THE FILE
RA              RA112                MODIFY A PROBLEM TYPE RECORD
RA              RA113                DELETE A PROBLEM TYPE FROM THE FILE
RA              RA114                LIST PROBLEM TYPES
RA              RA12                 CUSTOMER FILE MAINTENANCE
RA              RA121           E    ADD A CUSTOMER TO THE FILE
RA              RA122                MODIFY A CUSTOMER'S RECORD
RA              RA123                DELETE A CUSTOMER FROM THE FILE
RA              RA124                INQUIRE ABOUT A CUSTOMER
RA              RA2                  RETURN AUTHORIZATION PROCESSING
RA              RA21                 ASSIGN RETURN AUTHORIZATION NUMBER(S)
RA              RA22                 RECEIVE A PRODUCT FROM A CUSTOMER
RA              RA23                 CLOSE OUT A RETURN AUTHORIZATION
RA              RA24                 INQUIRE ABOUT AN RA'S CURRENT STATUS
RA              RA3                  REPORTING ACTIVITIES
RA              RA31                 LIST ALL CUSTOMERS
RA              RA32                 LIST ALL RETURN AUTHORIZATIONS
RA              RA33                 LIST ALL PROBLEM TYPES

RENUMBER        RENUMBER             MASTER CONTROL PROGRAM
RENUMBER        RENUMBER1            OVERLAY
```

3.1.6 LIST ALL MACROS WITHIN A PROGRAM

PURPOSE: Macro commands may be listed for any program which has been processed by the FORCE Linkage Editor. The report shows the macro command line and its line number in the program.

PROCEDURE: Enter the name of the system with which the program is associated. Then enter the program name.

NOTES: Below is a sample listing of macros within a program.



```
MMM DD,YYYY HH:MM                                         PAGE NUMBER - 1

                                   MACRO COMMANDS FOR - T.RA121
                                   -----------------------------

LINE NO    MACRO COMMAND LINE LOCATED WITHIN THE PROGRAM
-------    ----------------------------------------------
   30      MACRO;PROGRAM,RA,RA121
   80      MACRO;DEFINE,RA,RA121,ENTRY=027
  160      MACRO;DISPLAY,RA,121
 1850      MACRO;FIND,RA,RACUSTF1,RACUSTF1,RACUSTF1
 1980      MACRO;READ,RA,RACUSTF,AM=RAN,VAR=I2(1),#=00,ORG=C,SUB=Y
 4380      MACRO;WRITE,RA,RACUSTF,AM=RAN,VAR=I2(1),#=00,ORG=C,SUB=Y
 4440      MACRO;DELETE,RA,RACUSTF1,RACUSTF1,RACUSTF1
 4560      MACRO;INSERT,RA,RACUSTF1,RACUSTF1,RACUSTF1
```

3.1.7 LIST ALL REPORT FORMATS FOR A SYSTEM

<u>PURPOSE</u>: This function prints a selected list of report formats (used in report generation), denoting the format type and its description. Report formats may be listed for a specified system, or all report formats within the Dictionary may be listed. The list of report formats is arranged alphabetically by the associated system name.

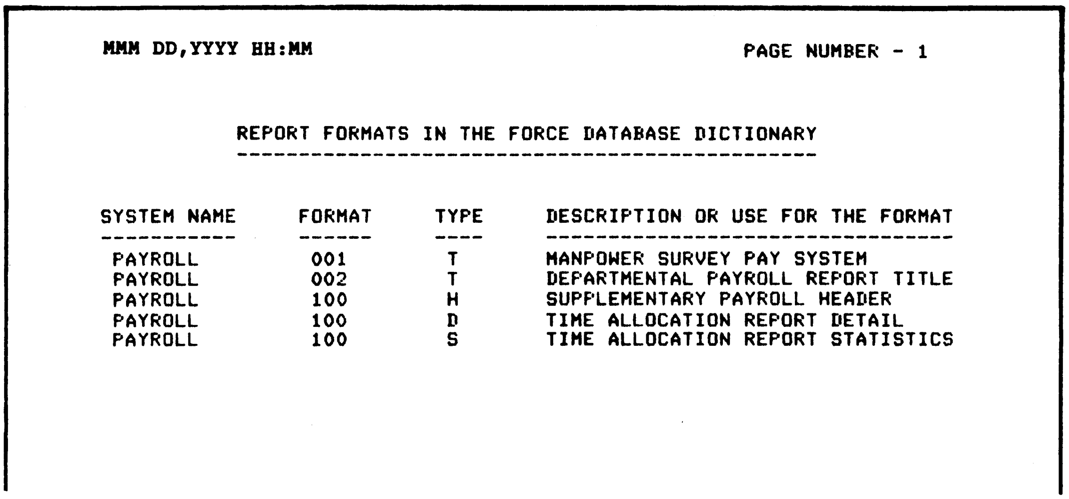
<u>PROCEDURE</u>: Enter the name of the system for which report formats are to be listed.

Enter RETURN to print a list of all report formats, beginning alphabetically with the first associated system name.

After specification of the report output, FORCE asks if the printer uses 8 1/2 x 11 paper. A (Y)es initiates printing, and a (N)o returns to the Data Base Documentation menu.

<u>NOTES</u>: Below is a sample listing of report formats.

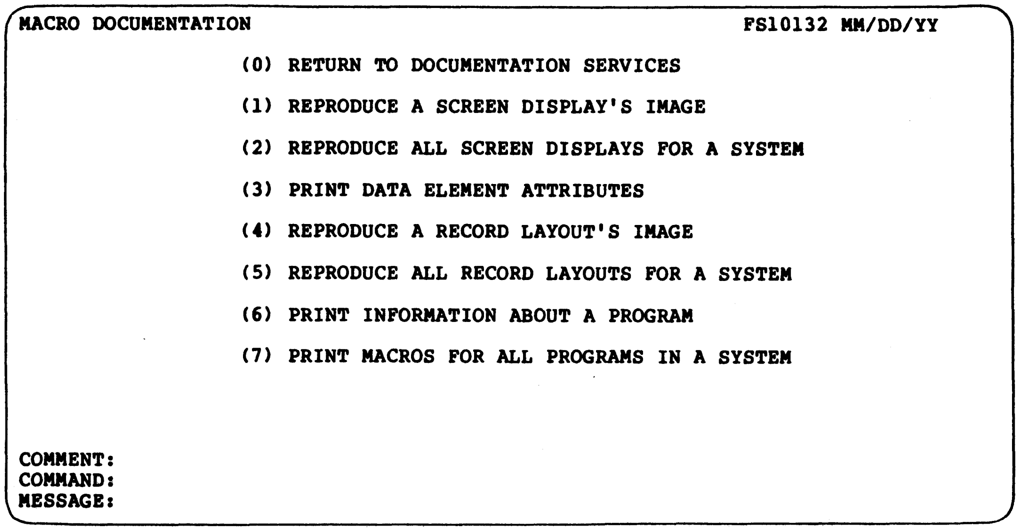
Format types are: (H)eader, (D)etail, (S)tatistics and (T)itle.



```
MMM DD,YYYY HH:MM                                      PAGE NUMBER - 1


                 REPORT FORMATS IN THE FORCE DATABASE DICTIONARY
                 -----------------------------------------------

SYSTEM NAME      FORMAT      TYPE      DESCRIPTION OR USE FOR THE FORMAT
-----------      ------      ----      ---------------------------------
 PAYROLL          001         T        MANPOWER SURVEY PAY SYSTEM
 PAYROLL          002         T        DEPARTMENTAL PAYROLL REPORT TITLE
 PAYROLL          100         H        SUPPLEMENTARY PAYROLL HEADER
 PAYROLL          100         D        TIME ALLOCATION REPORT DETAIL
 PAYROLL          100         S        TIME ALLOCATION REPORT STATISTICS
```

3.2 MACRO FACILITY DOCUMENTATION

PURPOSE: These menu selections provide detailed reports on system components. Screen reproductions, data element attributes, record layouts, program information and system macros are documented using these functions.



```
MACRO DOCUMENTATION                                      FS10132 MM/DD/YY

                    (0) RETURN TO DOCUMENTATION SERVICES

                    (1) REPRODUCE A SCREEN DISPLAY'S IMAGE

                    (2) REPRODUCE ALL SCREEN DISPLAYS FOR A SYSTEM

                    (3) PRINT DATA ELEMENT ATTRIBUTES

                    (4) REPRODUCE A RECORD LAYOUT'S IMAGE

                    (5) REPRODUCE ALL RECORD LAYOUTS FOR A SYSTEM

                    (6) PRINT INFORMATION ABOUT A PROGRAM

                    (7) PRINT MACROS FOR ALL PROGRAMS IN A SYSTEM



COMMENT:
COMMAND:
MESSAGE:
```

PROCEDURE: Enter the number which corresponds to the desired function.

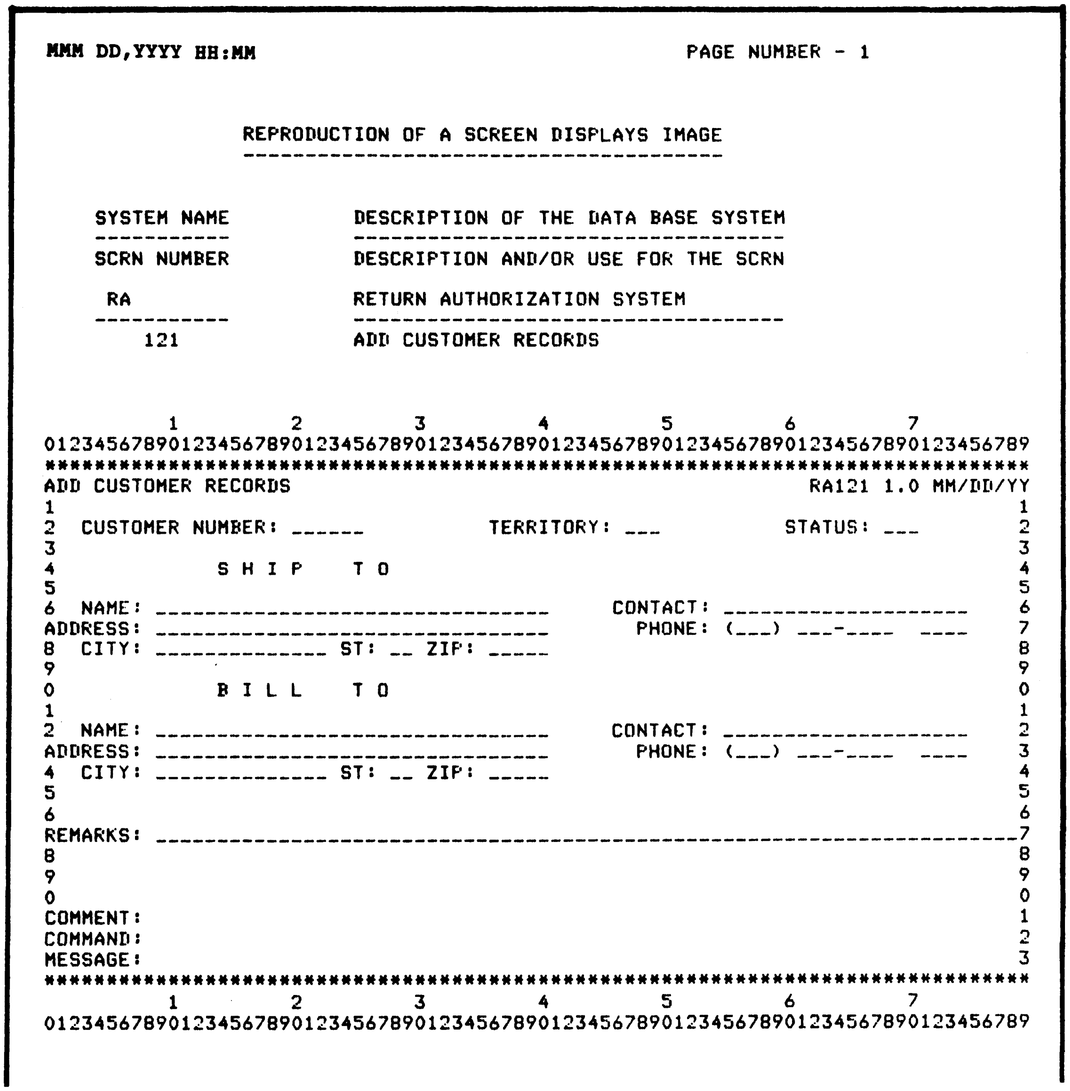
3.2.1 REPRODUCE A SCREEN DISPLAY'S IMAGE

PURPOSE: This function provides a reproduction of a screen format for any screen display in the Dictionary. The screen format is bordered with numbers representing the horizontal and vertical screen coordinates.

Each screen format is displayed on the CRT as it is printed.

PROCEDURE: Enter the name of the system with which the screen is associated. Then enter the assigned screen display number.

NOTES: Below is a sample screen format reproduction.



```
MMM DD,YYYY HH:MM                                     PAGE NUMBER - 1


                  REPRODUCTION OF A SCREEN DISPLAYS IMAGE
                  ---------------------------------------

    SYSTEM NAME          DESCRIPTION OF THE DATA BASE SYSTEM
    -----------          -----------------------------------
    SCRN NUMBER          DESCRIPTION AND/OR USE FOR THE SCRN

     RA                  RETURN AUTHORIZATION SYSTEM
    -----------          -----------------------------------
        121              ADD CUSTOMER RECORDS


          1         2         3         4         5         6         7
01234567890123456789012345678901234567890123456789012345678901234567890123456789
********************************************************************************
ADD CUSTOMER RECORDS                                          RA121 1.0 MM/DD/YY
1                                                                              1
2  CUSTOMER NUMBER: ______           TERRITORY: ___          STATUS: ___         2
3                                                                              3
4           S H I P   T O                                                      4
5                                                                              5
6  NAME: ________________________________     CONTACT: ____________________    6
ADDRESS: ________________________________       PHONE: (___) ___-____  ____    7
8  CITY: ______________ ST: __ ZIP: _____                                      8
9                                                                              9
0           B I L L   T O                                                      0
1                                                                              1
2  NAME: ________________________________     CONTACT: ____________________    2
ADDRESS: ________________________________       PHONE: (___) ___-____  ____    3
4  CITY: ______________ ST: __ ZIP: _____                                      4
5                                                                              5
6                                                                              6
REMARKS: ______________________________________________________________________7
8                                                                              8
9                                                                              9
0                                                                              0
COMMENT:                                                                       1
COMMAND:                                                                       2
MESSAGE:                                                                       3
********************************************************************************
          1         2         3         4         5         6         7
01234567890123456789012345678901234567890123456789012345678901234567890123456789
```

3.2.2 REPRODUCE ALL SCREEN DISPLAYS FOR A SYSTEM

PURPOSE: This function reproduces all screen formats for a specified system. The screen format is bordered with numbers representing horizontal and vertical screen coordinates.

Each screen is displayed on the CRT as it is printed.

PROCEDURE: Enter the name of the system for which screen formats are to be reproduced.

NOTES: See 3.2.1 for a sample screen display reproduction.

REPRODUCE
ALL SCREENS
3.2.2

3.2.3 PRINT DATA ELEMENT ATTRIBUTES

PURPOSE: This function prints the physical and logical attributes of all data elements within a system, or all data elements within the Dictionary.

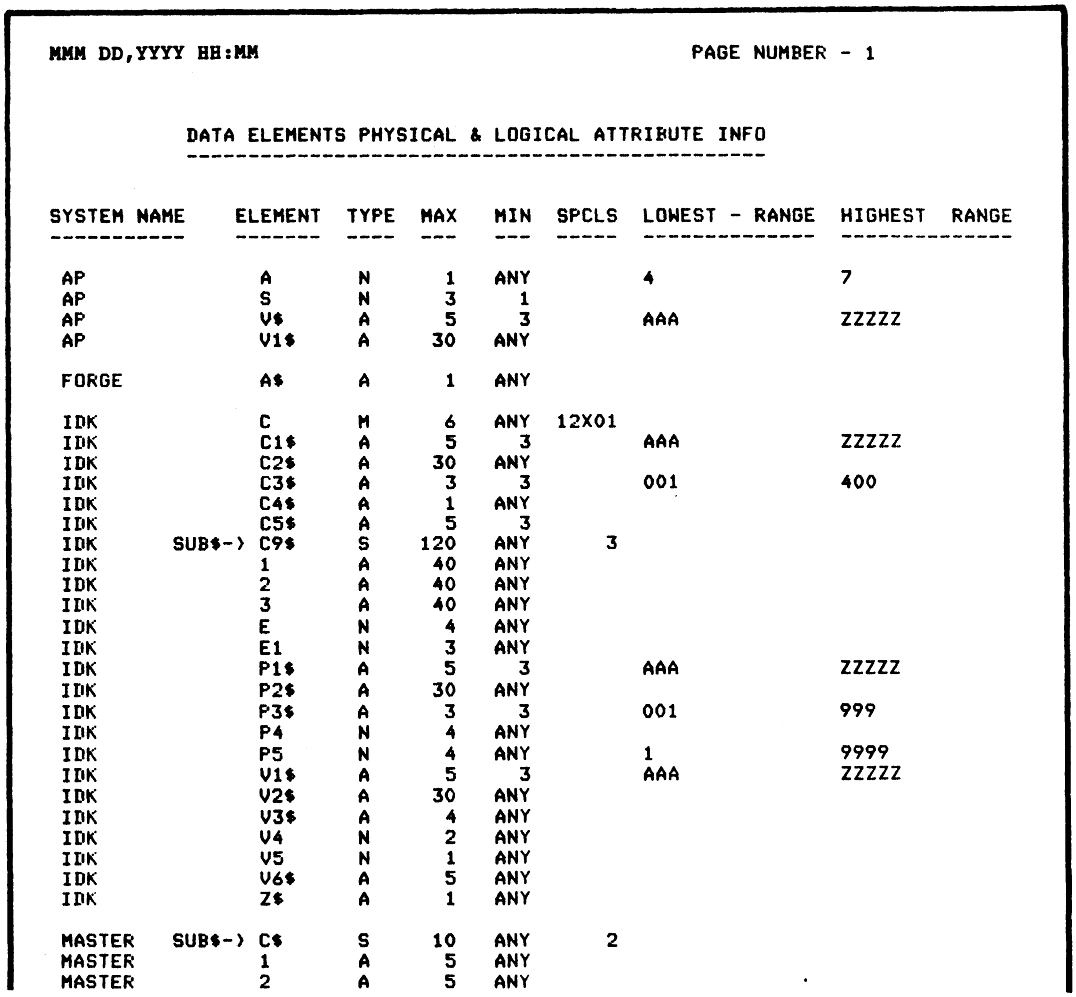
PROCEDURE: Enter the name of the system for which data element attributes are to be printed.

Enter RETURN to print information for all data elements in the Dictionary, beginning alphabetically with the first associated system name.

After specification of the output data, FORCE asks if the printer uses 8 1/2 x 11 paper. A (Y)es initiates printing of the report, and a (N)o returns to the Macro Facility Documentation menu.

NOTES: Below is a sample data element attributes report.

The SPCLS column denotes the number of subdivisions for a subdivided string, or the cell coordinates for a matrix.



```
MMM DD,YYYY HH:MM                                   PAGE NUMBER - 1


           DATA ELEMENTS PHYSICAL & LOGICAL ATTRIBUTE INFO
           -----------------------------------------------

SYSTEM NAME    ELEMENT  TYPE  MAX   MIN  SPCLS  LOWEST - RANGE  HIGHEST  RANGE
-----------    -------  ----  ---   ---  -----  --------------  --------------

 AP              A       N       1  ANY         4               7
 AP              S       N       3    1
 AP              V$      A       5    3         AAA             ZZZZZ
 AP              V1$     A      30  ANY

 FORGE           A$      A       1  ANY

 IDK             C       M       6  ANY  12X01
 IDK             C1$     A       5    3         AAA             ZZZZZ
 IDK             C2$     A      30  ANY
 IDK             C3$     A       3    3         001             400
 IDK             C4$     A       1  ANY
 IDK             C5$     A       5    3
 IDK      SUB$-> C9$     S     120  ANY      3
 IDK             1       A      40  ANY
 IDK             2       A      40  ANY
 IDK             3       A      40  ANY
 IDK             E       N       4  ANY
 IDK             E1      N       3  ANY
 IDK             P1$     A       5    3         AAA             ZZZZZ
 IDK             P2$     A      30  ANY
 IDK             P3$     A       3    3         001             999
 IDK             P4      N       4  ANY
 IDK             P5      N       4  ANY         1               9999
 IDK             V1$     A       5    3         AAA             ZZZZZ
 IDK             V2$     A      30  ANY
 IDK             V3$     A       4  ANY
 IDK             V4      N       2  ANY
 IDK             V5      N       1  ANY
 IDK             V6$     A       5  ANY
 IDK             Z$      A       1  ANY

 MASTER   SUB$-> C$      S      10  ANY      2
 MASTER          1       A       5  ANY
 MASTER          2       A       5  ANY
```

3.2.4 REPRODUCE A RECORD LAYOUT'S IMAGE

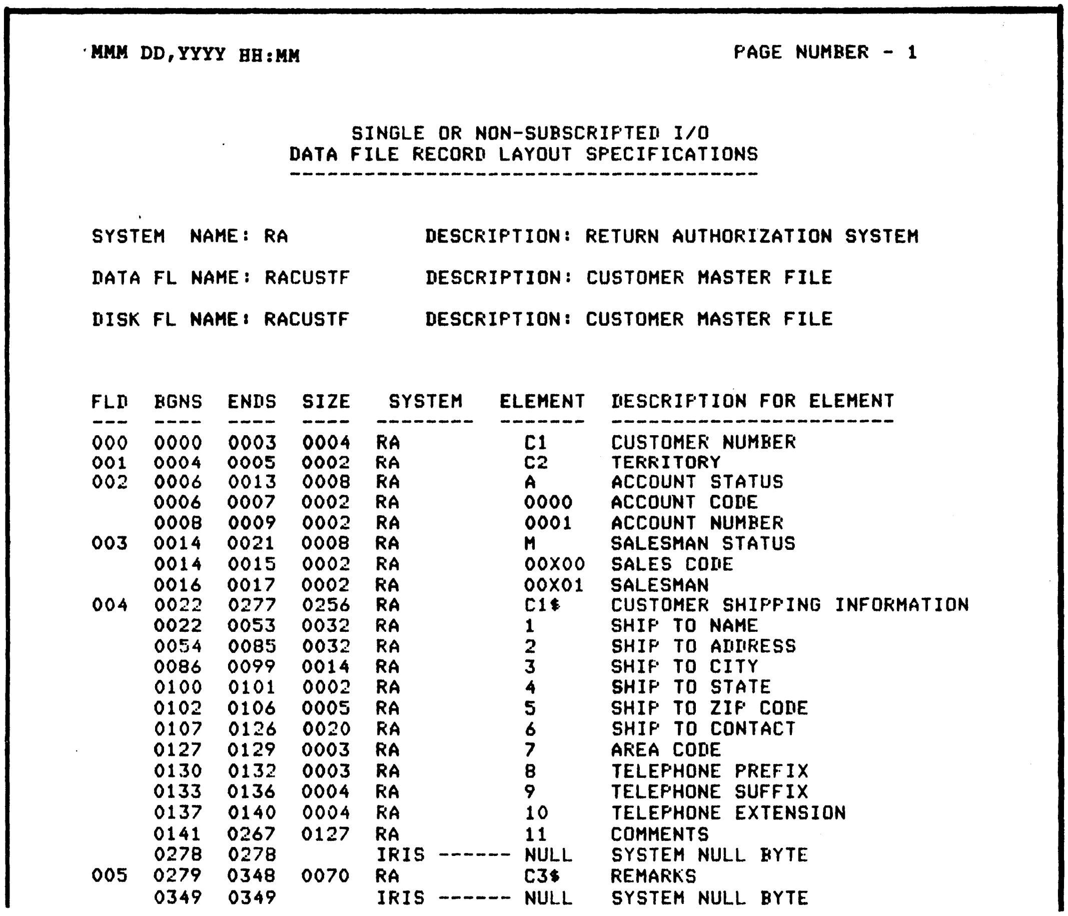
PURPOSE: This function prints record layout information for any FORCE data file or key construct file. Each field of the record layout is described in detail, denoting its coordinates, size, corresponding system data element and description.

PROCEDURE: Enter the name of the system with which the file is associated. Then enter the file name.

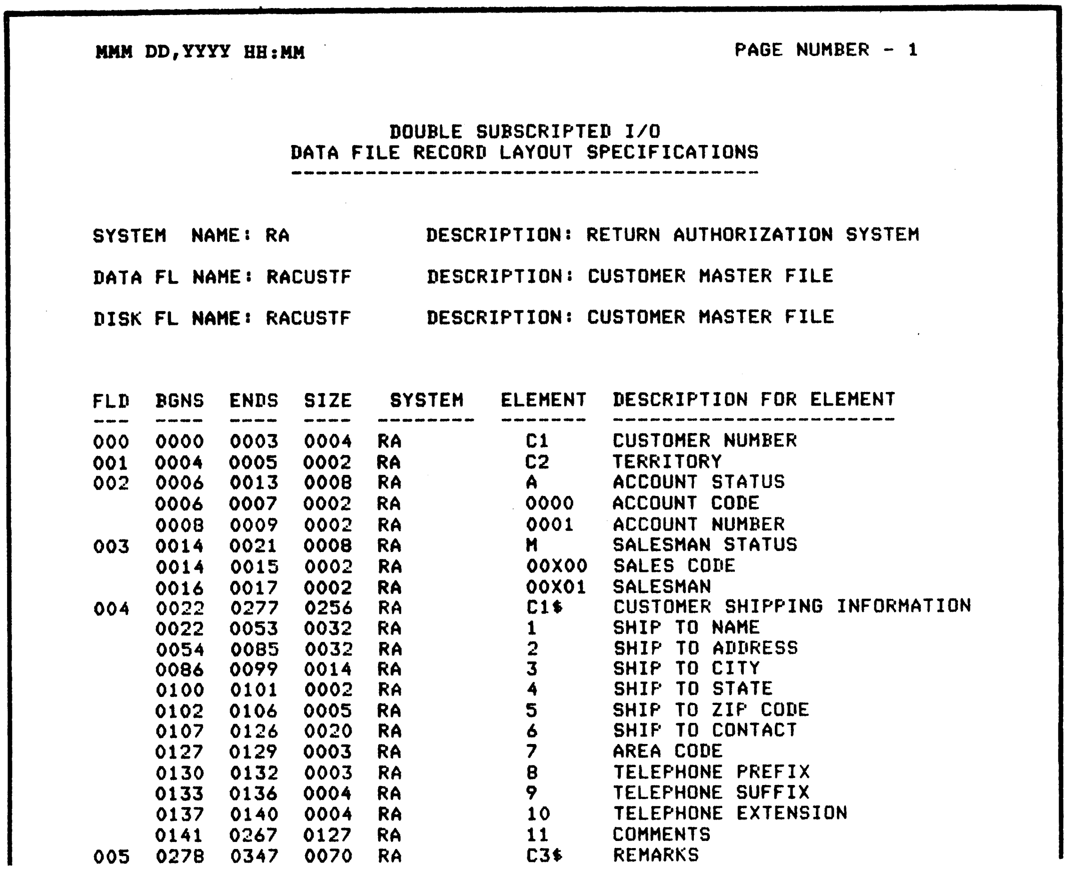
NOTES: The report output accommodates the differences between single subscription and double subscription for the record layout.

Below is a reproduction of a single subscripted file record layout. Reproductions of a double subscripted record layout and a key construct are shown on the following page.

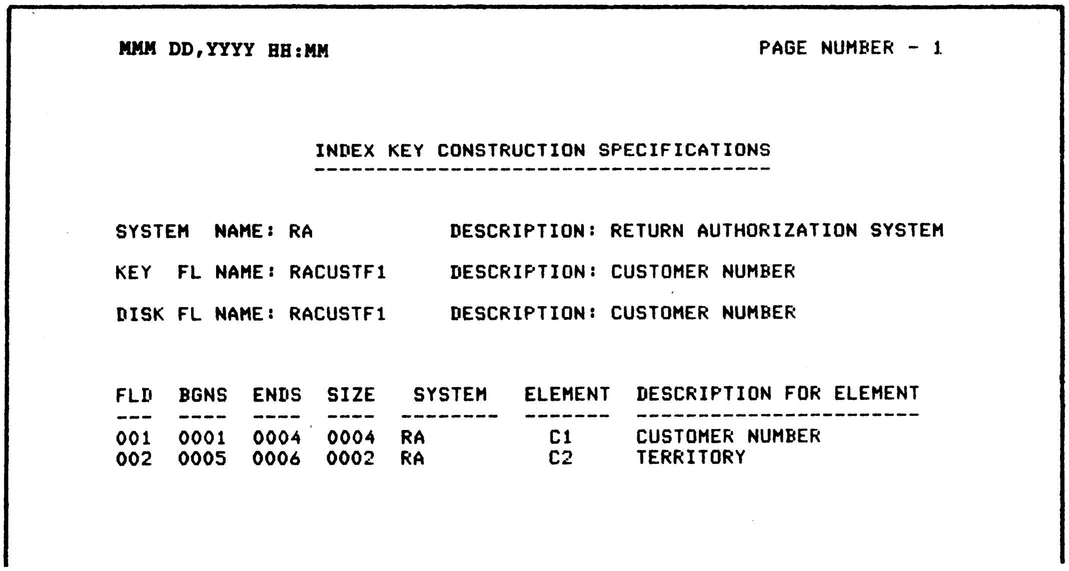
In these reports, the FLD column data represents the item number for formatted files. The byte offset for contiguous files is denoted by the BGNS column.



```
MMM DD,YYYY HH:MM                                            PAGE NUMBER - 1


                           SINGLE OR NON-SUBSCRIPTED I/O
                      DATA FILE RECORD LAYOUT SPECIFICATIONS
                      --------------------------------------

SYSTEM  NAME: RA            DESCRIPTION: RETURN AUTHORIZATION SYSTEM

DATA FL NAME: RACUSTF       DESCRIPTION: CUSTOMER MASTER FILE

DISK FL NAME: RACUSTF       DESCRIPTION: CUSTOMER MASTER FILE


FLD  BGNS  ENDS  SIZE   SYSTEM    ELEMENT  DESCRIPTION FOR ELEMENT
---  ----  ----  ----  --------   -------  -----------------------
000  0000  0003  0004  RA           C1     CUSTOMER NUMBER
001  0004  0005  0002  RA           C2     TERRITORY
002  0006  0013  0008  RA           A      ACCOUNT STATUS
     0006  0007  0002  RA           0000   ACCOUNT CODE
     0008  0009  0002  RA           0001   ACCOUNT NUMBER
003  0014  0021  0008  RA           M      SALESMAN STATUS
     0014  0015  0002  RA           00X00  SALES CODE
     0016  0017  0002  RA           00X01  SALESMAN
004  0022  0277  0256  RA           C1$    CUSTOMER SHIPPING INFORMATION
     0022  0053  0032  RA           1      SHIP TO NAME
     0054  0085  0032  RA           2      SHIP TO ADDRESS
     0086  0099  0014  RA           3      SHIP TO CITY
     0100  0101  0002  RA           4      SHIP TO STATE
     0102  0106  0005  RA           5      SHIP TO ZIP CODE
     0107  0126  0020  RA           6      SHIP TO CONTACT
     0127  0129  0003  RA           7      AREA CODE
     0130  0132  0003  RA           8      TELEPHONE PREFIX
     0133  0136  0004  RA           9      TELEPHONE SUFFIX
     0137  0140  0004  RA           10     TELEPHONE EXTENSION
     0141  0267  0127  RA           11     COMMENTS
     0278  0278        IRIS ------ NULL    SYSTEM NULL BYTE
005  0279  0348  0070  RA           C3$    REMARKS
     0349  0349        IRIS ------ NULL    SYSTEM NULL BYTE
```



```
MMM DD,YYYY HH:MM                                          PAGE NUMBER - 1


                           DOUBLE SUBSCRIPTED I/O
                     DATA FILE RECORD LAYOUT SPECIFICATIONS
                     --------------------------------------


SYSTEM  NAME: RA            DESCRIPTION: RETURN AUTHORIZATION SYSTEM

DATA FL NAME: RACUSTF       DESCRIPTION: CUSTOMER MASTER FILE

DISK FL NAME: RACUSTF       DESCRIPTION: CUSTOMER MASTER FILE


FLD  BGNS  ENDS  SIZE   SYSTEM   ELEMENT  DESCRIPTION FOR ELEMENT
---  ----  ----  ----  --------  -------  -----------------------
000  0000  0003  0004  RA          C1     CUSTOMER NUMBER
001  0004  0005  0002  RA          C2     TERRITORY
002  0006  0013  0008  RA          A      ACCOUNT STATUS
     0006  0007  0002  RA          0000   ACCOUNT CODE
     0008  0009  0002  RA          0001   ACCOUNT NUMBER
003  0014  0021  0008  RA          M      SALESMAN STATUS
     0014  0015  0002  RA          00X00  SALES CODE
     0016  0017  0002  RA          00X01  SALESMAN
004  0022  0277  0256  RA          C1$    CUSTOMER SHIPPING INFORMATION
     0022  0053  0032  RA          1      SHIP TO NAME
     0054  0085  0032  RA          2      SHIP TO ADDRESS
     0086  0099  0014  RA          3      SHIP TO CITY
     0100  0101  0002  RA          4      SHIP TO STATE
     0102  0106  0005  RA          5      SHIP TO ZIP CODE
     0107  0126  0020  RA          6      SHIP TO CONTACT
     0127  0129  0003  RA          7      AREA CODE
     0130  0132  0003  RA          8      TELEPHONE PREFIX
     0133  0136  0004  RA          9      TELEPHONE SUFFIX
     0137  0140  0004  RA          10     TELEPHONE EXTENSION
     0141  0267  0127  RA          11     COMMENTS
005  0278  0347  0070  RA          C3$    REMARKS
```



```
MMM DD,YYYY HH:MM                                          PAGE NUMBER - 1


                      INDEX KEY CONSTRUCTION SPECIFICATIONS
                      -------------------------------------

SYSTEM  NAME: RA            DESCRIPTION: RETURN AUTHORIZATION SYSTEM

KEY  FL NAME: RACUSTF1      DESCRIPTION: CUSTOMER NUMBER

DISK FL NAME: RACUSTF1      DESCRIPTION: CUSTOMER NUMBER


FLD  BGNS  ENDS  SIZE   SYSTEM   ELEMENT  DESCRIPTION FOR ELEMENT
---  ----  ----  ----  --------  -------  -----------------------
001  0001  0004  0004  RA          C1     CUSTOMER NUMBER
002  0005  0006  0002  RA          C2     TERRITORY
```

3.2.5 REPRODUCE ALL RECORD LAYOUTS FOR A SYSTEM

PURPOSE: This function prints record layout information for all data files and key construct files within a system. Each field of the record layout is described in detail, denoting its coordinates, size, corresponding system data element and description.

PROCEDURE: Enter the name of the system for which file record layout information is to be printed.

NOTES: The report output accommodates differences between single subscription and double subscription for the record layout.

Reproductions of a key construct file, and single and double subscripted record layouts are shown in Section 3.2.4.

REPRODUCE ALL
RECORD LAYOUTS
3.2.5

3.2.6 PRINT INFORMATION ABOUT A PROGRAM

PURPOSE: This function provides comprehensive information about any entry or report program within the Dictionary.

For a data entry program, the report denotes the associated system, input screen and output file(s) used by the program. Program inputs are then documented in detail.

Information listed for a report program summarizes each report level and describes in detail the processing activities to be performed at every I/O.

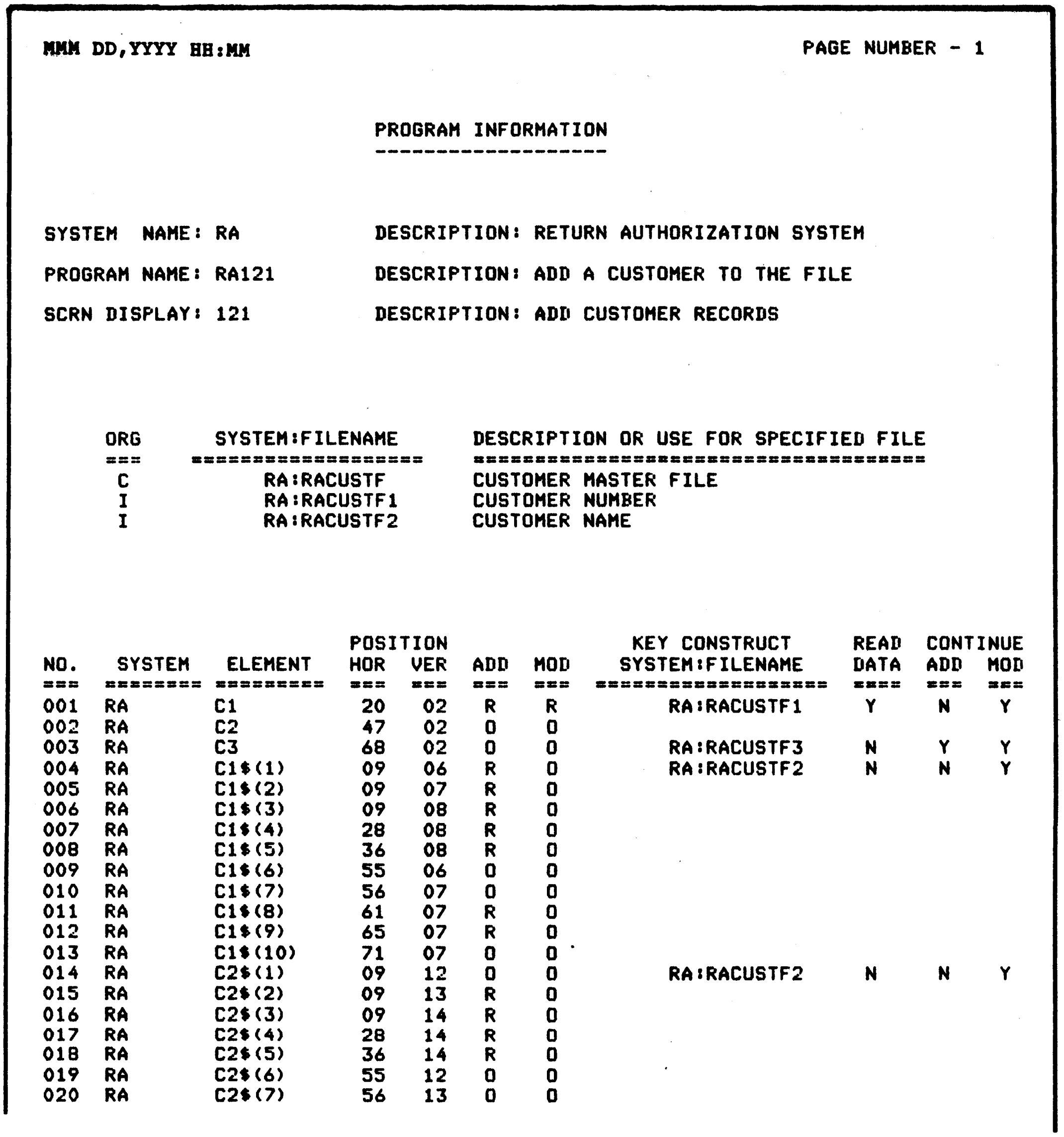
PROCEDURE: This documentation is available only after the associated program has been substantially developed within the dictionary.

Enter the system name with which the program is associated. Then enter the program name.

If the program is a report program, FORCE asks if the (S)ynopsis or (E)xpanded report should be printed. Enter S for an abridged report, or E for a detailed report.

PRINT PROGRAM
INFORMATION
3.2.6

NOTES: A sample program information report for an entry program is shown on the following page.



```
MMM DD,YYYY HH:MM                                         PAGE NUMBER - 1


                           PROGRAM INFORMATION
                           -------------------


SYSTEM  NAME: RA           DESCRIPTION: RETURN AUTHORIZATION SYSTEM

PROGRAM NAME: RA121        DESCRIPTION: ADD A CUSTOMER TO THE FILE

SCRN DISPLAY: 121          DESCRIPTION: ADD CUSTOMER RECORDS



     ORG      SYSTEM:FILENAME      DESCRIPTION OR USE FOR SPECIFIED FILE
     ===   ===================     =====================================
      C          RA:RACUSTF        CUSTOMER MASTER FILE
      I          RA:RACUSTF1       CUSTOMER NUMBER
      I          RA:RACUSTF2       CUSTOMER NAME



                            POSITION                KEY CONSTRUCT    READ  CONTINUE
NO.   SYSTEM   ELEMENT      HOR  VER   ADD  MOD    SYSTEM:FILENAME   DATA  ADD  MOD
===  ======== =========     ===  ===   ===  ===  ===================  ====  ===  ===
001  RA       C1             20   02    R    R         RA:RACUSTF1    Y     N    Y
002  RA       C2             47   02    O    O
003  RA       C3             68   02    O    O         RA:RACUSTF3    N     Y    Y
004  RA       C1$(1)         09   06    R    O         RA:RACUSTF2    N     N    Y
005  RA       C1$(2)         09   07    R    O
006  RA       C1$(3)         09   08    R    O
007  RA       C1$(4)         28   08    R    O
008  RA       C1$(5)         36   08    R    O
009  RA       C1$(6)         55   06    O    O
010  RA       C1$(7)         56   07    O    O
011  RA       C1$(8)         61   07    R    O
012  RA       C1$(9)         65   07    R    O
013  RA       C1$(10)        71   07    O    O
014  RA       C2$(1)         09   12    O    O         RA:RACUSTF2    N     N    Y
015  RA       C2$(2)         09   13    R    O
016  RA       C2$(3)         09   14    R    O
017  RA       C2$(4)         28   14    R    O
018  RA       C2$(5)         36   14    R    O
019  RA       C2$(6)         55   12    O    O
020  RA       C2$(7)         56   13    O    O
```

3.2.7 PRINT MACROS FOR ALL PROGRAMS IN A SYSTEM

PURPOSE: This function lists macro commands for programs which have been processed by the FORCE Linkage Editor. It lists macro command lines and their line numbers for all programs in a specified system.

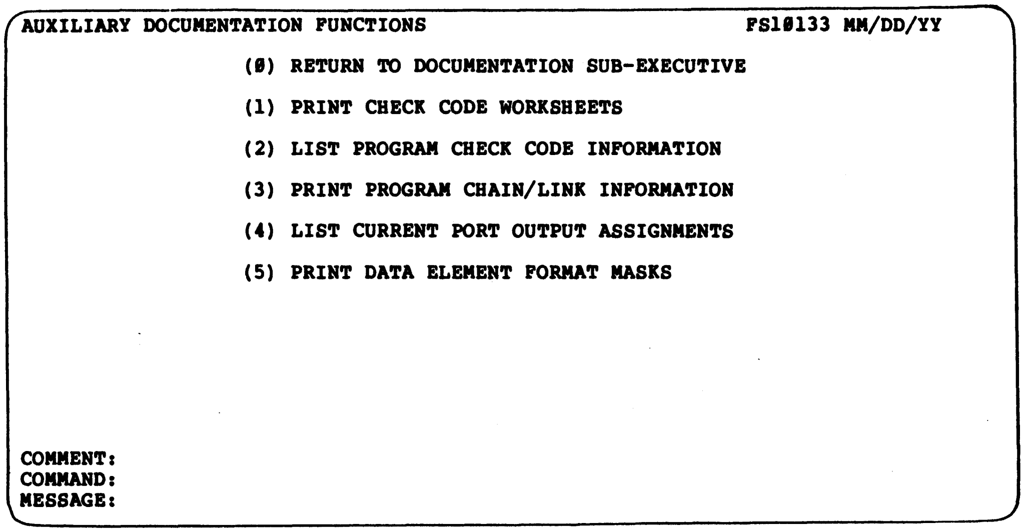
PROCEDURE: Enter the name of the system for which program macros are to be printed.

NOTES: A sample report on program macros is shown in Section 3.1.6.

PRINT PROGRAM
MACROS
3.2.7

3.3 AUXILIARY PRINT FUNCTIONS

PURPOSE: This menu provides selections to document port output assignments, check codes, program linkage and format masks.



```
AUXILIARY DOCUMENTATION FUNCTIONS                              FS1Ø133 MM/DD/YY

                        (Ø) RETURN TO DOCUMENTATION SUB-EXECUTIVE

                        (1) PRINT CHECK CODE WORKSHEETS

                        (2) LIST PROGRAM CHECK CODE INFORMATION

                        (3) PRINT PROGRAM CHAIN/LINK INFORMATION

                        (4) LIST CURRENT PORT OUTPUT ASSIGNMENTS

                        (5) PRINT DATA ELEMENT FORMAT MASKS







COMMENT:
COMMAND:
MESSAGE:
```

PROCEDURE: Enter the number which corresponds to the desired function.

3.3.1 PRINT CHECK CODE WORKSHEETS

PURPOSE: This function prints a check code worksheet which lists the programs for a specified system, or for all systems in the Dictionary. The worksheet is double-spaced, providing blanks for the programmer to manually catalog the SAVE or PROTECT codes.

This worksheet can be used to maintain check codes for any program that has been established in the Dictionary (see 1.5.1).

PROCEDURE: Enter the name of the system for which a check code worksheet is to be printed.

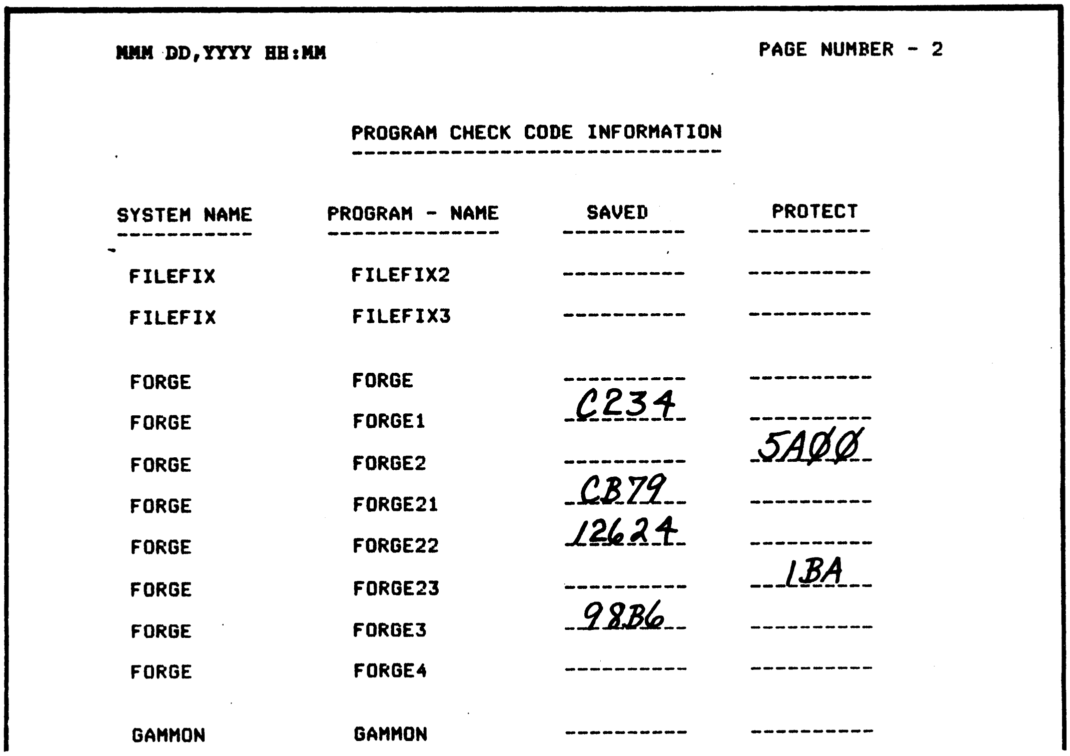
Enter RETURN to print a worksheet containing programs for all systems, beginning alphabetically at the first associated system name.

After specification of the output data, FORCE asks if the printer uses 8 1/2 x 11 paper. A (Y)es initiates printing of the report, and a (N)o returns to the Auxiliary Print Functions menu.

NOTES: Records of program check codes may be maintained within the Dictionary (see 1.5.5).

To include programs that are not generated by FORCE on the worksheet, define those programs to the Dictionary and associate them with the appropriate system.

Below is a sample check code worksheet.



```
MMM DD,YYYY HH:MM                                        PAGE NUMBER - 2

                         PROGRAM CHECK CODE INFORMATION
                         ------------------------------

SYSTEM NAME         PROGRAM - NAME        SAVED          PROTECT
-----------         --------------     ----------      ----------

 FILEFIX              FILEFIX2         ----------      ----------
 FILEFIX              FILEFIX3         ----------      ----------

 FORGE                FORGE            ----------      ----------
 FORGE                FORGE1           __C234____      ----------
 FORGE                FORGE2           ----------      __5AØØ____
 FORGE                FORGE21          __CB79____      ----------
 FORGE                FORGE22          _12624____      ----------
 FORGE                FORGE23          ----------      ___1BA____
 FORGE                FORGE3           __98B6____      ----------
 FORGE                FORGE4           ----------      ----------

 GAMMON               GAMMON           ----------      ----------
```

3.3.2 LIST PROGRAM CHECK CODE INFORMATION

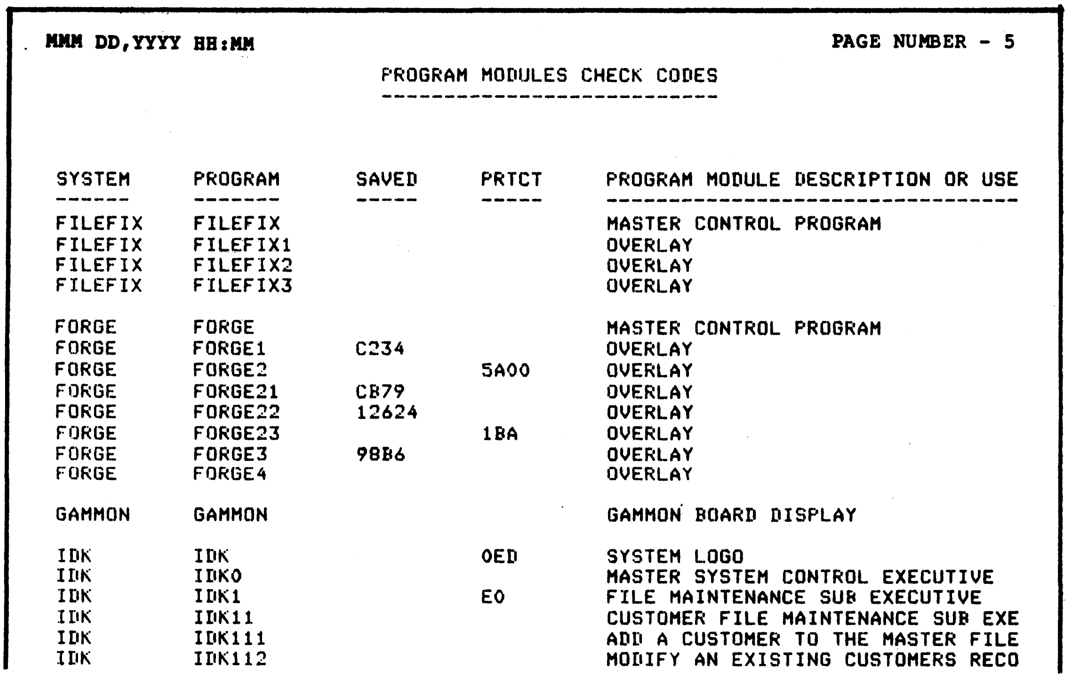
PURPOSE: This function prints stored check codes for each program of a specified system, or for programs in all systems. The report indicates the system program, its description, and its SAVE or PROTECT code.

PROCEDURE: Enter the name of the system for which check codes (see 1.5.5) are to be printed.

Press RETURN to print recorded check codes for all programs in all systems, beginning alphabetically with the first associated system name.

After specification of the report output, FORCE asks if the printer uses 8 1/2 x 11 paper. A (Y)es initiates printing of the report, and a (N)o returns to the Auxiliary Print Functions menu.

NOTES: Below is a sample report on stored check codes.



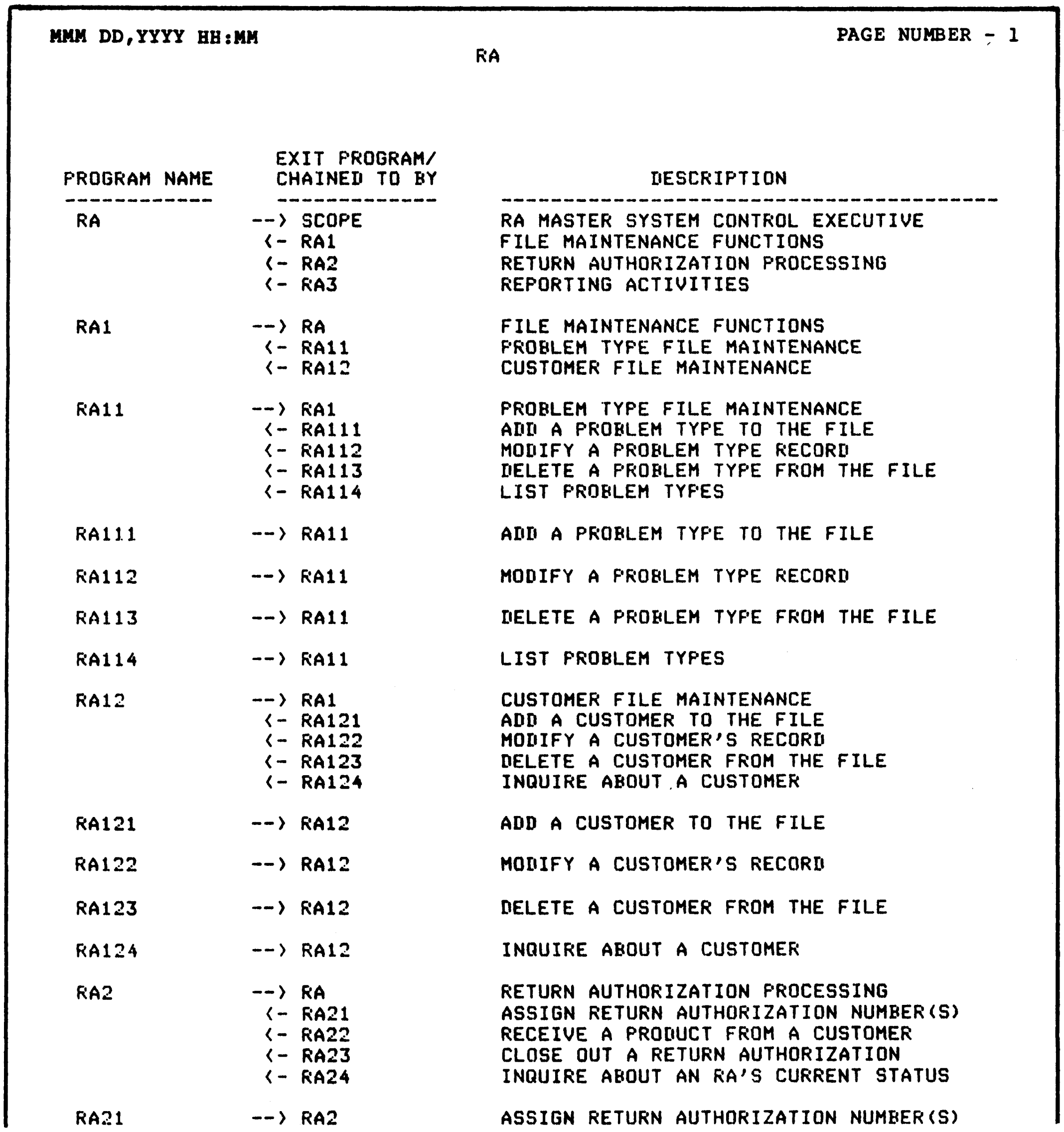
```
MMM DD,YYYY HH:MM                                                   PAGE NUMBER - 5
                                   PROGRAM MODULES CHECK CODES
                                   ---------------------------


SYSTEM      PROGRAM       SAVED      PRTCT      PROGRAM MODULE DESCRIPTION OR USE
------      -------       -----      -----      ---------------------------------
FILEFIX     FILEFIX                             MASTER CONTROL PROGRAM
FILEFIX     FILEFIX1                            OVERLAY
FILEFIX     FILEFIX2                            OVERLAY
FILEFIX     FILEFIX3                            OVERLAY

FORGE       FORGE                               MASTER CONTROL PROGRAM
FORGE       FORGE1        C234                  OVERLAY
FORGE       FORGE2                   5A00       OVERLAY
FORGE       FORGE21       CB79                  OVERLAY
FORGE       FORGE22       12624                 OVERLAY
FORGE       FORGE23                  1BA        OVERLAY
FORGE       FORGE3        98B6                  OVERLAY
FORGE       FORGE4                              OVERLAY

GAMMON      GAMMON                              GAMMON BOARD DISPLAY

IDK         IDK                      0ED        SYSTEM LOGO
IDK         IDK0                                MASTER SYSTEM CONTROL EXECUTIVE
IDK         IDK1                     E0         FILE MAINTENANCE SUB EXECUTIVE
IDK         IDK11                               CUSTOMER FILE MAINTENANCE SUB EXE
IDK         IDK111                              ADD A CUSTOMER TO THE MASTER FILE
IDK         IDK112                              MODIFY AN EXISTING CUSTOMERS RECO
```

3.3.3 PRINT PROGRAM CHAIN/LINK INFORMATION

PURPOSE: This function provides a report which illustrates the chaining pattern of a system's programs. For each system program, the report denotes its description, the program to which it exits, and the program(s) which chain to it.

PROCEDURE: Enter the name of the system for which the chain/link information is to be printed.

NOTES: Below is a sample chain/link report. Arrows illustrate the program chaining pattern.



```
MMM DD,YYYY HH:MM                                              PAGE NUMBER - 1
                                    RA



                     EXIT PROGRAM/
 PROGRAM NAME        CHAINED TO BY              DESCRIPTION
 ------------        -------------      ----------------------------------------
  RA               --> SCOPE            RA MASTER SYSTEM CONTROL EXECUTIVE
                    <- RA1              FILE MAINTENANCE FUNCTIONS
                    <- RA2              RETURN AUTHORIZATION PROCESSING
                    <- RA3              REPORTING ACTIVITIES

  RA1              --> RA               FILE MAINTENANCE FUNCTIONS
                    <- RA11             PROBLEM TYPE FILE MAINTENANCE
                    <- RA12             CUSTOMER FILE MAINTENANCE

  RA11             --> RA1              PROBLEM TYPE FILE MAINTENANCE
                    <- RA111            ADD A PROBLEM TYPE TO THE FILE
                    <- RA112            MODIFY A PROBLEM TYPE RECORD
                    <- RA113            DELETE A PROBLEM TYPE FROM THE FILE
                    <- RA114            LIST PROBLEM TYPES

  RA111            --> RA11             ADD A PROBLEM TYPE TO THE FILE

  RA112            --> RA11             MODIFY A PROBLEM TYPE RECORD

  RA113            --> RA11             DELETE A PROBLEM TYPE FROM THE FILE

  RA114            --> RA11             LIST PROBLEM TYPES

  RA12             --> RA1              CUSTOMER FILE MAINTENANCE
                    <- RA121            ADD A CUSTOMER TO THE FILE
                    <- RA122            MODIFY A CUSTOMER'S RECORD
                    <- RA123            DELETE A CUSTOMER FROM THE FILE
                    <- RA124            INQUIRE ABOUT A CUSTOMER

  RA121            --> RA12             ADD A CUSTOMER TO THE FILE

  RA122            --> RA12             MODIFY A CUSTOMER'S RECORD

  RA123            --> RA12             DELETE A CUSTOMER FROM THE FILE

  RA124            --> RA12             INQUIRE ABOUT A CUSTOMER

  RA2              --> RA               RETURN AUTHORIZATION PROCESSING
                    <- RA21             ASSIGN RETURN AUTHORIZATION NUMBER(S)
                    <- RA22             RECEIVE A PRODUCT FROM A CUSTOMER
                    <- RA23             CLOSE OUT A RETURN AUTHORIZATION
                    <- RA24             INQUIRE ABOUT AN RA'S CURRENT STATUS

  RA21             --> RA2              ASSIGN RETURN AUTHORIZATION NUMBER(S)
```

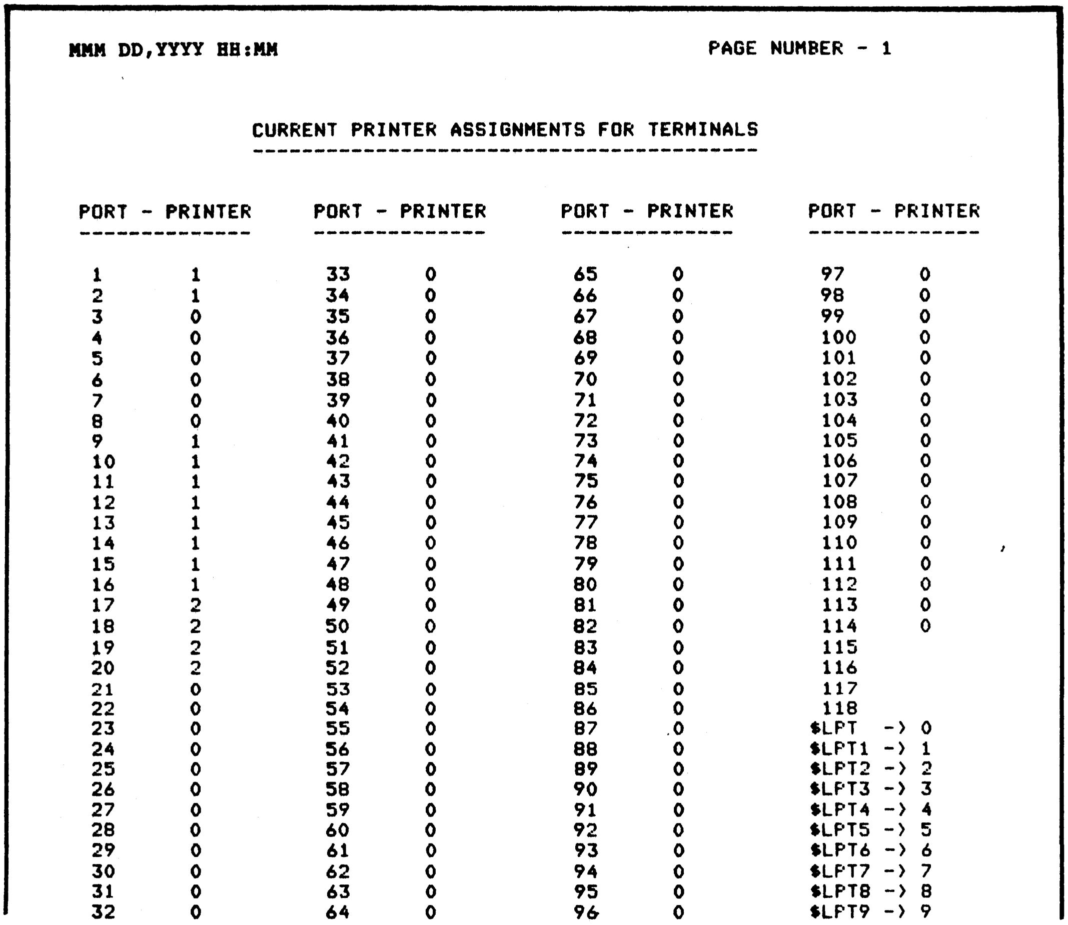
3.3.4 LIST CURRENT PORT OUTPUT ASSIGNMENTS

PURPOSE: This function prints a report which lists the current port-to-printer output assignments. This report specifies each port's assigned printer for FORCE documentation.

PROCEDURE: FORCE asks if the printer uses 8 1/2 by 11 paper. A (Y)es initiates printing of the report, and a (N)o returns to the Auxiliary Print Functions menu.

NOTES: Printer output assignments for FORCE documentation are maintained as a FORCE manager function (see Section 6.2.4).

Below is a sample report of port output assignments.



```
MMM DD,YYYY HH:MM                                    PAGE NUMBER - 1


                 CURRENT PRINTER ASSIGNMENTS FOR TERMINALS
                 -----------------------------------------

PORT - PRINTER     PORT - PRINTER     PORT - PRINTER     PORT - PRINTER
--------------     --------------     --------------     --------------

 1       1          33      0          65      0          97      0
 2       1          34      0          66      0          98      0
 3       0          35      0          67      0          99      0
 4       0          36      0          68      0          100     0
 5       0          37      0          69      0          101     0
 6       0          38      0          70      0          102     0
 7       0          39      0          71      0          103     0
 8       0          40      0          72      0          104     0
 9       1          41      0          73      0          105     0
 10      1          42      0          74      0          106     0
 11      1          43      0          75      0          107     0
 12      1          44      0          76      0          108     0
 13      1          45      0          77      0          109     0
 14      1          46      0          78      0          110     0
 15      1          47      0          79      0          111     0
 16      1          48      0          80      0          112     0
 17      2          49      0          81      0          113     0
 18      2          50      0          82      0          114     0
 19      2          51      0          83      0          115
 20      2          52      0          84      0          116
 21      0          53      0          85      0          117
 22      0          54      0          86      0          118
 23      0          55      0          87      0         $LPT  -> 0
 24      0          56      0          88      0         $LPT1 -> 1
 25      0          57      0          89      0         $LPT2 -> 2
 26      0          58      0          90      0         $LPT3 -> 3
 27      0          59      0          91      0         $LPT4 -> 4
 28      0          60      0          92      0         $LPT5 -> 5
 29      0          61      0          93      0         $LPT6 -> 6
 30      0          62      0          94      0         $LPT7 -> 7
 31      0          63      0          95      0         $LPT8 -> 8
 32      0          64      0          96      0         $LPT9 -> 9
```

3.3.5 PRINT DATA ELEMENT FORMAT MASKS

PURPOSE: This function prints selected reproductions of data element format masks, denoting the length of the mask and the associated data element. Data element format masks may be printed for a specific system, or for all systems. The report is arranged in ascending order by system data elements.

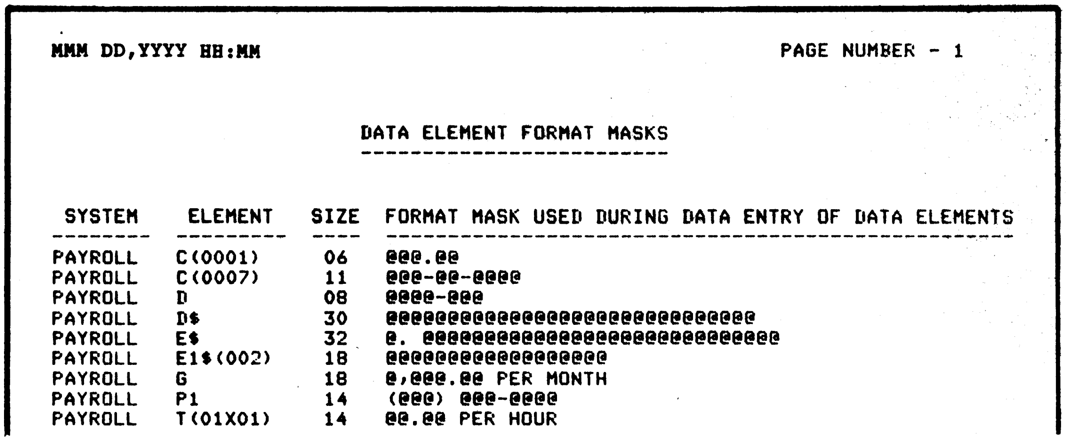
PROCEDURE: Enter the name of the system for which data element format masks are to be printed.

Enter RETURN to print data element format masks for all systems, beginning alphabetically with the first associated system name.

After specification of the report output, FORCE asks if the printer uses 8 1/2 x 11 paper. A (Y)es initiates printing of the report, and a (N)o returns to the Auxiliary Print Functions menu.

NOTES: Below is a sample format mask report.

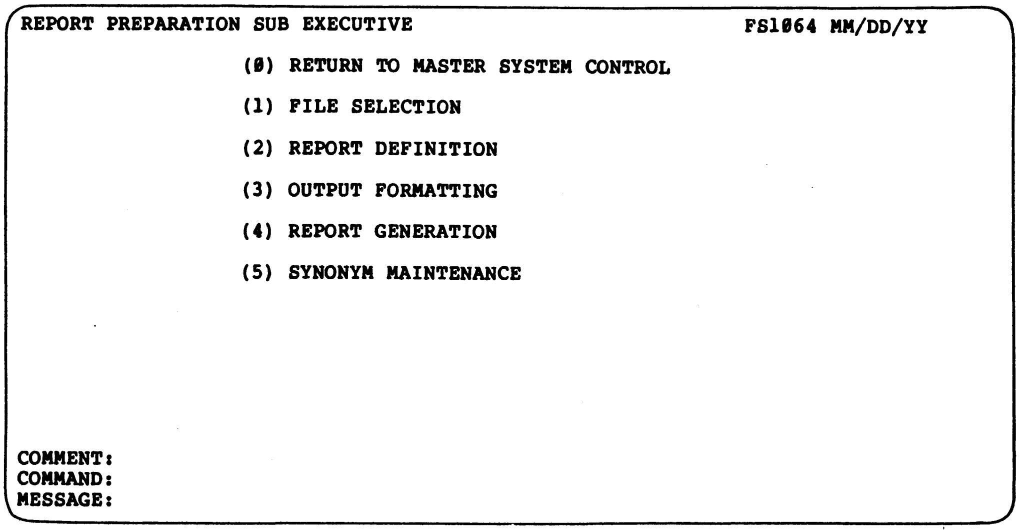
In order to fit on the report page, format masks are truncated at 50 characters. The SIZE column indicates the length of the mask in bytes (without truncation).



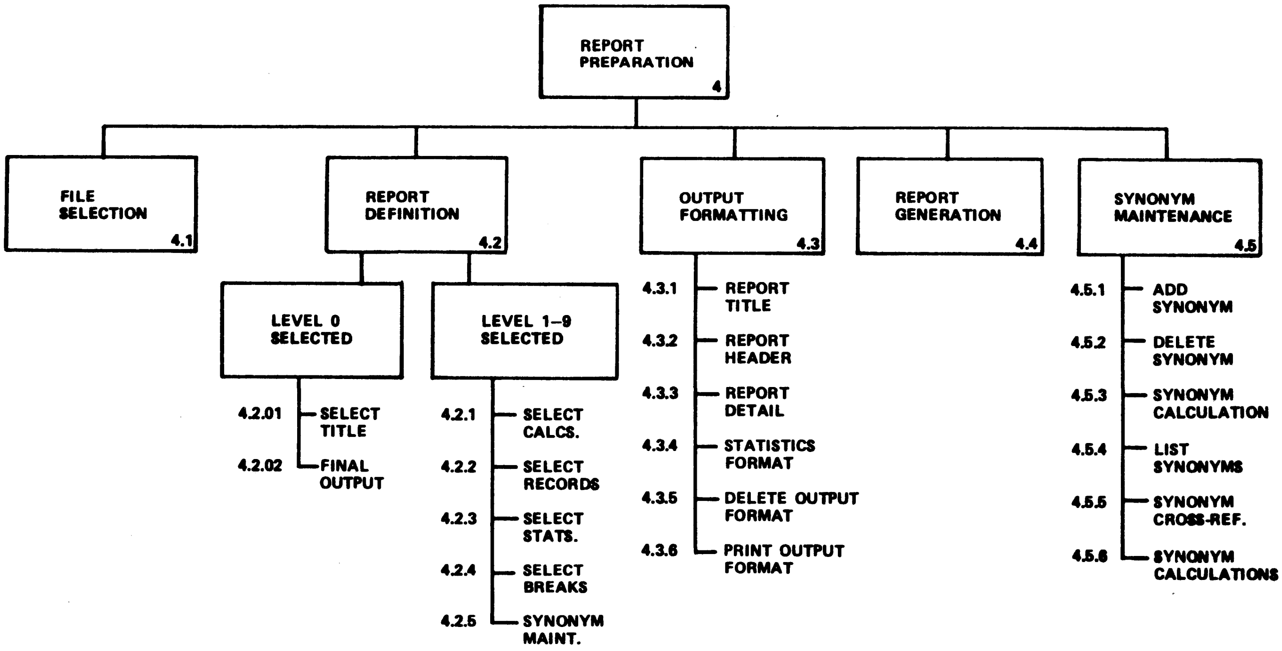
```
MMM DD,YYYY HH:MM                                        PAGE NUMBER - 1


                           DATA ELEMENT FORMAT MASKS
                           -------------------------


 SYSTEM     ELEMENT    SIZE  FORMAT MASK USED DURING DATA ENTRY OF DATA ELEMENTS
--------   ---------   ----  ----------------------------------------------------
PAYROLL    C(0001)      06   @@@.@@
PAYROLL    C(0007)      11   @@@-@@-@@@@
PAYROLL    D            08   @@@@-@@@
PAYROLL    D$           30   @@@@@@@@@@@@@@@@@@@@@@@@@@@@@@
PAYROLL    E$           32   @. @@@@@@@@@@@@@@@@@@@@@@@@@@@@@
PAYROLL    E1$(002)     18   @@@@@@@@@@@@@@@@@@
PAYROLL    G            18   @,@@@.@@ PER MONTH
PAYROLL    P1           14   (@@@) @@@-@@@@
PAYROLL    T(01X01)     14   @@.@@ PER HOUR
```

Section 4
REPORT PREPARATION

PURPOSE: The Report Preparation facility provides functions for generation of report program source code. Report program information is established and maintained within the Dictionary under this facility. Report programs, which may access up to nine files, use synonyms to reference all data. Output formats may be designed for the printing of titles, headers, details and statistics in the report output. Additionally, sophisticated record selection, break and sort criteria may be defined.



```
REPORT PREPARATION SUB EXECUTIVE                              FS1Ø64 MM/DD/YY

                         (Ø) RETURN TO MASTER SYSTEM CONTROL

                         (1) FILE SELECTION

                         (2) REPORT DEFINITION

                         (3) OUTPUT FORMATTING

                         (4) REPORT GENERATION

                         (5) SYNONYM MAINTENANCE







COMMENT:
COMMAND:
MESSAGE:
```



The Report Preparation menu selections provide functions for the establishment of report program Dictionary information, and the generation of report program source code.

1. File Selection designates the files to be used for each report level (I/O). Execution of this function establishes the program in the Dictionary as a report program. File Selection must precede all Report Preparation functions (except Synonym Maintenance).

2. Under Report Definition, each report level is defined in detail. For each level, the user may specify breaks, statistics and calculations, as well as record selection criteria. Final processing and synonym maintenance procedures may also be performed.

3. Output Formatting allows the user to design formats for the printing of titles, headers, details and statistics. Additionally, masks may be formatted for any printed data.

4. Report Generation is the facility which creates the report program source code based upon user-specified Dictionary information.

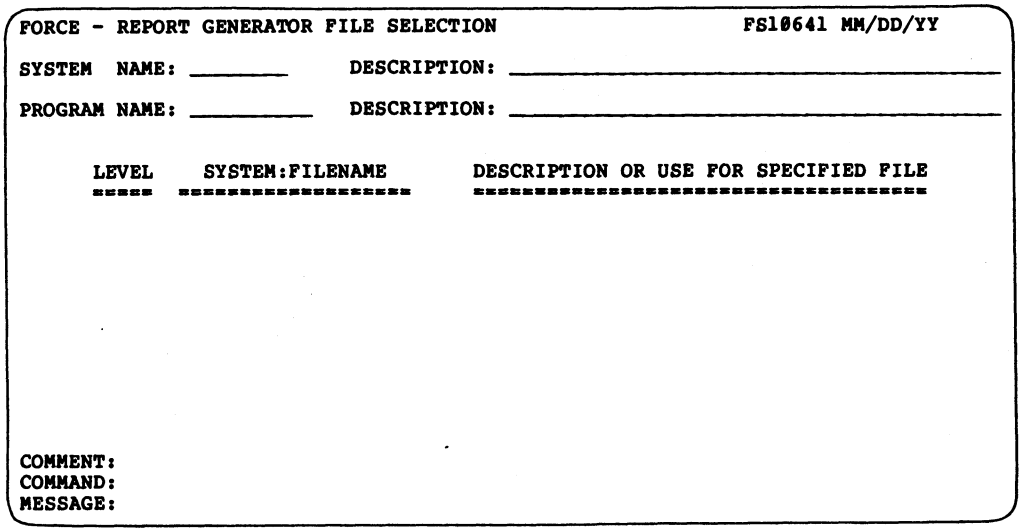
5. The Synonym Maintenance facility is used to define and maintain synonyms used in report program generation.

PROCEDURE: Enter the number which corresponds to the desired function. Subsequent menu choices are shown in the Report Preparation flowchart.

NOTES: Each report program requires at least one file I/O and one printout of accumulated data.

4.1 FILE SELECTION

PURPOSE: This function defines the specified program to FORCE as a report, and designates files to be used for each file I/O (report level). Additionally, file access information for each level is specified with this function.



```
FORCE - REPORT GENERATOR FILE SELECTION                    FS10641 MM/DD/YY

SYSTEM  NAME: _______     DESCRIPTION: __________________________________

PROGRAM NAME: _________   DESCRIPTION: __________________________________

      LEVEL    SYSTEM:FILENAME       DESCRIPTION OR USE FOR SPECIFIED FILE
      =====  ==================     =====================================









COMMENT:
COMMAND:
MESSAGE:
```

PROCEDURE: Prior to executing this function, components of the report (the program, files and synonyms) must be established within the Dictionary.

Enter the name of the system with which the report program is associated. Then enter the program name.

At the Level 1 entry, specify the file to be accessed at the first level of the report. If the file is associated with the same system as the report program, enter the filename. FORCE uses the system associated with the report program as a default for the system name entry.

If the file is associated with another system, specify the system and the file, using the format:

SYSTEM:FILENAME

Files used for report I/O must have been defined to FORCE as having an organization of (C)ontiguous, (F)ormatted or (I)ndexed.

Repeat file specification for each required level of the report program (up to nine levels).

When initial file specification is complete, FORCE displays the assigned file and access method for the first level. If the file's organization is indexed, the access method is indexed. For formatted or contiguous files, the access method is random.

FILE SELECTION 4.1

Under the SYSTEM:SYNONYM column now displayed, designate the synonym to store the file I/O value retrieved from an indexed access. The same synonym, representing the relative record number to be accessed, should be used for the next random read to the associated data file.

If a RETURN is entered at this field, FORCE relates a default synonym to the data.

If the file to be accessed is an index file, specify the Access Mode. Valid entries are AS IS, NEXT and EXACT. AS IS searches for a record that contains the key (regardless of its length). NEXT retrieves the record location following that of the previous search. EXACT searches for a record that matches (including the length) the key.

The default value for this entry field is AS IS.

Repeat these specifications for each level of the report.

NOTES: File Selection establishes the specified program within FORCE as a report program. Subsequent program development can then proceed.

Execution of this function with a previously established program invalidates the program's current assignments.

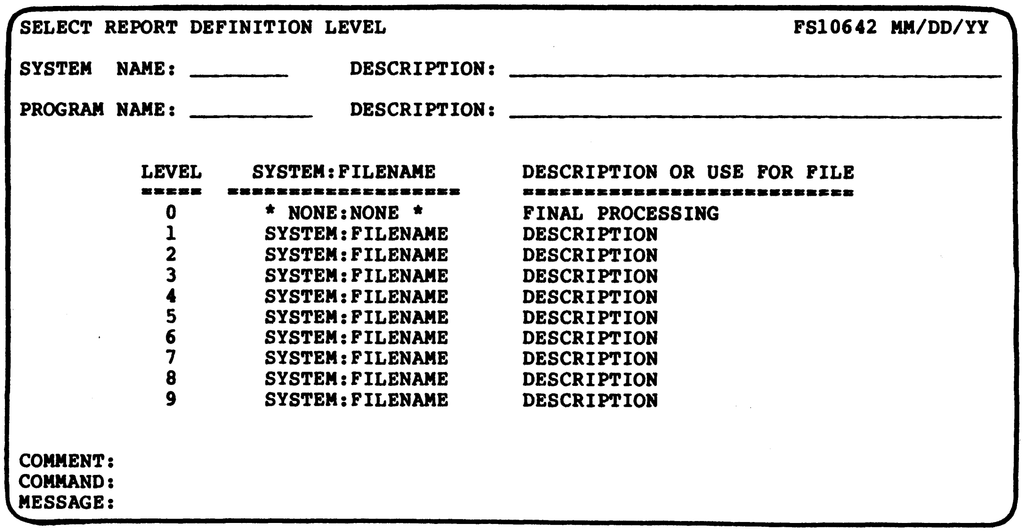
A file may be accessed repeatedly in a report.

4.2 REPORT DEFINITION

PURPOSE: This function initiates the definition of each report level. It provides an entry point to a menu for definition of Level 0 (final report processing), and a menu for definition of all other report levels (Levels 1-9).

Report Definition functions are accessed through this menu.

SPECIAL: Report Definition combines three operations depending on the report level to be defined. The initial screen chains to a menu for Level 0 definition, or a menu for Levels 1-9 definition. Both menus are documented in this section, with a separate PURPOSE, PROCEDURE and NOTES entry for each.



```
SELECT REPORT DEFINITION LEVEL                                  FS10642 MM/DD/YY

SYSTEM  NAME: ________      DESCRIPTION: ______________________________________

PROGRAM NAME: __________    DESCRIPTION: ______________________________________

          LEVEL    SYSTEM:FILENAME        DESCRIPTION OR USE FOR FILE
          =====  ===================      ===========================
            0       * NONE:NONE *         FINAL PROCESSING
            1       SYSTEM:FILENAME       DESCRIPTION
            2       SYSTEM:FILENAME       DESCRIPTION
            3       SYSTEM:FILENAME       DESCRIPTION
            4       SYSTEM:FILENAME       DESCRIPTION
            5       SYSTEM:FILENAME       DESCRIPTION
            6       SYSTEM:FILENAME       DESCRIPTION
            7       SYSTEM:FILENAME       DESCRIPTION
            8       SYSTEM:FILENAME       DESCRIPTION
            9       SYSTEM:FILENAME       DESCRIPTION

COMMENT:
COMMAND:
MESSAGE:
```

PROCEDURE: Enter the name of the system with which the report program is associated. Then enter the name of the program.

FORCE displays the program's file access information by level.

Enter the number of the report level to be defined. Selection of Level 0 invokes one menu screen; selection of Levels 1-9 invokes another.

NOTES: Level 0 processing provides functions which are performed at the end of a report.

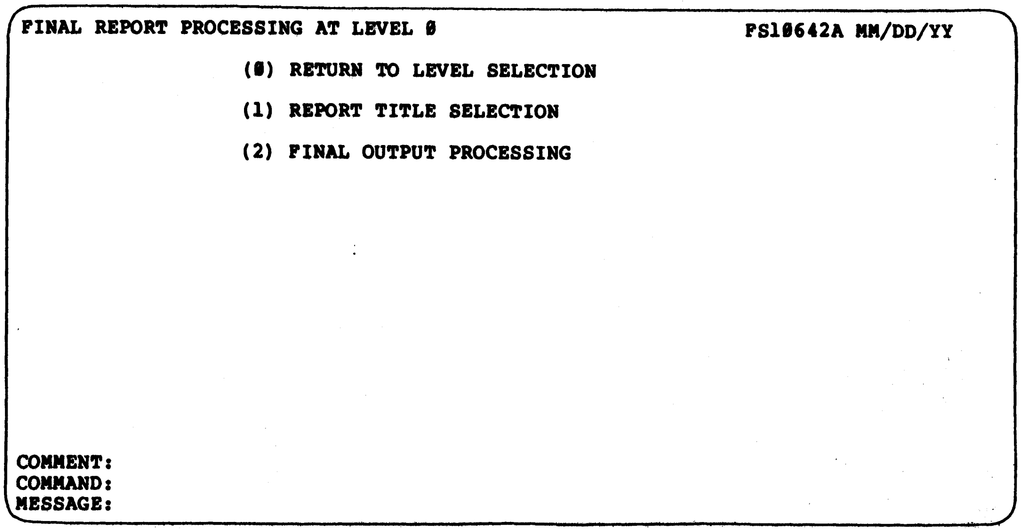
The facilities provided by the Report Definition menus allow specification of sophisticated report criteria.

Each report program requires at least one file I/O and one printout of accumulated data.

PURPOSE: This menu, presented after specification of Level 0 report definition, provides optional functions for final processing of a report program.

Title Selection allows the user to designate a previously-defined title format to be printed on each page of the report.

Final Output Processing establishes formats to be printed after the last report level is executed (i.e., after the final break).



```
FINAL REPORT PROCESSING AT LEVEL Ø                          PS1Ø642A MM/DD/YY

                         (Ø) RETURN TO LEVEL SELECTION

                         (1) REPORT TITLE SELECTION

                         (2) FINAL OUTPUT PROCESSING









COMMENT:
COMMAND:
MESSAGE:
```

PROCEDURE: Enter the number which corresponds to the desired function.

NOTES: Any report program may employ Level 0 processing.

PURPOSE: This menu is invoked after specification of Levels 1-9. It provides level-specific processing functions which may be applied to any report I/O.

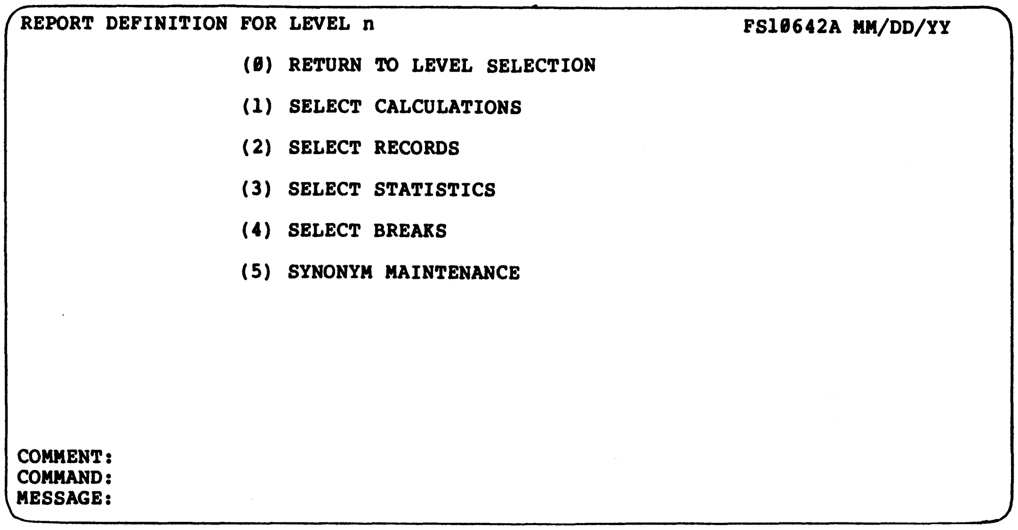
Select Calculations establishes a synonym calculation to be performed at the specified level of the report program.

Select Records allows the user to construct a conditional statement(s) for selection or rejection of records during level processing.

Select Statistics establishes a statistical computation to be performed at the specified report level.

Select Breaks sets a procedural break at the designated report level, and initiates printing of specified output formats and the associated data.

Synonym Maintenance is the facility with which Dictionary data elements are assigned and maintained as synonyms.



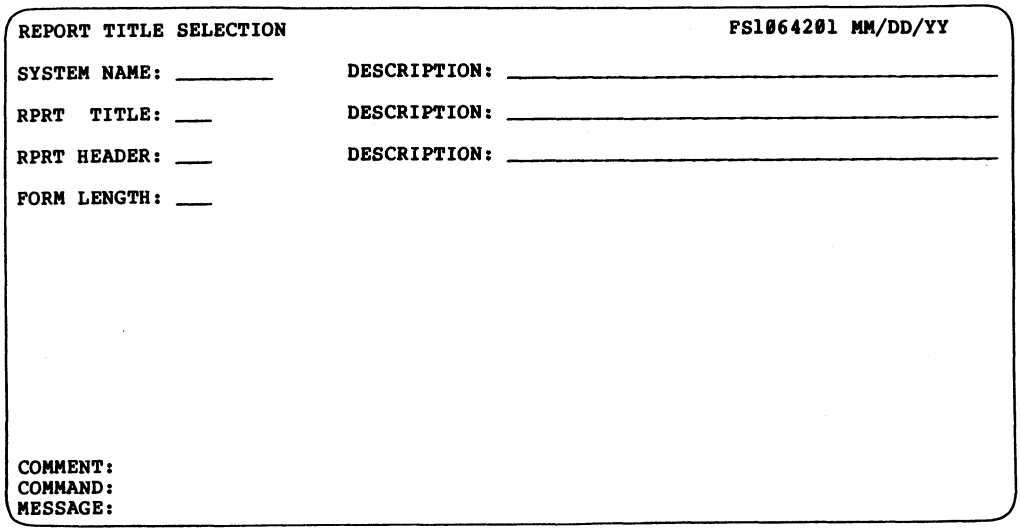
```
REPORT DEFINITION FOR LEVEL n                                  FS1Ø642A MM/DD/YY

                       (Ø) RETURN TO LEVEL SELECTION

                       (1) SELECT CALCULATIONS

                       (2) SELECT RECORDS

                       (3) SELECT STATISTICS

                       (4) SELECT BREAKS

                       (5) SYNONYM MAINTENANCE







COMMENT:
COMMAND:
MESSAGE:
```

PROCEDURE: Enter the number which corresponds to the desired function.

4.2.01 REPORT TITLE SELECTION

PURPOSE: This function selects a title format to be printed at the top of every report page. A header may be designated for printing beneath the title format. The length of the form on which the report is to be printed can also be specified.

This function is also used to modify or delete an established title selection.



```
REPORT TITLE SELECTION                                    FS1064201 MM/DD/YY

SYSTEM NAME: ________     DESCRIPTION: ______________________________

RPRT  TITLE: ___          DESCRIPTION: ______________________________

RPRT HEADER: ___          DESCRIPTION: ______________________________

FORM LENGTH: ___









COMMENT:
COMMAND:
MESSAGE:
```

PROCEDURE: Enter the name of the system with which the title to be printed is associated. Then enter the title format number.

The Report Header field allows the user to designate a header to be printed beneath the title format when detail length forces a new page and title. No header is printed in this situation if a RETURN is entered at the Report Header field.

REPORT TITLE
SELECTION
4.2.01

At the Form Length field, specify the number of vertical lines on the paper used for printing of the report. This entry ensures that the report output (printed data) accommodates the paper length.

A RETURN at the Form Length entry field establishes a default page length of 66 vertical lines.

When modifying an established title selection, the current information is displayed on the screen. New information may be entered, or current information left intact by pressing RETURN at each entry field.

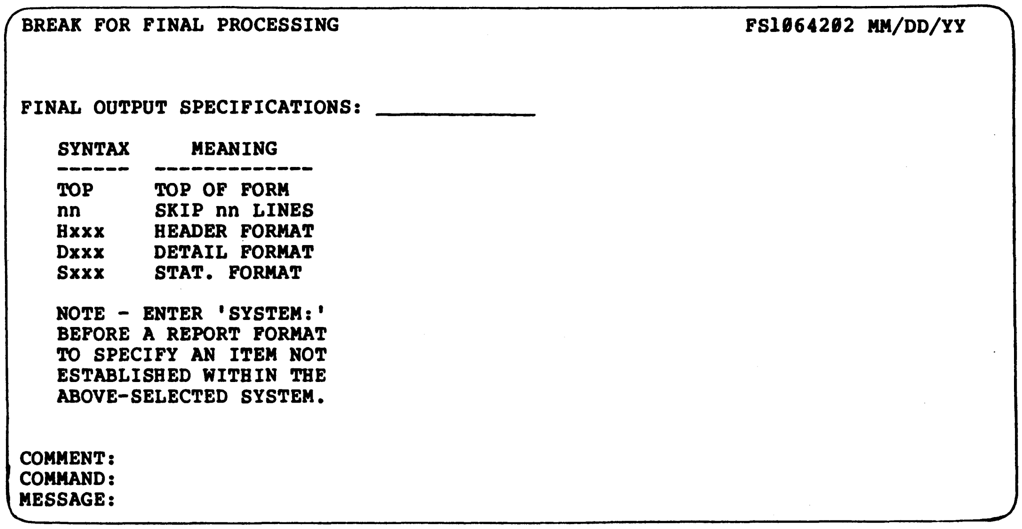
To delete an established title selection, press ESCAPE at System Name entry field. FORCE then prompts for deletion of the title from the report.

NOTES: A report may use only one title format.

4.2.02 FINAL OUTPUT PROCESSING

PURPOSE: This function establishes output formats to be printed after completion of the report levels. Header, detail and statistics formats may be printed in any combination, from any system. Additionally, vertical spacing commands may be specified.

This function may also be used to modify or delete an established final output.



```
BREAK FOR FINAL PROCESSING                                    FS1Ø642Ø2 MM/DD/YY


FINAL OUTPUT SPECIFICATIONS: ____________

   SYNTAX     MEANING
   ------  -------------
   TOP     TOP OF FORM
   nn      SKIP nn LINES
   Hxxx    HEADER FORMAT
   Dxxx    DETAIL FORMAT
   Sxxx    STAT. FORMAT

   NOTE - ENTER 'SYSTEM:'
   BEFORE A REPORT FORMAT
   TO SPECIFY AN ITEM NOT
   ESTABLISHED WITHIN THE
   ABOVE-SELECTED SYSTEM.


COMMENT:
COMMAND:
MESSAGE:
```

PROCEDURE: In the SYNTAX and MEANING columns, FORCE describes the output specifications that may be selected.

At the Final Output Specifications entry field, enter the desired print information. Each completed entry causes another blank input field to appear on the next line.

Output specifications are printed in the order in which they are entered (from top to bottom).

The TOP and Skip nn commands format the report output. Use the TOP function to issue a top of form (new page) at the specified report location. If a title is assigned to the report program, it will be printed at the top of the page. The Skip nn lines command prints the specified number of vertical blank lines at the designated location.

Use the information in the SYNTAX column to enter the report format(s) to be printed. The first character of the entry identifies the format type, and the following three characters denote the format number. Valid format type entries are: (H)eader, (D)etail and (S)tatistics.

To specify a report format from a different system (other than the system associated with the report program), enter SYSTEM: before the format type and number.

Enter RETURN at the entry field to indicate completion of output specifications.

FORCE checks the specified formats to ensure that synonym data used in the formats is accessed in the report levels.

When modifying a final output, the current information is displayed. New information may be entered, or current information may be left intact by a RETURN at the entry field. A * symbol entered over an existing input deletes that specification and all subsequent entries.

To delete an entire output, press ESCAPE at the top Final Output Specifications field. FORCE then prompts for deletion of the final output.

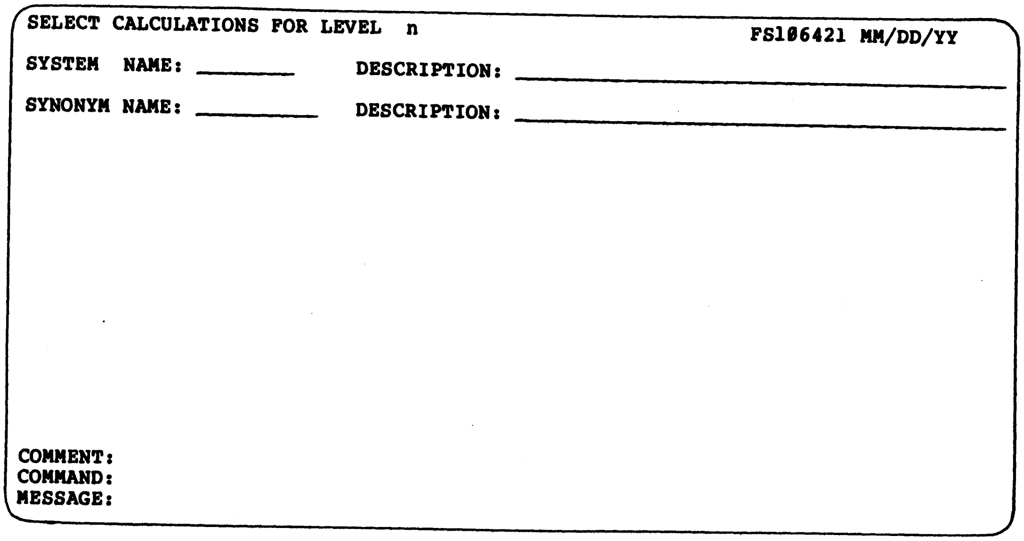
NOTES: Throughout the print specification process, FORCE checks to ensure that the synonyms used by the specified formats are available in the report program.

At least one output specification must be entered for a break.

Up to 16 print specifications may be entered at each break. Any combination of valid formats may be printed.

4.2.1 SELECT CALCULATIONS

PURPOSE: This function establishes calculated synonyms within a report level. Subsequently, the synonym calculation may be used in report level definition, or it may be printed in an output format.



```
SELECT CALCULATIONS FOR LEVEL  n                               FS1Ø6421 MM/DD/YY
SYSTEM  NAME: ________      DESCRIPTION: ______________________________________
SYNONYM NAME: _________     DESCRIPTION: ______________________________________











COMMENT:
COMMAND:
MESSAGE:
```

PROCEDURE: Prior to executing this function, the synonym calculation must be defined (see 4.5.3).

Enter the name of the system with which the calculated synonym is associated. Then enter the synonym name.

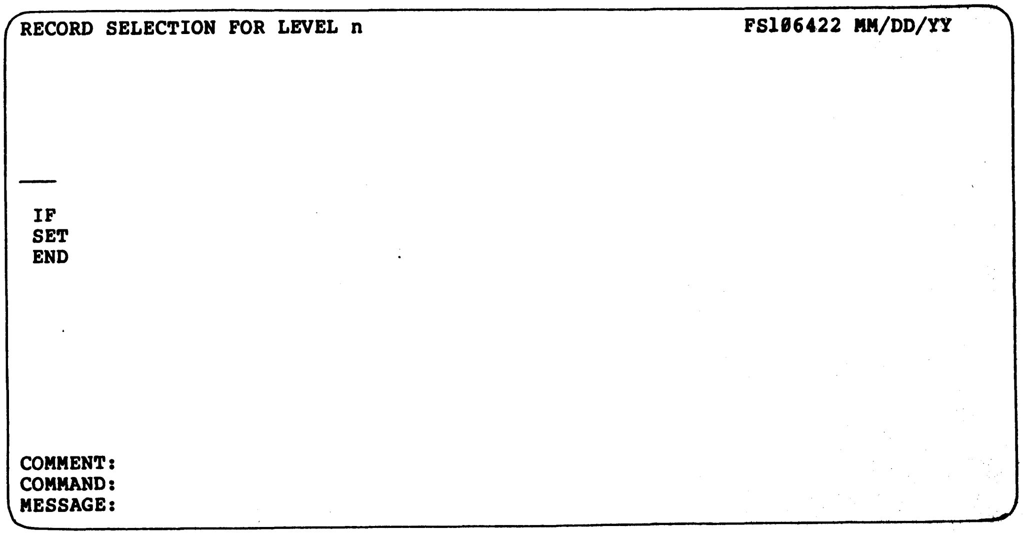
NOTES: FORCE displays the selected synonym calculation as it checks to ensure that synonyms used in the calculation are available at the specified report level.

SELECT CALCULATIONS 4.2.1

4.2.2 SELECT RECORDS

PURPOSE: This function is used to selectively accept or reject records for printing at each report level. Selection criteria is established by constructing conditional, English-language sentences which may nest logical AND/OR conjunctions.

This function may also be used to modify or delete established record selection criteria for a level.



```
RECORD SELECTION FOR LEVEL n                              FS1Ø6422 MM/DD/YY




___

 IF
 SET
 END





COMMENT:
COMMAND:
MESSAGE:
```

PROCEDURE: Record selection criteria is specified in sentence form along the displayed horizontal line. A completed selection specification represents an executable, high-level statement.

Prompts for valid entries are displayed beneath each input location on the line.

All selection line entries may be keyed in manually, and some may be input through the use of cursor tracking. This type of entry may be made at any selection line field that accepts literal input of the displayed prompts. At these fields, a RETURN on the selection line moves the cursor down, next to the first prompt. Vertical cursor tracking is then enabled. An entry may be input by positioning the cursor next to the desired selection and pressing RETURN. For example, the first vertical choice (IF) may be selected by entering two successive RETURNs.

Entry of ? at the first entry field displays a comprehensive help module explaining the record selection process.

Enter the first record selection specification.

Entry of SET at the initial field extends the selection line for entry of REJECT or TRANSFER operations. Upon entry of REJECT or TRANSFER, each level of the report is displayed, and the user specifies the level to which the selected operation transfers control. (See 8.4.1 for more on TRANSFER and REJECT.)

The following procedures apply to an IF sentence construction.

Enter IF at the first field.

At the second record selection field, specify the synonym, statistical function or sub-function to be evaluated. Valid functions and sub-functions are: (A)verage, (C)ount, (H)igh Value, (L)ow Value and (T)otal.

Input of ? at this entry field displays a list of synonyms available from the system associated with the report program. Use the format ?SYNONYM to begin the list alphabetically at a specific synonym. Enter ?SYSTEM: to list available synonyms from a specified system, or ?SYSTEM:SYNONYM to begin the list at a specific synonym within the system.

Only those synonyms accessed by the current or previous report levels are listed.

The third record selection line entry establishes a logical relationship for comparison of the selected values. Key in the entry, use cursor tracking and a RETURN to establish the input, or enter the symbol(s) which represent the desired logical relation (= <> < <= > >=).

Entry of a ? invokes a help module that explains the logical relationship specification.

At the next field, enter the data to be compared against the second field of the selection line. This data may be a synonym, a statistical function, sub-function or a literal value. Literal values may contain numerics or alphanumerics and must be enclosed within quotation marks. A literal value must conform to the field type and size of the synonym to which it is compared.

Enter ? to display a list of available synonyms.

The final specification establishes the selection operation to be performed. Valid entries are ACCEPT, REJECT, SELECT, AND and OR. AND and OR specifications indicate that another selection line will follow, and continue the selection statement.

Entry of ? at the final selection line field invokes a help module that explains the operation entry.

An END entry at the first field indicates that selection criteria is complete.

Any number of record-selection statements may be constructed for a level. Each statement may have any number of clauses, connected by AND or OR specifications. But each completed statement must conclude with ACCEPT, REJECT or SELECT.

Multiple selection statements are executed in the order in which they are specified.

To modify a record selection statement, use the ESCAPE key to back up to the field to be modified. (The ESCAPE key backs up between statements if required.) Then enter the modified information.

Note that each back-up with the ESCAPE key deletes the existing information. Information deleted by use of the ESCAPE key must be re-entered if it is to be retained.

To delete record selection criteria, use the ESCAPE key to back up over all selection line fields. Then enter END at the initial field of the first line.

NOTES: The ESCAPE key may be used to back up to previous entry fields, and to the last field of previous selection statements (if any). Each back-up deletes the existing entry. An ESCAPE will not exit from the first field of the first statement. The END entry must be input to exit the record selection function.

A selection statement may not mix AND and OR specifications within its clauses.

Specification of a statistical function or sub-function within record selection establishes (if not previously established - see 4.2.4) performance of that function within the program level. A statistical function established within the record selection process is performed before record selection criteria is applied.

There is no record selection involved in Level 0 processing.

Sample record selection statements are shown below.

```
IF ACCTNUM EQUALS "2891" ACCEPT

IF COMPNAME CONTAINS "CORP" OR
IF COMPNAME CONTAINS "INC." REJECT

IF ACCTBAL GREATER THAN CREDLMT ACCEPT
```

Table 4-1 describes each record selection-line specification.

TABLE 4-1. RECORD SELECTION SPECIFICATIONS

Prompt	Description
IF	Format - IF VALUE RELATED TO VALUE ACTION An IF statement establishes conditional record selection criteria to be evaluated.
SET	Format - SET CONTROL TO LEVEL n A SET statement designates program control to a specified report level for transfer and reject operations.
END	Format - END The END entry indicates completion of selection statement(s).
	Note: Subsequent field prompts on the selection line are determined by the choice of an IF or a SET entry.

Table continues on next page.

TABLE 4-1. RECORD SELECTION SPECIFICATIONS (Cont)

IF Statement - IF <u>**VALUE**</u> RELATED TO <u>**VALUE**</u> ACTION

VALUE Prompt	Description
SYNONYM	Compares the value of the specified synonym data. The specified synonym must be available in the current or a previous report level.
SYSTEM:SYNONYM	Specifies a synonym from a system other than the one associated with the report program.
F=SYNONYM	Indicates statistical functions performed on synonym data within the selection criteria. Substitute the desired function for the F entry. For example, A=SALARY is the input to reference the average of the SALARY synonym data.
F=SYSTEM:SYNONYM	Specifies a statistical function to be performed on a synonym from a system other than the one associated with the report program.
F-SYNONYM	Specifies statistical sub-functions. For example, C-EMPNUM is the input to specify the sub-count of the EMPNUM synonym data.
F-SYSTEM:SYNONYM	Specifies a statistical sub-function from a system other than the one associated with the report program.
"LITERAL"	Compares any literal information (alphanumeric or numeric) to the first specified value. Literals must be enclosed within quotes.

Table continues on next page.

TABLE 4-1. RECORD SELECTION SPECIFICATIONS (Cont)

IF Statement - IF VALUE **RELATED TO** VALUE ACTION

RELATED TO Prompt	Description
EQUAL TO	= may also be input.
NOT EQUAL TO	<> may also be input.
LESS THAN	< may also be input.
LESS THAN OR EQUAL TO	<= may also be input.
GREATER THAN	> may also be input.
GREATER THAN OR EQUAL TO	>= may also be input.
CONTAINS	Determine if the first value includes (literally) any occurrence of the compared value.

IF Statement - IF VALUE RELATED TO VALUE **ACTION**

ACTION Prompt	Description
ACCEPT	If record conforms to selection statement, continue record evaluation. If not, invoke the current reject level.
REJECT	If record conforms to selection statement, invoke the current reject level. If not, continue record evaluation.
SELECT	If record conforms to selection statement, select it, discontinue evaluation at the current level and proceed to the next. If not, continue record evaluation.
AND	Statement includes additional selection lines. Record must meet criteria specified in each line.
OR	Statement includes additional selection lines. Record may meet criteria specified in any line of the statement.

TABLE 4-1. RECORD SELECTION SPECIFICATIONS (Cont)

SET Statement - SET <u>CONTROL TO</u> LEVEL n

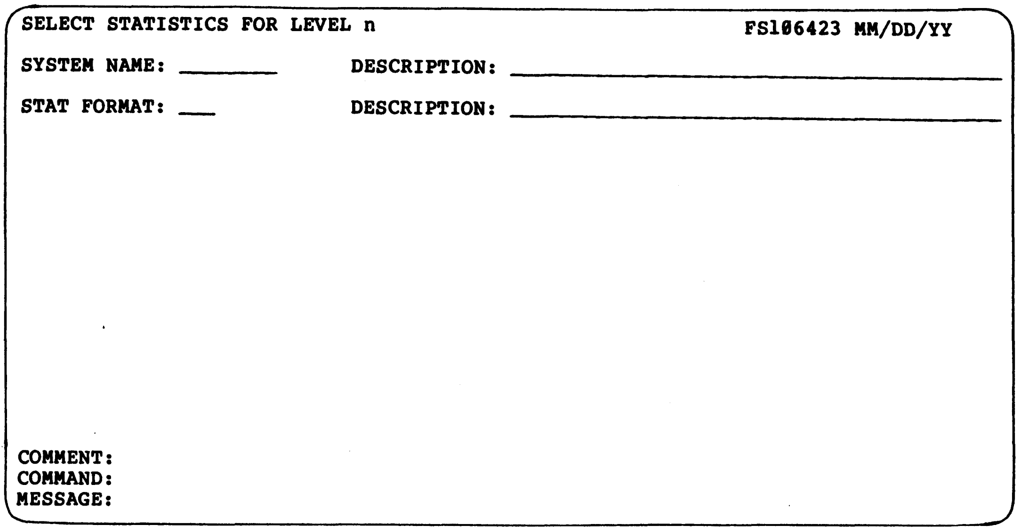
CONTROL TO Prompt	Description
REJECT	During record selection, establish a change of program control to the specified level for a reject operation. The reject is also invoked after a record has been processed by all report levels.
TRANSFER	Transfer control to specified level if indexed access is unsuccessful (e.g., record not found, end of file).

SET Statement - SET CONTROL TO <u>LEVEL n</u>

LEVEL n Prompt	Description
n	Specifies the level of the report to which control is to be transferred.

4.2.3 SELECT STATISTICS

PURPOSE: This function establishes computation of statistical values within a report level. The user specifies a statistics format which contains the statistical values to be maintained at the designated report level. Subsequently, the statistic may be used in report level definition, or it may be printed.



```
SELECT STATISTICS FOR LEVEL n                                   FS106423 MM/DD/YY

SYSTEM NAME: ________        DESCRIPTION: ______________________________________

STAT FORMAT: ___             DESCRIPTION: ______________________________________















COMMENT:
COMMAND:
MESSAGE:
```

PROCEDURE: Prior to executing this function, the referenced statistics format must be designed (see 4.3.4).

Enter the name of the system with which the statistics format is associated. Then enter the format number.

NOTES: FORCE scans the selected statistics format to ensure that the synonyms used are available at the specified report level.

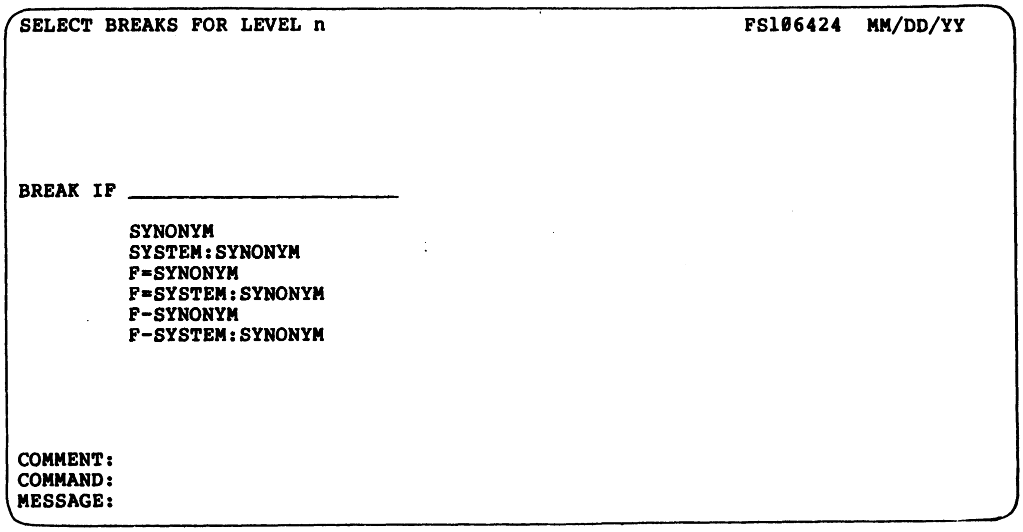
SELECT
STATISTICS
4.2.3

4.2.4 SELECT BREAKS

PURPOSE: This function establishes a procedural break(s) at the designated report level, and initiates printing of a specified output format(s) and the associated data. Breaks are selected by constructing a statement containing conditional break criteria, and specifying the output format(s) to be printed.

Break criteria may also be modified or deleted using this function.

SPECIAL: Select Breaks combines two operations. Each operation is documented in this section with a separate PROCEDURE and NOTES entry. The operations are presented as they occur during program execution.



```
SELECT BREAKS FOR LEVEL n                                  FS1Ø6424  MM/DD/YY




BREAK IF ____________________

           SYNONYM
           SYSTEM:SYNONYM
           F=SYNONYM
           F=SYSTEM:SYNONYM
           F-SYNONYM
           F-SYSTEM:SYNONYM




COMMENT:
COMMAND:
MESSAGE:
```

PROCEDURE: Break selection criteria is specified in sentence form along the displayed horizontal line. A completed break specification represents an executable, high-level statement.

Prompts for valid entries are displayed beneath each input location on the line.

At the BREAK IF prompt, enter the synonym, statistical function or sub-function to be evaluated by the break statement. Valid functions and sub-functions are: (A)verage, (C)ount, (H)igh Value, (L)ow Value and (T)otal.

Entry of ? at this field displays a list of synonyms available from the system associated with the report program. Use the format ?SYNONYM to begin the list alphabetically at a specific synonym. Enter ?SYSTEM: to list available synonyms from a specified system, or ?SYSTEM:SYNONYM to begin the list at a specific synonym within the system.

The second selection line field establishes a logical relationship for execution of break criteria. The user may key in a prompted entry, or enter the symbol(s) which represent the desired logical relation (= <> < <= > >=). An entry may also be input by pressing RETURN, which enables vertical cursor tracking, and pressing RETURN again after the cursor is positioned next to an entry.

Input of CHANGES at the second field completes the selection line entry. If any other logical relation is input, a third entry field is displayed.

At the third selection line field, specify the synonym, statistical function, sub-function or literal value to be compared against the initial field.

A literal value may be alphanumeric or numeric, but it must conform to the field type and size of the synonym with which it is compared. Enclose literal values within quotation marks.

An unconditional break may be used to initiate printing (see 8.4.2).

Upon completion of the break selection line, FORCE displays a print specification screen. The user then specifies the data to be printed at the break.

To modify or delete an established break statement, enter its selection line information. FORCE then prompts for deletion of that break criteria. An entry of (Y)es at the deletion prompt deletes the break statement entirely. A RETURN displays the break print specifications which may then be modified.

<u>NOTES</u>: The break statement is executed within the level each time a record conforms to the selection line criteria. Breaks are executed in the order in which they are specified.

A level may employ up to 26 break statements. The order in which they are specified is the order in which they are executed by the generated report program.

Throughout the break selection process, FORCE edits for the validity of the break statement by ensuring that all referenced synonyms are available at the specified report level.

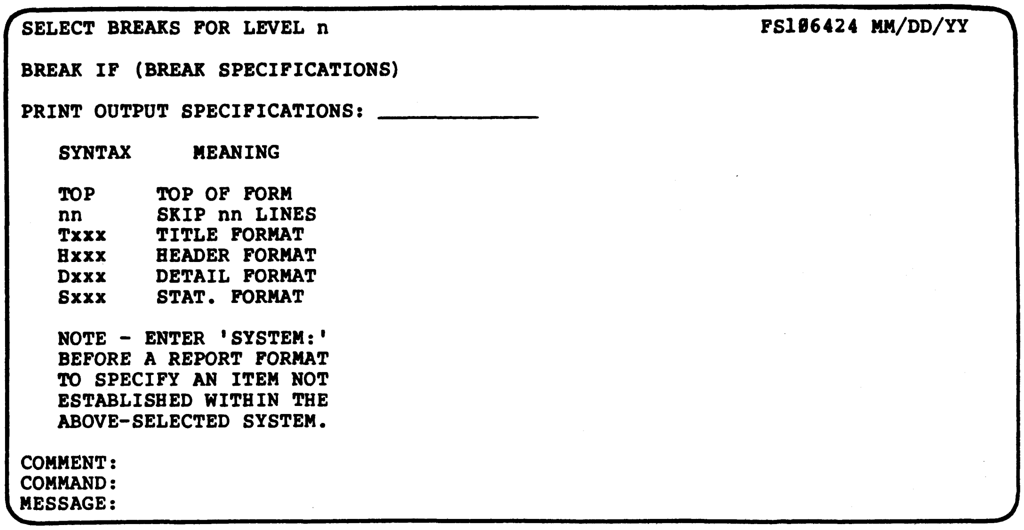
Sample break statements are shown below.

BREAK IF COST EQUALS PRICE

BREAK IF STATUS CHANGES

BREAK IF C=ITEM GREATER THAN "100"

BREAK IF ITEM EQUALS ITEM



```
SELECT BREAKS FOR LEVEL n                                        FS1Ø6424 MM/DD/YY

BREAK IF (BREAK SPECIFICATIONS)

PRINT OUTPUT SPECIFICATIONS: ____________

   SYNTAX     MEANING

   TOP      TOP OF FORM
   nn       SKIP nn LINES
   Txxx     TITLE FORMAT
   Hxxx     HEADER FORMAT
   Dxxx     DETAIL FORMAT
   Sxxx     STAT. FORMAT

   NOTE - ENTER 'SYSTEM:'
   BEFORE A REPORT FORMAT
   TO SPECIFY AN ITEM NOT
   ESTABLISHED WITHIN THE
   ABOVE-SELECTED SYSTEM.

COMMENT:
COMMAND:
MESSAGE:
```

PROCEDURE: In the SYNTAX and MEANING columns, FORCE describes the output specifications that may be selected. Selected specifications are printed each time a record conforms to the established break criteria.

At the Print Output Specifications prompt, enter the desired print information. Each completed entry causes another blank input field to appear on the next line.

Output specifications are printed in the order in which they are entered (from top to bottom).

The TOP and Skip nn commands format the report output. Use the TOP function to issue a top of form (new page) at the specified report location. If a title is assigned to the report program, it will be printed at the top of the page. The Skip nn lines command prints the specified number of vertical blank lines at the designated location.

Use the information in the SYNTAX column to enter the report format(s) to be printed. The first character of the entry identifies the format type, and the following three characters denote the format number. Valid format entries are: (H)eader, (D)etail and (S)tatistics.

To specify a report format from a different system (other than the system associated with the report program), enter SYSTEM: before the format type and number.

A RETURN at the print specification entry field indicates that selection is complete.

When modifying a break print specification, current entries are displayed. New information may be entered, or each field may be left intact by pressing RETURN. Input of the * symbol over an entry deletes that entry and all succeeding entries.

NOTES: Throughout the print specification process, FORCE checks to ensure that the synonyms used by the specified formats are available at that level of the report program.

At least one output specification must be entered for a break.

Up to 16 print specifications may be entered at each break. Any combination of valid formats may be printed.

4.2.5 SYNONYM MAINTENANCE

PURPOSE: Synonyms are used throughout the Report Preparation process to reference data elements within the Dictionary. This menu provides functions to establish and maintain synonyms for use in report generation.

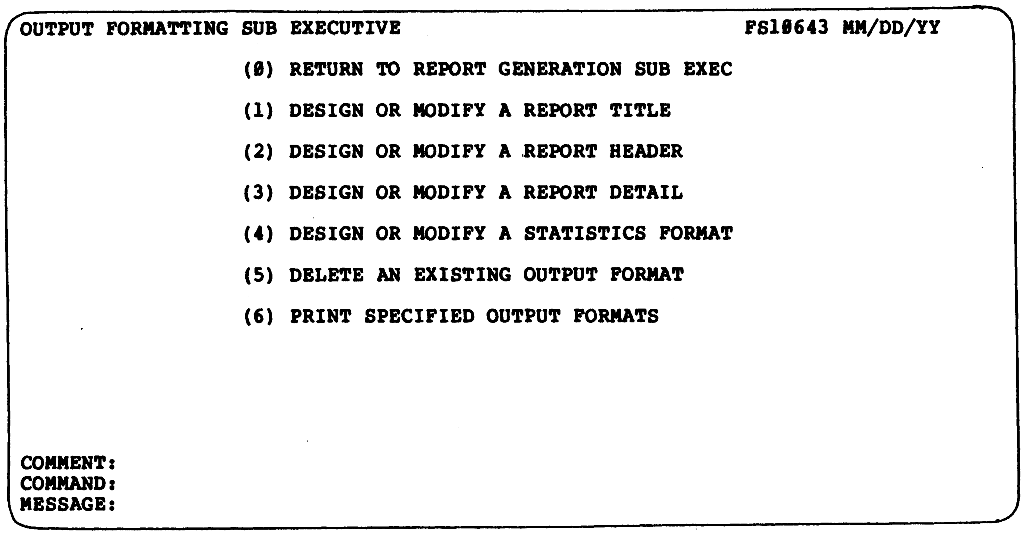
SPECIAL: This function provides a second entry point to the Synonym Maintenance facility so that the user can manage synonym information without having to exit Report Definition. If the user enters Synonym Maintenance through the Report Definition menu, an exit transfers control back to Report Definition.

See Section 4.5 for complete Synonym Maintenance documentation.

4.3 OUTPUT FORMATTING

PURPOSE: Output Formatting procedures enable the user to structure the appearance of a report. Four types of output formats may be designed for use in report generation: Title, Header, Detail and Statistics.

The Output Formatting menu provides functions to design, modify, delete and document each type of report format.



```
OUTPUT FORMATTING SUB EXECUTIVE                          FS1Ø643 MM/DD/YY

                    (Ø) RETURN TO REPORT GENERATION SUB EXEC

                    (1) DESIGN OR MODIFY A REPORT TITLE

                    (2) DESIGN OR MODIFY A REPORT HEADER

                    (3) DESIGN OR MODIFY A REPORT DETAIL

                    (4) DESIGN OR MODIFY A STATISTICS FORMAT

                    (5) DELETE AN EXISTING OUTPUT FORMAT

                    (6) PRINT SPECIFIED OUTPUT FORMATS




COMMENT:
COMMAND:
MESSAGE:
```

PROCEDURE: Enter the number which corresponds to the desired function.

NOTES: FORCE stores the image of a designed report format, and may subsequently generate the corresponding source code.

4.3.1 DESIGN OR MODIFY A REPORT TITLE

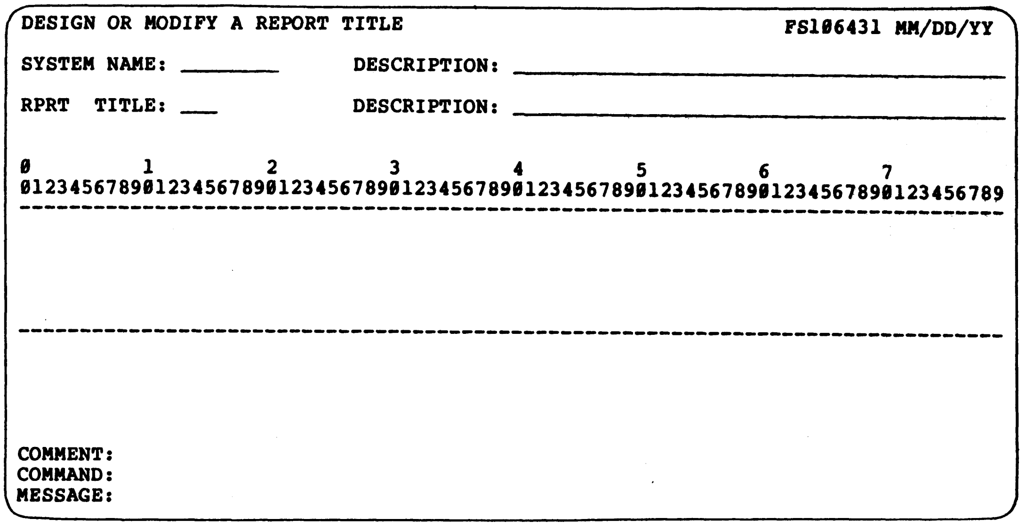
PURPOSE: This function designs a title for use in any report output by specifying title contents and their print positions. Titles are designed by creating the desired format on the CRT.

Title formats are printed at the top of every page of a report. Literals, synonyms, as well as date and page functions may be printed in a report title.

Report titles are assigned a number and stored in the Dictionary with their associated system. FORCE may then generate title-formatting source code by referencing the image of the title in the Dictionary.

This function also may be used to modify a previously-designed title.

SPECIAL: Design or Modify a Report Title combines three operations. Each operation is documented in this section with a separate PROCEDURE and NOTES entry. The operations are presented as they occur during execution of this function.



```
DESIGN OR MODIFY A REPORT TITLE                              FS1Ø6431 MM/DD/YY

SYSTEM NAME: ________     DESCRIPTION: ______________________________

RPRT  TITLE: ___          DESCRIPTION: ______________________________

Ø         1         2         3         4         5         6         7
Ø123456789Ø123456789Ø123456789Ø123456789Ø123456789Ø123456789Ø123456789Ø123456789
--------------------------------------------------------------------------------



--------------------------------------------------------------------------------




COMMENT:
COMMAND:
MESSAGE:
```

PROCEDURE: To design a title, enter the name of the system with which the title format is associated. Then enter a three-character, alphanumeric title number and a description.

FORCE displays a title formatting area. The program is in Screen Formatting Mode as indicated on the Comment Line. While in Screen Formatting Mode, use the terminal's directional keys to position the cursor.

Literal information, synonym data, and system page and date functions may be printed in a title. The @ symbol is used to specify each location where synonym data is to be printed.

Except for the @ symbol, any entry in the formatting area is duplicated literally when printed in the title.

Locations for page and date functions are denoted with the @ symbol, and formatted later in the design process.

Print locations (@ symbols) or literals may be specified anywhere within the five bordered vertical lines and the 132 denoted horizontal positions. The screen is refreshed accordingly when formatting extends beyond the defined area.

Press RETURN after each entry. The first RETURN updates cursor position status and stores the entry in memory. A second RETURN transfers the cursor to the first position of the next line, then updates cursor position.

An ESCAPE transters control to Command Input Mode, as indicated on the Comment Line. With the cursor positioned on the Command Line, the user may enter commands which facilitate the title formatting process. The commands are listed below:

Command	Activity
A	Abort (used only in title modification)
C	Center a line on the screen
D	Delete characters from within a line
E	Erase a line from the screen
G	Get a report header format
H	Help module
I	Insert characters within a line
M	Move one line to another line
S	Save the screen in the Dictionary
V	View title format at an offset
X	Extract and refresh screen display
?	Command Line summary information

The command H presents a help module for title formatting. Table 4-2 describes the formatting commands in detail.

Control is transferred between Screen Formatting Mode and Command Input Mode by pressing the ESCAPE key. A RETURN also transfers control from Command Input to Screen Formatting.

Enter the information, either literals or @ symbols, at the appropriate location in the formatting area.

After the title is designed, issue a Save command. FORCE then displays synonym specification entry fields.

To modify a previously-formatted title, enter the system name and the title number. The title format description may be modified, or left intact by a RETURN at the Description entry field. FORCE then displays the current title format on the screen.

The title format may be modified using any of the methods available when designing a format.

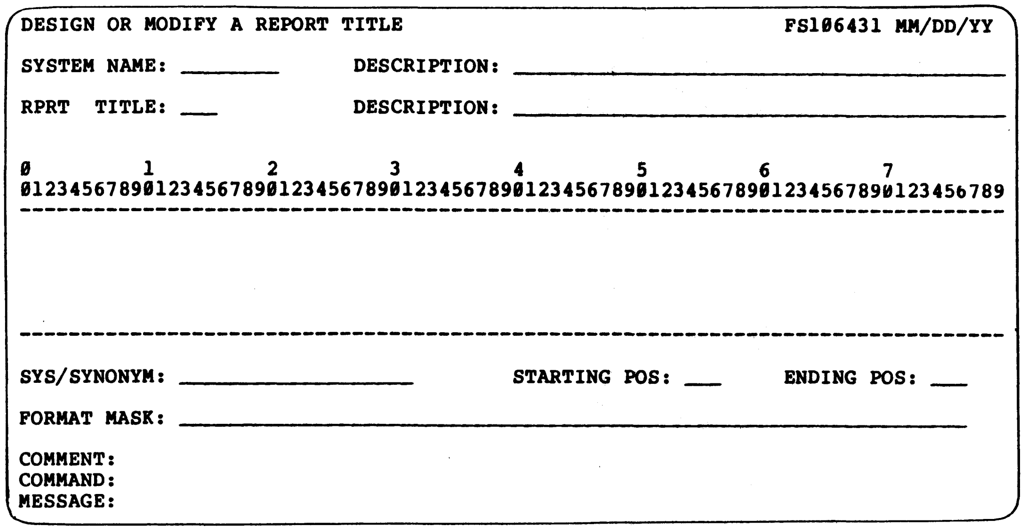
NOTES: The Save command is the only way to exit the title design function. When modifying a format, the Save or the Abort command may be used to exit.

A report may use only one title format.

Any blank lines at the top of the formatting area are printed in the report output above the title. Blank lines in the bottom of the title formatting area are not printed in the report output.

The Get command displays a selected header (dimmed) in the title formatting area. Use the Get command to facilitate alignment of a title position over a header format.

Title numbers within a system must be unique. The same number, however, may be assigned to related header, detail and statistics formats.



```
DESIGN OR MODIFY A REPORT TITLE                                FS1Ø6431 MM/DD/YY

SYSTEM NAME: ________        DESCRIPTION: ________________________________________

RPRT  TITLE: ___             DESCRIPTION: ________________________________________

Ø         1         2         3         4         5         6         7
Ø123456789Ø123456789Ø123456789Ø123456789Ø123456789Ø123456789Ø123456789Ø123456789
--------------------------------------------------------------------------------




--------------------------------------------------------------------------------

SYS/SYNONYM: ___________________          STARTING POS: ___    ENDING POS: ___

FORMAT MASK: ____________________________________________________________

COMMENT:
COMMAND:
MESSAGE:
```

PROCEDURE: FORCE prompts for specification of the related synonym data, page function, or date function to be printed at each @ symbol. Print locations are prompted by order of their occurrence in the format area; from left to right, top to bottom.

FORCE substitutes a question mark (?) for the @ symbol with which a value is to be related.

At the System/Synonym entry field, specify the synonym or the system function to be printed at the location denoted by a question mark. To print the system date or the page, use the format:

    @DATE   or   @PAGE

The System/Synonym entry also may be used to assign synonym data to a @ symbol in the title format. If the synonym is associated with the same system as the report title, enter the synonym name. If the synonym is associated with a different system, enter the system name and the synonym name using the format:

```
    SYSTEM:SYNONYM
```

Starting and ending print positions are then displayed, based upon the length of the associated data element. For synonyms containing string data, this information may be left intact by a RETURN at the print position entry fields, or modified information may be entered. For synonyms related to numeric data, the print positions may not be modified.

Synonym specification may be aided by displaying selective lists of synonyms on the screen (see NOTES).

At the Format Mask entry field, format the printed synonym data. The @ symbol designates each print position of the represented data. Any other displayable keyboard characters (including blanks) may be placed within, before or following the printed data.

For example, consider a synonym for a telephone number. The synonym, PHONENO, may use the following format mask for printing in a title:

```
    (@@@) @@@-@@@@
```

If PHONENO contains the number 4084585278, the data is printed in the format:

```
    (408) 458-5278
```

A RETURN at the Format Mask entry field establishes a mask of @ symbols for the length of the synonym data.

FORCE uses the | symbol to denote the end of a formatted mask. This symbol is not printed on the report output.

Format masking is repeated for each synonym to be printed in the report. When masking is completed, FORCE clears the screen and displays prompts for field formatting.

When modifying a title, the current synonym information is displayed after the format is reviewed. New information may be entered, or the information may be left intact by a RETURN at each entry field.

<u>NOTES</u>: Page number functions are printed up to six digits long.

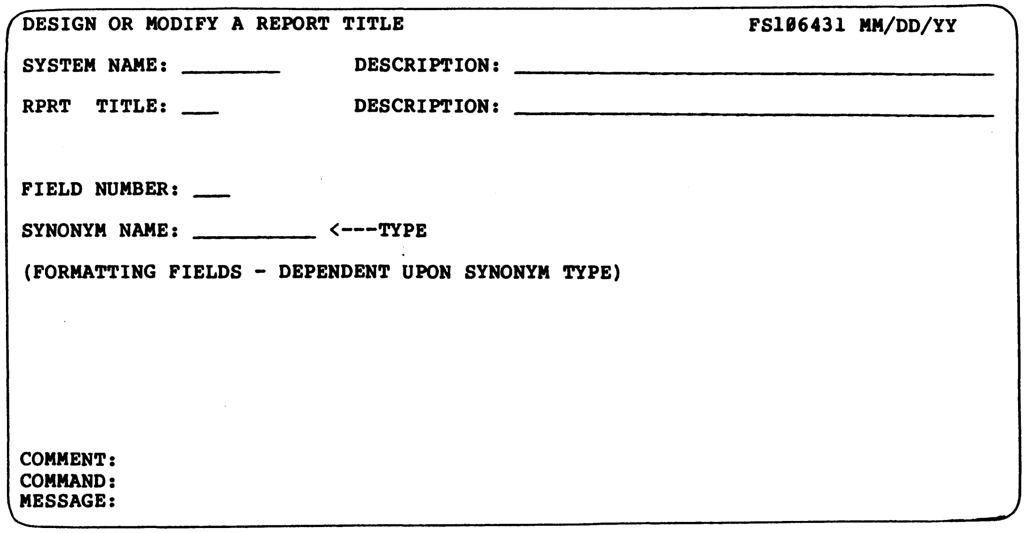
The date function in title formatting prints the system date up to the minutes, using the format:

```
    JUL 25, 1982  14:21
```

To assist referencing of synonyms that correspond to title format @ symbols, a selective list of available synonyms may be displayed on the screen. Entry of ? displays an alphabetic list of synonyms from the system associated with the title format.

Use the format ?SYNONYM to begin the synonym list alphabetically with the specified synonym. An entry of ?SYSTEM: lists all synonyms within the specified system. Enter ?SYSTEM:SYNONYM to begin the list at a selected synonym within a specified system. Calculated synonyms are denoted by an asterisk.

The ESCAPE key may be used to exit the synonym query list.



```
DESIGN OR MODIFY A REPORT TITLE                              FS106431 MM/DD/YY

SYSTEM NAME: _______       DESCRIPTION: ______________________________________

RPRT  TITLE: ___           DESCRIPTION: ______________________________________


FIELD NUMBER: ___

SYNONYM NAME: _________ <---TYPE

(FORMATTING FIELDS - DEPENDENT UPON SYNONYM TYPE)







COMMENT:
COMMAND:
MESSAGE:
```

PROCEDURE: Each of the previously specified title fields may be formatted additionally for printing. FORCE displays each field number, its associated system and synonym, and the field type.

Field types are Numeric, Alphanumeric and System (for date and page fields).

Field formatting procedures differ slightly for alphanumeric and numeric fields.

For alphanumeric data, specify a Justify Mode of (L)eft, (R)ight or (N)one. A RETURN at the Justify Mode entry defaults to (N)one. The data is printed as it is stored in the file.

At the Filler Character entry field, specify any keyboard character to fill the length of the synonym when the current data occupies only a portion of the specified length. If (N)one is entered at the Justify Mode field, this entry is bypassed. No filler is used if a RETURN is pressed at the Filler Character entry field.

Numeric data is always right justified when printed, so filler characters for numerics are printed to the left of the first digit.

The Float Character entry field allows specification of a character to be printed immediately before numeric data, regardless of its length. For example, printed financial figures may use a floating $ sign. Enter RETURN at the Float Character entry field if no float character is required.

At the Field Format entry specify (R)ounding or (T)runcating operations to be performed on numeric data before it is printed. A RETURN at the Field Format entry causes (N)one to be used.

If rounding or truncation is to be performed, a mask is displayed which represents the synonym length as established by the previously-defined format mask. The space bar is then used to move the cursor above the position at which the synonym is to be truncated or rounded. Enter X at the desired location. The Format Mask is then displayed with the specified value.

A RETURN at the Format Mask entry field rounds or truncates to the length of the synonym.

The synonym formatting process is repeated for each synonym in the title.

When modifying a title format, the current field formatting information is diplayed after review of synonym information. New data may be entered at each entry field, or the current information may be left intact by pressing RETURN.

Enter a blank space to remove a single-character input from an entry field and establish the default entry.

NOTES: Table 4-2 describes report output formatting commands in detail.

TABLE 4-2. REPORT OUTPUT FORMATTING COMMANDS

Command	Function
A	(A)bort a screen modification Used only when modifying an output format. To abort modification of an output format: ● Enter A and RETURN at the Command Line. Control returns to the Screen Formatting Menu.
C	(C)enter a line on the screen ● Enter C while in Command Input Mode. ● FORCE prompts for the Last Print Column, which indicates the width of the paper. Enter the number of horizontal print positions on the page, so that the center of the line can be calculated. ● Enter the number of the line to be centered and RETURN. Control returns to Command Input Mode.
D	(D)elete characters from a line ● Position cursor at the desired deletion point. ● Press RETURN to set cursor position line coordinates in the status information. ● Press ESCAPE to return to Command Input Mode. ● Enter D at the Command Line, then press RETURN to enter Delete Character Mode. Enter a second D, followed by a RETURN, to delete a character at the specified cursor coordinates. The remaining text on that line is moved back one character position. Additional D entries delete succeeding characters which have been moved into the specified cursor coordinate. ● Enter RETURN at the Command Line to return to Command Input Mode.

Table continues on next page.

TABLE 4-2. REPORT OUTPUT FORMATTING COMMANDS (Cont)

Command	Function
E	(E)rase an entire line from the screen • Enter E and RETURN while in Command Input Mode. • FORCE prompts for the line number to be deleted. Entry of a line number plus RETURN deletes the entire line and returns control to Command Input Mode. Entry of a RETURN only returns control to Command Input Mode without deleting a line.
G	(G)et another output format This command facilitates the vertical alignment of output formats to be printed together (header over a detail, title over a header, etc.) by displaying a specified format when creating its counterpart. To display (dimmed) a selected output format in the formatting area: • Enter G while in Command Input Mode. • FORCE prompts for the output format (from the same system) to be displayed. When creating a title or a detail format, only header formats may be displayed. Detail formats are displayed when creating header and statistics formats. • Enter the output format number to be displayed. A RETURN defaults to the same format number as the one being created (different format types may use the same number). • The specified format is displayed (dimmed). It may be typed over, and it is not retained when the current format is saved.
H	(H)elp module To display a list of commands with an explanation of each function: • Enter H and RETURN at the Command Line. A list of formatting commands and their functions is displayed on the screen. • Press RETURN to return to Command Input Mode.

TABLE 4-2. REPORT OUTPUT FORMATTING COMMANDS (Cont)

Command	Function
I	(I)nsert characters within a line ● Position cursor at desired insert location. ● Press RETURN to set cursor position and line coordinates in the status information. ● Press ESCAPE to return to Command Input Mode. ● Enter I at the Command Line and press RETURN. Data between the cursor and the end of the line disappears. ● Enter data to be inserted and press RETURN. The remainder of the original line is appended to the insertion.
M	(M)ove one line to another line ● Enter M while in Command Input Mode. FORCE prompts for the line to be moved, and the line on the screen to which it is to be moved. ● Enter the prompted line numbers and RETURN. Control returns to Command Input Mode after the line is moved.
S	(S)ave a designed ouput format To store an output format in the Dictionary, associated with the system name and screen number specified when the screen was created: ● Enter S and RETURN at the Command Line. Control then returns to the Screen Formatting Menu.

Table continues on next page.

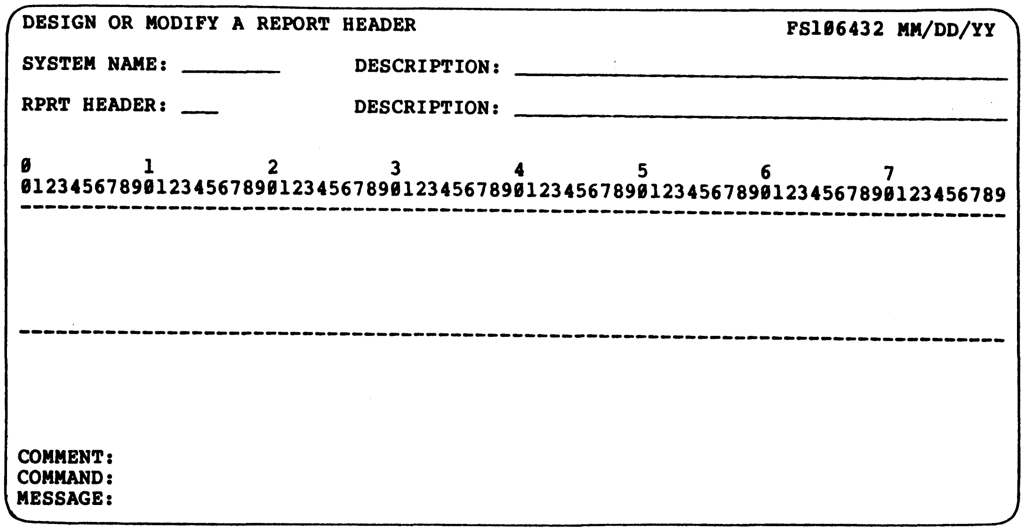
TABLE 4-2. REPORT OUTPUT FORMATTING COMMANDS (Cont)

Command	Function
V	(V)iew format from a specified offset This command enables the user to selectively view a format whose width exceeds that of the CRT screen. To view a format beginning with a specified horizontal position: ● Enter V while in Command Input Mode. ● FORCE prompts for the horizontal offset from which the format is to be displayed. Enter the horizontal offset number (these numbers are displayed bordering the formatting area). A RETURN defaults to an entry of offset zero. ● The format is displayed accordingly, and control returns to Command Input Mode.
X	Extract and refresh screen display To reinstate a format's original state if it has been disorganized by overrun of screen dimensions when using up-and-down arrows, or by other operator input error: ● Press RETURN before leaving Screen Formatting Mode to save completed work and prevent loss of additions or modifications. ● Use the ESCAPE key to enter Command Input Mode. ● Enter X and RETURN at the Command Line. The screen clears and the correct screen format is displayed. Control returns to Command Input Mode.
?	Command Line summary For a quick reference summary of the commands: ● Enter ? and RETURN at the Command Line. A summary of valid commands is displayed, and control transfers to Command Input Mode.

4.3.2 DESIGN OR MODIFY A REPORT HEADER

PURPOSE: This function designs a header for use in any report output by creating the desired format on the CRT. Formatted headers are assigned a number and stored in the Dictionary with their associated system. FORCE may then generate header formatting source code by referencing the image of the header in the Dictionary.

This function also may be used to modify a previously-designed header.



```
DESIGN OR MODIFY A REPORT HEADER                                FS1Ø6432 MM/DD/YY

SYSTEM NAME: ________       DESCRIPTION: _________________________________________

RPRT HEADER: ___            DESCRIPTION: _________________________________________

Ø         1         2         3         4         5         6         7
Ø123456789Ø123456789Ø123456789Ø123456789Ø123456789Ø123456789Ø123456789Ø123456789
--------------------------------------------------------------------------------




--------------------------------------------------------------------------------




COMMENT:
COMMAND:
MESSAGE:
```

PROCEDURE: Enter the name of the system associated with the header. Then enter a three-character, alphanumeric header number and a description.

DESIGN/MODIFY
REPORT HEADER
4.3.2

FORCE displays a header-formatting area. The program is in Screen Formatting Mode as indicated on the Comment Line. While in Screen Formatting Mode use the terminal's cursor control keys to position the cursor at the desired location and enter literal information exactly as it is to appear in the header.

Formatting may be performed anywhere within the five bordered vertical lines and the 132 denoted horizontal positions (see Table 4-2, View Command). The screen is refreshed accordingly when formatting extends beyond the defined area.

Press RETURN after each completed line entry. The first RETURN updates cursor position status and enters the previous data string into memory. A second RETURN transfers the cursor to the first position of the next line, then updates the cursor position.

An ESCAPE transfers control to Command Input Mode, as indicated on the Comment Line. With the cursor positioned on the Command Line, the user may enter commands which facilitate the header formatting process. The commands are listed below:

Command	Activity
A	Abort (used only in header modification)
C	Center a line on the screen
D	Delete characters from within a line
E	Erase a line from the screen.
G	Get a report detail format
H	Help module
I	Insert characters within a line
M	Move one line to another line
S	Save the screen in the Dictionary
V	View header format at an offset
X	Extract and refresh screen display
?	Command Line summary information

The command H presents a help module for header formatting. Table 4-2 describes the formatting commands in detail.

Control is transferred between Screen Formatting Mode and Command Input Mode by pressing the ESCAPE key. A RETURN also transfers control from Command Input to Screen Formatting.

To modify a previously-formatted header, enter the system name and the header number. FORCE displays the current header format on the screen. The header format description may be modified, or left intact by a RETURN at the Description entry field.

The header format may then be modified using any of the methods available when designing a format.

NOTES: The Save command is the only way to exit the header design function. When modifying a format, the Save or the Abort command may be used to exit.

One blank line is always printed above a header, regardless of the number of blank lines at the top of the header formatting area. Blank lines at the bottom of the formatting area are not printed in the report output.

A Get command displays a selected report detail (dimmed) in the header formatting area. Use the Get command (see Table 4-2) to facilitate alignment of a header position over a detail format.

Header numbers within a system must be unique. The same number may be assigned to related title, detail and statistics formats.

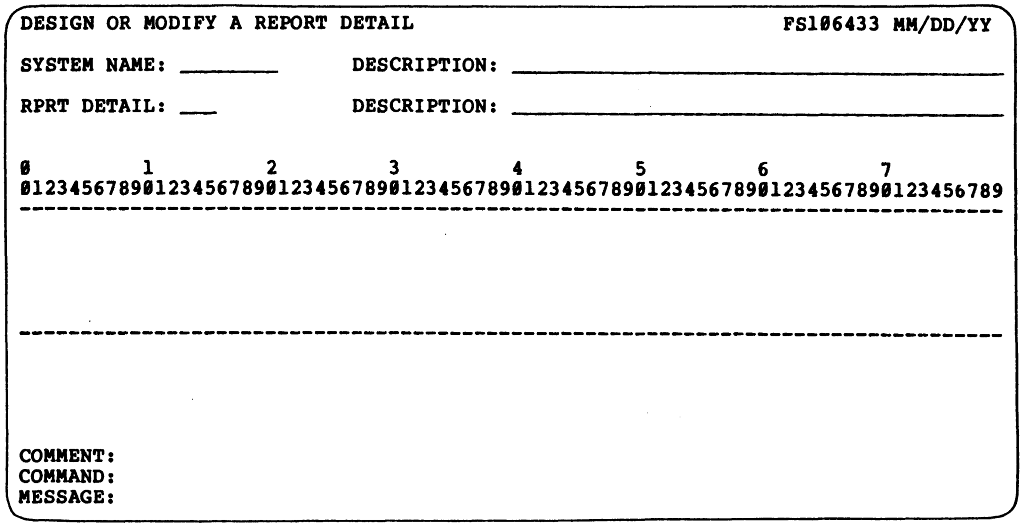
4.3.3 DESIGN OR MODIFY A REPORT DETAIL

PURPOSE: This function formats a detail for use in any report output by designating print positions for the report data. The print positions are associated with synonyms, and the data itself is then formatted within the detail.

Formatted details are assigned a number and stored in the Dictionary with their associated system. FORCE may then generate detail formatting source code by referencing the image of the detail in the Dictionary.

This function also may be used to modify a previously-designed detail.

SPECIAL: Design or Modify a Report Detail combines three operations. Each operation is documented in this section with a separate PROCEDURE and NOTES entry. The operations are presented in the order of their occurrence during execution of this function.



```
DESIGN OR MODIFY A REPORT DETAIL                                FS1Ø6433 MM/DD/YY

SYSTEM NAME: ________      DESCRIPTION: ____________________________________

RPRT DETAIL: ___           DESCRIPTION: ____________________________________

Ø         1         2         3         4         5         6         7
Ø123456789Ø123456789Ø123456789Ø123456789Ø123456789Ø123456789Ø123456789Ø123456789
--------------------------------------------------------------------------------




--------------------------------------------------------------------------------




COMMENT:
COMMAND:
MESSAGE:
```

PROCEDURE: Enter the name of the system associated with the detail format. Then enter a three-character, alphanumeric detail number and a description.

FORCE displays a detail formatting area. The program is in Screen Formatting Mode as indicated on the Comment Line. While in Screen Formatting Mode, use the terminal's cursor control keys to position the cursor.

The @ symbol is used to specify each location where data is to begin printing. All other entries are disregarded when the format is saved. The @ symbol is the only entry that can be stored in the detail formatting area.

Print locations may be specified anywhere within the five bordered vertical lines and the 132 denoted horizontal positions (see Table 4-2, View Command). The screen is refreshed accordingly when formatting extends beyond the defined area.

Press RETURN after input of each @ symbol. The first RETURN updates cursor position status and enters the print location into memory. A second RETURN transfers the cursor to the first position of the next line, then updates cursor position.

An ESCAPE transfers control to Command Input Mode, as indicated on the Comment Line. With the cursor positioned on the Command Line, the user may enter commands which facilitate the detail formatting process. The commands are listed below:

Command	Activity
A	Abort (used only in detail modification)
C	Center a line on the screen
D	Delete characters from within a line
E	Erase a line from the screen
G	Get a report header format
H	Help module
I	Insert characters within a line
M	Move one line to another line
S	Save the screen in the Dictionary
V	View detail format at an offset
X	Extract and refresh screen display
?	Command Line summary information

The command H presents a help module for detail formatting. Table 4-2 describes the formatting commands in detail.

Control is transferred between Screen Formatting Mode and Command Input Mode by pressing the ESCAPE key. A RETURN also transfers control from Command Input to Screen Formatting.

Enter the @ symbol to specify each location where printed data is to begin.

After the detail format has been designed, issue a Save command. FORCE then displays synonym specification entry fields.

To modify a previously-formatted detail, enter the system name and the detail number. The detail format description may be modified, or left intact by a RETURN at the Description entry field. FORCE then displays the current detail format on the screen.

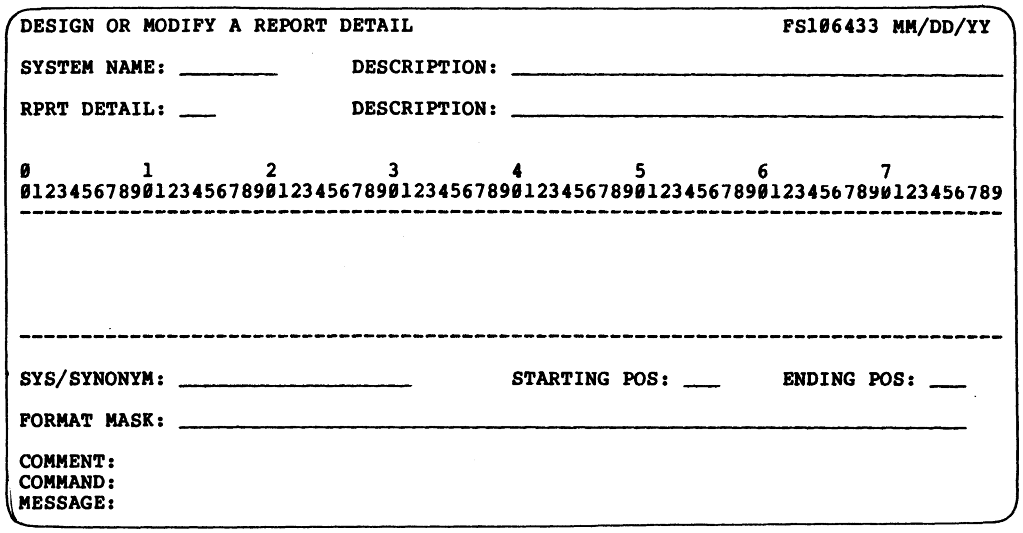
The detail format may be modified using any of the methods available when designing a format.

NOTES: The Save command is the only way to exit the detail design function. When modifying a format, the Save or the Abort command may be used to exit.

Any blank lines at the top of the formatting area are printed in the report output above the detail. If the detail is to be printed directly beneath a header, it should be entered on the top line of the formatting area. Blank lines at the bottom of the formatting area are not printed in the report output.

The Get command displays a selected header (dimmed) in the detail formatting area. Use the Get command (see Table 4-2) to facilitate alignment of a detail position beneath a header format.

Detail numbers within a system must be unique. The same number may be assigned to related title, header and statistics formats.



```
DESIGN OR MODIFY A REPORT DETAIL                               FS1Ø6433 MM/DD/YY

SYSTEM NAME: ________    DESCRIPTION: ______________________________________

RPRT DETAIL: ___         DESCRIPTION: ______________________________________

Ø         1         2         3         4         5         6         7
Ø123456789Ø123456789Ø123456789Ø123456789Ø123456789Ø123456789Ø123456789Ø123456789
--------------------------------------------------------------------------------



--------------------------------------------------------------------------------

SYS/SYNONYM: __________________     STARTING POS: ___     ENDING POS: ___

FORMAT MASK: _____________________________________________________________

COMMENT:
COMMAND:
MESSAGE:
```

PROCEDURE: FORCE prompts for entry of the related synonym data to be printed at each specified print location (@ symbol). Print locations are prompted by order of their occurrence in the format area; from left to right, top to bottom. FORCE substitutes a question mark (?) for the @ symbol with which a synonym is to be related.

At the System/Synonym entry field, specify the related synonym to be printed at the location denoted by question mark. If the synonym is associated with the same system as the report detail, enter the synonym name. If the synonym is associated with a different system, enter the system name and the synonym name using the format:

SYSTEM:SYNONYM

Starting and ending print positions are then displayed, based upon the length of the associated data element. For synonyms containing string data, this information may be left intact by a RETURN at the print position entry fields, or modified information may be entered. For synonyms related to numeric data, the print positions may not be modified.

Synonym specification may be aided by displaying selective lists of synonyms on the screen (see NOTES).

At the Format Mask entry field, the user may format the printed synonym data. The @ symbol designates each print position of the represented data. Any other displayable keyboard character (including blanks) may be placed before, within or following the printed data.

For example, consider a five-digit, alphanumeric synonym for a customer number. The synonym, defined by the user as CUSTNO, could use a mask like the following:

```
    CUSTOMER: (@) @@-@@
```

If the associated synonym, CUSTNO, contains the number 2JI80, the data is printed in this format:

```
    CUSTOMER: (2) JI-80
```

A RETURN at the Format Mask entry field establishes a mask of @ symbols for the length of the synonym data.

FORCE uses the | symbol to denote the end of a formatted mask. This symbol is not printed on the report output.

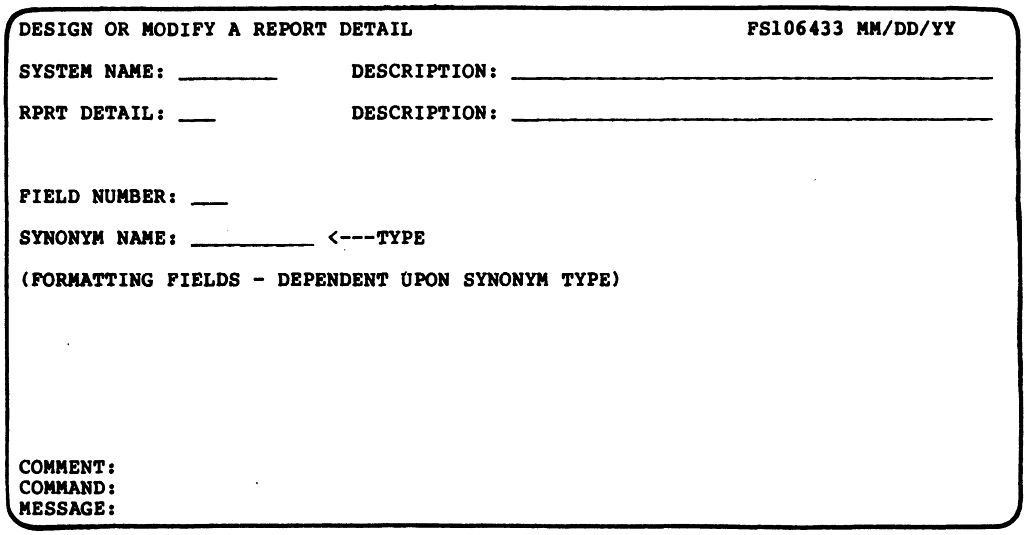
Format masking is repeated for each synonym to be printed in the report. When format masking is completed, FORCE clears the screen and displays prompts for field formatting.

When modifying a detail, the current synonym information is displayed after the format is reviewed. The associated synonym and its mask may be left intact by a RETURN at each entry field, or new information may be entered.

NOTES: To assist referencing of the synonyms that correspond to the detail format @ symbols, a selective list of available synonyms may be displayed on the screen. Entry of ? displays an alphabetic list of synonyms from the system associated with the detail format.

Use the format ?SYNONYM to begin the synonym list alphabetically with the specified synonym. An entry of ?SYSTEM: lists all synonyms within the specified system. Enter ?SYSTEM:SYNONYM to begin the list at a selected synonym within a specified system. Calculated synonyms are denoted by an asterisk.

The ESCAPE key may be used to exit the synonym query list.



```
DESIGN OR MODIFY A REPORT DETAIL                                  FS106433 MM/DD/YY

SYSTEM NAME: ________        DESCRIPTION: ____________________________________

RPRT DETAIL: ___             DESCRIPTION: ____________________________________


FIELD NUMBER: ___

SYNONYM NAME: __________ <---TYPE

(FORMATTING FIELDS - DEPENDENT UPON SYNONYM TYPE)






COMMENT:
COMMAND:
MESSAGE:
```

PROCEDURE: Each of the previously-specified detail fields may be additionally formatted. FORCE displays each field number, its associated system and synonym, and the data element type (Numeric and Alphanumeric).

Field formatting procedures differ slightly for alphanumeric and numeric fields.

For alphanumeric data, specify a Justify Mode of (L)eft, (R)ight or (N)one. A RETURN at the Justify Mode entry defaults to (N)one.

At the Filler Character entry field, specify any keyboard character to fill the length of the synonym when the current data occupies only a portion of the specified length. This field is bypassed if (N)one is input at the Justify Mode field. No filler is used if a RETURN is pressed at the Filler Character entry field.

Numeric data is always right justified when printed, so filler characters for numerics are printed to the left of the first digit.

The Float Character entry field allows specification of a character to be printed immediately before numeric synonym data, regardless of its length. Enter RETURN at the Float Character entry field if no float character is required.

At the Field Format entry field, specify (R)ounding or (T)runcating operations to be performed on numeric data before it is printed. A RETURN at the Field Format entry causes (N)one to be used.

If rounding or truncation is to be performed, a mask is displayed which represents the synonym length as established by the previously-defined format mask. The space bar is then used to move the cursor above the position at which the data is to be truncated or rounded. Enter X at the desired location. The Format Mask is then displayed with the specified value.

A RETURN at the Format Mask entry field rounds or truncates to the length of the synonym.

The field formatting process is repeated for each field in the report.

When modifying a detail format, the current field formatting information is diplayed after review of synonym information. New data may be entered at each entry field, or the current information may be left intact by pressing RETURN.

Enter a blank space to remove a single input from any field and establish the default value as the entry.

4.3.4 DESIGN OR MODIFY A STATISTICS FORMAT

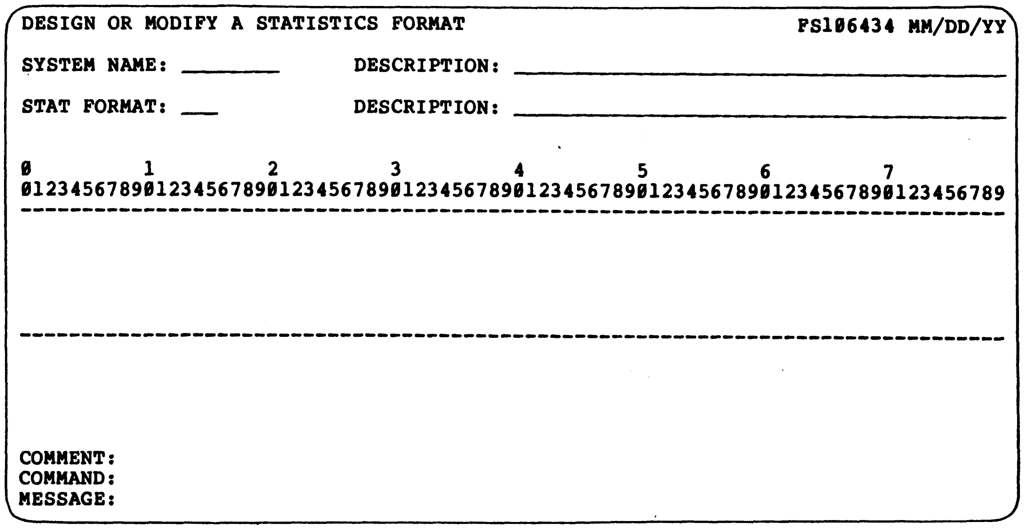
PURPOSE: This function designs a statistics format for use in any report output by designating print positions for computed statistics.

Print positions are associated with numeric synonyms, and formatted masks are established for the printed statistical data.

Statistics formats are assigned a number and stored in the Dictionary with their associated system. FORCE may then generate statistics formatting source code by referencing the image of the format in the Dictionary.

This function also may be used to modify a previously-designed statistics format.

SPECIAL: Design or Modify a Statistics Format combines three operations. Each operation is documented in this section with a separate PROCEDURE and NOTES entry. The operations are presented as they occur during program execution.



```
DESIGN OR MODIFY A STATISTICS FORMAT                           FS1Ø6434 MM/DD/YY

SYSTEM NAME: ________        DESCRIPTION: ______________________________________

STAT FORMAT: ___             DESCRIPTION: ______________________________________


Ø         1         2         3         4         5         6         7
Ø123456789Ø123456789Ø123456789Ø123456789Ø123456789Ø123456789Ø123456789Ø123456789
--------------------------------------------------------------------------------





--------------------------------------------------------------------------------





COMMENT:
COMMAND:
MESSAGE:
```

DESIGN/MODIFY
STAT. FORMAT
4.3.4

PROCEDURE: Enter the name of the system with which the statistics format is associated. Then enter a three-character, alphanumeric statistics format number and a description.

FORCE displays a statistics formatting area. The program is in Screen Formatting Mode as indicated on the Comment Line. While in Screen Formatting Mode, use the terminal's cursor control keys to position the cursor.

The @ symbol is used to specify each starting location where statistical data is to be printed. All other entries within the formatting area are disregarded when the statistics format is saved.

Synonyms and statistical functions are subsequently associated with each @ symbol.

Print locations may be specified anywhere within the five bordered vertical lines and the 132 denoted horizontal positions (see Table 4-2, View Command). The screen is refreshed accordingly when formatting extends beyond the defined area.

Press RETURN after entry of each @ symbol. The first RETURN updates cursor position status and enters the print location into memory. A second RETURN transfers the cursor to the first position of the next line, then updates cursor position.

An ESCAPE transfers control to Command Input Mode, as indicated on the Comment Line. With the cursor positioned on the Command Line, the user may enter commands which facilitate the statistics formatting process. The commands are listed below:

Command	Activity
A	Abort (used only in format modification)
C	Center a line on the screen
D	Delete characters from within a line
E	Erase a line from the screen
G	Get a report detail
H	Help module
I	Insert characters within a line
M	Move one line to another line
S	Save the screen in the Dictionary
V	View statistics format at an offset
X	Extract and refresh screen display
?	Command Line summary information

The command H presents a help module for statistics formatting. Table 4-2 describes the formatting commands in detail.

Control is transferred between Screen Formatting Mode and Command Input Mode by pressing the ESCAPE key. A RETURN also transfers control from Command Input to Screen Formatting.

Enter the @ symbol to specify each location where printed data is to begin.

After the statistics format has been designed, issue a Save command. FORCE then displays synonym specification entry fields.

To modify a previously-designed statistics format, enter the system name and the format number. The statistics description may be modified, or left intact by a RETURN at the Description entry field. FORCE then displays the current statistics format information on the screen.

The format may be modified using any of the methods available when designing a statistics format.

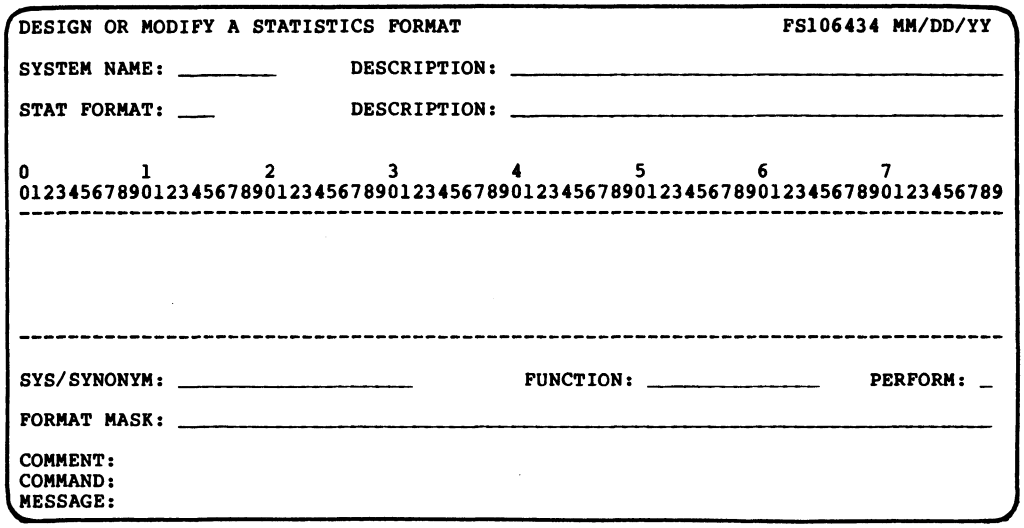
NOTES: The Save command is the only way to exit the statistics formatting function. When modifying a designed statistics format, the Save or the Abort command may be used to exit.

Any blank lines at the top of the formatting area are printed in the report output above the statistics format. If the format is to be printed directly beneath a detail format, it should be entered on the top line of the formatting area. Blank lines at the bottom of the formatting area are not printed in the report output.

The Get command displays a selected detail format (dimmed) in the statistics formatting area. Use the Get command (see Table 4-2) to facilitate alignment of a statistics position beneath a detail format.

Statistics format numbers within a system must be unique. The same number may be assigned to related title, header, detail and statistics formats.

Statistical functions may be performed on numeric data only.



```
DESIGN OR MODIFY A STATISTICS FORMAT                              FS106434 MM/DD/YY

SYSTEM NAME: ________     DESCRIPTION: ____________________________________________

STAT FORMAT: ___          DESCRIPTION: ____________________________________________

0         1         2         3         4         5         6         7
01234567890123456789012345678901234567890123456789012345678901234567890123456789
--------------------------------------------------------------------------------




--------------------------------------------------------------------------------

SYS/SYNONYM: ___________________         FUNCTION: _____________       PERFORM: _

FORMAT MASK: ______________________________________________________________

COMMENT:
COMMAND:
MESSAGE:
```

PROCEDURE: FORCE prompts for entry of the related synonym data to be printed at each specified print location (@ symbol). Print locations are prompted by order of their occurrence in the formatting area; from left to right, top to bottom. FORCE substitutes a question mark (?) for the @ symbol with which a synonym is to be related.

At the System/Synonym entry field, specify the synonym to be printed at the location denoted by a question mark. If the synonym is associated with the same system as the statistics format, enter the synonym name. If the synonym is associated with a different system, enter the system and the synonym name using the format:

```
SYSTEM:SYNONYM
```

Synonym assignment may be aided by displaying selective lists of synonyms on the screen (see NOTES).

At the Function entry field, specify the statistical operation to be performed on the synonym data. The available functions and sub-functions are: Average, Count, High Value, Low Value and Total.

The Average function computes the mean average of the synonym data. Count maintains the number of specified synonym values. High Value and Low Value functions record the highest and lowest values for the synonym data. Total prints the sum of the synonym data.

A standard function maintains a statistical value throughout the entire report. Specification of a sub-function clears the statistical value after printing.

To specify a sub-function, enter SUB- before the function type.

At the Perform field, specify whether the statistical function is to be performed (B)efore or (A)fter record selection. A RETURN defaults to an entry of (A)fter.

The Format Mask entry field allows formatting of printed statistical data. Use the @ symbol to designate each print position of the represented data. Any other displayable keyboard character may be included before, within or following the printed data.

For example, consider a statistical function which prints a four-digit number representing the total number of employees in a department. A mask like the following could be used:

```
NUMBER OF EMPLOYEES - @,@@@
```

If 4900 is the synonym value, the data is printed in this form:

```
NUMBER OF EMPLOYEES - 4,900
```

A RETURN at the Format Mask entry field establishes a mask of @ symbols for the length of the synonym data.

FORCE uses the | symbol to denote the end of a formatted mask. This symbol is not printed in the report output.

Format masking is performed for each synonym to be printed in the report. When masking is completed, FORCE clears the screen and displays prompts for field formatting.

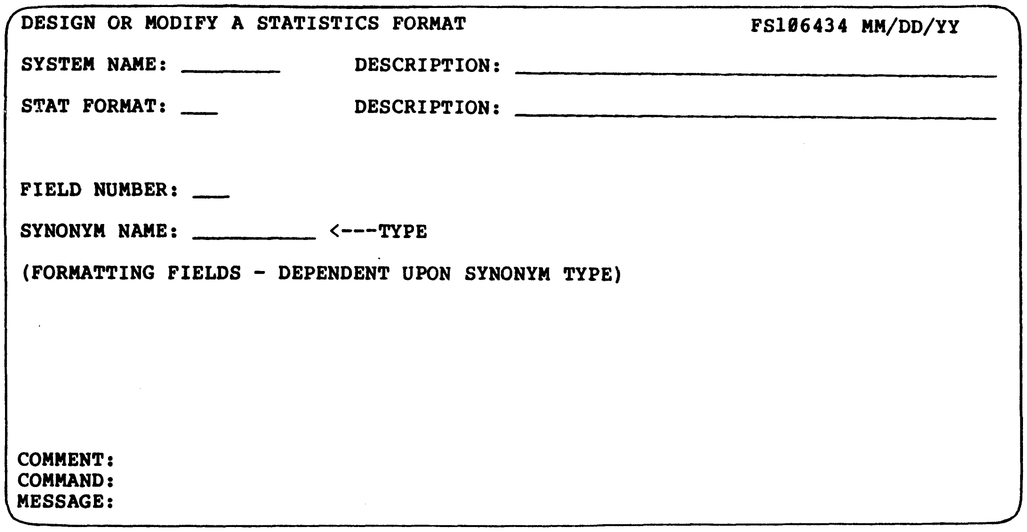
When modifying a previously-designed statistics format, current synonym information is displayed after the format is reviewed. New information may be entered, or the associated synonym and its mask may be left intact by a RETURN at each entry field.

__NOTES__: To assist in referencing synonyms that correspond to the @ symbols in the formatting area, a selective list of available synonyms may be displayed on the screen. Entry of ? displays an alphabetic list of synonyms from the system associated with the statistics format.

Use the format ?SYNONYM to begin the synonym list alphabetically with the specified synonym. An entry of ?SYSTEM: lists all synonyms within the specified system. Enter ?SYSTEM:SYNONYM to begin the list at a selected synonym within a specified system.

Calculated synonyms are denoted in the list by an asterisk.

The ESCAPE key may be used to exit the synonym query list.



```
DESIGN OR MODIFY A STATISTICS FORMAT                    FS106434 MM/DD/YY

SYSTEM NAME: ________      DESCRIPTION: ______________________________

STAT FORMAT: ___           DESCRIPTION: ______________________________


FIELD NUMBER: ___

SYNONYM NAME: __________ <---TYPE

(FORMATTING FIELDS - DEPENDENT UPON SYNONYM TYPE)






COMMENT:
COMMAND:
MESSAGE:
```

__PROCEDURE__: Additional formatting may be specified for each of the established statistical functions. FORCE displays each field number, its associated system and the function type.

At the Filler Character entry field, specify any keyboard character to fill the length of the synonym when the current data occupies only a portion of the specified length. No filler is used if a RETURN is pressed at the Filler Character entry field.

Numeric data is always right justified when printed, so filler characters for numerics are printed to the left of the first digit.

The Float Character entry field allows specification of a character to be printed immediately before synonym data, regardless of its length. For example, printed financial figures may use a floating $ sign. Enter RETURN at the Float Character entry field if no float character is required.

At the Field Format entry the user may specify (R)ounding or (T)runcating operations to be performed on numeric data before it is printed. A RETURN at the Field Format entry causes (N)one to be used.

If rounding or truncation is to be performed, a mask is displayed which represents the synonym length as established by the previously-defined mask. The space bar is then used to move the cursor above the position at which the data is to be truncated or rounded.

Enter X at the desired location. The Format Mask is then displayed with the specified value.

A RETURN at the Format Mask entry field rounds or truncates to the length of the synonym.

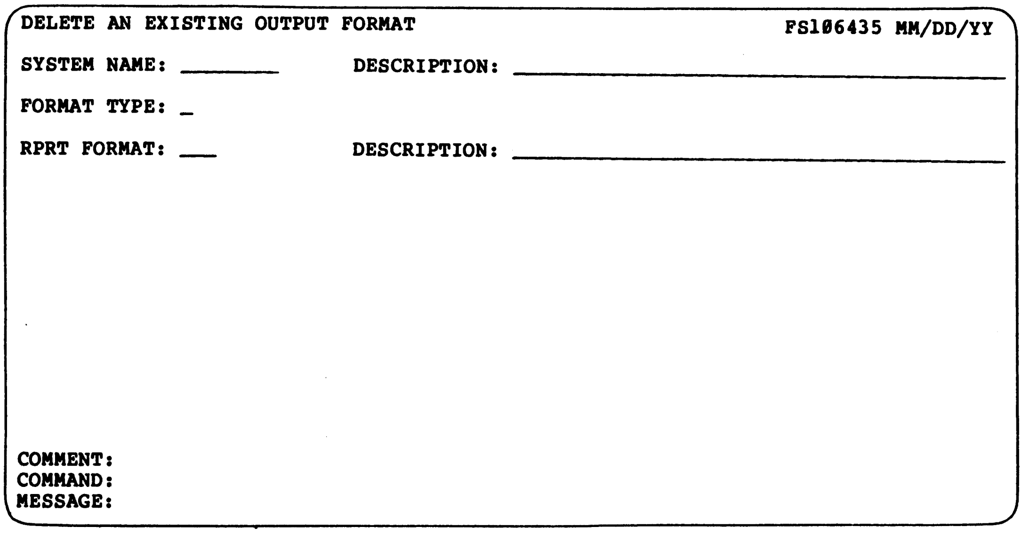
The field formatting process is repeated for each field in the report.

When modifying a statistics format, the current field formatting information is displayed after review of the synonym information. New data may be entered at each entry field, or the current information may be left intact by pressing RETURN.

Enter a blank space to remove a single input from any field and establish the default value as the entry.

4.3.5 DELETE AN EXISTING OUTPUT FORMAT

PURPOSE: This function allows deletion of a specified report format from the Dictionary.



```
DELETE AN EXISTING OUTPUT FORMAT                                  FS106435 MM/DD/YY

SYSTEM NAME: ________       DESCRIPTION: ______________________________________

FORMAT TYPE: _

RPRT FORMAT: ___            DESCRIPTION: ______________________________________










COMMENT:
COMMAND:
MESSAGE:
```

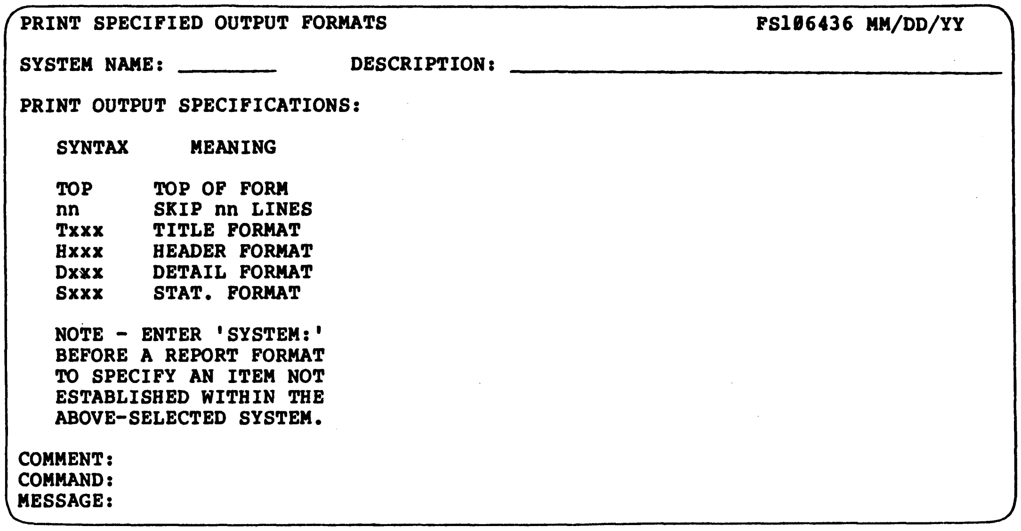
PROCEDURE: Enter the name of the system associated with the format to be deleted. Then enter the format type. Valid output format types are: (T)itle, (H)eader, (D)etail and (S)tatistics.

At the Report Format field, specify the number of the report format to be deleted.

NOTES: This function invalidates Dictionary reference to the affected output format.

4.3.6 PRINT SPECIFIED OUTPUT FORMATS

PURPOSE: This function prints a comprehensive set of information about specified report formats. Title, header, detail and statistics formats from any system may be documented in one report. The documentation, which may itself be formatted, reproduces the selected format(s) and lists print positioning information for each field within each format.



```
PRINT SPECIFIED OUTPUT FORMATS                              FS106436 MM/DD/YY

SYSTEM NAME: ________       DESCRIPTION: ____________________________________

PRINT OUTPUT SPECIFICATIONS:

   SYNTAX     MEANING

   TOP        TOP OF FORM
   nn         SKIP nn LINES
   Txxx       TITLE FORMAT
   Hxxx       HEADER FORMAT
   Dxxx       DETAIL FORMAT
   Sxxx       STAT. FORMAT

   NOTE - ENTER 'SYSTEM:'
   BEFORE A REPORT FORMAT
   TO SPECIFY AN ITEM NOT
   ESTABLISHED WITHIN THE
   ABOVE-SELECTED SYSTEM.

COMMENT:
COMMAND:
MESSAGE:
```

PROCEDURE: Enter the system name with which the output format is associated. If formats from different systems are to be printed, enter the name of the system with which the greatest number of formats are associated.

FORCE describes the available report functions in the SYNTAX and MEANING columns.

The TOP and nn (skip lines) commands format the report output. Use the TOP function to issue a top of form (new page) at the specified report location. If a title is assigned to the report, it will be printed. The Skip nn lines command prints the specified number of blank lines at the designated location.

TOP and nn entries are executed in the order in which they are entered.

Enter the report format(s) to be printed, using the information in the SYNTAX column. The first character of the entry identifies the format type, and the following three characters denote the format number.

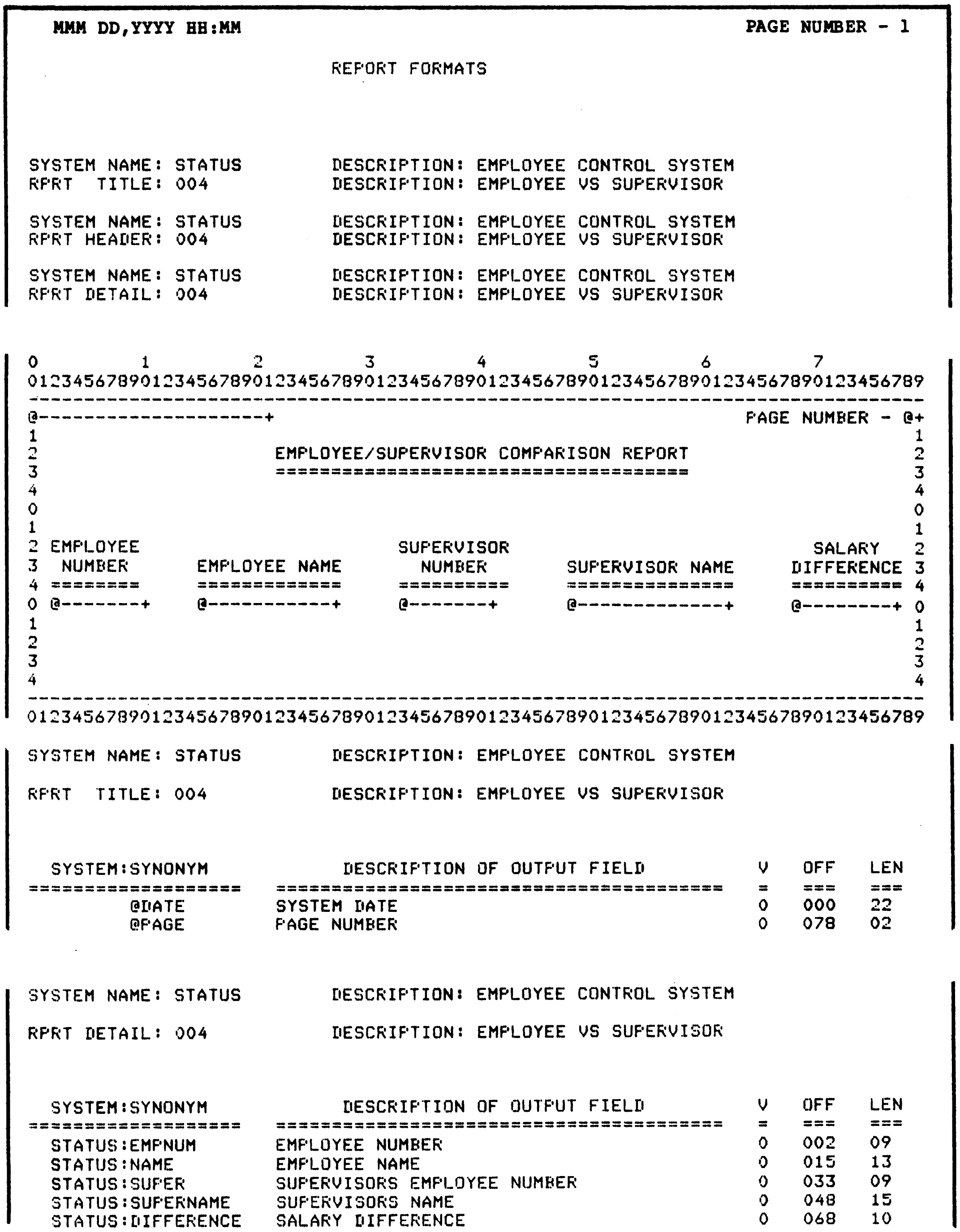
To specify a format from a different system, enter SYSTEM: immediately before the format type number.

Upon entry of a valid report format, FORCE displays the established format description.

FORCE scans the specified output(s) to determine the required paper width, then displays the appropriate forms mount message.

<u>NOTES</u>: Each report may contain any combination of title, header, detail and statistic formats.

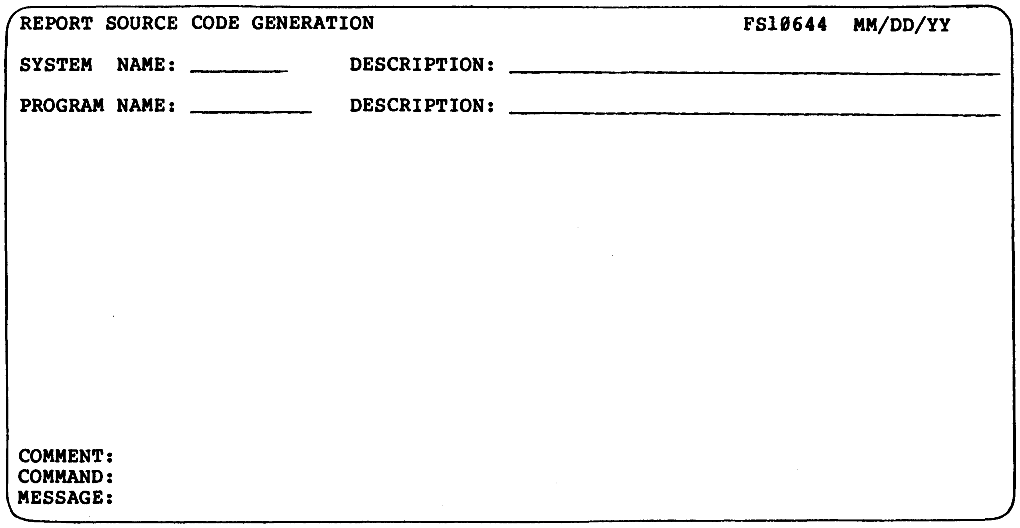
A printed report output is shown below.



```
  MMM DD,YYYY HH:MM                                               PAGE NUMBER - 1

                                 REPORT FORMATS



SYSTEM NAME: STATUS          DESCRIPTION: EMPLOYEE CONTROL SYSTEM
RPRT  TITLE: 004             DESCRIPTION: EMPLOYEE VS SUPERVISOR

SYSTEM NAME: STATUS          DESCRIPTION: EMPLOYEE CONTROL SYSTEM
RPRT HEADER: 004             DESCRIPTION: EMPLOYEE VS SUPERVISOR

SYSTEM NAME: STATUS          DESCRIPTION: EMPLOYEE CONTROL SYSTEM
RPRT DETAIL: 004             DESCRIPTION: EMPLOYEE VS SUPERVISOR


0         1         2         3         4         5         6         7
01234567890123456789012345678901234567890123456789012345678901234567890123456789
--------------------------------------------------------------------------------
@--------------------+                                          PAGE NUMBER - @+
1                                                                              1
2                     EMPLOYEE/SUPERVISOR COMPARISON REPORT                    2
3                     =====================================                    3
4                                                                              4
0                                                                              0
1                                                                              1
2 EMPLOYEE                       SUPERVISOR                           SALARY   2
3  NUMBER      EMPLOYEE NAME       NUMBER       SUPERVISOR NAME     DIFFERENCE 3
4 ========     =============     ==========     ===============     ========== 4
0 @-------+    @-----------+     @-------+      @-------------+     @--------+ 0
1                                                                              1
2                                                                              2
3                                                                              3
4                                                                              4
--------------------------------------------------------------------------------
01234567890123456789012345678901234567890123456789012345678901234567890123456789

SYSTEM NAME: STATUS          DESCRIPTION: EMPLOYEE CONTROL SYSTEM

RPRT  TITLE: 004             DESCRIPTION: EMPLOYEE VS SUPERVISOR


  SYSTEM:SYNONYM             DESCRIPTION OF OUTPUT FIELD          V   OFF   LEN
==================   ======================================       =   ===   ===
          @DATE      SYSTEM DATE                                  0   000   22
          @PAGE      PAGE NUMBER                                  0   078   02


SYSTEM NAME: STATUS          DESCRIPTION: EMPLOYEE CONTROL SYSTEM

RPRT DETAIL: 004             DESCRIPTION: EMPLOYEE VS SUPERVISOR


  SYSTEM:SYNONYM             DESCRIPTION OF OUTPUT FIELD          V   OFF   LEN
==================   ======================================       =   ===   ===
  STATUS:EMPNUM      EMPLOYEE NUMBER                              0   002   09
  STATUS:NAME        EMPLOYEE NAME                                0   015   13
  STATUS:SUPER       SUPERVISORS EMPLOYEE NUMBER                  0   033   09
  STATUS:SUPERNAME   SUPERVISORS NAME                             0   048   15
  STATUS:DIFFERENCE  SALARY DIFFERENCE                            0   068   10
```

4.4 REPORT GENERATION

PURPOSE: Report programs are generated by execution of this function. The Report Generation facility accesses established report information in the Dictionary and generates labeled program source code accordingly. This facility may also be used to re-generate a report program that has had its Dictionary information modified.



```
REPORT SOURCE CODE GENERATION                              FS10644  MM/DD/YY

SYSTEM  NAME: ________     DESCRIPTION: ________________________________

PROGRAM NAME: _________    DESCRIPTION: ________________________________










COMMENT:
COMMAND:
MESSAGE:
```

PROCEDURE: Prior to execution of this function, the report program information must be developed within the Dictionary. Generation of a report program requires specification of at least one file I/O (see 4.1) and one print specification (see 4.2.02 and 4.2.4).

Enter the name of the system with which the program to be generated is associated. Then enter the name of the program.

NOTES: All FORCE-generated program text files use the same naming conventions. The file has the same name as the program, with an L. prefix to denote that it is a labeled text file. For example, a program defined to the Dictionary as SWI57 is generated on disk as L.SWI57 in a program text file.

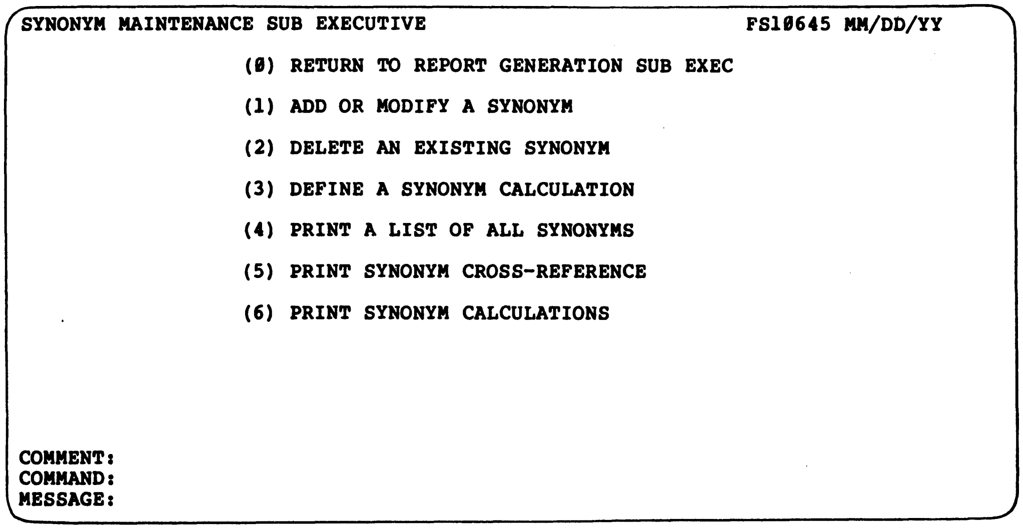
Subsequently, the user may process the L. file through the Linkage Editor (see 2 .1), which expands the file into an IRIS Business BASIC source text file. The file expanded by the Linkage Editor is written on disk with a T. prefix. For example, the program text file L.SWI57 is expanded by the Linkage Editor as T.SWI57 in a program text file.

A re-generated L. program overwrites the previous L. program text file on disk.

REPORT GENERATION 4.4

4.5 SYNONYM MAINTENANCE

PURPOSE: Synonyms are used throughout the Report Preparation process to reference data elements within the Dictionary. Synonyms allow the user to address data elements by descriptive words, often in the precise terminology of the application. This menu provides functions to establish and maintain synonyms for use in report generation.



```
SYNONYM MAINTENANCE SUB EXECUTIVE                              FS10645 MM/DD/YY

                        (0) RETURN TO REPORT GENERATION SUB EXEC

                        (1) ADD OR MODIFY A SYNONYM

                        (2) DELETE AN EXISTING SYNONYM

                        (3) DEFINE A SYNONYM CALCULATION

                        (4) PRINT A LIST OF ALL SYNONYMS

                        (5) PRINT SYNONYM CROSS-REFERENCE

                        (6) PRINT SYNONYM CALCULATIONS




COMMENT:
COMMAND:
MESSAGE:
```

PROCEDURE: Enter the number which corresponds to the desired function.

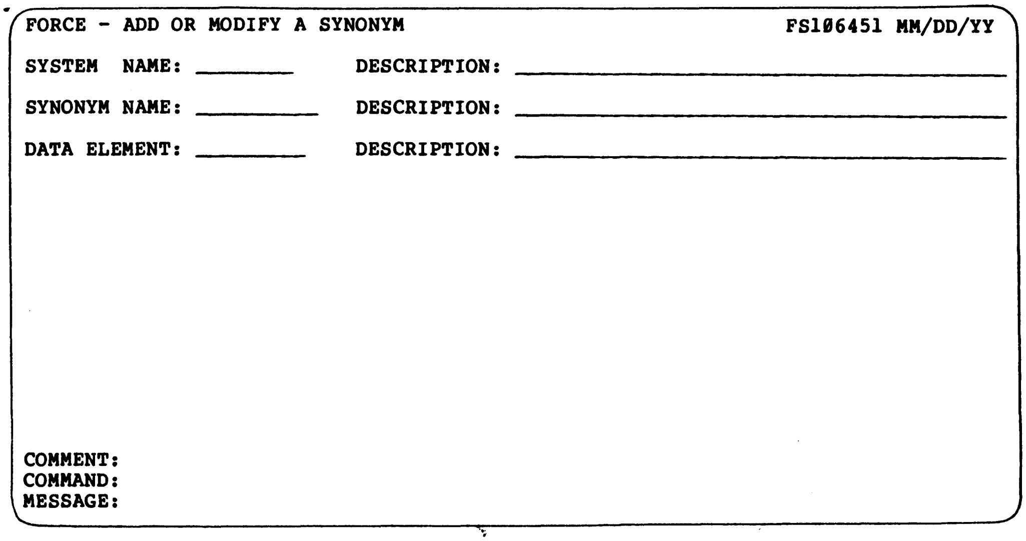
NOTES: Synonyms also may be assigned under Data Element Definition (see 1.3.1).

Every synonym must be associated with a data element that has been defined to the Dictionary.

4.5.1 ADD OR MODIFY A SYNONYM

PURPOSE: Synonyms are used in report program specifications to reference data elements. This function enables the user to establish a synonym by associating it with a data element in the Dictionary.

Synonyms may also be modified with this function by altering the description or the associated data element.



```
FORCE - ADD OR MODIFY A SYNONYM                          FS1Ø6451 MM/DD/YY

SYSTEM  NAME: _______     DESCRIPTION: ________________________________

SYNONYM NAME: _________   DESCRIPTION: ________________________________

DATA ELEMENT: ________    DESCRIPTION: ________________________________









COMMENT:
COMMAND:
MESSAGE:
```

PROCEDURE: Enter the name of the system with which the synonym is to be associated. Then enter the name of the synonym.

At the Data Element entry field, specify the name of the data element to be associated with the synonym. The associated data element may be numeric, a string, a string subdivision or a matrix cell.

Upon entry of the associated data element, the synonym description may be entered. A RETURN at the synonym description entry field duplicates the description of the associated data element.

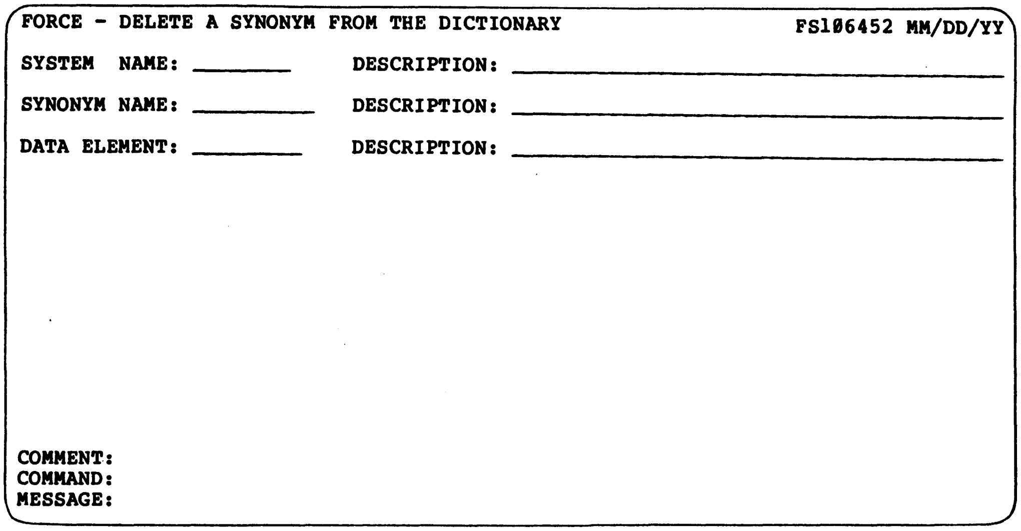
To modify a synonym, enter the name of the associated system and the synonym name. The current synonym information is displayed. The associated data element or the synonym description may be changed. New information may be entered, or the current information may be left intact by a RETURN at the entry field.

NOTES: Synonym names may not exceed ten characters and may be used only once within a system. However, multiple synonyms may be assigned to the same system data element.

Synonyms may be assigned to string subdivisions or matrix cells that have been linked (see 1.3.4 and 1.3.5).

4.5.2 DELETE AN EXISTING SYNONYM

PURPOSE: This function deletes a system synonym from the Dictionary.



```
FORCE - DELETE A SYNONYM FROM THE DICTIONARY              FS1Ø6452 MM/DD/YY
SYSTEM  NAME: ________     DESCRIPTION: ____________________________________
SYNONYM NAME: __________   DESCRIPTION: ____________________________________
DATA ELEMENT: _________    DESCRIPTION: ____________________________________













COMMENT:
COMMAND:
MESSAGE:
```

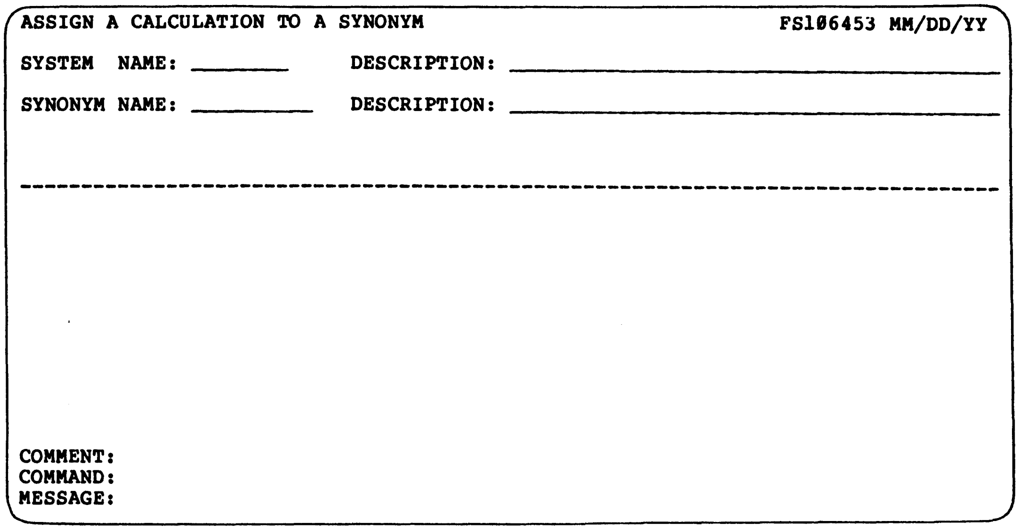
PROCEDURE: Enter the name of the system with which the synonym is associated. Then enter the synonym name.

NOTES: This function has no effect on the data element associated with the deleted synonym, or any other synonym that is related to the data element.

4.5.3 DEFINE A SYNONYM CALCULATION

PURPOSE: System synonyms may be defined as the result of mathematical operations performed with other synonyms and/or numeric values. This function enables the user to construct a calculation and relate its value to a synonym.

Synonym calculations may also be modified with this function.



```
ASSIGN A CALCULATION TO A SYNONYM                           FS1Ø6453 MM/DD/YY

SYSTEM  NAME: _______      DESCRIPTION: __________________________________

SYNONYM NAME: _________    DESCRIPTION: __________________________________


-----------------------------------------------------------------------------









COMMENT:
COMMAND:
MESSAGE:
```

PROCEDURE: Prior to executing this function, the affected synonym must be established within the Dictionary (see 4.5.1).

Enter the name of the system with which the synonym is associated. Then enter the synonym for which calculations are to be specified.

FORCE displays a line on which the synonym calculation is to be specified. The calculated value of the synonym is determined by the expression following the SYNONYM= prompt. In specifying the calculations, the user may enter:

- valid synonym names
- numeric values
- symbols to add, subtract, multiply or divide (+ - * /)
- grouping symbols (brackets or parentheses)

Lateral cursor tracking is enabled along the calculation line. The terminal's directional keys may be used to move in either direction on the line.

Enter the calculation which yields the value to be stored in the synonym. Mathematical operations are performed as entered on the calculation line, from left to right, unless otherwise indicated by grouping symbols.

Throughout this process, invalid entries on the calculation line are denoted by a pointer and the appropriate error message is displayed.

A listing of available synonyms may be obtained at any time by entering a question mark (?). FORCE automatically places a question mark on the line after input of invalid data. A RETURN then invokes a list of all available synonyms.

(See NOTES for more information on synonym listings.)

Upon entry of a valid calculation, the user may specify format operations to be performed on each synonym value used in the calculation.

Any one of the following operations may be performed on the each synonym value within the calculation: (R)ounded, (T)runcated or (N)one. A RETURN at the Field Format entry causes (N)one to be used.

If rounding or truncation is to be performed, a mask is displayed which represents the synonym length both before and after a decimal point. An arrow over the calculation line points to the synonym for which field formatting information is requested.

The space bar is then used to position the cursor above the position at which the synonym is to be truncated or rounded. Enter X at the desired location. The Format Mask is then displayed with the specified value.

A RETURN at the Format Mask entry field rounds or truncates to an integer.

Repeat this process for each synonym used in the calculation.

To modify a synonym calculation, enter the associated system name and the synonym name. The current information is displayed. Each entry may be modified by entering new information, or left intact by a RETURN at the entry field.

NOTES: When constructing a synonym calculation, a selective list of available synonyms may be displayed on the screen. Entry of ? on the calculation line displays an alphabetic list of system synonyms and their descriptions.

The format ?SYNONYM begins the system synonym list alphabetically with the specified synonym. An entry of ?SYSTEM: lists all synonyms within the specified system. Enter ?SYSTEM:SYNONYM to begin the list at a selected synonym within a specified system. Calculated synonyms are denoted by an asterisk.

A synonym calculation equation may include other calculated synonyms (nested calculations). A synonym may not be used in the calculation of itself (circular definition).

Only synonyms related to numeric data elements may be used in synonym calculations.

4.5.4 PRINT A LIST OF ALL SYNONYMS

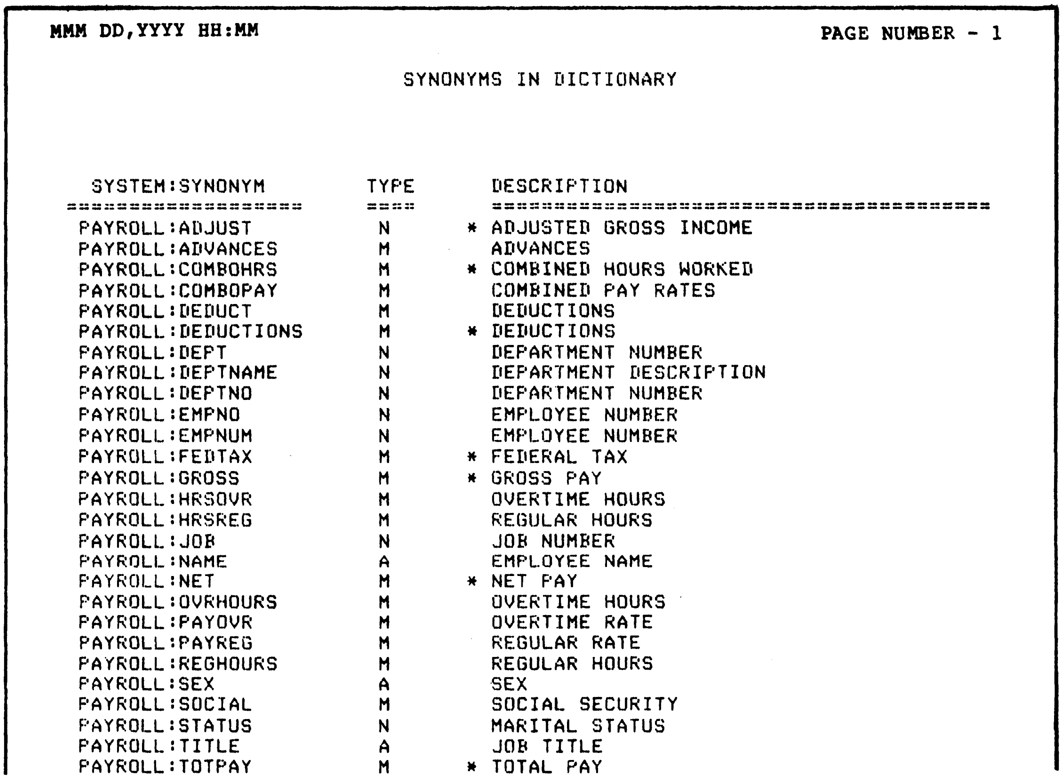
PURPOSE: This function prints a list of all synonyms within a system, or all synonyms in the Dictionary. The list is arranged alphabetically within the associated system, and shows the synonym, description and associated data element type. Calculated synonyms are denoted by an asterisk.

PROCEDURE: Specify the system name for which synonyms are to be printed.

Enter RETURN to print a list of all synonyms in the Dictionary, beginning alphabetically with the first associated system name.

NOTES: Synonym types are: (A)lphanumeric, (N)umeric, and (M)atrix.

Below is a sample listing of synonyms.



```
MMM DD,YYYY HH:MM                                          PAGE NUMBER - 1

                          SYNONYMS IN DICTIONARY



   SYSTEM:SYNONYM        TYPE       DESCRIPTION
  ===================    ====       ========================================
   PAYROLL:ADJUST         N      *  ADJUSTED GROSS INCOME
   PAYROLL:ADVANCES       M         ADVANCES
   PAYROLL:COMBOHRS       M      *  COMBINED HOURS WORKED
   PAYROLL:COMBOPAY       M         COMBINED PAY RATES
   PAYROLL:DEDUCT         M         DEDUCTIONS
   PAYROLL:DEDUCTIONS     M      *  DEDUCTIONS
   PAYROLL:DEPT           N         DEPARTMENT NUMBER
   PAYROLL:DEPTNAME       N         DEPARTMENT DESCRIPTION
   PAYROLL:DEPTNO         N         DEPARTMENT NUMBER
   PAYROLL:EMPNO          N         EMPLOYEE NUMBER
   PAYROLL:EMPNUM         N         EMPLOYEE NUMBER
   PAYROLL:FEDTAX         M      *  FEDERAL TAX
   PAYROLL:GROSS          M      *  GROSS PAY
   PAYROLL:HRSOVR         M         OVERTIME HOURS
   PAYROLL:HRSREG         M         REGULAR HOURS
   PAYROLL:JOB            N         JOB NUMBER
   PAYROLL:NAME           A         EMPLOYEE NAME
   PAYROLL:NET            M      *  NET PAY
   PAYROLL:OVRHOURS       M         OVERTIME HOURS
   PAYROLL:PAYOVR         M         OVERTIME RATE
   PAYROLL:PAYREG         M         REGULAR RATE
   PAYROLL:REGHOURS       M         REGULAR HOURS
   PAYROLL:SEX            A         SEX
   PAYROLL:SOCIAL         M         SOCIAL SECURITY
   PAYROLL:STATUS         N         MARITAL STATUS
   PAYROLL:TITLE          A         JOB TITLE
   PAYROLL:TOTPAY         M      *  TOTAL PAY
```

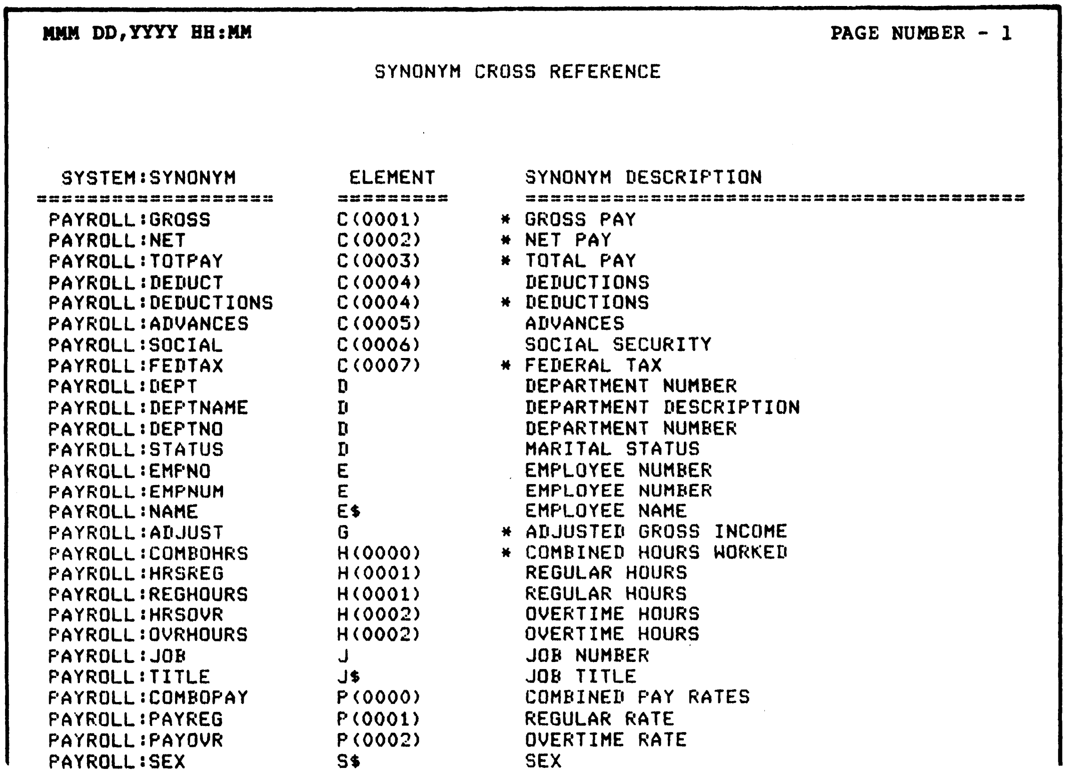
4.5.5 PRINT SYNONYM CROSS-REFERENCE

PURPOSE: This function prints a cross-reference of synonyms and data elements within a system, or within all systems. The list is arranged in ascending order by system data elements and shows the synonym, its description and associated data element. Calculated synonyms are denoted by an asterisk.

PROCEDURE: Specify the system name for which the cross-reference is to be printed.

Enter RETURN to print a cross-reference for all synonyms in the Dictionary, beginning alphabetically with the first associated system name.

NOTES: Below is a sample synonym cross-reference report.



```
MMM DD,YYYY HH:MM                                              PAGE NUMBER - 1
                              SYNONYM CROSS REFERENCE



 SYSTEM:SYNONYM          ELEMENT       SYNONYM DESCRIPTION
====================    =========     ========================================
PAYROLL:GROSS           C(0001)     * GROSS PAY
PAYROLL:NET             C(0002)     * NET PAY
PAYROLL:TOTPAY          C(0003)     * TOTAL PAY
PAYROLL:DEDUCT          C(0004)       DEDUCTIONS
PAYROLL:DEDUCTIONS      C(0004)     * DEDUCTIONS
PAYROLL:ADVANCES        C(0005)       ADVANCES
PAYROLL:SOCIAL          C(0006)       SOCIAL SECURITY
PAYROLL:FEDTAX          C(0007)     * FEDERAL TAX
PAYROLL:DEPT            D             DEPARTMENT NUMBER
PAYROLL:DEPTNAME        D             DEPARTMENT DESCRIPTION
PAYROLL:DEPTNO          D             DEPARTMENT NUMBER
PAYROLL:STATUS          D             MARITAL STATUS
PAYROLL:EMPNO           E             EMPLOYEE NUMBER
PAYROLL:EMPNUM          E             EMPLOYEE NUMBER
PAYROLL:NAME            E$            EMPLOYEE NAME
PAYROLL:ADJUST          G           * ADJUSTED GROSS INCOME
PAYROLL:COMBOHRS        H(0000)     * COMBINED HOURS WORKED
PAYROLL:HRSREG          H(0001)       REGULAR HOURS
PAYROLL:REGHOURS        H(0001)       REGULAR HOURS
PAYROLL:HRSOVR          H(0002)       OVERTIME HOURS
PAYROLL:OVRHOURS        H(0002)       OVERTIME HOURS
PAYROLL:JOB             J             JOB NUMBER
PAYROLL:TITLE           J$            JOB TITLE
PAYROLL:COMBOPAY        P(0000)       COMBINED PAY RATES
PAYROLL:PAYREG          P(0001)       REGULAR RATE
PAYROLL:PAYOVR          P(0002)       OVERTIME RATE
PAYROLL:SEX             S$            SEX
```

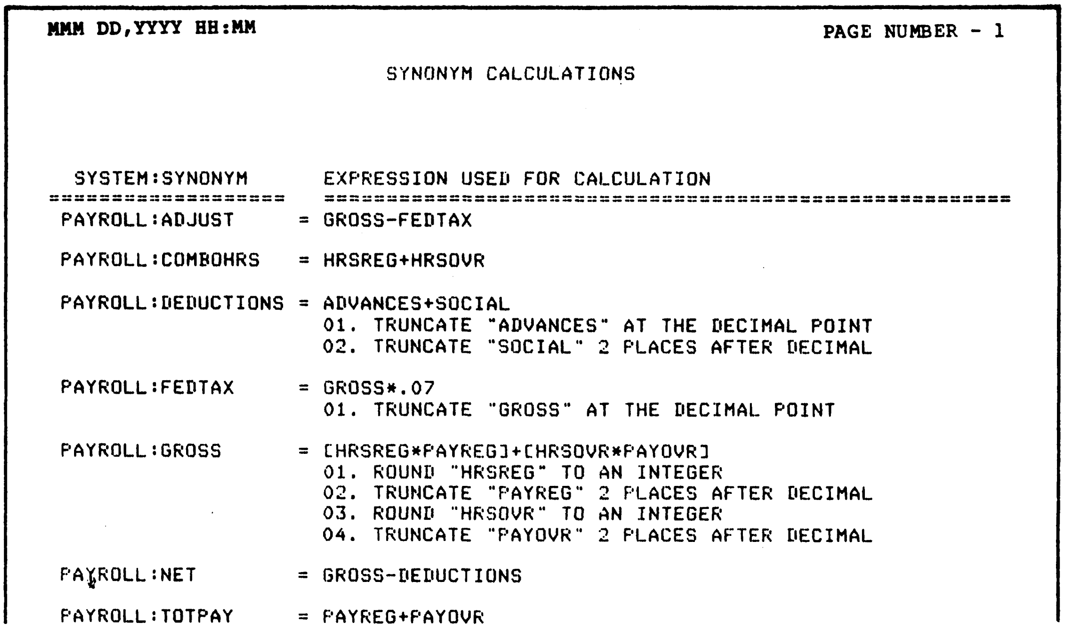
4.5.6 PRINT SYNONYM CALCULATIONS

PURPOSE: This function prints information on calculated synonyms for a specified system or for all systems. The list is arranged alphabetically by system synonyms. Synonym calculations are described, and field formatting specifications are defined for each synonym used in the calculation.

PROCEDURE: Specify the system name for which the calculated synonym information is to be printed.

Enter RETURN to print calculated synonym information for all synonyms in the Dictionary, beginning alphabetically with the first associated system name.

NOTES: Below is a sample calculated synonyms report.



```
MMM DD,YYYY HH:MM                                              PAGE NUMBER - 1

                              SYNONYM CALCULATIONS



  SYSTEM:SYNONYM       EXPRESSION USED FOR CALCULATION
===================    ==========================================================
 PAYROLL:ADJUST      = GROSS-FEDTAX

 PAYROLL:COMBOHRS    = HRSREG+HRSOVR

 PAYROLL:DEDUCTIONS  = ADVANCES+SOCIAL
                       01. TRUNCATE "ADVANCES" AT THE DECIMAL POINT
                       02. TRUNCATE "SOCIAL" 2 PLACES AFTER DECIMAL

 PAYROLL:FEDTAX      = GROSS*.07
                       01. TRUNCATE "GROSS" AT THE DECIMAL POINT

 PAYROLL:GROSS       = [HRSREG*PAYREG]+[HRSOVR*PAYOVR]
                       01. ROUND "HRSREG" TO AN INTEGER
                       02. TRUNCATE "PAYREG" 2 PLACES AFTER DECIMAL
                       03. ROUND "HRSOVR" TO AN INTEGER
                       04. TRUNCATE "PAYOVR" 2 PLACES AFTER DECIMAL

 PAYROLL:NET         = GROSS-DEDUCTIONS

 PAYROLL:TOTPAY      = PAYREG+PAYOVR
```

Section 5
MACRO COMMANDS

FORCE macro commands are statements which direct the Linkage Editor to generate source code. They are embedded within a program text file, then interpreted as the program is processed by the Linkage Editor. The Linkage Editor expands the source code within the program text file at the location of the inserted macro, then renumbers the program accordingly.

Macro commands are also used in FORCE-generated source code. They may subsequently by modified using the same procedures as with manually inserted macros.

Each macro type generates source code that performs a specific function. At generation time, the expanded source code is adapted to the application by accessing the data Dictionary, and by implementing functional parameters specified within the macro command line.

Macro expansion expedites the task of writing a diverse range of well-documented, consistent, maintainable source code. Macros may be inserted within any Business BASIC or labeled program text file.

FORCE macro commands enable customized application development by providing the user with a variety of adaptable, error-free source code segments. The User Macro is particularly powerful in that it allows the insertion of any number of user-written routines with variable parameter values (see 5.2.4).

Macro commands streamline program development and modification because the user can generate many lines of error-free, efficient code by writing a single macro command. Macro usage facilitates system maintenance in that Dictionary modifications are reflected in each macro expansion. Program maintenance is further enhanced because of the extensive documentation (commented source code) that is generated with each macro.

FORCE macros are categorized and documented in three groups: Levels 1-4 Macros; Level 5 Macros; and Level 6 Macros.

For further information on macros see:

5.1 - Macro Implementation
5.2 - Levels 1-4 Macros
5.3 - Level 5 Macros
5.4 - Level 6 Macros

Table 5-1 shows the FORCE macro types, their functions, and the FORCE Level at which they are available.

TABLE 5-1. MACRO COMMANDS

Type	Function	Level
DISPLAY	Screen Display Format	1
DATA	Dimension Selected Elements	2
INPUT	Input and Edit Routine	2
USER	User-Defined Alternate Files	2
VARIABLES	Dimension all File Elements	3
READ	Read from a File	3
WRITE	Write to a File	3
DEFINE	Establish Global Variables by Program Type	4
FIND	Locate Matching Key	5
INSERT	Place Key in Directory	5
DELETE	Remove Key from Directory	5
GET	Perform FIND, Read Data Record	5
PUT	Write Data Record, Perform INSERT	5
UPDATE	Perform FIND, Write Data Record	5
LINK	Extract Available Location, Perform PUT	5
FREE	Perform DELETE, Place Location on Free List	5
PROGRAM	Dimension/Assign Variables (Report or Entry Programs)	6
HEADER	Generate a Report Header Format	6
DETAIL	Generate a Report Detail Format	6

5.1 MACRO IMPLEMENTATION

Macros are implemented by inserting the required macro command at the appropriate location within the program text file. The text file is then processed by the Linkage Editor, which expands the macro into source code based upon macro command line specifications and associated Dictionary information. The program text file is renumbered accordingly by the Linkage Editor.

A program may contain any number of macros. Macros may be inserted within labeled source code exactly as within Business BASIC source code, except that they do not require a line number.

The procedures for macro insertion are outlined as follows:

1. Select the macro type that performs the required function.

2. Determine the location within the program source code where the macro is to be inserted.

3. Edit the program text file into which the macro is to be inserted. Insert the macro command line(s) at the appropriate location within the program.

4. Process the program through the Linkage Editor (see 2.1). The Linkage Editor expands the macro command line(s) into the corresponding source code, and overwrites the original program text file.

5. Load and save the expanded program text file. The program may then be executed.

5.1.1 MACRO EXPANSION

Macro expansion is the process by which functional source code is generated from macro commands. Expansion is performed by the Linkage Editor, which processes macros inserted within IRIS Business BASIC or labeled source code (see 2.1).

During expansion, when the Linkage Editor encounters a REMACRO statement, it interprets the command line contents and generates the corresponding source code. When the entire text file has been parsed, the Linkage Editor renumbers the source code. Except for renumbering, macros are the only portion of the source code affected by the expansion process.

Every macro in a text file is re-expanded when the program is processed by the Linkage Editor. When a text file is expanded through the Linkage Editor, previously-expanded macro source code is deleted (between the REMACRO and the REMSTOP lines). Each macro is then re-expanded to accommodate any changes that may have been made to the macro or the Dictionary information it references.

After a macro command line has been processed by the Linkage Editor, it functions as a reference point for the corresponding expanded source code. In a BASIC text file, a macro command line is loaded as a non-executable REM line (REM + MACRO). Macro lines may be subsequently overstored at execution time (see 2.4).

The system date is printed to the right of a REMACRO line each time it is expanded. That date and the hyphen that precedes it should never be edited.

5.1.2 PROGRAM MODIFICATION WITH MACROS

A program that contains macros may be modified by altering the referenced Dictionary information or changing a macro command line, then re-expanding the text file through the Linkage Editor.

The Linkage Editor expands macros into source code based upon current Dictionary information (see 5.1.1). Therefore, if referenced Dictionary information is changed, a re-expansion of the source code incorporates those changes. Similarly, a program may be modified by altering a macro command line so that it references different Dictionary information, and then re-expanding the text file.

For example, consider a program that contains the following Display Macro:

```
    REMACRO;DISPLAY,LEDGER,121
```

This macro directs the Linkage Editor to generate the source code which creates screen display number 121 from the LEDGER system. The image of this screen display is referenced within the Dictionary, and the corresponding source code is generated in the form of PRINT statements.

To change the screen display source code in the program text file, the user may modify the screen format in the Dictionary (see 1.2.2) and re-expand the program. The screen display source code will then produce the screen as it currently appears in the Dictionary.

If an entirely different screen is to be used in the program, the user may change the macro command line to reference another screen. For example, the macro command line may be edited to read:

```
    REMACRO;DISPLAY,ACCOUNTS,300
```

When the program text file is processed through the Linkage Editor, the macro is re-expanded into source code, which creates screen display 300 from the ACCOUNTS system.

A hyphen and the system date are printed to the right of an expanded macro command line. When modifying a macro command line, these items should never be altered. They will be changed by the Linkage Editor when the macro is re-expanded.

5.2 LEVELS 1-4 MACRO COMMANDS

The Levels 1-4 macro commands are listed below, with the corresponding manual section:

Type	Function	Section
DISPLAY	Screen Display Format	5.2.1
DATA	Dimension Selected Elements	5.2.2
INPUT	Input and Edit Routine	5.2.3
USER	User-Defined Alternate Files	5.2.4
VARIABLES	Dimension all File Elements	5.2.5
READ	Read from a File	5.2.6
WRITE	Write to a File	5.2.7
DEFINE	Establish Global Variables	5.2.8

Levels 1-4 macros are single-line commands which use the same format up to and including the system name. The standard Levels 1-4 macro command line format is:

```
nnnn REMACRO;TYPE,SYSTEM,CONTROLS(,PARAMETERS)
  │     │      │     │       │         └Optional parameters
  │     │      │     │       │
  │     │      │     │       └Required control items, which
  │     │      │     │        vary among macro types
  │     │      │     │
  │     │      │     └Associated system name
  │     │      │
  │     │      └Macro type specification
  │     │
  │     └Macro notation (REM + MACRO), followed by a semicolon
  │
  └BASIC line number up to four digits.  Not required for macros
   inserted within labeled source code (see 8.1).
```

Macro command line entries must adhere to this order and format. A semicolon must follow the REMACRO notation, and subsequent command line entries must be separated by commas.

See Table 5-2 for Levels 1-4 macro command set-up; see Table 5-3 for Levels 1-4 parameters.

See Table 5-4 for an explanation of macro command line entries.

TABLE 5-2. MACRO COMMAND SET-UP - LEVELS 1-4

Command Line Entry*	Macro Type							
	DISPLAY	DATA	INPUT	USER	VARIABLES	READ	WRITE	DEFINE
Line Number**	R	R	R	R	R	R	R	R
REMACRO; Notation	R	R	R	R	R	R	R	R
Macro Type	R	R	R	R	R	R	R	R
System Name	R	R	R	R	R	R	R	O
Screen Number	R							
Data File Name					R	R	R	
Program Name								O
Program Text File				R				
Data Element		R	R					

R = Required command line components
O = Optional parameters

*Levels 1-4 macro command line components must be entered in the order presented above, from top to bottom.

**A BASIC line number is not required for macros inserted within labeled source code (see 8.1).

TABLE 5-3. MACRO COMMAND CONTROLS/PARAMETERS - LEVELS 1-4

Controls/ Parameters	Macro Type							
	DISPLAY	DATA	INPUT	USER	VARIABLES	READ	WRITE	DEFINE
AM= Access Method						R	R	
VAR= Access Variable						R	R	
#= Channel Number						R	R	
H:V Cursor Position			R					
ORG= File Organization						R	R	
G=nnnn Set GOSUB Line nnnn	O			O		O	O	
L=nnnn Set Line nnnn	O	O	O	O	O	O	O	O
CS=N No Clear Screen	O							
SUB=N Do Not Use Subscription						O	O	
LOCK=N Do Not Lock Record						O	O	

R = Required command line components
O = Optional parameters

TABLE 5-4. MACRO COMMAND LINE ENTRIES (ALL MACROS)

Entry	Definition
System Name	The name of an established FORCE system, with which the command line entries are associated.
Data File Name	An established FORCE data file, belonging to the specified system. This file's record layout is referenced from the Dictionary for reads and writes.
Index File Name	The name of an established FORCE index file that contains the required directory.
Key Construct File	An established FORCE file whose record layout is used to construct the key used for accessing an index.
Program Name	The name of a program as it is defined to FORCE.
Program Text File	The name of the program as recorded on disk.
Data Element	The name of the data element(s) to be used in the expanded source code.
AM=	Access Method - establishes the mode of file access. Valid entries are RAN,DIR,RND for a Random access, and SEQ for a Sequential access (e.g., AM=SEQ, AM=RAN).
VAR=	Access Variable - Identifies a variable, expression or number to point to a record number in a READ, WRITE or SEARCH statement (e.g., VAR=X9, VAR=M(4)+5, VAR=3).
@H:V	Cursor Position - horizontal and vertical coordinates, separated by a colon, denoting the input location on the screen display (e.g., @14:21).
CS=N	Do Not Clear Screen - do not clear the CRT before displaying a screen display format (particularly useful when using a GOSUB).

TABLE 5-4. MACRO COMMAND LINE ENTRIES (Cont)

Entry	Definition
G=nnnn	Establish GOSUB - sets the expanded source code as a GOSUB, beginning at line nnnn, and ending in a RETURN (e.g., G=1840).
L=nnnn	Change to Line nnnn - establishes the first line number of the expanded code as nnnn (e.g., L=1040).
#=	Channel Number - to be used for file I/O.
#=nn:nn	Channel Numbers - separated by a colon, to be used for file access. The first number refers to the channel for the data file access; the number after the colon is the channel for directory access (e.g., #=0:1).
DIR=nn	Directory Number - specifies the index file directory to be accessed (e.g., DIR=3).
ORG=	File Organization - specifies the organization of the file used in a read or write. Valid entries are (F)ormatted, (C)ontiguous and (T)ext (i.e., ORG=F, ORG=C, ORG=T).
PACK=N	No Index Optimize - stipulates that the index should not be reoptimized after a key is deleted or an insert operation is unsuccessful (this reoptimization is otherwise performed automatically).
KEY=	Index Access Mode - on a directory search, this parameter is used to specify an identical key match (EXACT), or the next key in the file (NEXT). Valid entries are EXACT and NEXT (i.e., KEY=EXACT, KEY=NEXT). If this parameter is not specified, the default search is for a partial key match.
LOCK=N	Do Not Lock Record - specifies that the I/O to a data file should not lock any records.
SUB=N	Do Not Use Subscriptions - indicates that string data elements should not be double-subscripted during file I/O.

TABLE 5-4. MACRO COMMAND LINE ENTRIES (Cont)

Entry	Definition
SKIP=nn	Directive to skip nn vertical lines after print of output format (header or detail).
ERR=nnnn:nnnn	Error Branching - may be used to set an "IF ERR 0 GOTO nnnn" branch during execution of the routine, and to reset it after its completion. The first number represents a line to which control should be transferred if an error is detected during the routine. If this number is zero, the error trap is cleared during the execution of the routine. The second number has a similar meaning, only the trap is initiated after execution of the routine (e.g., ERR=0:2180, ERR=680:0).
FOUND= nnnn:nnnn(M)	Search Branching - used to conditionally transfer control to line number(s) based on the success or failure of a SEARCH command. The first number indicates where control should be transferred if the SEARCH command was successful (e.g., if a find succeeded). The number after the colon shows where to go if the SEARCH command failed (e.g., if the record was not found). Placing the (M) parameter to the right of either line number (but not both) causes an appropriate message to be generated for the specified case (e.g., FOUND=930:620(M), FOUND=2100(M):180).

5.2.1 DISPLAY MACRO

The Display Macro is expanded into source code which creates a screen display format. The image of the screen display specified in the Display Macro command line is referenced from the Dictionary. PRINT statements are generated which duplicate the screen format.

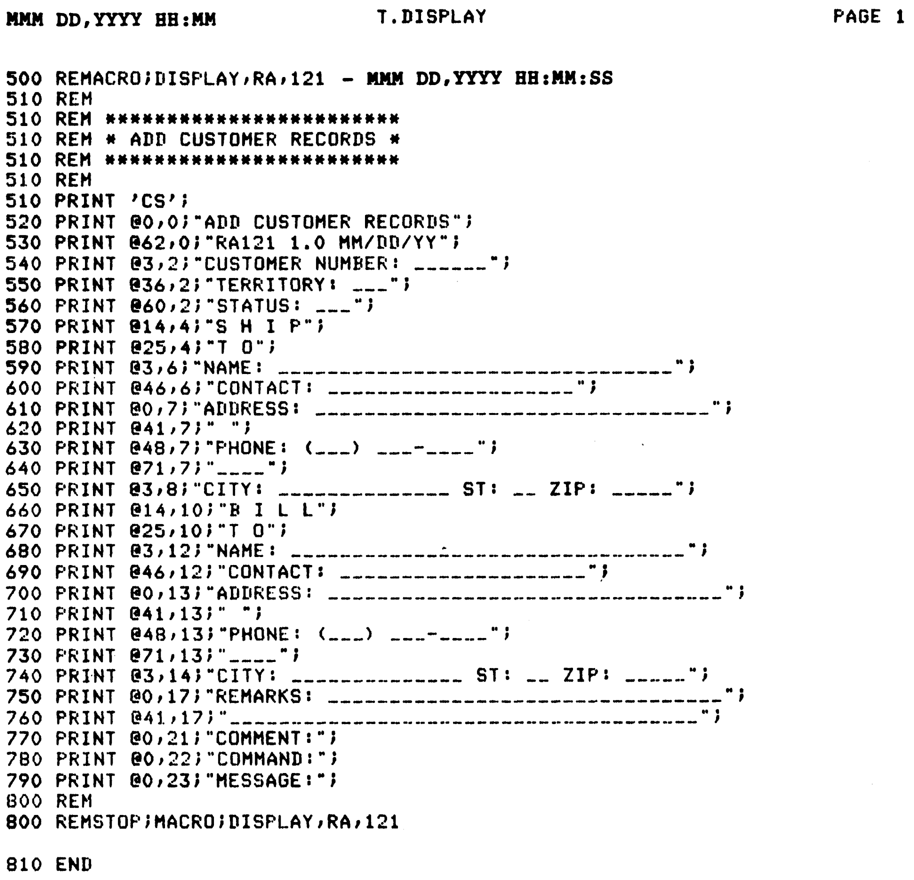
The command line format for the Display Macro is:

nnnn REMACRO;DISPLAY,SYSTEM,nnn(,Parameters)

Command line entries unique to the Display Macro are:

- Macro Type - DISPLAY
- Screen Display Number - the number of the screen format in the Dictionary
- Optional parameters - (see Tables 5-2, 5-3)

A sample Display Macro and its expanded code are shown below.



```
MMM DD,YYYY HH:MM            T.DISPLAY                    PAGE 1

500 REMACRO;DISPLAY,RA,121 - MMM DD,YYYY HH:MM:SS
510 REM
510 REM ************************
510 REM * ADD CUSTOMER RECORDS *
510 REM ************************
510 REM
510 PRINT 'CS';
520 PRINT @0,0;"ADD CUSTOMER RECORDS";
530 PRINT @62,0;"RA121 1.0 MM/DD/YY";
540 PRINT @3,2;"CUSTOMER NUMBER: ______";
550 PRINT @36,2;"TERRITORY: ___";
560 PRINT @60,2;"STATUS: ___";
570 PRINT @14,4;"S H I P";
580 PRINT @25,4;"T O";
590 PRINT @3,6;"NAME: ________________________________";
600 PRINT @46,6;"CONTACT: ____________________";
610 PRINT @0,7;"ADDRESS: ________________________________";
620 PRINT @41,7;" ";
630 PRINT @48,7;"PHONE: (___) ___-____";
640 PRINT @71,7;"____";
650 PRINT @3,8;"CITY: ______________ ST: __ ZIP: _____";
660 PRINT @14,10;"B I L L";
670 PRINT @25,10;"T O";
680 PRINT @3,12;"NAME: ________________________________";
690 PRINT @46,12;"CONTACT: ____________________";
700 PRINT @0,13;"ADDRESS: ________________________________";
710 PRINT @41,13;" ";
720 PRINT @48,13;"PHONE: (___) ___-____";
730 PRINT @71,13;"____";
740 PRINT @3,14;"CITY: ______________ ST: __ ZIP: _____";
750 PRINT @0,17;"REMARKS: ________________________________";
760 PRINT @41,17;"______________________________________";
770 PRINT @0,21;"COMMENT:";
780 PRINT @0,22;"COMMAND:";
790 PRINT @0,23;"MESSAGE:";
800 REM
800 REMSTOP;MACRO;DISPLAY,RA,121

810 END
```

5.2.2 DATA MACRO

An expanded Data Macro can dimension any number of system data elements. Data elements are dimensioned and documented based upon their attributes as currently defined to the Dictionary.

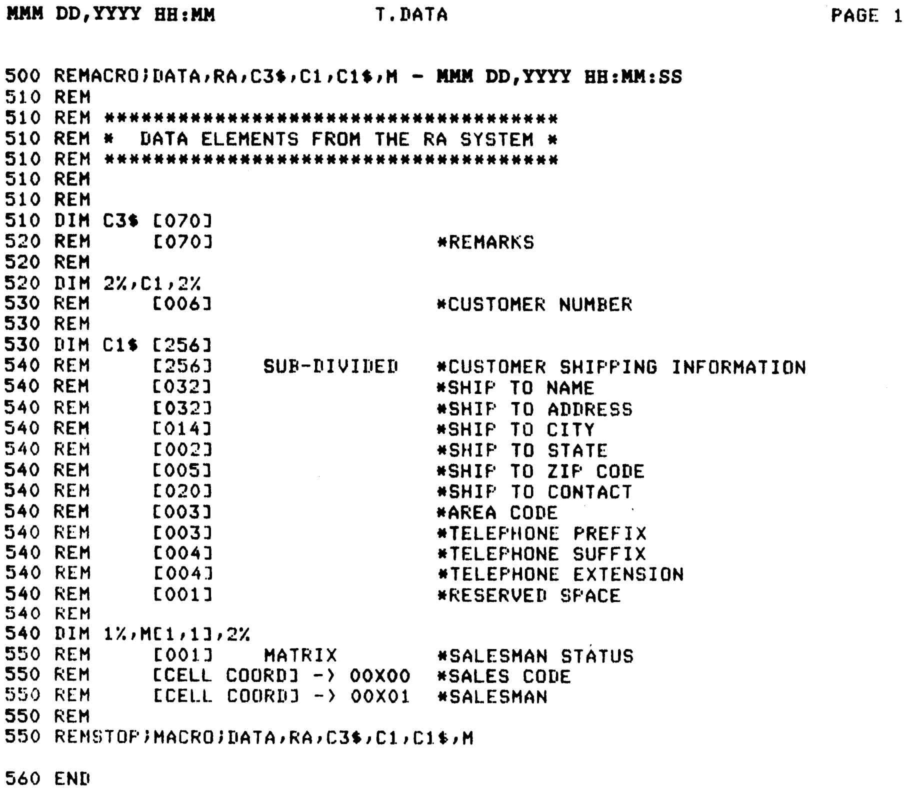
The command line format for the Data Macro is:

nnnn REMACRO;DATA,SYSTEM,ELEMENTS(,Parameters)

Command line entries unique to the Display Macro are:

- Macro Type - DATA
- Data Elements - selected data elements to be dimensioned, from the specified system, separated by commas
- Optional parameters - (see Tables 5-2, 5-3)

A sample Data Macro and its expanded code are shown below.



```
MMM DD,YYYY HH:MM             T.DATA                                   PAGE 1

500 REMACRO;DATA,RA,C3$,C1,C1$,M - MMM DD,YYYY HH:MM:SS
510 REM
510 REM *************************************
510 REM *  DATA ELEMENTS FROM THE RA SYSTEM *
510 REM *************************************
510 REM
510 REM
510 DIM C3$ [070]
520 REM     [070]                 *REMARKS
520 REM
520 DIM 2%,C1,2%
530 REM     [006]                 *CUSTOMER NUMBER
530 REM
530 DIM C1$ [256]
540 REM     [256]    SUB-DIVIDED  *CUSTOMER SHIPPING INFORMATION
540 REM     [032]                 *SHIP TO NAME
540 REM     [032]                 *SHIP TO ADDRESS
540 REM     [014]                 *SHIP TO CITY
540 REM     [002]                 *SHIP TO STATE
540 REM     [005]                 *SHIP TO ZIP CODE
540 REM     [020]                 *SHIP TO CONTACT
540 REM     [003]                 *AREA CODE
540 REM     [003]                 *TELEPHONE PREFIX
540 REM     [004]                 *TELEPHONE SUFFIX
540 REM     [004]                 *TELEPHONE EXTENSION
540 REM     [001]                 *RESERVED SPACE
540 REM
540 DIM 1%,M[1,1],2%
550 REM     [001]    MATRIX       *SALESMAN STATUS
550 REM     [CELL COORD] -> 00X00 *SALES CODE
550 REM     [CELL COORD] -> 00X01 *SALESMAN
550 REM
550 REMSTOP;MACRO;DATA,RA,C3$,C1,C1$,M

560 END
```

5.2.3 INPUT MACRO

The INPUT Macro is expanded into source code for the input and edit of a data element. Complete edit checks with explicit error messages are established, based upon the data element's attributes as currently defined to the Dictionary.

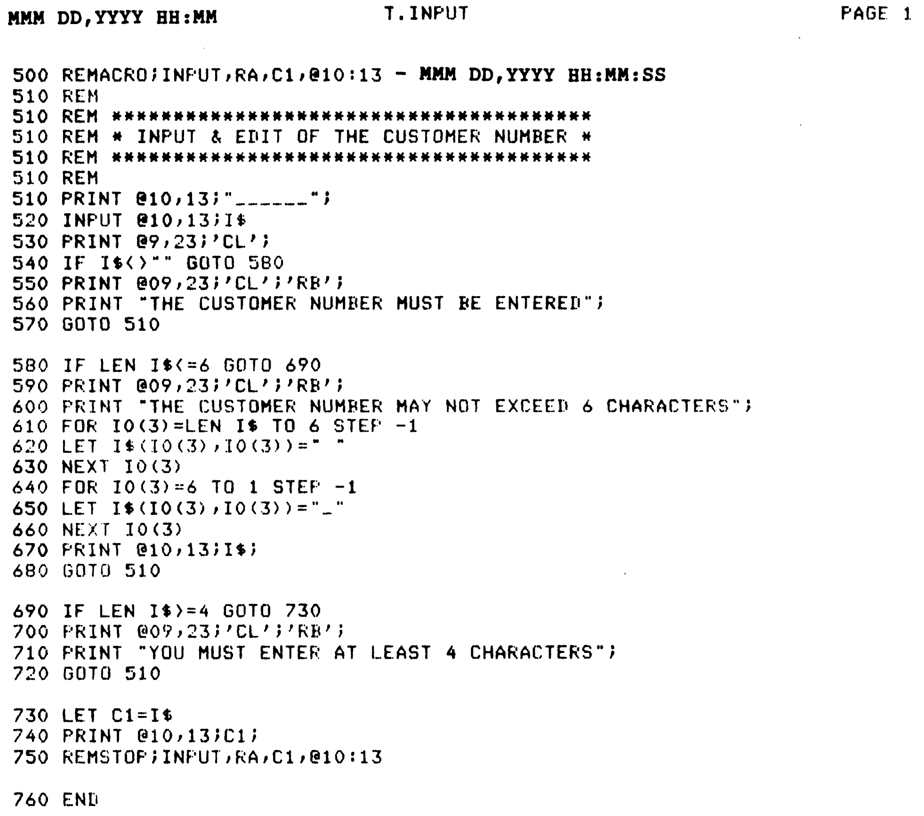
The command line format for the Input Macro is:

nnnn REMACRO;INPUT,SYSTEM,ELEMENT,@H:V(,Parameters)

Command line entries unique to the Input Macro are:

- Macro Type - INPUT
- Data Element - name of data element to be input (data element may not be a matrix)
- Input Position - @H:V
- Optional parameters - (see Tables 5-2, 5-3)

A sample Input Macro and its expanded code are shown below.



```
MMM DD,YYYY HH:MM              T.INPUT                           PAGE 1

500 REMACRO;INPUT,RA,C1,@10:13 - MMM DD,YYYY HH:MM:SS
510 REM
510 REM ***************************************
510 REM * INPUT & EDIT OF THE CUSTOMER NUMBER *
510 REM ***************************************
510 REM
510 PRINT @10,13;"______";
520 INPUT @10,13;I$
530 PRINT @9,23;'CL';
540 IF I$<>"" GOTO 580
550 PRINT @09,23;'CL';'RB';
560 PRINT "THE CUSTOMER NUMBER MUST BE ENTERED";
570 GOTO 510

580 IF LEN I$<=6 GOTO 690
590 PRINT @09,23;'CL';'RB';
600 PRINT "THE CUSTOMER NUMBER MAY NOT EXCEED 6 CHARACTERS";
610 FOR IO(3)=LEN I$ TO 6 STEP -1
620 LET I$(IO(3),IO(3))=" "
630 NEXT IO(3)
640 FOR IO(3)=6 TO 1 STEP -1
650 LET I$(IO(3),IO(3))="_"
660 NEXT IO(3)
670 PRINT @10,13;I$;
680 GOTO 510

690 IF LEN I$>=4 GOTO 730
700 PRINT @09,23;'CL';'RB';
710 PRINT "YOU MUST ENTER AT LEAST 4 CHARACTERS";
720 GOTO 510

730 LET C1=I$
740 PRINT @10,13;C1;
750 REMSTOP;INPUT,RA,C1,@10:13

760 END
```

5.2.4 USER MACRO

The User Macro enables inclusion of user-written source code within program text files. A text file that contains the user-written source code is defined to FORCE as a program. The text file is subsequently specified in a User Macro command line, and may be expanded within any text file in which the User Macro is inserted. Any number of text files may be employed as User Macros.

FORCE Level 5 implements the ability to substitute parameter values each time the User Macro is inserted. Parameter passing procedures are described in this section.

User Macro Requirements:

1. User-written code to be implemented with a User Macro must be in text file form. The code must be renumberable by itself without any errors. All referenced line numbers (or labels) within user-written code must be contained within the user-written code.

2. Routines or sections of code to be implemented by the User Macro must be established (see 1.5.1) as programs within the Dictionary prior to expansion.

3. The program name defined to FORCE must be the same as the associated text file name on disk.

4. The text file name specified in the User Macro command line must be the same as the program name defined to FORCE. Therefore, the disk text file name, the program name defined to FORCE, and the name specified in the User Macro command line must all be the same.

The command line format for the User Macro is:

nnnn REMACRO;USER,SYSTEM,TEXTFILE(,Parameters)

Command line entries unique to the User Macro are:

- Macro Type - USER
- Text File name - name of the text file on disk to be implemented
- Optional parameters - (see Tables 5-2, 5-3)

USER MACRO PARAMETER PASSING

The User Macro in FORCE Level 5 is enhanced with the ability to pass parameters. Portions of any statement(s) in the User Macro text file can be designated for variable information each time the program/routine is implemented. Parameter substitutions are specified in REMINFO command lines, which accompany the User REMACRO command line (see 5.3 for more on REMINFO command lines).

The following is a brief, three-line example of a User Macro expansion, including substituted parameter values. The user-written routine to be implemented is named S.DELETE, and it is associated with the PAYROLL system. Note the syntactical structure of the substituted literal parameters.

User-written routine

```
    10 SEARCH #{CHANNEL},5,1;{KEY},V1,{STATUS}
    20 IF {STATUS}=0 RETURN
    30 PRINT @09,23;'CLRB';{MESSAGE};
```

Macro command lines

```
    100 REMACRO;USER,PAYROLL,S.DELETE,G=200
    110 REMINFO;{CHANNEL=4},{KEY=I$},{STATUS=S(3,4)}
    120 REMINFO;{MESSAGE="KEY WAS NOT FOUND"}
```

Expanded source code

```
    200 SEARCH #4,5,1;I$,V1,S(3,4)
    210 IF S(3,4)=0 RETURN
    220 PRINT @09,23;'CLRB';"KEY WAS NOT FOUND";
    210 RETURN
```

Valid User Macro Substitution Parameters	Example (above)
Absolute Values	{CHANNEL=4}
Variables	{KEY=I$}
Subscripted Variables	{STATUS=S(3,4)}
Quoted Literals	{MESSAGE="KEY WAS NOT FOUND"}

Parameters must be grouped by { } symbols (braces) in the implemented program text file and the User Macro REMINFO command lines. The ASCII representations for these symbols are 173 and 175.

Each parameter in the text file must be defined in the REMINFO command lines. A substituted value cannot be more than 60 characters long, and the parameter keyword (i.e., CHANNEL, STATUS) cannot exceed 10 characters in length.

The next page shows a text file that is to be implemented with the User Macro. On the following page is the corresponding expanded User Macro (note the substituted parameters).

```
MMM DD,YYYY HH:MM             T.USER                              PAGE 1

10 REM ***********************************************
10 REM * CALLABLE ROUTINE:  BUILD AN INDEX WORK FILE *
10 REM ***********************************************
10 REM
10 REM     INPUT:  FILNAM - NAME TO BE USED FOR DIRECTORY BUILD.
10 REM             FILNUM - CHANNEL ON WHICH TO OPEN DIRECTORY.
10 REM             KEYLEN - LENGTH OF DIRECTORY KEY (IN WORDS).
10 REM             KEYNUM - MAXIMUM NUMBER OF ALLOWABLE INSERTS.
10 REM
10 REM VARIABLES:   Z = STATUS RETURNED BY SEARCH.
10 REM             Z1 = NUMBER OF KEYS IN EACH BLOCK.
10 REM             Z2 = NUMBER OF FINE BLOCKS REQUIRED.
10 REM             Z3 = NUMBER OF COARSE BLOCKS REQUIRED.
10 REM             I$ = COMPLETE NAME FOR DIRECTORY BUILD.
10 REM
10 DEF FNR(Z)=INT(Z)+SGN(FRA Z)
20 LET Z1=FNR[254/({KEYLEN}+1)]
30 LET Z2=FNR[{KEYNUM}*2/(Z1+1)]
40 IF Z2<2 LET Z2=2
50 LET Z3=FNR[Z2/(Z1-1)]
60 IF Z3<2 LET Z3=2
70 REM
70 REM CONSTRUCT FILE NAME AND BUILD THE DIRECTORY.
70 REM
70 LET I$="<00> ",Z2+Z3+1
80 LET I$=I$[1,5],"[",I$[7,LEN I$-1],":256] ",{FILNAM}
90 BUILD #{FILNUM},I$
100 REM
100 REM ESTABLISH DIRECTORY STRUCTURE AND RETURN.
100 REM
100 SEARCH #{FILNUM},0,1;I$,{KEYLEN},Z
110 IF NOT Z SEARCH #{FILNUM},0,0;I$,{KEYNUM},Z
120 RETURN
```

```
MMM DD,YYYY HH:MM             T.USER1                                PAGE 1

500 REMACRO;USER,RA,T.USER - MMM DD,YYYY HH:MM:SS
510 REMINFO;{FILNAM=N$},{FILNUM=0},{KEYLEN=K0},{KEYNUM=R0}
520 REM
520 REM ***************************************************
520 REM * USER - T.USER - CALLABLE ROUTINE FOR USER MACRO *
520 REM ***************************************************
520 REM
520 REM  PARAMETER   NO. USED   EQUATED EXPRESSION
520 REM  ---------   --------   ------------------
520 REM  FILNAM        0001     N$
520 REM  FILNUM        0003     0
520 REM  KEYLEN        0002     K0
520 REM  KEYNUM        0002     R0
520 REM
520 REM ***********************************************
520 REM * CALLABLE ROUTINE:  BUILD AN INDEX WORK FILE *
520 REM ***********************************************
520 REM
520 REM     INPUT:  FILNAM - NAME TO BE USED FOR DIRECTORY BUILD.
520 REM             FILNUM - CHANNEL ON WHICH TO OPEN DIRECTORY.
520 REM             KEYLEN - LENGTH OF DIRECTORY KEY (IN WORDS).
520 REM             KEYNUM - MAXIMUM NUMBER OF ALLOWABLE INSERTS.
520 REM
520 REM VARIABLES:   Z = STATUS RETURNED BY SEARCH.
520 REM             Z1 = NUMBER OF KEYS IN EACH BLOCK.
520 REM             Z2 = NUMBER OF FINE BLOCKS REQUIRED.
520 REM             Z3 = NUMBER OF COARSE BLOCKS REQUIRED.
520 REM             I$ = COMPLETE NAME FOR DIRECTORY BUILD.
520 REM
520 DEF FNR(Z)=INT(Z)+SGN(FRA Z)
530 LET Z1=FNR[254/(K0+1)]
540 LET Z2=FNR[R0*2/(Z1+1)]
550 IF Z2<2 LET Z2=2
560 LET Z3=FNR[Z2/(Z1-1)]
570 IF Z3<2 LET Z3=2
580 REM
580 REM CONSTRUCT FILE NAME AND BUILD THE DIRECTORY.
580 REM
580 LET I$="<00> ",Z2+Z3+1
590 LET I$=I$[1,5],"[",I$[7,LEN I$-1],":256] ",N$
600 BUILD #0,I$
610 REM
610 REM ESTABLISH DIRECTORY STRUCTURE AND RETURN.
610 REM
610 SEARCH #0,0,1;I$,K0,Z
620 IF NOT Z SEARCH #0,0,0;I$,R0,Z
630 RETURN

640 REM
640 REMSTOP;MACRO;USER,RA,T.USER

650 END
```

5.2.5 VARIABLES MACRO

An expanded Variables Macro dimensions all data elements in a specified file. FORCE generates this documented source code based upon current Dictionary information on the data elements' attributes, and the file's record layout.

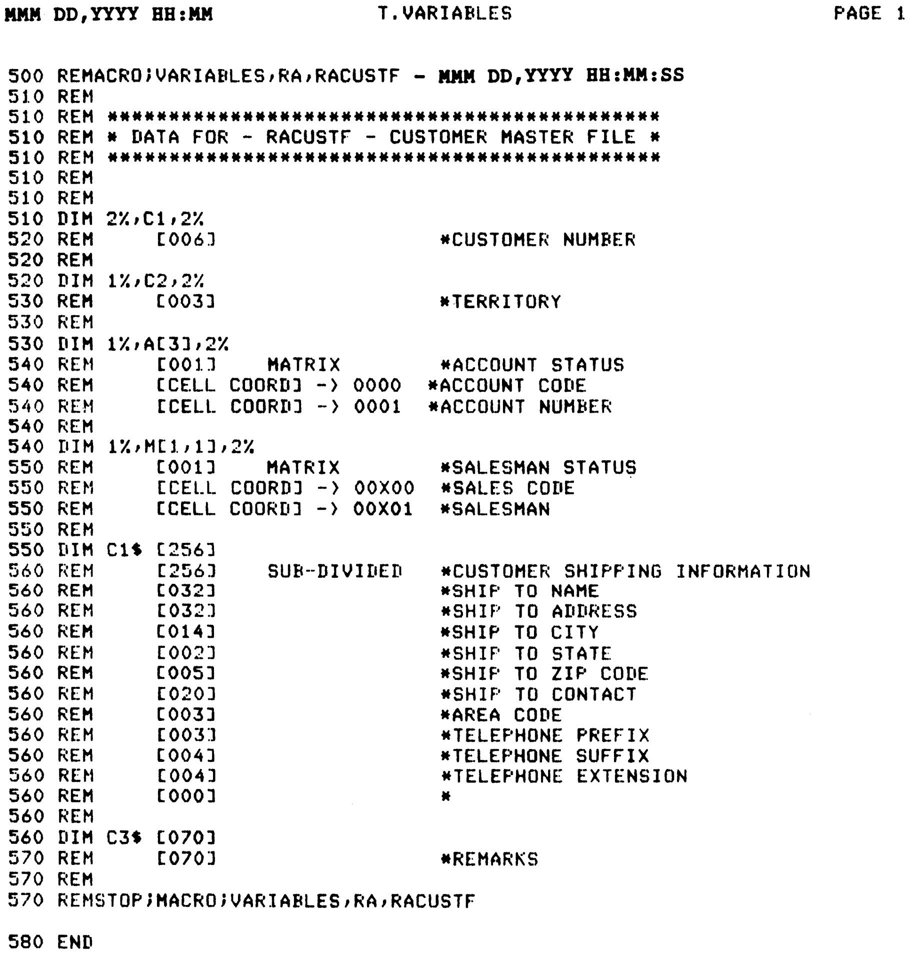
The command line format for the Variables Macro is:

nnnn REMACRO;VARIABLES,SYSTEM,DATAFILE(,Parameters)

Command line entries unique to the Variables Macro are:

- Macro Type - VARIABLES
- Data File Name - the name of the data file (as defined to FORCE) that contains the elements to be dimensioned
- Optional parameters - (see Tables 5-2, 5-3)

A sample Variables Macro and its expanded code are shown below.



```
MMM DD,YYYY HH:MM              T.VARIABLES                          PAGE 1

500 REMACRO;VARIABLES,RA,RACUSTF - MMM DD,YYYY HH:MM:SS
510 REM
510 REM ********************************************
510 REM * DATA FOR - RACUSTF - CUSTOMER MASTER FILE *
510 REM ********************************************
510 REM
510 REM
510 DIM 2%,C1,2%
520 REM     [006]                    *CUSTOMER NUMBER
520 REM
520 DIM 1%,C2,2%
530 REM     [003]                    *TERRITORY
530 REM
530 DIM 1%,A[3],2%
540 REM     [001]    MATRIX          *ACCOUNT STATUS
540 REM     [CELL COORD] -> 0000  *ACCOUNT CODE
540 REM     [CELL COORD] -> 0001  *ACCOUNT NUMBER
540 REM
540 DIM 1%,M[1,1],2%
550 REM     [001]    MATRIX          *SALESMAN STATUS
550 REM     [CELL COORD] -> 00X00  *SALES CODE
550 REM     [CELL COORD] -> 00X01  *SALESMAN
550 REM
550 DIM C1$ [256]
560 REM     [256]    SUB-DIVIDED     *CUSTOMER SHIPPING INFORMATION
560 REM     [032]                    *SHIP TO NAME
560 REM     [032]                    *SHIP TO ADDRESS
560 REM     [014]                    *SHIP TO CITY
560 REM     [002]                    *SHIP TO STATE
560 REM     [005]                    *SHIP TO ZIP CODE
560 REM     [020]                    *SHIP TO CONTACT
560 REM     [003]                    *AREA CODE
560 REM     [003]                    *TELEPHONE PREFIX
560 REM     [004]                    *TELEPHONE SUFFIX
560 REM     [004]                    *TELEPHONE EXTENSION
560 REM     [000]                    *
560 REM
560 DIM C3$ [070]
570 REM     [070]                    *REMARKS
570 REM
570 REMSTOP;MACRO;VARIABLES,RA,RACUSTF

580 END
```

5.2.6 READ MACRO

The Read Macro is expanded into source code, which performs a READ to a selected file. Information regarding the file's record layout is obtained by accessing the Dictionary, and documented in the expanded source code.

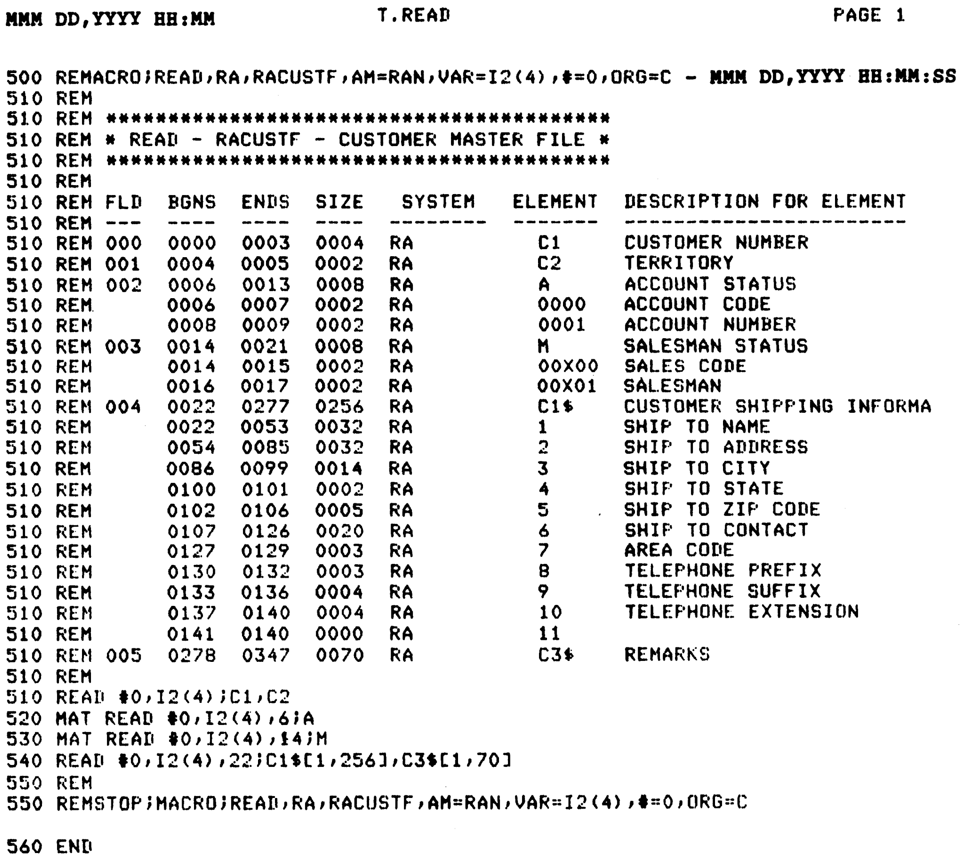
The command line format for the Read Macro is:

nnnn REMACRO;READ,SYSTEM,DATAFILE,AM=,VAR=,#=,ORG=(,Parameters)

Command line entries unique to the Read Macro are:

- Macro Type - READ
- Data File Name - the name of the data file to be read
- Access Method - AM=
- Access Variable - VAR=
- Channel Number - #=
- File Organization - ORG=
- Optional parameters - (see Tables 5-2, 5-3)

A sample Read Macro and its expanded code are shown below.



```
MMM DD,YYYY HH:MM              T.READ                                    PAGE 1

500 REMACRO;READ,RA,RACUSTF,AM=RAN,VAR=I2(4),#=0,ORG=C - MMM DD,YYYY HH:MM:SS
510 REM
510 REM ****************************************
510 REM * READ - RACUSTF - CUSTOMER MASTER FILE *
510 REM ****************************************
510 REM
510 REM FLD  BGNS  ENDS  SIZE   SYSTEM   ELEMENT  DESCRIPTION FOR ELEMENT
510 REM ---  ----  ----  ----  --------  -------  -----------------------
510 REM 000  0000  0003  0004  RA          C1     CUSTOMER NUMBER
510 REM 001  0004  0005  0002  RA          C2     TERRITORY
510 REM 002  0006  0013  0008  RA          A      ACCOUNT STATUS
510 REM      0006  0007  0002  RA          0000   ACCOUNT CODE
510 REM      0008  0009  0002  RA          0001   ACCOUNT NUMBER
510 REM 003  0014  0021  0008  RA          M      SALESMAN STATUS
510 REM      0014  0015  0002  RA          00X00  SALES CODE
510 REM      0016  0017  0002  RA          00X01  SALESMAN
510 REM 004  0022  0277  0256  RA          C1$    CUSTOMER SHIPPING INFORMA
510 REM      0022  0053  0032  RA          1      SHIP TO NAME
510 REM      0054  0085  0032  RA          2      SHIP TO ADDRESS
510 REM      0086  0099  0014  RA          3      SHIP TO CITY
510 REM      0100  0101  0002  RA          4      SHIP TO STATE
510 REM      0102  0106  0005  RA          5      SHIP TO ZIP CODE
510 REM      0107  0126  0020  RA          6      SHIP TO CONTACT
510 REM      0127  0129  0003  RA          7      AREA CODE
510 REM      0130  0132  0003  RA          8      TELEPHONE PREFIX
510 REM      0133  0136  0004  RA          9      TELEPHONE SUFFIX
510 REM      0137  0140  0004  RA          10     TELEPHONE EXTENSION
510 REM      0141  0140  0000  RA          11
510 REM 005  0278  0347  0070  RA          C3$    REMARKS
510 REM
510 READ #0,I2(4);C1,C2
520 MAT READ #0,I2(4),6;A
530 MAT READ #0,I2(4),14;M
540 READ #0,I2(4),22;C1$[1,256],C3$[1,70]
550 REM
550 REMSTOP;MACRO;READ,RA,RACUSTF,AM=RAN,VAR=I2(4),#=0,ORG=C

560 END
```

5.2.7 WRITE MACRO

The Write Macro is expanded into source code, which performs a WRITE to a selected file. Information regarding the file's record layout is obtained by accessing the Dictionary, and documented in the expanded source code.

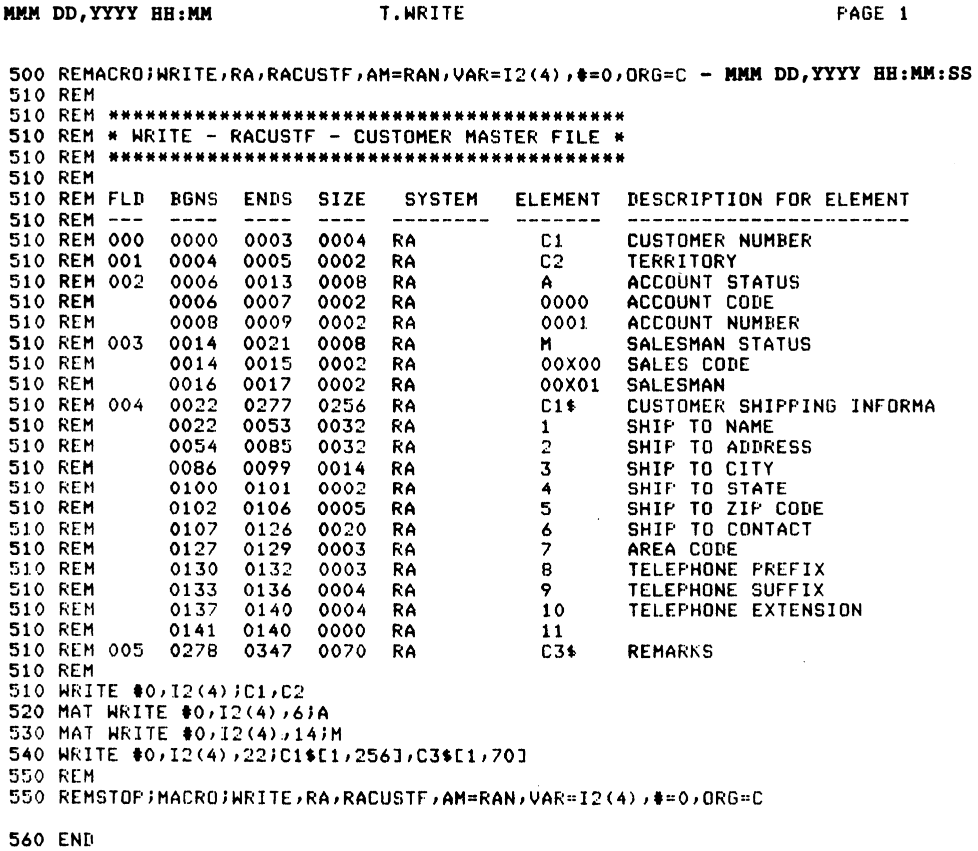
The command line format for the Write Macro is:

nnnn REMACRO;WRITE,SYSTEM,DATAFILE,AM=,VAR=,#=,ORG=(,Parameters)

Command line entries unique to the Write Macro are:

- Macro Type - WRITE
- Data File Name - the name of the data file to be read
- Access Method - AM=
- Access Variable - VAR=
- Channel Number - #=
- File Organization - ORG=
- Optional parameters - (see Tables 5-2, 5-3)

A sample Write Macro and its expanded code are shown below.



```
MMM DD,YYYY HH:MM              T.WRITE                              PAGE 1

500 REMACRO;WRITE,RA,RACUSTF,AM=RAN,VAR=I2(4),#=0,ORG=C - MMM DD,YYYY HH:MM:SS
510 REM
510 REM ******************************************
510 REM * WRITE - RACUSTF - CUSTOMER MASTER FILE *
510 REM ******************************************
510 REM
510 REM FLD  BGNS  ENDS  SIZE   SYSTEM   ELEMENT  DESCRIPTION FOR ELEMENT
510 REM ---  ----  ----  ----  --------  -------  -----------------------
510 REM 000  0000  0003  0004  RA          C1     CUSTOMER NUMBER
510 REM 001  0004  0005  0002  RA          C2     TERRITORY
510 REM 002  0006  0013  0008  RA          A      ACCOUNT STATUS
510 REM      0006  0007  0002  RA          0000   ACCOUNT CODE
510 REM      0008  0009  0002  RA          0001   ACCOUNT NUMBER
510 REM 003  0014  0021  0008  RA          M      SALESMAN STATUS
510 REM      0014  0015  0002  RA          00X00  SALES CODE
510 REM      0016  0017  0002  RA          00X01  SALESMAN
510 REM 004  0022  0277  0256  RA          C1$    CUSTOMER SHIPPING INFORMA
510 REM      0022  0053  0032  RA          1      SHIP TO NAME
510 REM      0054  0085  0032  RA          2      SHIP TO ADDRESS
510 REM      0086  0099  0014  RA          3      SHIP TO CITY
510 REM      0100  0101  0002  RA          4      SHIP TO STATE
510 REM      0102  0106  0005  RA          5      SHIP TO ZIP CODE
510 REM      0107  0126  0020  RA          6      SHIP TO CONTACT
510 REM      0127  0129  0003  RA          7      AREA CODE
510 REM      0130  0132  0003  RA          8      TELEPHONE PREFIX
510 REM      0133  0136  0004  RA          9      TELEPHONE SUFFIX
510 REM      0137  0140  0004  RA          10     TELEPHONE EXTENSION
510 REM      0141  0140  0000  RA          11
510 REM 005  0278  0347  0070  RA          C3$    REMARKS
510 REM
510 WRITE #0,I2(4);C1,C2
520 MAT WRITE #0,I2(4),6;A
530 MAT WRITE #0,I2(4),,14;M
540 WRITE #0,I2(4),22;C1$[1,256],C3$[1,70]
550 REM
550 REMSTOP;MACRO;WRITE,RA,RACUSTF,AM=RAN,VAR=I2(4),#=0,ORG=C

560 END
```

5.2.8 DEFINE MACRO

The expanded Define Macro dimensions a set of global variables for program use. Extensive documentation is included in the generated source code, to facilitate program development and maintenance.

Depending upon the parameters specified in the command line, three sets of global variables may be generated: report program, entry program or standard program global variables. To generate global variables for a report or entry program, the system and program names should be specified as parameters in the command line.

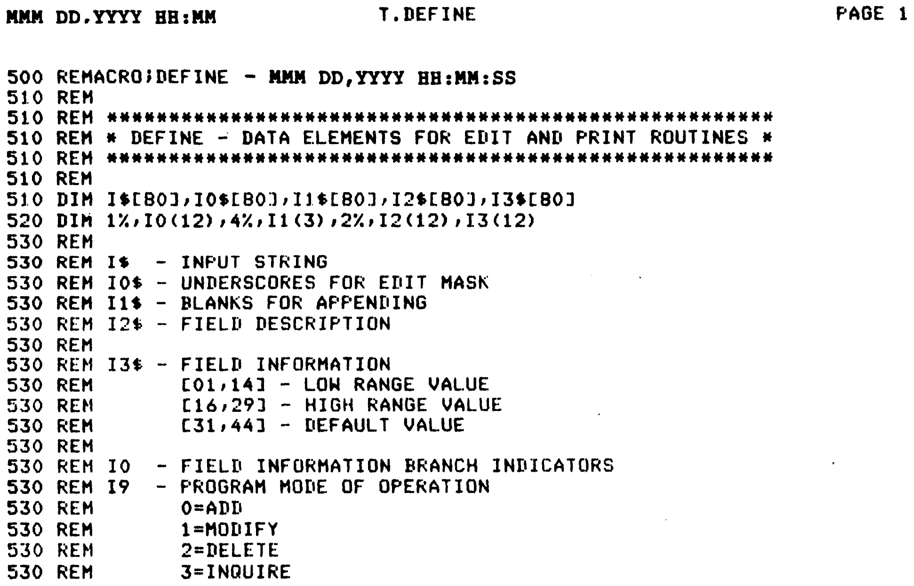
The command line format for the Define Macro is:

nnnn REMACRO;DEFINE(,Parameters)

Command line entries unique to the Define Macro are:

- Macro Type - DEFINE
- Optional parameters - (see Tables 5-2, 5-3)

The expanded code for a Define Macro in an entry program is shown below.



```
MMM DD.YYYY HH:MM              T.DEFINE                                  PAGE 1

500 REMACRO;DEFINE - MMM DD,YYYY HH:MM:SS
510 REM
510 REM ******************************************************
510 REM * DEFINE - DATA ELEMENTS FOR EDIT AND PRINT ROUTINES *
510 REM ******************************************************
510 REM
510 DIM I$[80],I0$[80],I1$[80],I2$[80],I3$[80]
520 DIM 1%,I0(12),4%,I1(3),2%,I2(12),I3(12)
530 REM
530 REM I$  - INPUT STRING
530 REM I0$ - UNDERSCORES FOR EDIT MASK
530 REM I1$ - BLANKS FOR APPENDING
530 REM I2$ - FIELD DESCRIPTION
530 REM
530 REM I3$ - FIELD INFORMATION
530 REM       [01,14] - LOW RANGE VALUE
530 REM       [16,29] - HIGH RANGE VALUE
530 REM       [31,44] - DEFAULT VALUE
530 REM
530 REM I0  - FIELD INFORMATION BRANCH INDICATORS
530 REM I9  - PROGRAM MODE OF OPERATION
530 REM       0=ADD
530 REM       1=MODIFY
530 REM       2=DELETE
530 REM       3=INQUIRE
```

Continues on next page.

```
530 REM
530 REM I1  - NUMERIC RANGE CHECK MATRIX
530 REM       [00] - NUMERIC VALUE OF INPUT DATA
530 REM       [01] - NUMERIC LOW RANGE VALUE
530 REM       [02] - NUMERIC HIGH RANGE VALUE
530 REM       [03] - VARIABLE VALUE
530 REM
530 REM I2  - FILE ACCESS INFORMATION
530 REM       [00] - KEY LENGTH BEFORE SEARCH
530 REM       [01] - DATA FILE RECORD NUMBER
530 REM       [02] -
530 REM       [03] -
530 REM       [04] - RECORD NUMBER
530 REM       [05] - SEARCH STATUS VALUES
530 REM                 0=OPERATION WAS SUCCESSFUL
530 REM                 1=OPERATION WAS NOT SUCCESSFUL
530 REM                 2=END OF DIRECTORY
530 REM                 3=END OF DATA LOCATED
530 REM                 4=FILE IS NOT INDEXED
530 REM                 5=UNDETERMINED ERROR
530 REM       [06] - SEARCH FUNCTION VARIABLE
530 REM       [07] - LENGTH OF MESSAGE
530 REM
530 REMSTOP;MACRO;DEFINE

540 END
```

5.3 LEVEL 5 MACRO COMMANDS

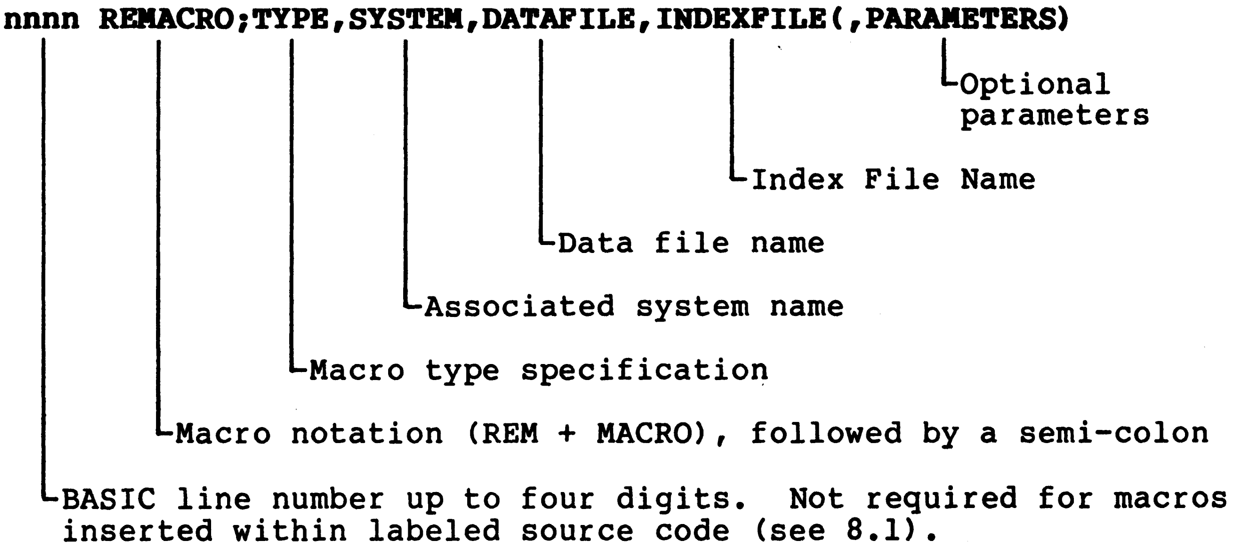
Level 5 macros are expanded into source code for indexed file access. These macros enable extensive utilization of the SEARCH facility in IRIS Business BASIC.

Each Level 5 macro consists of a set of command lines, including the standard REMACRO command line and any number of REMINFO command lines. REMINFO command lines allow the user to specify supplemental information about a macro.

The Level 5 macro commands are listed below, with their corresponding function and section in the manual:

Type	Function	Section
FIND	Locate Matching Key	5.3.1
INSERT	Place Key in Directory	5.3.2
DELETE	Remove Key from Directory	5.3.3
GET	Perform FIND, Read Data Record	5.3.4
PUT	Write Data Record, Perform INSERT	5.3.5
UPDATE	Perform FIND, Write Data Record	5.3.6
LINK	Extract Available Location, Perform PUT	5.3.7
FREE	Perform DELETE, Update Free List	5.3.8

The standard Level 5 REMACRO command line format is shown below:



nnnn REMACRO;TYPE,SYSTEM,DATAFILE,INDEXFILE(,PARAMETERS)

Optional parameters
Index File Name
Data file name
Associated system name
Macro type specification
Macro notation (REM + MACRO), followed by a semi-colon
BASIC line number up to four digits. Not required for macros inserted within labeled source code (see 8.1).

REMACRO command-line entries must adhere to this order and format. A semicolon must follow the REMACRO notation, and subsequent command-line entries must be separated by commas.

REMACRO command-line structure is the same for all Level 5 macros, up to and including the index file name. REMINFO command-line structure is shown on the following page.

A Level 5 macro must contain at least one, and may contain any number of REMINFO command lines. The REMINFO command lines combine with the REMACRO command line to give additional information about a macro. They are similar in structure to the REMACRO command line.

The standard Level 5 REMINFO command line structure is shown below:

```
nnnn REMINFO;CHANNELS,DIRECTORY(,PARAMETERS)
 |     |        |        |         |
 |     |        |        |         └Parameters
 |     |        |        |          (required/optional)
 |     |        |        |
 |     |        |        └Directory number
 |     |        |
 |     |        └Channel numbers (data & index files)
 |     |
 |     └Macro notation (REM + INFO), followed by a semi-colon
 |
 └BASIC line number up to four digits (not required for macros
  inserted within labeled source code).
```

REMINFO command lines follow immediately after the associated REMACRO command line. The BASIC line number must represent a line number following that of the preceding REMACRO command line, or a preceding REMINFO command line.

The REMINFO line contents are the same for all Level 5 macros, up to and including the directory number. REMINFO command-line contents may be entered in any order after the REMINFO; notation. Each REMINFO line element may be entered as a separate command-line, if it is preceded by the appropriate nnnn REMINFO; statement.

For each macro, the channel numbers and the directory number only need to be entered once, and may be included in any of the REMINFO command-lines.

See Table 5-4 for an explanation of macro command-line entries.

See Table 5-5 for Level 5 REMACRO command line set-up; see Table 5-6 for Level 5 REMINFO command line set-up.

TABLE 5-5. REMACRO LINE SET-UP - LEVEL 5 MACROS

Command Line Entry*	Macro Type							
	FIND	INSERT	DELETE	GET	PUT	UPDATE	LINK	FREE
Line Number**	R	R	R	R	R	R	R	R
REMACRO; Notation	R	R	R	R	R	R	R	R
Macro Type	R	R	R	R	R	R	R	R
System Name	R	R	R	R	R	R	R	R
Data File Name	R	R	R	R	R	R	R	R
Index File Name	R	R	R	R	R	R	R	R
Key File Name	O	O	O	O	O	O	O	O
G=nnnn Set GOSUB Line nnnn	O	O	O	O	O	O	O	O
L=nnnn Set Line nnnn	O	O	O	O	O	O	O	O

R = Required
O = Optional

*Level 5 macro command line components, when used, must be entered in the order presented above, from top to bottom.

**A BASIC line number is not required for macros inserted within labeled source code (see 8.1).

TABLE 5-6. REMINFO LINE SET-UP - LEVEL 5 MACROS

Command Line Entry	Macro Type							
	FIND	INSERT	DELETE	GET	PUT	UPDATE	LINK	FREE
Line Number*	R	R	R	R	R	R	R	R
REMINFO; Notation	R	R	R	R	R	R	R	R
#=nn:nn Channel Numbers	R	R	R	R	R	R	R	R
DIR= Directory Number	R	R	R	R	R	R	R	R
ORG= File Organization				R	R	R	R	
VAR= Access Variable	O	R	O	O	R	O	R	O
PACK=N No Index Optimize		O	O		O		O	O
KEY= Index Access Mode	O			O				
LOCK=N No Record Lock				O	O	O	O	
SUB=N No Subscription				O	O	O	O	
ERR=nnnn:nnnn Error Branching	O	O	O	O	O	O	O	O
FOUND=nnnn:nnnn(M) Search Branching	O	O	O	O	O	O	O	O

R = Required
O = Optional

*A BASIC line number is not required for macros inserted within labeled source code (see 8.1).

5.3.1 FIND MACRO

The Find Macro is expanded into source code, which checks a specified index to determine if a designated key is on file. The generated source code is constructed, based upon macro-line parameters and the corresponding Dictionary information.

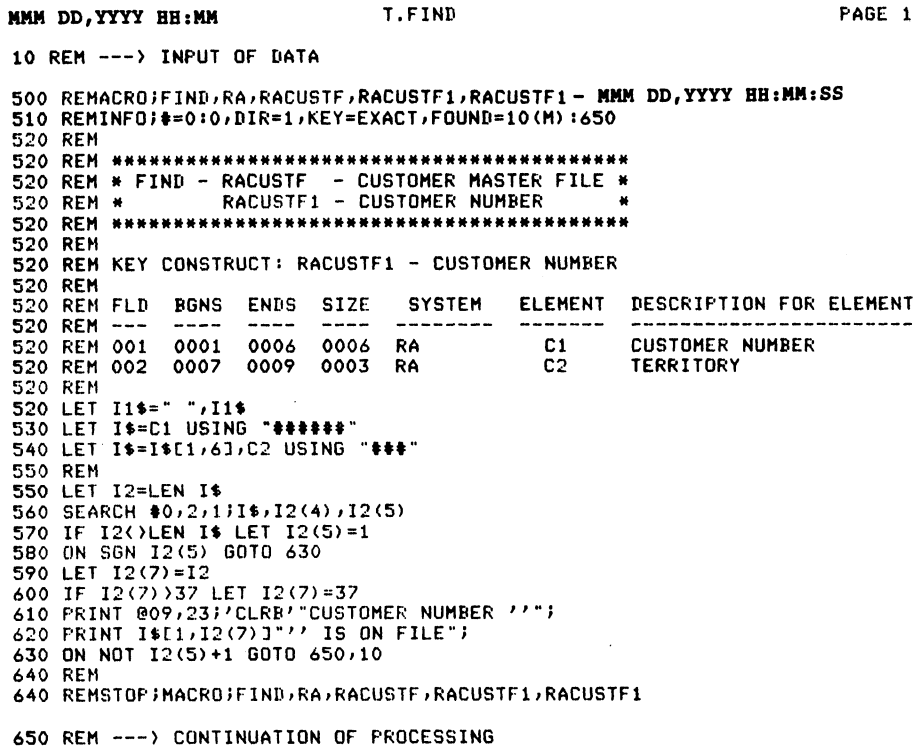
The command line format for the Find Macro is:

nnnn REMACRO;FIND,SYSTEM,DATAFILE,INDEXFILE(,Parameters)
nnnn REMINFO;CHANNELS,DIRECTORY(,Parameters)

Command line entries unique to the Find Macro are:

- Macro Type - FIND
- Parameters - (see Tables 5-5, 5-6)

A sample Find Macro and its expanded code are shown below.



```
MMM DD,YYYY HH:MM              T.FIND                              PAGE 1

10 REM ---> INPUT OF DATA

500 REMACRO;FIND,RA,RACUSTF,RACUSTF1,RACUSTF1- MMM DD,YYYY HH:MM:SS
510 REMINFO;#=0:0,DIR=1,KEY=EXACT,FOUND=10(M):650
520 REM
520 REM ******************************************
520 REM * FIND - RACUSTF  - CUSTOMER MASTER FILE *
520 REM *        RACUSTF1 - CUSTOMER NUMBER      *
520 REM ******************************************
520 REM
520 REM KEY CONSTRUCT: RACUSTF1 - CUSTOMER NUMBER
520 REM
520 REM FLD  BGNS  ENDS  SIZE   SYSTEM   ELEMENT  DESCRIPTION FOR ELEMENT
520 REM ---  ----  ----  ----  --------  -------  -----------------------
520 REM 001  0001  0006  0006  RA          C1     CUSTOMER NUMBER
520 REM 002  0007  0009  0003  RA          C2     TERRITORY
520 REM
520 LET I1$=" ",I1$
530 LET I$=C1 USING "######"
540 LET I$=I$[1,6],C2 USING "###"
550 REM
550 LET I2=LEN I$
560 SEARCH #0,2,1;I$,I2(4),I2(5)
570 IF I2<>LEN I$ LET I2(5)=1
580 ON SGN I2(5) GOTO 630
590 LET I2(7)=I2
600 IF I2(7)>37 LET I2(7)=37
610 PRINT @09,23;'CLRB'"CUSTOMER NUMBER ''";
620 PRINT I$[1,I2(7)]"'' IS ON FILE";
630 ON NOT I2(5)+1 GOTO 650,10
640 REM
640 REMSTOP;MACRO;FIND,RA,RACUSTF,RACUSTF1,RACUSTF1

650 REM ---> CONTINUATION OF PROCESSING
```

5.3.2 INSERT MACRO

The Insert Macro is expanded into source code, which places a key in a specified directory. Macro command line specifications are used to access the associated Dictionary information.

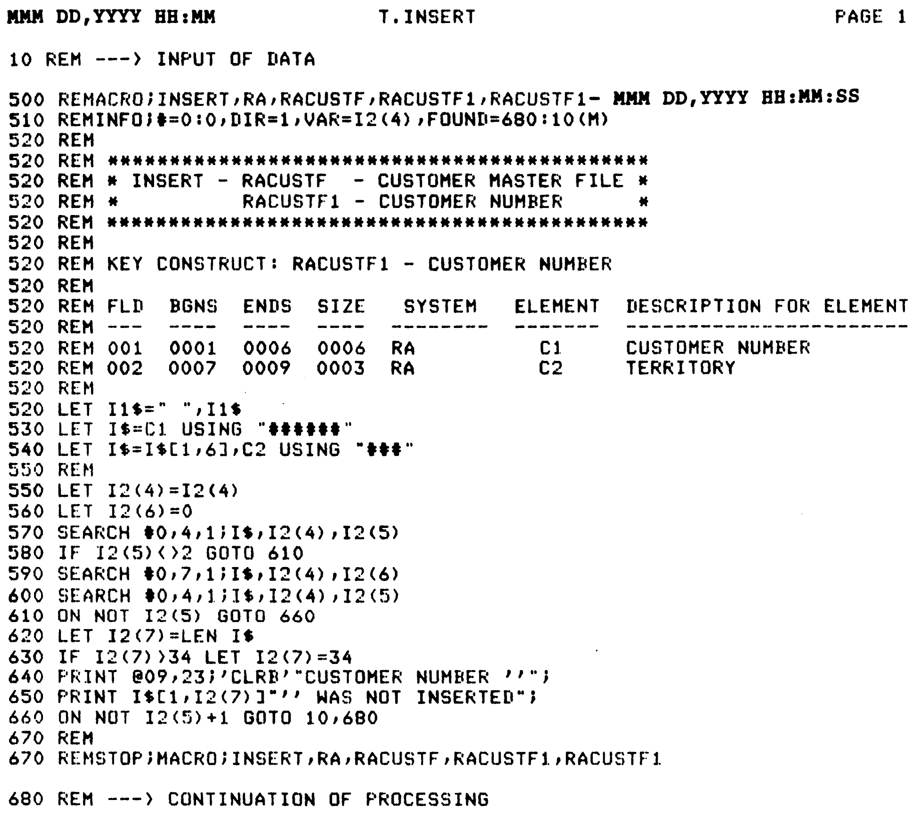
The command line format for the Insert Macro is:

nnnn REMACRO;INSERT,SYSTEM,DATAFILE,INDEXFILE(,Parameters)
nnnn REMINFO;CHANNELS,DIRECTORY,VAR=(,Parameters)

Command line entries unique to the Insert Macro are:

- Macro Type - INSERT
- Access Variable - VAR=
- Parameters - (see Tables 5-5, 5-6)

A sample Insert Macro and its expanded code are shown below.



```
MMM DD,YYYY HH:MM             T.INSERT                          PAGE 1

10 REM ---> INPUT OF DATA

500 REMACRO;INSERT,RA,RACUSTF,RACUSTF1,RACUSTF1- MMM DD,YYYY HH:MM:SS
510 REMINFO;#=0:0,DIR=1,VAR=I2(4),FOUND=680:10(M)
520 REM
520 REM ********************************************
520 REM * INSERT - RACUSTF  - CUSTOMER MASTER FILE *
520 REM *          RACUSTF1 - CUSTOMER NUMBER      *
520 REM ********************************************
520 REM
520 REM KEY CONSTRUCT: RACUSTF1 - CUSTOMER NUMBER
520 REM
520 REM FLD  BGNS  ENDS  SIZE   SYSTEM   ELEMENT  DESCRIPTION FOR ELEMENT
520 REM ---  ----  ----  ----  --------  -------  -----------------------
520 REM 001  0001  0006  0006  RA          C1     CUSTOMER NUMBER
520 REM 002  0007  0009  0003  RA          C2     TERRITORY
520 REM
520 LET I1$=" ",I1$
530 LET I$=C1 USING "######"
540 LET I$=I$[1,6],C2 USING "###"
550 REM
550 LET I2(4)=I2(4)
560 LET I2(6)=0
570 SEARCH #0,4,1;I$,I2(4),I2(5)
580 IF I2(5)<>2 GOTO 610
590 SEARCH #0,7,1;I$,I2(4),I2(6)
600 SEARCH #0,4,1;I$,I2(4),I2(5)
610 ON NOT I2(5) GOTO 660
620 LET I2(7)=LEN I$
630 IF I2(7)>34 LET I2(7)=34
640 PRINT @09,23;'CLRB'"CUSTOMER NUMBER ''";
650 PRINT I$[1,I2(7)]"'' WAS NOT INSERTED";
660 ON NOT I2(5)+1 GOTO 10,680
670 REM
670 REMSTOP;MACRO;INSERT,RA,RACUSTF,RACUSTF1,RACUSTF1

680 REM ---> CONTINUATION OF PROCESSING
```

5.3.3 DELETE MACRO

A Delete Macro is expanded into source code which removes a key from a directory, based upon information specified within the macro command lines. The generated code searches an index and deletes a specified key.

The command line format for the Delete Macro is:

nnnn REMACRO;DELETE,SYSTEM,DATAFILE,INDEXFILE(,Parameters)
nnnn REMINFO;CHANNELS,DIRECTORY(,Parameters)

Command line entries unique to the Delete Macro are:

- Macro Type - DELETE
- Parameters - (see Tables 5-5, 5-6)

A sample Delete Macro and its expanded code are shown below.

```
MMM DD,YYYY HH:MM            T.DELETE                              PAGE 1

10 REM ---> INPUT OF DATA

500 REMACRO;DELETE,RA,RACUSTF,RACUSTF1,RACUSTF1- MMM DD,YYYY HH:MM:SS
510 REMINFO;#=0:0,DIR=1,FOUND=640:10(M)
520 REM
520 REM ********************************************
520 REM * DELETE - RACUSTF  - CUSTOMER MASTER FILE *
520 REM *          RACUSTF1 - CUSTOMER NUMBER      *
520 REM ********************************************
520 REM
520 REM KEY CONSTRUCT: RACUSTF1 - CUSTOMER NUMBER
520 REM
520 REM FLD  BGNS  ENDS  SIZE   SYSTEM   ELEMENT  DESCRIPTION FOR ELEMENT
520 REM ---  ----  ----  ----  --------  -------  -----------------------
520 REM 001  0001  0006  0006  RA          C1     CUSTOMER NUMBER
520 REM 002  0007  0009  0003  RA          C2     TERRITORY
520 REM
520 LET I1$=" ",I1$
530 LET I$=C1 USING "######"
540 LET I$=I$[1,6],C2 USING "###"
550 REM
550 SEARCH #0,5,1;I$,I2(4),I2(5)
560 ON NOT I2(5) GOTO 610
570 LET I2(7)=LEN I$
580 IF I2(7)>37 LET I2(7)=37
590 PRINT @09,23;'CLRB'"CUSTOMER NUMBER ''";
600 PRINT I$[1,I2(7)]"'' WAS NOT FOUND";
610 IF I2(5)=0 SEARCH #0,7,1;I$,I2(4),I2(6)
620 ON NOT I2(5)+1 GOTO 10,640
630 REM
630 REMSTOP;MACRO;DELETE,RA,RACUSTF,RACUSTF1,RACUSTF1

640 REM ---> CONTINUATION OF PROCESSING
```

5.3.4 GET MACRO

The Get Macro is expanded into source code which performs a search for a matching key, then reads a data record on the retrieved pointer. The expanded source code is constructed based upon macro command-line specifications, and associated Dictionary information.

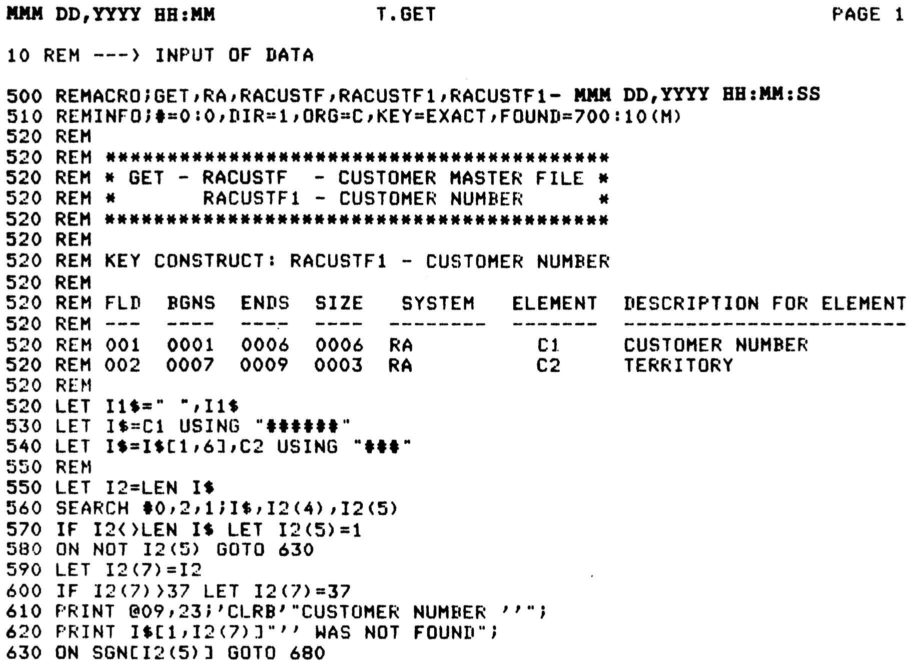
The command line format for the Get Macro is:

nnnn REMACRO;GET,SYSTEM,DATAFILE,INDEXFILE(,Parameters)
nnnn REMINFO;CHANNELS,DIRECTORY,ORG=(,Parameters)

Command line entries unique to the Get Macro are:

- Macro Type - GET
- File Organization - ORG=
- Parameters - (see Tables 5-5, 5-6)

A sample Get Macro and its expanded code are shown below.



```
MMM DD,YYYY HH:MM            T.GET                              PAGE 1

10 REM ---> INPUT OF DATA

500 REMACRO;GET,RA,RACUSTF,RACUSTF1,RACUSTF1- MMM DD,YYYY HH:MM:SS
510 REMINFO;#=0:0,DIR=1,ORG=C,KEY=EXACT,FOUND=700:10(M)
520 REM
520 REM ****************************************
520 REM * GET - RACUSTF  - CUSTOMER MASTER FILE *
520 REM *       RACUSTF1 - CUSTOMER NUMBER      *
520 REM ****************************************
520 REM
520 REM KEY CONSTRUCT: RACUSTF1 - CUSTOMER NUMBER
520 REM
520 REM FLD  BGNS  ENDS  SIZE   SYSTEM   ELEMENT  DESCRIPTION FOR ELEMENT
520 REM ---  ----  ----  ----  --------  -------  -----------------------
520 REM 001  0001  0006  0006  RA          C1     CUSTOMER NUMBER
520 REM 002  0007  0009  0003  RA          C2     TERRITORY
520 REM
520 LET I1$=" ",I1$
530 LET I$=C1 USING "######"
540 LET I$=I$[1,6],C2 USING "###"
550 REM
550 LET I2=LEN I$
560 SEARCH #0,2,1;I$,I2(4),I2(5)
570 IF I2<>LEN I$ LET I2(5)=1
580 ON NOT I2(5) GOTO 630
590 LET I2(7)=I2
600 IF I2(7)>37 LET I2(7)=37
610 PRINT @09,23;'CLRB'"CUSTOMER NUMBER ''";
620 PRINT I$[1,I2(7)]"'' WAS NOT FOUND";
630 ON SGN[I2(5)] GOTO 680
```

Continues on next page.

```
640 REM
640 REM FLD  BGNS  ENDS  SIZE   SYSTEM   ELEMENT  DESCRIPTION FOR ELEMENT
640 REM ---  ----  ----  ----  --------  -------  -----------------------
640 REM 000  0000  0003  0004  RA          C1     CUSTOMER NUMBER
640 REM 001  0004  0005  0002  RA          C2     TERRITORY
640 REM 002  0006  0013  0008  RA          A      ACCOUNT STATUS
640 REM      0006  0007  0002  RA        0000     ACCOUNT CODE
640 REM      0008  0009  0002  RA        0001     ACCOUNT NUMBER
640 REM 003  0014  0021  0008  RA          M      SALESMAN STATUS
640 REM      0014  0015  0002  RA        00X00    SALES CODE
640 REM      0016  0017  0002  RA        00X01    SALESMAN
640 REM 004  0022  0277  0256  RA         C1$     CUSTOMER SHIPPING INFORMA
640 REM      0022  0053  0032  RA          1      SHIP TO NAME
640 REM      0054  0085  0032  RA          2      SHIP TO ADDRESS
640 REM      0086  0099  0014  RA          3      SHIP TO CITY
640 REM      0100  0101  0002  RA          4      SHIP TO STATE
640 REM      0102  0106  0005  RA          5      SHIP TO ZIP CODE
640 REM      0107  0126  0020  RA          6      SHIP TO CONTACT
640 REM      0127  0129  0003  RA          7      AREA CODE
640 REM      0130  0132  0003  RA          8      TELEPHONE PREFIX
640 REM      0133  0136  0004  RA          9      TELEPHONE SUFFIX
640 REM      0137  0140  0004  RA          10     TELEPHONE EXTENSION
640 REM      0141  0141  0001  RA          11     RESERVED SPACE
640 REM 005  0278  0347  0070  RA          C3$    REMARKS
640 REM
640 READ #0,I2(4);C1,C2
650 MAT READ #0,I2(4),6;A
660 MAT READ #0,I2(4),14;M
670 READ #0,I2(4),22;C1$[1,256],C3$[1,70]
680 ON NOT I2(5)+1 GOTO 10,700
690 REM
690 REMSTOP;MACRO;GET,RA,RACUSTF,RACUSTF1,RACUSTF1

700 REM ---> CONTINUATION OF PROCESSING
```

5.3.5 PUT MACRO

The Put Macro is expanded into source code, which writes a data record, then inserts the key in a directory. The generated source code is constructed, based upon macro-line parameters and the corresponding Dictionary information.

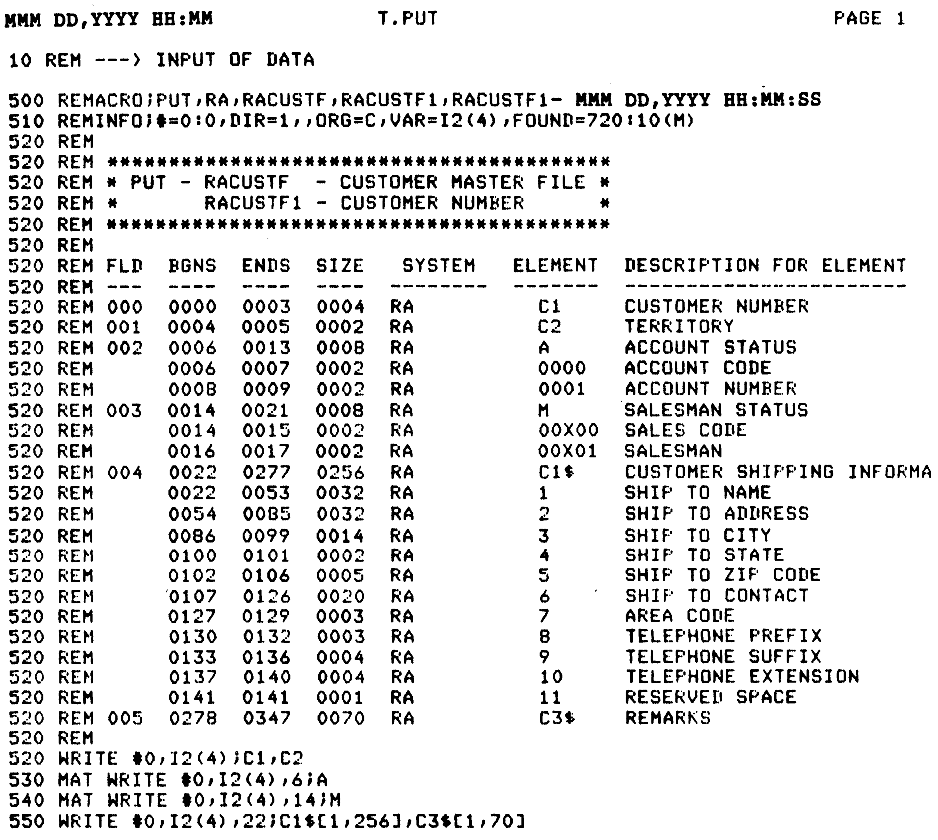
The command line format for the Put Macro is:

nnnn REMACRO;PUT,SYSTEM,DATAFILE,INDEXFILE(,Parameters)
nnnn REMINFO;CHANNELS,DIRECTORY,ORG=,VAR=(,Parameters)

Command line entries unique to the Put Macro are:

- Macro Type - PUT
- File Organization - ORG=
- Access Variable - VAR=
- Parameters - (see Tables 5-5, 5-6)

A sample Put Macro and its expanded code are shown below.



```
MMM DD,YYYY HH:MM                 T.PUT                              PAGE 1

10 REM ---> INPUT OF DATA

500 REMACRO;PUT,RA,RACUSTF,RACUSTF1,RACUSTF1- MMM DD,YYYY HH:MM:SS
510 REMINFO;#=0:0,DIR=1,,ORG=C,VAR=I2(4),FOUND=720:10(M)
520 REM
520 REM ****************************************
520 REM * PUT - RACUSTF  - CUSTOMER MASTER FILE *
520 REM *       RACUSTF1 - CUSTOMER NUMBER      *
520 REM ****************************************
520 REM
520 REM FLD  BGNS  ENDS  SIZE   SYSTEM    ELEMENT  DESCRIPTION FOR ELEMENT
520 REM ---  ----  ----  ----  ---------  -------  -----------------------
520 REM 000  0000  0003  0004  RA           C1     CUSTOMER NUMBER
520 REM 001  0004  0005  0002  RA           C2     TERRITORY
520 REM 002  0006  0013  0008  RA           A      ACCOUNT STATUS
520 REM      0006  0007  0002  RA           0000   ACCOUNT CODE
520 REM      0008  0009  0002  RA           0001   ACCOUNT NUMBER
520 REM 003  0014  0021  0008  RA           M      SALESMAN STATUS
520 REM      0014  0015  0002  RA           00X00  SALES CODE
520 REM      0016  0017  0002  RA           00X01  SALESMAN
520 REM 004  0022  0277  0256  RA           C1$    CUSTOMER SHIPPING INFORMA
520 REM      0022  0053  0032  RA           1      SHIP TO NAME
520 REM      0054  0085  0032  RA           2      SHIP TO ADDRESS
520 REM      0086  0099  0014  RA           3      SHIP TO CITY
520 REM      0100  0101  0002  RA           4      SHIP TO STATE
520 REM      0102  0106  0005  RA           5      SHIP TO ZIP CODE
520 REM      0107  0126  0020  RA           6      SHIP TO CONTACT
520 REM      0127  0129  0003  RA           7      AREA CODE
520 REM      0130  0132  0003  RA           8      TELEPHONE PREFIX
520 REM      0133  0136  0004  RA           9      TELEPHONE SUFFIX
520 REM      0137  0140  0004  RA           10     TELEPHONE EXTENSION
520 REM      0141  0141  0001  RA           11     RESERVED SPACE
520 REM 005  0278  0347  0070  RA           C3$    REMARKS
520 REM
520 WRITE #0,I2(4);C1,C2
530 MAT WRITE #0,I2(4),6;A
540 MAT WRITE #0,I2(4),14;M
550 WRITE #0,I2(4),22;C1$[1,256],C3$[1,70]
```

Continues on next page.

```
560 REM
560 REM KEY CONSTRUCT: RACUSTF1 - CUSTOMER NUMBER
560 REM
560 REM FLD  BGNS  ENDS  SIZE   SYSTEM   ELEMENT  DESCRIPTION FOR ELEMENT
560 REM ---  ----  ----  ----  --------  -------  ------------------------
560 REM 001  0001  0006  0006  RA          C1     CUSTOMER NUMBER
560 REM 002  0007  0009  0003  RA          C2     TERRITORY
560 REM
560 LET I1$=" ",I1$
570 LET I$=C1 USING "######"
580 LET I$=I$[1,6],C2 USING "###"
590 REM
590 LET I2(4)=I2(4)
600 LET I2(6)=0
610 SEARCH #0,4,1;I$,I2(4),I2(5)
620 IF I2(5)<>2 GOTO 650
630 SEARCH #0,7,1;I$,I2(4),I2(6)
640 SEARCH #0,4,1;I$,I2(4),I2(5)
650 ON NOT I2(5) GOTO 700
660 LET I2(7)=LEN I$
670 IF I2(7)>34 LET I2(7)=34
680 PRINT @09,23;'CLRB'"CUSTOMER NUMBER ''";
690 PRINT I$[1,I2(7)]"'' WAS NOT INSERTED";
700 ON NOT I2(5)+1 GOTO 10,720
710 REM
710 REMSTOP;MACRO;PUT,RA,RACUSTF,RACUSTF1,RACUSTF1

720 REM ---> CONTINUATION OF PROCESSING
```

5.3.6 UPDATE MACRO

An Update Macro is expanded into source code which locates a matching key in an index, and performs a WRITE to the corresponding data record. The generated source code is constructed, based upon macro line parameters and the corresponding Dictionary information.

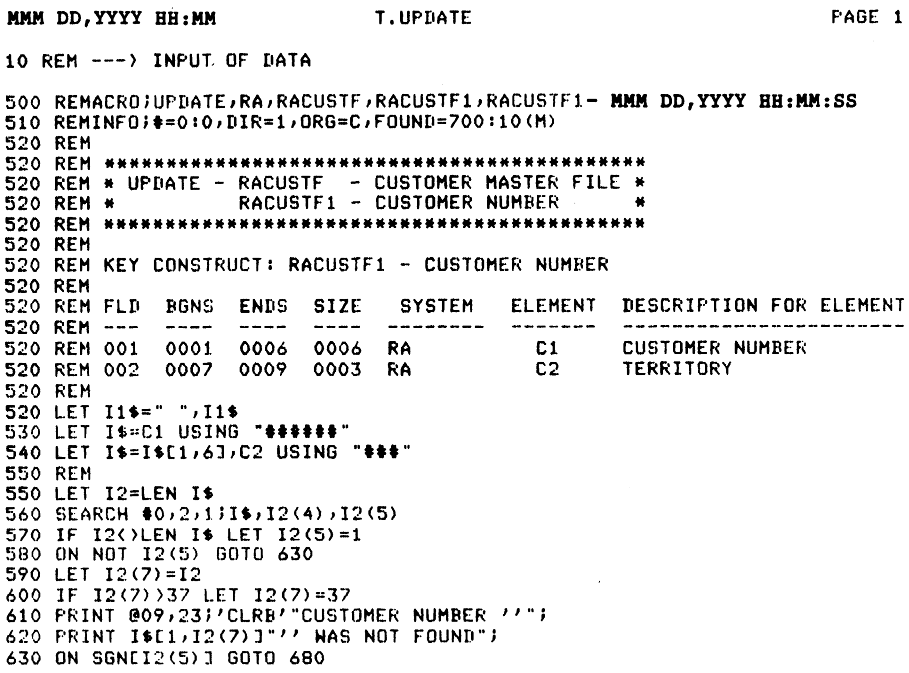
The command line format for the Update Macro is:

nnnn REMACRO;UPDATE,SYSTEM,DATAFILE,INDEXFILE(,Parameters)
nnnn REMINFO;CHANNELS,DIRECTORY,ORG=(,Parameters)

Command line entries unique to the Update Macro are:

- Macro Type - UPDATE
- File Organization - ORG=
- Parameters - (see Tables 5-5, 5-6)

A sample Update Macro and its expanded code are shown below.



```
MMM DD,YYYY HH:MM            T.UPDATE                           PAGE 1

10 REM ---> INPUT OF DATA

500 REMACRO;UPDATE,RA,RACUSTF,RACUSTF1,RACUSTF1- MMM DD,YYYY HH:MM:SS
510 REMINFO;#=0:0,DIR=1,ORG=C,FOUND=700:10(M)
520 REM
520 REM ********************************************
520 REM * UPDATE - RACUSTF  - CUSTOMER MASTER FILE *
520 REM *          RACUSTF1 - CUSTOMER NUMBER      *
520 REM ********************************************
520 REM
520 REM KEY CONSTRUCT: RACUSTF1 - CUSTOMER NUMBER
520 REM
520 REM FLD  BGNS  ENDS  SIZE   SYSTEM   ELEMENT  DESCRIPTION FOR ELEMENT
520 REM ---  ----  ----  ----  --------  -------  -----------------------
520 REM 001  0001  0006  0006  RA          C1     CUSTOMER NUMBER
520 REM 002  0007  0009  0003  RA          C2     TERRITORY
520 REM
520 LET I1$=" ",I1$
530 LET I$=C1 USING "######"
540 LET I$=I$[1,6],C2 USING "###"
550 REM
550 LET I2=LEN I$
560 SEARCH #0,2,1;I$,I2(4),I2(5)
570 IF I2<>LEN I$ LET I2(5)=1
580 ON NOT I2(5) GOTO 630
590 LET I2(7)=I2
600 IF I2(7)>37 LET I2(7)=37
610 PRINT @09,23;'CLRB'"CUSTOMER NUMBER ''";
620 PRINT I$[1,I2(7)]"'' WAS NOT FOUND";
630 ON SGN[I2(5)] GOTO 680
```

Continues on next page.

```
640 REM
640 REM FLD  BGNS  ENDS  SIZE   SYSTEM   ELEMENT  DESCRIPTION FOR ELEMENT
640 REM ---  ----  ----  ----  --------  -------  -----------------------
640 REM 000  0000  0003  0004  RA          C1     CUSTOMER NUMBER
640 REM 001  0004  0005  0002  RA          C2     TERRITORY
640 REM 002  0006  0013  0008  RA          A      ACCOUNT STATUS
640 REM      0006  0007  0002  RA          0000   ACCOUNT CODE
640 REM      0008  0009  0002  RA          0001   ACCOUNT NUMBER
640 REM 003  0014  0021  0008  RA          M      SALESMAN STATUS
640 REM      0014  0015  0002  RA          00X00  SALES CODE
640 REM      0016  0017  0002  RA          00X01  SALESMAN
640 REM 004  0022  0277  0256  RA          C1$    CUSTOMER SHIPPING INFORMA
640 REM      0022  0053  0032  RA          1      SHIP TO NAME
640 REM      0054  0085  0032  RA          2      SHIP TO ADDRESS
640 REM      0086  0099  0014  RA          3      SHIP TO CITY
640 REM      0100  0101  0002  RA          4      SHIP TO STATE
640 REM      0102  0106  0005  RA          5      SHIP TO ZIP CODE
640 REM      0107  0126  0020  RA          6      SHIP TO CONTACT
640 REM      0127  0129  0003  RA          7      AREA CODE
640 REM      0130  0132  0003  RA          8      TELEPHONE PREFIX
640 REM      0133  0136  0004  RA          9      TELEPHONE SUFFIX
640 REM      0137  0140  0004  RA          10     TELEPHONE EXTENSION
640 REM      0141  0141  0001  RA          11     RESERVED SPACE
640 REM 005  0278  0347  0070  RA          C3$    REMARKS
640 REM
640 WRITE #0,I2(4);C1,C2
650 MAT WRITE #0,I2(4),6;A
660 MAT WRITE #0,I2(4),14;M
670 WRITE #0,I2(4),22;C1$[1,256],C3$[1,70]
680 ON NOT I2(5)+1 GOTO 10,700
690 REM
690 REMSTOP;MACRO;UPDATE,RA,RACUSTF,RACUSTF1,RACUSTF1

700 REM ---> CONTINUATION OF PROCESSING
```

5.3.7 LINK MACRO

The Link Macro is expanded into source code to extract an available location from a data-record Free List, write a data record, and place the key in a directory.

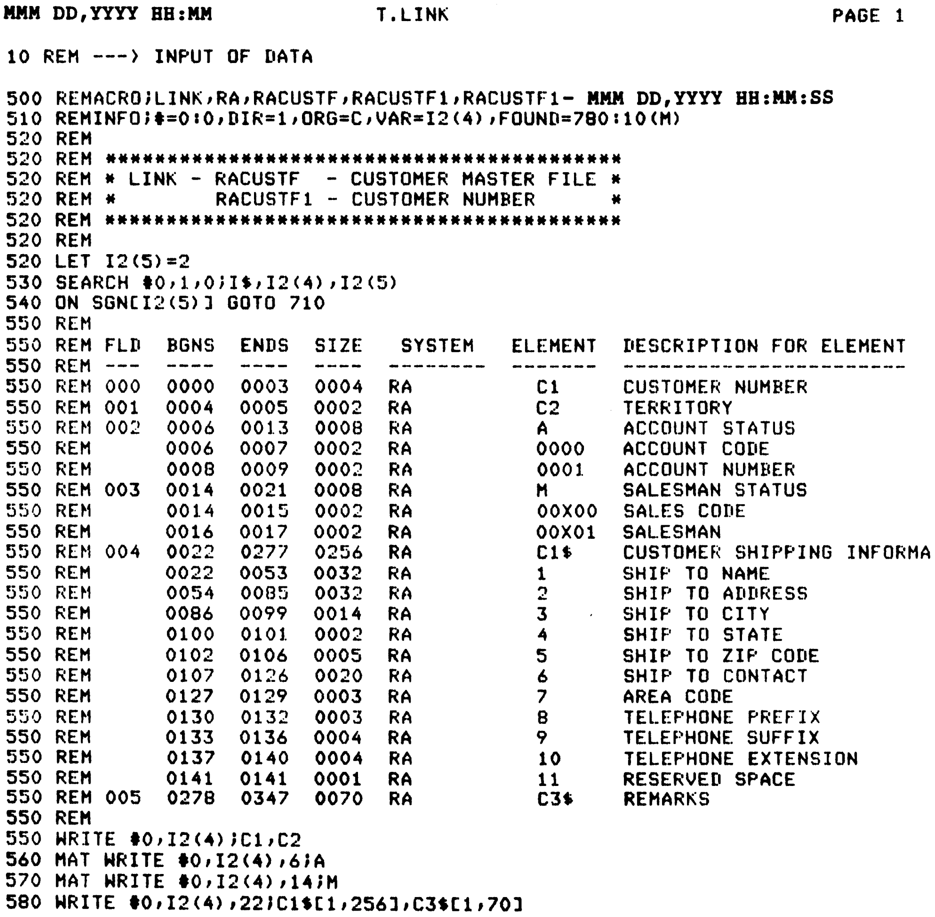
The command line format for the Link Macro is:

nnnn REMACRO;LINK,SYSTEM,DATAFILE,INDEXFILE(,Parameters)
nnnn REMINFO;CHANNELS,DIRECTORY,ORG=,VAR=(,Parameters)

Command line entries unique to the Link Macro are:

- Macro Type - LINK
- File Organization - ORG=
- Access Variable - VAR=
- Parameters - (see Tables 5-5, 5-6)

A sample Link Macro and its expanded code are shown below.



```
MMM DD,YYYY HH:MM              T.LINK                              PAGE 1

10 REM ---> INPUT OF DATA

500 REMACRO;LINK,RA,RACUSTF,RACUSTF1,RACUSTF1- MMM DD,YYYY HH:MM:SS
510 REMINFO;#=0:0,DIR=1,ORG=C,VAR=I2(4),FOUND=780:10(M)
520 REM
520 REM ******************************************
520 REM * LINK - RACUSTF  - CUSTOMER MASTER FILE *
520 REM *        RACUSTF1 - CUSTOMER NUMBER      *
520 REM ******************************************
520 REM
520 LET I2(5)=2
530 SEARCH #0,1,0;I$,I2(4),I2(5)
540 ON SGN[I2(5)] GOTO 710
550 REM
550 REM FLD  BGNS  ENDS  SIZE   SYSTEM    ELEMENT  DESCRIPTION FOR ELEMENT
550 REM ---  ----  ----  ----  --------   -------  ------------------------
550 REM 000  0000  0003  0004  RA           C1     CUSTOMER NUMBER
550 REM 001  0004  0005  0002  RA           C2     TERRITORY
550 REM 002  0006  0013  0008  RA           A      ACCOUNT STATUS
550 REM      0006  0007  0002  RA           0000   ACCOUNT CODE
550 REM      0008  0009  0002  RA           0001   ACCOUNT NUMBER
550 REM 003  0014  0021  0008  RA           M      SALESMAN STATUS
550 REM      0014  0015  0002  RA           00X00  SALES CODE
550 REM      0016  0017  0002  RA           00X01  SALESMAN
550 REM 004  0022  0277  0256  RA           C1$    CUSTOMER SHIPPING INFORMA
550 REM      0022  0053  0032  RA           1      SHIP TO NAME
550 REM      0054  0085  0032  RA           2      SHIP TO ADDRESS
550 REM      0086  0099  0014  RA           3      SHIP TO CITY
550 REM      0100  0101  0002  RA           4      SHIP TO STATE
550 REM      0102  0106  0005  RA           5      SHIP TO ZIP CODE
550 REM      0107  0126  0020  RA           6      SHIP TO CONTACT
550 REM      0127  0129  0003  RA           7      AREA CODE
550 REM      0130  0132  0003  RA           8      TELEPHONE PREFIX
550 REM      0133  0136  0004  RA           9      TELEPHONE SUFFIX
550 REM      0137  0140  0004  RA           10     TELEPHONE EXTENSION
550 REM      0141  0141  0001  RA           11     RESERVED SPACE
550 REM 005  0278  0347  0070  RA           C3$    REMARKS
550 REM
550 WRITE #0,I2(4);C1,C2
560 MAT WRITE #0,I2(4),6;A
570 MAT WRITE #0,I2(4),14;M
580 WRITE #0,I2(4),22;C1$[1,256],C3$[1,70]
```

Continues on next page.

```
590 REM
590 REM KEY CONSTRUCT: RACUSTF1 - CUSTOMER NUMBER
590 REM
590 REM FLD  BGNS  ENDS  SIZE   SYSTEM    ELEMENT  DESCRIPTION FOR ELEMENT
590 REM ---  ----  ----  ----  --------   -------  ------------------------
590 REM 001  0001  0006  0006  RA           C1     CUSTOMER NUMBER
590 REM 002  0007  0009  0003  RA           C2     TERRITORY
590 REM
590 LET I1$=" ",I1$
600 LET I$=C1 USING "######"
610 LET I$=I$[1,6],C2 USING "###"
620 REM
620 LET I2(4)=I2(4)
630 LET I2(6)=0
640 SEARCH #0,4,1;I$,I2(4),I2(5)
650 IF I2(5)<>2 GOTO 680
660 SEARCH #0,7,1;I$,I2(4),I2(6)
670 SEARCH #0,4,1;I$,I2(4),I2(5)
680 ON NOT I2(5) GOTO 710
690 LET I2(6)=3
700 SEARCH #0,1,0;I$,I2(4),I2(6)
710 ON NOT I2(5) GOTO 760
720 LET I2(7)=LEN I$
730 IF I2(7)>34 LET I2(7)=34
740 PRINT @09,23;'CLRB'"CUSTOMER NUMBER ''";
750 PRINT I$[1,I2(7)]"'' WAS NOT INSERTED";
760 ON NOT I2(5)+1 GOTO 10,780
770 REM
770 REMSTOP;MACRO;LINK,RA,RACUSTF,RACUSTF1,RACUSTF1

780 REM ---> CONTINUATION OF PROCESSING
```

5.3.8 FREE MACRO

A Free Macro is expanded into source code, which removes a key from a directory, and places the vacated location on a data record Free List.

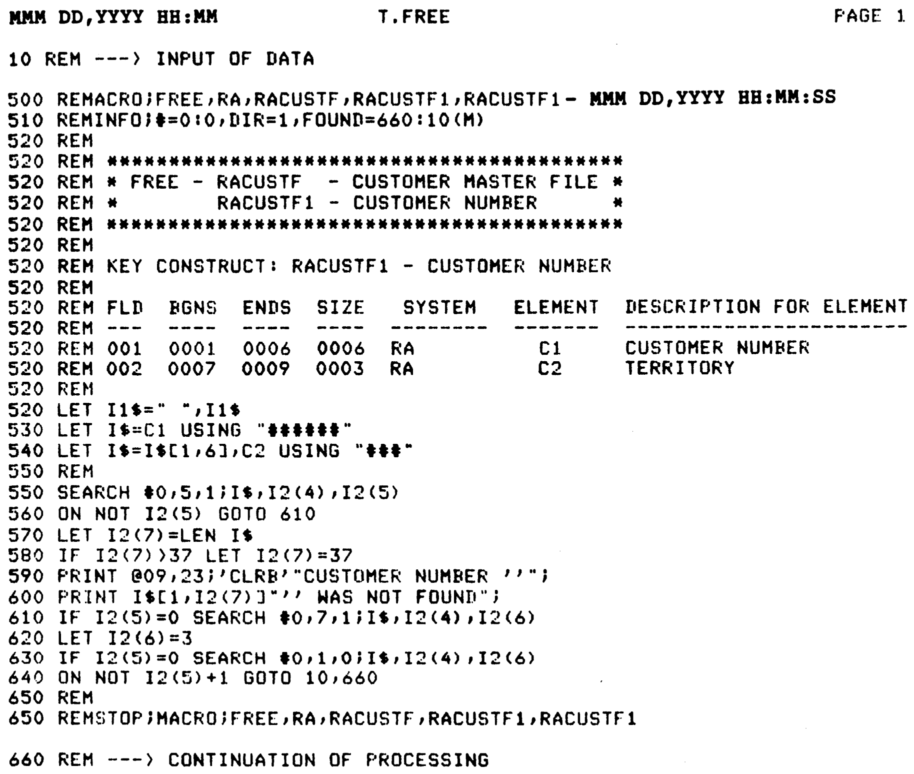
The command line format for the Free Macro is:

nnnn REMACRO;FREE,SYSTEM,DATAFILE,INDEXFILE(,Parameters)
nnnn REMINFO;CHANNELS,DIRECTORY(,Parameters)

Command line entries unique to the Free Macro are:

- Macro Type - FREE
- Parameters - (see Tables 5-5, 5-6)

A sample Free Macro and its expanded code are shown below.



```
MMM DD,YYYY HH:MM            T.FREE                              PAGE 1

10 REM ---> INPUT OF DATA

500 REMACRO;FREE,RA,RACUSTF,RACUSTF1,RACUSTF1- MMM DD,YYYY HH:MM:SS
510 REMINFO;#=0:0,DIR=1,FOUND=660:10(M)
520 REM
520 REM ******************************************
520 REM * FREE - RACUSTF  - CUSTOMER MASTER FILE *
520 REM *        RACUSTF1 - CUSTOMER NUMBER      *
520 REM ******************************************
520 REM
520 REM KEY CONSTRUCT: RACUSTF1 - CUSTOMER NUMBER
520 REM
520 REM FLD  BGNS  ENDS  SIZE   SYSTEM   ELEMENT  DESCRIPTION FOR ELEMENT
520 REM ---  ----  ----  ----  --------  -------  -----------------------
520 REM 001  0001  0006  0006  RA           C1     CUSTOMER NUMBER
520 REM 002  0007  0009  0003  RA           C2     TERRITORY
520 REM
520 LET I1$=" ",I1$
530 LET I$=C1 USING "######"
540 LET I$=I$[1,6],C2 USING "###"
550 REM
550 SEARCH #0,5,1;I$,I2(4),I2(5)
560 ON NOT I2(5) GOTO 610
570 LET I2(7)=LEN I$
580 IF I2(7)>37 LET I2(7)=37
590 PRINT @09,23;'CLRB'"CUSTOMER NUMBER ''";
600 PRINT I$[1,I2(7)]"'' WAS NOT FOUND";
610 IF I2(5)=0 SEARCH #0,7,1;I$,I2(4),I2(6)
620 LET I2(6)=3
630 IF I2(5)=0 SEARCH #0,1,0;I$,I2(4),I2(6)
640 ON NOT I2(5)+1 GOTO 10,660
650 REM
650 REMSTOP;MACRO;FREE,RA,RACUSTF,RACUSTF1,RACUSTF1

660 REM ---> CONTINUATION OF PROCESSING
```

5.4 LEVEL 6 MACRO COMMANDS

The Level 6 macro commands are listed below, with the corresponding manual section:

Type	Function	Section
HEADER	Report Header Format	5.4.1
DETAIL	Report Detail Format	5.4.2
PROGRAM	Dimension/Assign Variables	5.4.3

Level 6 macros are single-line commands which use the same format up to and including the system name. The standard Level 6 macro command line format is:

```
nnnn REMACRO;TYPE,SYSTEM,CONTROLS(,PARAMETERS)
  |     |      |     |       |         |
  |     |      |     |       |         └Optional parameters
  |     |      |     |       |
  |     |      |     |       └Required control items, which
  |     |      |     |        vary among macro types
  |     |      |     |
  |     |      |     └Associated system name
  |     |      |
  |     |      └Macro type specification
  |     |
  |     └Macro notation (REM + MACRO), followed by a semi-colon
  |
  └BASIC line number up to four digits.  Not required for macros
   inserted within labeled source code (see 8.1).
```

Macro command line entries must adhere to this order and format. A semi-colon must follow the REMACRO notation, and subsequent command line entries must be separated by commas.

See Table 5-4 for an explanation of macro command line entries.

See Table 5-7 for Level 6 macro command set-up.

TABLE 5-7. MACRO COMMAND SET-UP - LEVEL 6 MACROS

Command Line Entry*	Macro Type		
	HEADER	DETAIL	PROGRAM
Line Number**	R	R	R
REMACRO; Notation	R	R	R
Macro Type	R	R	R
System Name	R	R	R
Program Name			R
Format Number	R	R	
#= Channel Number	R	R	
L=nnnn Set Line nnnn	O	O	O
G=nnnn Set GOSUB Line nnnn	O	O	
Skip=nn Skip nn Lines	O	O	

*Required Level 6 macro command line components must be entered in the order represented above, from top to bottom.

**A BASIC line number is not required for macros inserted within labeled source code (see 8.1).

5.4.1 HEADER MACRO

A Header Macro is expanded into source code which prints a header format for use in a report. Code is generated by referencing the Dictionary for previously-specified (see 4.3.2) header format information.

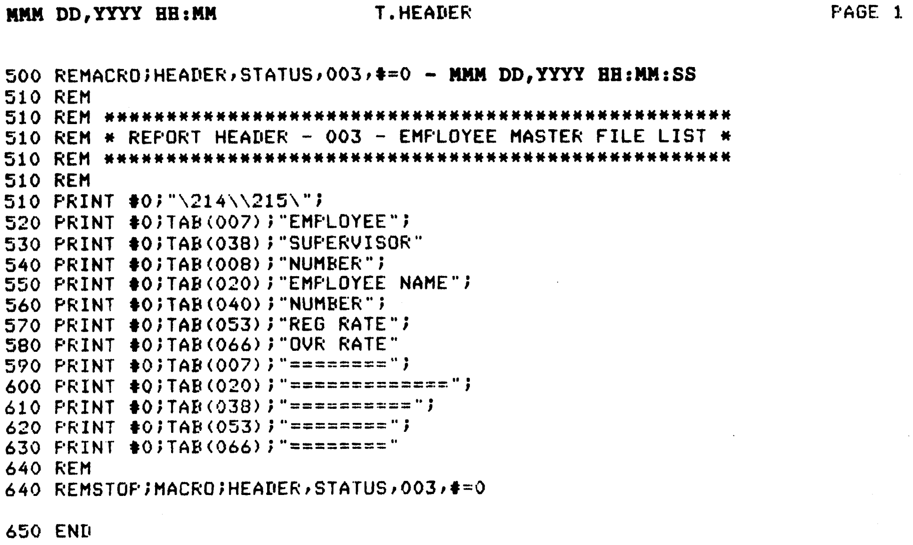
The command line format for the Header Macro is:

nnnn REMACRO;HEADER,SYSTEM,nnn,#=(,Parameters)

Command line entries unique to the Header Macro are:

- Macro Type - HEADER
- Header Format Number - the number of the header format in the Dictionary
- Channel Number - Channel number used for the report output
- Optional parameters - (see Table 5-7)

A sample Header Macro and its expanded code are shown below.



```
MMM DD,YYYY HH:MM              T.HEADER                          PAGE 1

500 REMACRO;HEADER,STATUS,003,#=0 - MMM DD,YYYY HH:MM:SS
510 REM
510 REM ****************************************************
510 REM * REPORT HEADER - 003 - EMPLOYEE MASTER FILE LIST *
510 REM ****************************************************
510 REM
510 PRINT #0;"\214\\215\";
520 PRINT #0;TAB(007);"EMPLOYEE";
530 PRINT #0;TAB(038);"SUPERVISOR"
540 PRINT #0;TAB(008);"NUMBER";
550 PRINT #0;TAB(020);"EMPLOYEE NAME";
560 PRINT #0;TAB(040);"NUMBER";
570 PRINT #0;TAB(053);"REG RATE";
580 PRINT #0;TAB(066);"OVR RATE"
590 PRINT #0;TAB(007);"========";
600 PRINT #0;TAB(020);"=============";
610 PRINT #0;TAB(038);"==========";
620 PRINT #0;TAB(053);"========";
630 PRINT #0;TAB(066);"========"
640 REM
640 REMSTOP;MACRO;HEADER,STATUS,003,#=0

650 END
```

5.4.2 DETAIL MACRO

A Detail Macro is expanded into source code which prints a detail format for use in a report. Code is generated by referencing the Dictionary for previously-specified (see 4.3.3) detail format information.

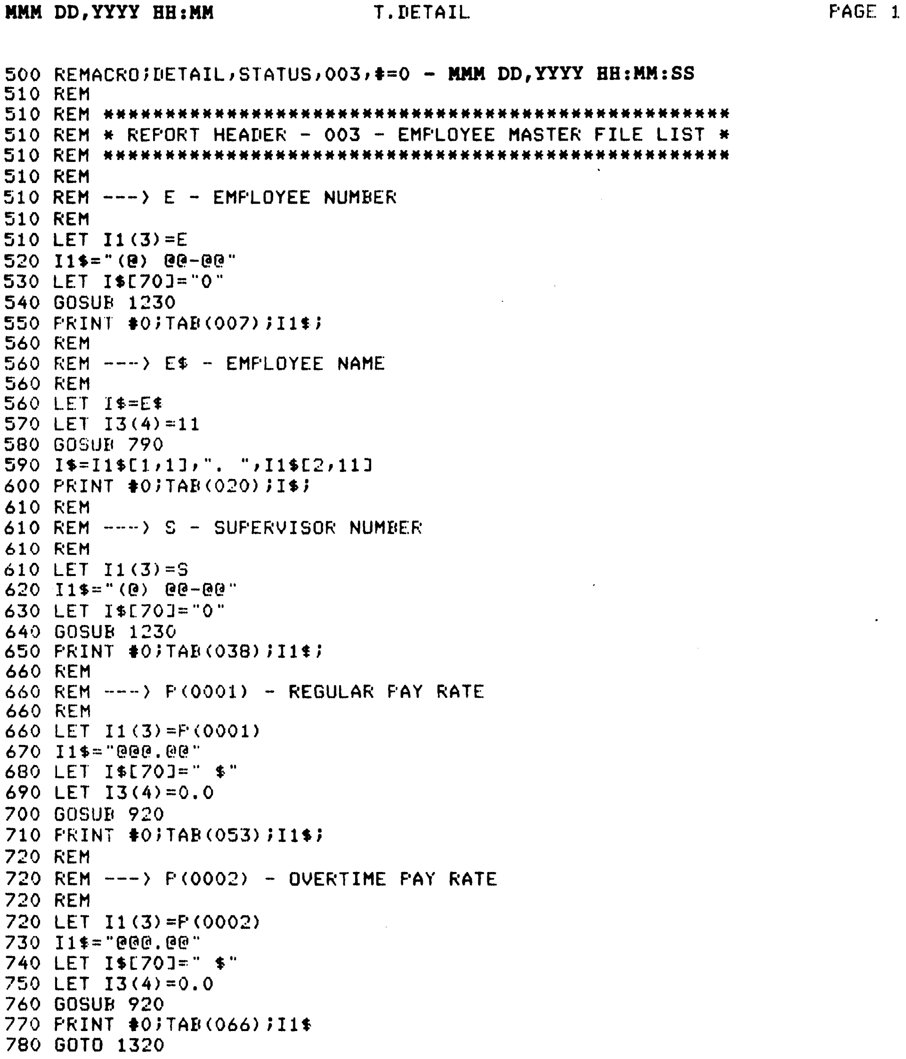
The command line format for the Detail Macro is:

nnnn REMACRO;DETAIL,SYSTEM,nnn,#=(,Parameters)

Command line entries unique to the Detail Macro are:

- Macro Type - DETAIL
- Detail Format Number - the number of the detail format in the Dictionary
- Channel Number - Channel number used for the report output
- Optional parameters - (see Table 5-7)

A sample Detail Macro and its expanded code are shown below.



```
MMM DD,YYYY HH:MM            T.DETAIL                          PAGE 1

500 REMACRO;DETAIL,STATUS,003,#=0 - MMM DD,YYYY HH:MM:SS
510 REM
510 REM ***************************************************
510 REM * REPORT HEADER - 003 - EMPLOYEE MASTER FILE LIST *
510 REM ***************************************************
510 REM
510 REM ---> E - EMPLOYEE NUMBER
510 REM
510 LET I1(3)=E
520 I1$="(@) @@-@@"
530 LET I$[70]="0"
540 GOSUB 1230
550 PRINT #0;TAB(007);I1$;
560 REM
560 REM ---> E$ - EMPLOYEE NAME
560 REM
560 LET I$=E$
570 LET I3(4)=11
580 GOSUB 790
590 I$=I1$[1,1],". ",I1$[2,11]
600 PRINT #0;TAB(020);I$;
610 REM
610 REM ----> S - SUPERVISOR NUMBER
610 REM
610 LET I1(3)=S
620 I1$="(@) @@-@@"
630 LET I$[70]="0"
640 GOSUB 1230
650 PRINT #0;TAB(038);I1$;
660 REM
660 REM ---> P(0001) - REGULAR PAY RATE
660 REM
660 LET I1(3)=P(0001)
670 I1$="@@@.@@"
680 LET I$[70]=" $"
690 LET I3(4)=0.0
700 GOSUB 920
710 PRINT #0;TAB(053);I1$;
720 REM
720 REM ---> P(0002) - OVERTIME PAY RATE
720 REM
720 LET I1(3)=P(0002)
730 I1$="@@@.@@"
740 LET I$[70]=" $"
750 LET I3(4)=0.0
760 GOSUB 920
770 PRINT #0;TAB(066);I1$
780 GOTO 1320
```

```
790 REM
790 REM ---> ROUTINE:  JUSTIFY AN ALPHANUMERIC VARIABLE.
790 REM
790 LET I$[70]=" "
800 IF LEN I$>I3(4) LET I$=I$[1,I3(4)]
810 LET I1$=""
820 IF I$<>"" LET I1$=I$[70],I1$
830 IF I$<>"" LET I1$=I1$[1,I3(4)]
840 IF FRA[I3(4)]=0 FOR I3(5)=1 TO LEN I$
850 IF FRA[I3(4)]>0 FOR I3(5)=LEN I$ TO 1 STEP -1
860   IF I$[I3(5),I3(5)]<>" " GOTO 880
870 NEXT I3(5)
880 IF FRA[I3(4)]=0 IF I3(5)<=LEN I$ LET I1$[1,LEN I$-I3(5)+1]=I$[I3(5)]
890 IF FRA[I3(4)]>0 IF I3(5) LET I1$[LEN I1$-I3(5)+1]=I$[1,I3(5)]
900 RETURN

910 REM
910 REM ---> ROUTINE:  TRUNCATE/ROUND NUMBER AND PLACE IN WORK STRING.
910 REM
910 LET I$[70]="  "
920 LET I1=10^[SGN I3(4)*INT ABS I3(4)]
930 IF FRA I3(4)=0 IF FRA ABS[I1(3)*I1]>=.5 LET I1(3)=I1(3)+SGN I1(3)*.5/I1
940 IF FRA ABS I3(4)<=.1 LET I1(3)=I1(3)-FRA[I1(3)*I1]/I1
950 LET I$=I1(3) USING " -------------#.##############"
960 REM
960 REM ---> PLACE FLOATING CHARACTER AT START OF NUMBER.
960 REM
960 IF I$[71]=" " GOTO 1010
970 FOR I3=2 TO LEN I$
980   IF I$[I3,I3]<>" " GOTO 1000
990 NEXT I3
1000 IF I3<=LEN I$ LET I$[I3-1,I3-1]=I$[71]
1010 REM
1010 REM ---> MOVE NUMBER INTO CORRECT AREA OF FORMAT MASK.
1010 REM
1010 LET I3(4)=0
1020 LET I3(5)=0
1030 FOR I3=LEN I1$ TO 1 STEP -1
1040   IF I1$[I3,I3]="." LET I3(5)=I3
1050   IF I1$[I3,I3]="@" IF I3(5)=0 LET I3(5)=I3+1
1060 IF I3(5)-I3 NEXT I3
1070 FOR I3=LEN I1$ TO 1 STEP -1
1080   LET I1=I3+I3(4)-I3(5)+16
1090   IF I1$[I3,I3]<>"@" GOTO 1140
1100   IF ABS[16-I1]<=14 IF I$[I1,I1]<>" " LET I1$[I3,I3]=I$[I1]
1110   IF I1$[I3,I3]="@" IF I3<I3(5) LET I1$[I3,I3]=I$[70]
1120   IF I1$[I3,I3]="@" IF I3>I3(5) LET I1$[I3,I3]="0"
1130   GOTO 1200
1140   IF I1$[I3,I3]<>"," GOTO 1200
1150   IF I1>=1 IF I$[I1,I1]>="0" IF I$[I1,I1]<="9" GOTO 1190
1160   IF I1>=1 IF I$[I1,I1]=I$[71] IF I$[71]<>" " GOTO 1100
1170   IF I1>=1 IF I$[I1,I1]="-" GOTO 1100
1180   LET I1$[I3,I3]=I$[70]
1190   LET I3(4)=I3(4)+1
1200 NEXT I3
1210 RETURN
1220 REM
1220 REM ---> RIGHT JUSTIFY A NUMERIC WITHIN THE FORMAT MASK.
1220 REM
1220 LET I$[70]=" "
1230 LET I$=I1(3)
1240 LET I3(4)=LEN I$-1
1250 FOR I3=LEN I1$ TO 1 STEP -1
1260   IF I1$[I3,I3]<>"@" GOTO 1300
1270   IF I3(4)>=2-NOT[SGN I1(3)+1] LET I1$[I3,I3]=I$[I3(4)]
1280   IF I3(4)<2-NOT[SGN I1(3)+1] LET I1$[I3,I3]=I$[70]
1290   LET I3(4)=I3(4)-1
1300 NEXT I3
1310 RETURN


1320 REM
1320 REMSTOP;MACRO;DETAIL,STATUS,003,#=0


1330 END
```

5.4.3 PROGRAM MACRO

A Program Macro is expanded to dimension the data elements expanded in a report or an entry program. The expanded code is based upon the program's required data elements as recorded in the Dictionary. Complete documentation of the data elements is included in the generated source code.

The Program Macro does not dimension global variables (see 5.2.8).

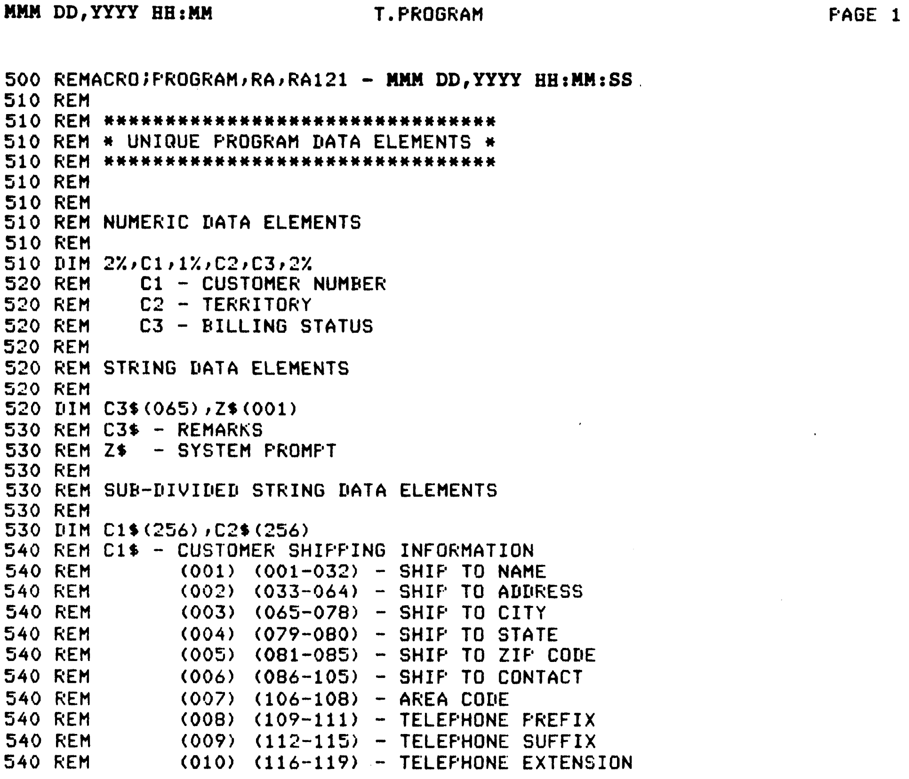
The command line format for the Program Macro is:

nnnn REMACRO;PROGRAM,SYSTEM,PROGRAM NAME(,Parameters)

Command line entries unique to the Program Macro are:

- o Macro Type - PROGRAM
- o Program Name - The name of the program, as defined to FORCE, for which the data elements are to be dimensioned
- o Optional parameters - (see Table 5-7)

A sample report Program Macro and its expanded code are shown below. (A sample entry Program Macro is shown on the following page.)



```
MMM DD,YYYY HH:MM            T.PROGRAM                         PAGE 1


500 REMACRO;PROGRAM,RA,RA121 - MMM DD,YYYY HH:MM:SS
510 REM
510 REM ********************************
510 REM * UNIQUE PROGRAM DATA ELEMENTS *
510 REM ********************************
510 REM
510 REM
510 REM NUMERIC DATA ELEMENTS
510 REM
510 DIM 2%,C1,1%,C2,C3,2%
520 REM    C1 - CUSTOMER NUMBER
520 REM    C2 - TERRITORY
520 REM    C3 - BILLING STATUS
520 REM
520 REM STRING DATA ELEMENTS
520 REM
520 DIM C3$(065),Z$(001)
530 REM C3$ - REMARKS
530 REM Z$  - SYSTEM PROMPT
530 REM
530 REM SUB-DIVIDED STRING DATA ELEMENTS
530 REM
530 DIM C1$(256),C2$(256)
540 REM C1$ - CUSTOMER SHIPPING INFORMATION
540 REM        (001) (001-032) - SHIP TO NAME
540 REM        (002) (033-064) - SHIP TO ADDRESS
540 REM        (003) (065-078) - SHIP TO CITY
540 REM        (004) (079-080) - SHIP TO STATE
540 REM        (005) (081-085) - SHIP TO ZIP CODE
540 REM        (006) (086-105) - SHIP TO CONTACT
540 REM        (007) (106-108) - AREA CODE
540 REM        (008) (109-111) - TELEPHONE PREFIX
540 REM        (009) (112-115) - TELEPHONE SUFFIX
540 REM        (010) (116-119) - TELEPHONE EXTENSION
```

Continues on next page.

```
540 REM C2$ - CUSTOMER BILLING INFORMATION
540 REM       (001) (001-032) - BILL TO NAME
540 REM       (002) (033-064) - BILL TO ADDRESS
540 REM       (003) (065-078) - BILL TO CITY
540 REM       (004) (079-080) - BILL TO STATE
540 REM       (005) (081-085) - BILL TO ZIP CODE
540 REM       (006) (086-105) - CONTACT NAME
540 REM       (007) (106-108) - AREA CODE
540 REM       (008) (109-111) - TELEPHONE PREFIX
540 REM       (009) (112-115) - TELEPHONE SUFFIX
540 REM       (010) (116-119) - TELEPHONE EXTENSION
540 REM
540 REM
540 REM
540 REMSTOP;PROGRAM,RA,RA121

550 END
```

Section 6
DATA DICTIONARY/MANAGER FUNCTIONS

The FORCE Dictionary stores and maintains system/application specifications. These specifications, which make each application unique, are retrieved from the Dictionary during source code generation. Specifications may also be retrieved by the user through FORCE documentation and interrogation functions.

FORCE manager functions control and optimize Dictionary performance. An initial FORCE manager program, FS101MAKE, establishes required Dictionary files. Re-indexing and reorganization operations maintain an efficient Dictionary structure. A purge facility enables the FORCE manager to eradicate a system from the Dictionary, and the printer-assignment program selectively designates port-to-printer output.

Section 6 is organized as follows:

6.1 DATA DICTIONARY

The FORCE Data Dictionary is a complex set of files for dynamic storage and management of system/application specifications. FORCE programs under Data Base Administration provide a user-friendly method for defining system components to the Dictionary.

Application specifications are related to a system name, which FORCE uses to catalog the information within the Dictionary. System specifications that must be defined to the Dictionary include screens, data elements, files and programs. These items represent the unique features and operations of each application. System specifications are validated against current Dictionary information and IRIS regulations.

During source code generation, FORCE language processing modules access the Dictionary and incorporate system specifications into the generated code.

Generated program source code exists independently of the FORCE Dictionary and is executable on any IRIS system.

The FORCE Dictionary retains system specifications after source code is generated (unless purged by the manager, see 6.2.5). More than thirty reports on any or all system components may be obtained through the FORCE documentation facility. System specifications may be shared or copied between systems within the Dictionary (e.g., two programs from different systems may employ the same screen displays or data elements).

To modify a program, the user changes the affected Dictionary specifications and regenerates the program. The original labeled program source code is overwritten on disk by a new program which incorporates the required modifications. This method of program modification utilizes the Dictionary to its fullest advantage.

For example, a data element for a zip code may need to be expanded from 5 to 9 digits. By modifying the data element's Dictionary attributes (maximum size, range checks, etc.) and regenerating the program, the user ensures that the data element is dimensioned and verified properly throughout the program. Additionally, the updated system information in the Dictionary reflects current application specifications.

6.1.1 DICTIONARY ACCESS

System components in the Dictionary may be accessed by any FORCE user. There are no established user privilege levels within the Dictionary. Application specifications (screens, files, data elements) may be shared or copied between systems. One FORCE function allows the user to copy an entire set of specifications for a program (see 1.5.6).

LOCK-OUTS

To ensure the integrity of Dictionary information, FORCE disallows simultaneous unit level access by multiple users. A user experiences a lock-out when attempting to access a Dictionary record that is currently being accessed by another user. During a lock-out, the invoked FORCE program waits for the first user to complete processing before enabling access by another user.

This does not mean that two users cannot work simultaneously within the same system. One user could create a screen display while the other is defining a data element. But both users cannot simultaneously define the same record layout.

GENERATION/REGENERATION

Dictionary specifications are incorporated into the source code during program generation (see 2.2 or 4.4). Once generated, the source code requires no further association with FORCE or the Dictionary. It may be loaded, executed and/or modified like any Business BASIC program.

However, the Dictionary provides an opportunity for efficient regeneration to accommodate program enhancement and modification. Because all program specifications remain in the Dictionary after generation, the task of any future development is greatly reduced. Substantial program alterations may be implemented simply by changing Dictionary information and regenerating the source code.

Regenerated source code reflects any Dictionary modifications and overwrites the current labeled source code on disk. Any manual modifications to a program are lost when a program is regenerated. Manual program modifications should be carefully logged before regeneration, so that they can be implemented within the regenerated program.

Source code may also be enhanced by the insertion of macros (see 5.1.2).

6.1.2 DICTIONARY FILES

FORCE Dictionary files must be created on Logical Unit 5. There are ten FORCE Dictionary files, each with a contiguous organization and a protection of <11>. Eight of the Dictionary files are indexes.

The FORCE logo screen (see 7.1) displays a calculation which indicates the percentage of utilized Dictionary file space. This figure increases with the number of applications defined to the Dictionary. When the Dictionary fills to 85% of its capacity, the manager must enlarge the files. Table 6-1 provides guidelines for size computation of enlarged Dictionary files.

TABLE 6-1. DICTIONARY FILE INFORMATION

Filename	Description	Key Length (Words/Bytes)	Size Computation
FS10FHD	Display Index	6/12	systems x 100
FS10FHE	Element Index	6/12	systems x 90
FS10FHF	File Index	9/18	systems x 25
FS10FHM	Matrix Index	8/16	systems x 90
FS10FHP	Program Index	9/18	systems x 85
FS10FHR	Report Index	6/12	systems x 50
FS10FHS	System Index	4/8	systems x 1
FS10FHX	Synonym X-Ref.	9/18	systems x 90
FS10FH - Header Master File Retrieval Information Block (RIB)			
FS10FD - Detail Master File Retrieval Information Block (RIB)			

Before enlarging any FORCE Dictionary file, back up all current files (see 6.2.3).

The Size Computation column indicates a recommended number of additional records per system. Multiply the given figure by the number of additional systems (complete applications) to be added to the Dictionary.

To increase the size of the data files (FS10FH and FS10FD), use the COPY processor to recreate the files on another logical unit. Specify the required parameters to enlarge the files. Then kill the old files and copy the new, enlarged files to Logical Unit 5.

To increase the size of the FORCE index files, first kill the current files. Then use an index file creation program, such as BUILDXF, to create new index files with the required names and sizes. Then run the FORCE reorganization function (see 6.2.3) to re-establish the original indexed information in the new files.

6.2 MANAGER FUNCTIONS

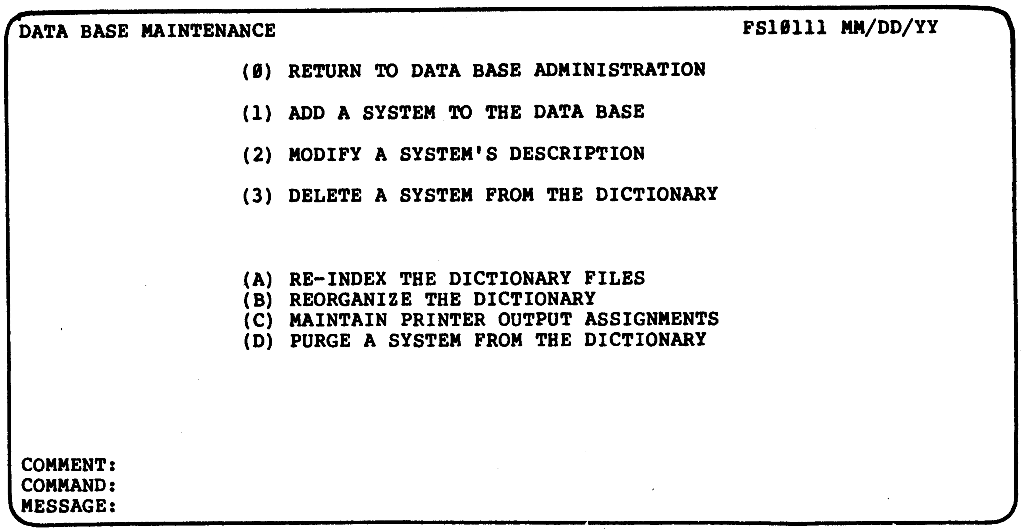
There are four standard FORCE manager functions. They are included on the Data Base Maintenance menu (1.1) as functions A through D. These FORCE manager functions appear only when a user has signed onto the system under the manager I.D. FORCE recognizes the system manager account as the FORCE manager I.D.

FS101MAKE is the initial manager program which must be executed for a new FORCE system. This program establishes Dictionary files.

FORCE manager functions control and optimize Dictionary performance. Re-indexing and reorganization operations maintain an efficient Dictionary structure. The purge facility removes all components of a system from the Dictionary, and the printer-assignment program selectively directs port-to-printer output.

All FORCE users should be logged off before executing any manager functions.

Below is the Data Base Maintenance menu for the FORCE manager.



```
DATA BASE MAINTENANCE                                      FS1Ø111 MM/DD/YY

                     (Ø) RETURN TO DATA BASE ADMINISTRATION

                     (1) ADD A SYSTEM TO THE DATA BASE

                     (2) MODIFY A SYSTEM'S DESCRIPTION

                     (3) DELETE A SYSTEM FROM THE DICTIONARY


                     (A) RE-INDEX THE DICTIONARY FILES
                     (B) REORGANIZE THE DICTIONARY
                     (C) MAINTAIN PRINTER OUTPUT ASSIGNMENTS
                     (D) PURGE A SYSTEM FROM THE DICTIONARY




COMMENT:
COMMAND:
MESSAGE:
```

For further information on manager functions, see:

6.2.1 FS101MAKE
6.2.2 Re-index the Dictionary Files
6.2.3 Reorganize the Dictionary
6.2.4 Maintain Printer Output Assignments
6.2.5 Purge a System from the Dictionary

6.2.1 FS101MAKE

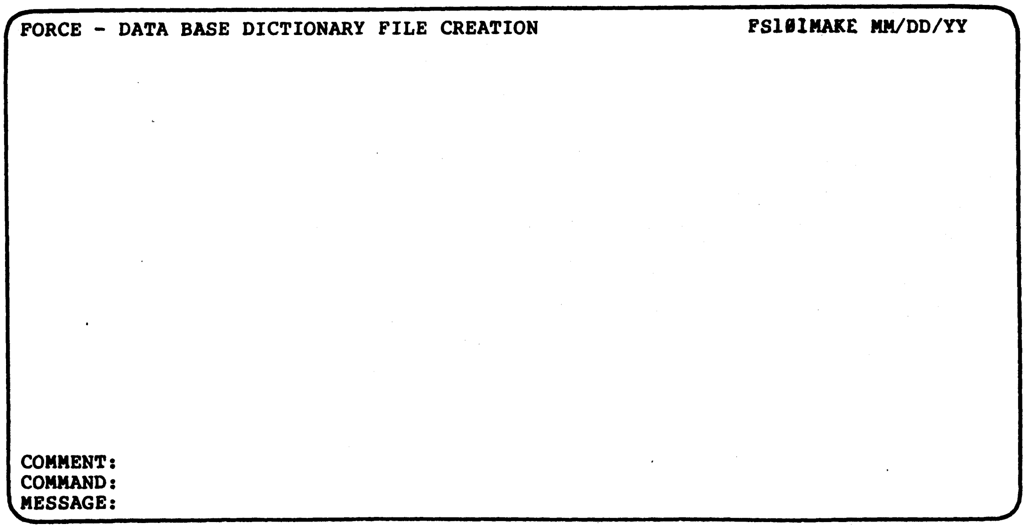
PURPOSE: FS101MAKE is a program that automatically builds the required Dictionary files for a new FORCE installation. FORCE does not function without a Dictionary, which must be located on Logical Unit 5.

This program represents the first step in installing FORCE. It is automatically initiated by attempting to invoke FORCE (see 7.1) before the Dictionary has been established.

FS101MAKE may also be used to delete an entire Dictionary and establish a new one.

PROCEDURE: FS101MAKE deletes any current Dictionary files before establishing new ones. If the manager wants to create a new Dictionary, the program FS101MAKE should be invoked directly from SCOPE. Once executed, the original Dictionary is deleted.

To establish a Dictionary for a new FORCE installation, use the following procedure: Sign onto the system under the manager's account. Enter 5/FORCE while in SCOPE. FS101MAKE is automatically invoked and the following screen is displayed.



```
FORCE - DATA BASE DICTIONARY FILE CREATION                FS1Ø1MAKE MM/DD/YY



COMMENT:
COMMAND:
MESSAGE:
```

Execution of this function requires that the manager respond to two prompts on the Comment Line. The first prompt ensures that the manager understands the invoked program. It reads:

DO YOU UNDERSTAND WHAT YOU ARE ABOUT TO DO?

Enter (Y)es or (N)o.

FORCE then asks for the number of systems that the manager plans to put into the Dictionary. Each system represents a complete application. This entry is used by FORCE to compute the required size of the Dictionary files (see Table 6-1).

For a standard Dictionary specify three systems. Consider that a ten-system Dictionary requires allocation of seven megabytes of contiguous disk for the Detail File (FS10FD) alone.

Enter the number of systems to be put into the Dictionary.

FORCE then begins building the Dictionary files. If there is not sufficient disk space on LU 5, FORCE displays the appropriate message and attempts to build a Dictionary of half the specified size. This process is repeated until successful, or until the number of systems is reduced to one, at which point the program is aborted.

Each file name is displayed as it is being created. When the Dictionary has been built, FS101MAKE chains to the FORCE logo. FORCE may then be used.

NOTES: Table 6-1 lists the FORCE Dictionary files.

When specifying the number of systems to be contained within the Dictionary, note that the figure given is used only for calculating Dictionary size. This entry does not translate as a direct restriction on the number of system names, screens, etc. For example, specification of one system enables the insertion of 32 system names. Table 6-1 indicates the calculation used to determine number of records allocated for each system.

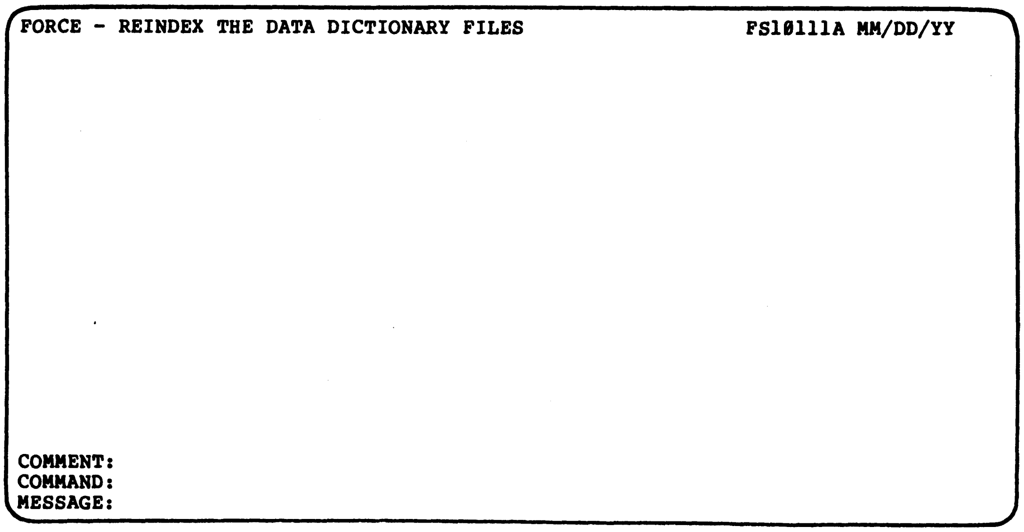
FORCE accepts an original specification for a one-system Dictionary. However, when reducing the number of systems because of insufficient disk space, FS101MAKE aborts if disk space cannot accommodate two systems.

6.2.2 RE-INDEX THE DICTIONARY FILES

PURPOSE: Dictionary index files require periodic re-indexing to ensure the integrity of their records. This function, which takes less that five minutes, should be run daily.

PROCEDURE: Prior to executing this function, all FORCE users should be logged off the system.

Enter selection A from the Data Base Administration Menu. The following screen is displayed.



```
FORCE - REINDEX THE DATA DICTIONARY FILES                 FS10111A MM/DD/YY




COMMENT:
COMMAND:
MESSAGE:
```

On the Comment Line, FORCE displays the following prompt:

DO YOU WANT TO DELETE ALL NON REFERENCED ITEMS?

A (Y)es removes Dictionary records for which the associated system name has been deleted (see 1.1.3). A (N)o causes the data to be retained after re-indexing. Enter (Y)es or (N)o.

Upon completion of the re-indexing process, control returns to the Data Base Maintenance Menu.

NOTES: The re-index program erases the pointers from all FORCE indexes, reads the FORCE data files and re-inserts the record pointers in the indexes. It is comparable to the IRIS REHASH program.

6.2.3 REORGANIZE THE DICTIONARY

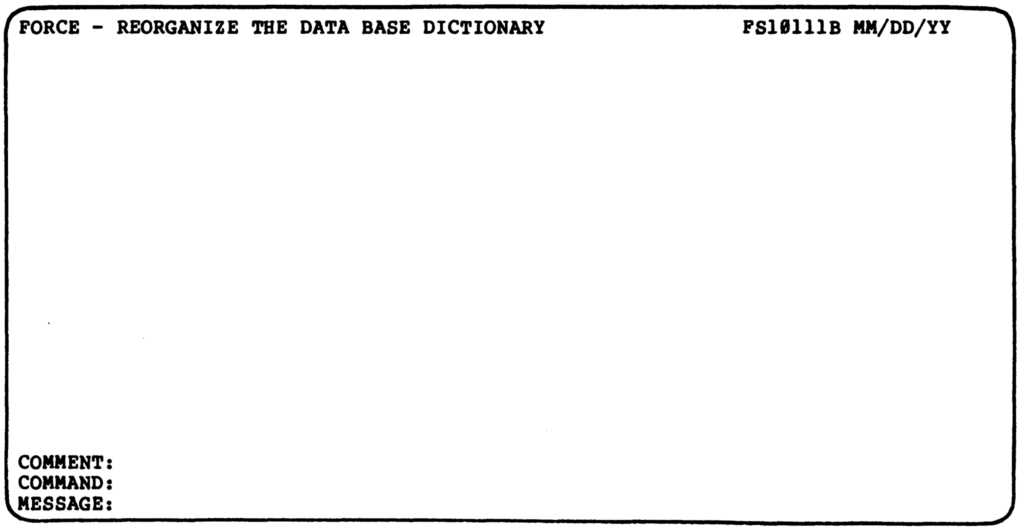
PURPOSE: This function analyzes and restructures the Dictionary data files to increase accessing efficiency and validate records. It then prints a report indicating FORCE program errors encountered since the last reorganization. When these processes are completed, the program invokes an automatic re-index of the Dictionary index files.

The reorganization program may also be used to facilitate back-up of the Dictionary files.

Reorganization constitutes a total Dictionary maintenance process and should be performed weekly.

PROCEDURE: Prior to executing this function, all FORCE users should be logged off the system.

Enter selection B from the Data Base Administration Menu. The following screen is displayed.



```
FORCE - REORGANIZE THE DATA BASE DICTIONARY              FS1Ø111B MM/DD/YY

















COMMENT:
COMMAND:
MESSAGE:
```

At the Command Line, FORCE displays the following prompt:

DO YOU HAVE A BACKUP COPY OF THE DICTIONARY FILES?

The Dictionary should always be backed up prior to running a reorganization. If there is a current backup of the Dictionary, enter (Y)es. If not, enter (N)o.

A (N)o entry causes FORCE to initiate the Dictionary backup process. FORCE prompts for the Logical Unit to which the Dictionary should be copied. Specify a Logical Unit for copying of the Dictionary.

This segment of the reorganization function facilitates copying of the Dictionary. To copy the Dictionary (without performing a complete reorganization), execute the program through the copying process, then press ESCAPE after the Dictionary has been copied. Do not press ESCAPE until the program returns to the original prompt asking for a backup of the Dictionary.

If the backup fails for any reason, the Dictionary is ruined. Copy over and re-index the backup Dictionary. Do not attempt another reorganization before determining the original problem.

When the Dictionary has been copied, the program once again prompts to determine if there is a backup copy of the Dictionary files. Enter (Y)es to initiate the reorganization process if the backup process was successful.

Upon completion of reorganization, FORCE prints a report that logs program errors encountered since the last reorganization. The report is followed by a printed, addressed mailer, which expedites the task of sending the report to POINT 4's Automated Software Products Facility in Carmel, CA. This report provides direction to the ASP Facility for customer support and product enhancements.

Fold, staple and mail the reorganization report.

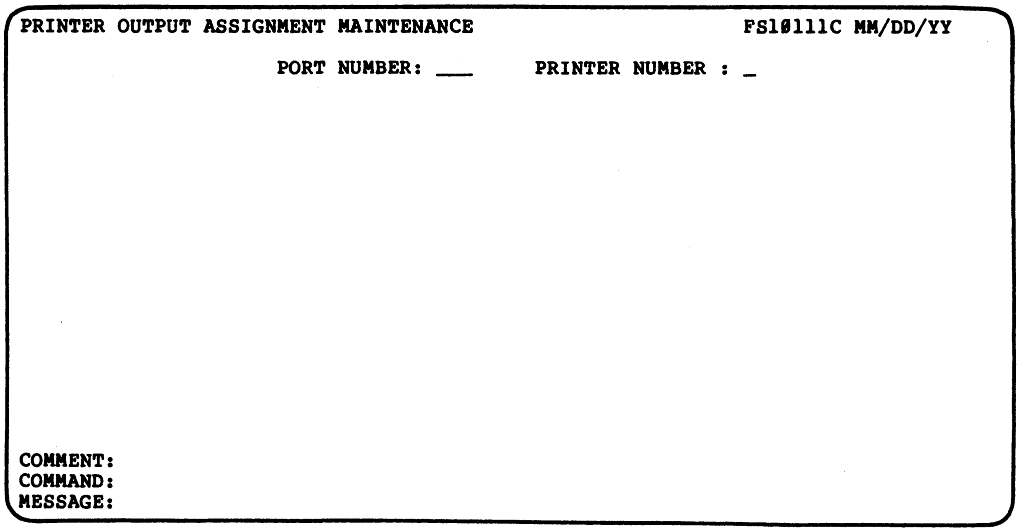
NOTES: The reorganization program deletes invalid Dictionary data, closes up vacated file space and resets record pointers accordingly. Its function is similar to the IRIS CLEANUP program.

6.2.4 MAINTAIN PRINTER OUTPUT ASSIGNMENTS

PURPOSE: This function selectively assigns port-to-printer output for FORCE documentation functions. This enables FORCE documentation printouts to be directed to the most convenient system printer.

PROCEDURE: Prior to executing this function, all FORCE users should be logged off the system.

Enter selection C from the Data Base Administration Menu. The following screen is displayed.



```
PRINTER OUTPUT ASSIGNMENT MAINTENANCE                   FS1Ø111C MM/DD/YY
                          PORT NUMBER: ___     PRINTER NUMBER : _















COMMENT:
COMMAND:
MESSAGE:
```

At the Port Number entry field, specify the system port for which a new printer assignment is to be made. This must be a three-digit number from 1 to 118.

Assign a system printer at the Printer Number field. This number, which must range from 0 to 9, represents the numeric portion of the associated printer driver (i.e., $LPT to $LPT9).

NOTES: A complete list of printer assignments may be obtained through FORCE Documentation Services (see 3.3.4).

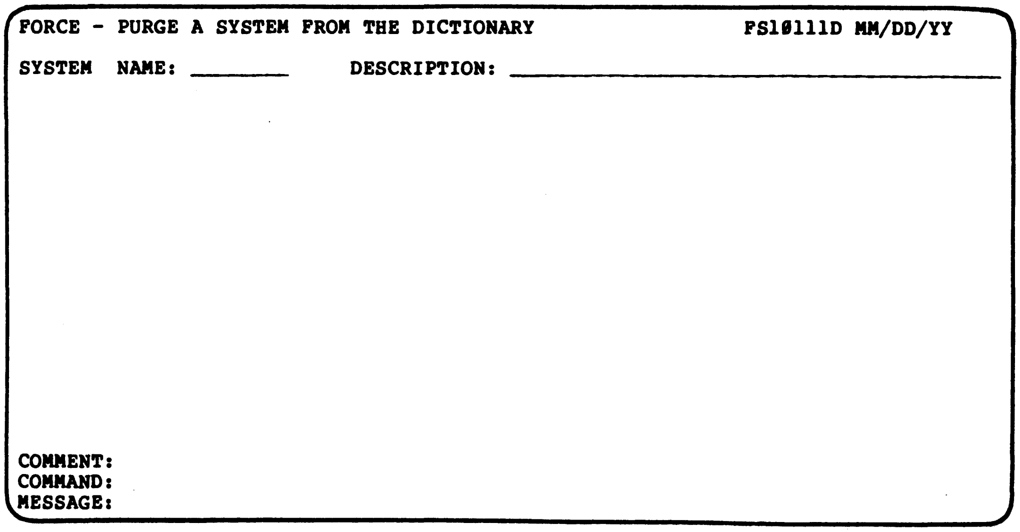
The name of any line printer driver to be assigned with this function must begin with $LPT. Printer Number 0 is assigned to $LPT.

6.2.5 PURGE A SYSTEM FROM THE DICTIONARY

PURPOSE: Execution of this function completely removes all components of the specified system from the Dictionary. A system purge differs from a system deletion (see 1.1.3) in that a deletion removes only a system name, while a purge eradicates all system components from the Dictionary.

PROCEDURE: Prior to executing this function, all FORCE users should be logged off the system.

Enter selection D from the Data Base Administration Menu. The following screen is invoked.



```
FORCE - PURGE A SYSTEM FROM THE DICTIONARY                 FS10111D MM/DD/YY

SYSTEM  NAME: ________      DESCRIPTION: ________________________________________











COMMENT:
COMMAND:
MESSAGE:
```

Enter the name of the system to be purged. FORCE displays a message as each category of system components (screens, data elements, etc.) is deleted from the Dictionary.

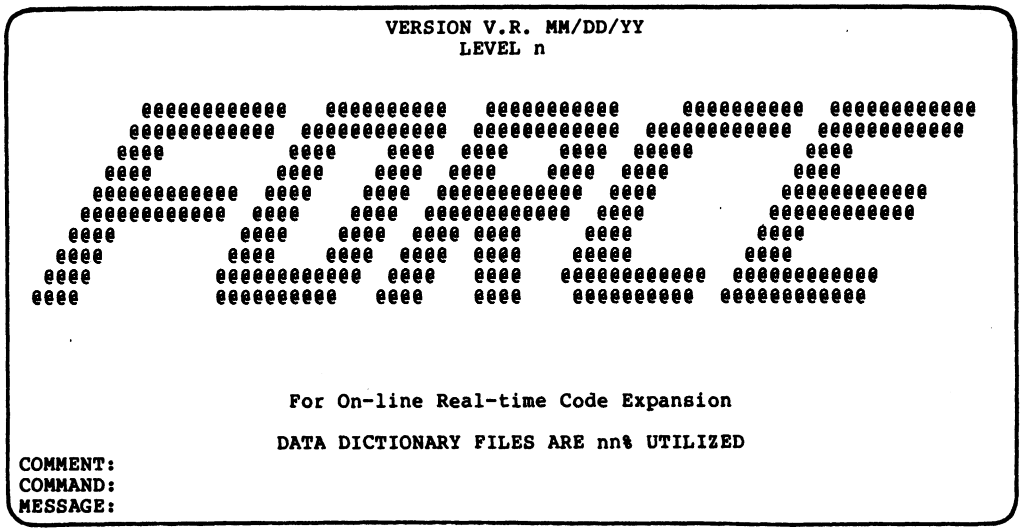
Section 7
OPERATIONAL GUIDELINES

To achieve the greatest degree of operating proficiency, the user should become familiar with standardized FORCE procedures and requirements. Operational Guidelines describes the procedure for invoking FORCE, interactive modes and methods, and the use of various terminal keys. A summary of required hardware/software is also included.

Subsections of Operational Guidelines are listed below:

7.1 INVOKING FORCE

To invoke the FORCE System, enter FORCE at the system prompt. The FORCE logo is then displayed.

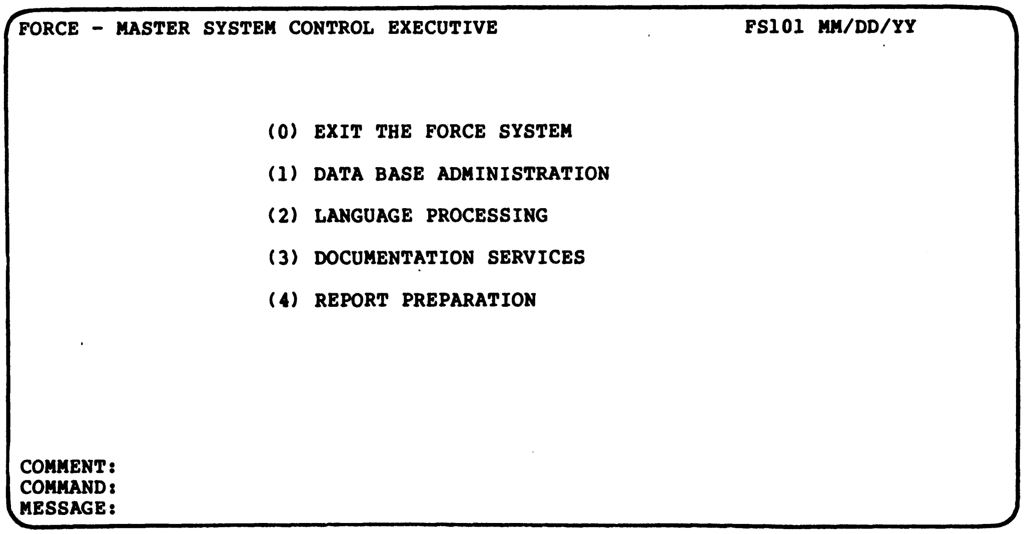


```
                              VERSION V.R. MM/DD/YY
                                    LEVEL n

         @@@@@@@@@@@@   @@@@@@@@@@   @@@@@@@@@@@     @@@@@@@@@@  @@@@@@@@@@@@
        @@@@@@@@@@@@  @@@@@@@@@@@@  @@@@@@@@@@@@  @@@@@@@@@@@@  @@@@@@@@@@@@
       @@@@          @@@@    @@@@  @@@@    @@@@  @@@@@         @@@@
      @@@@          @@@@    @@@@  @@@@    @@@@  @@@@          @@@@
     @@@@@@@@@@@@  @@@@    @@@@  @@@@@@@@@@@@  @@@@          @@@@@@@@@@@@
    @@@@@@@@@@@@  @@@@    @@@@  @@@@@@@@@@@@  @@@@          @@@@@@@@@@@@
   @@@@          @@@@    @@@@  @@@@ @@@@     @@@@          @@@@
  @@@@          @@@@    @@@@  @@@@  @@@@    @@@@@         @@@@
 @@@@          @@@@@@@@@@@@  @@@@   @@@@   @@@@@@@@@@@@  @@@@@@@@@@@@
@@@@           @@@@@@@@@@   @@@@    @@@@    @@@@@@@@@@  @@@@@@@@@@@@


                     For On-line Real-time Code Expansion

                     DATA DICTIONARY FILES ARE nn% UTILIZED
COMMENT:
COMMAND:
MESSAGE:
```

The logo screen denotes the version/revision, release date and Level of the user's copy of FORCE. Additional information provided by the logo screen includes the utilized percentage of allocated Dictionary file space.

Press ESCAPE to proceed from the logo to the FORCE master menu, called the Master System Control Executive. This menu provides access to the four primary functional areas of FORCE:

- Data Base Administration
- Language Processing
- Documentation Services
- Report Preparation



```
FORCE - MASTER SYSTEM CONTROL EXECUTIVE                      FS101 MM/DD/YY



                     (0) EXIT THE FORCE SYSTEM

                     (1) DATA BASE ADMINISTRATION

                     (2) LANGUAGE PROCESSING

                     (3) DOCUMENTATION SERVICES

                     (4) REPORT PREPARATION







COMMENT:
COMMAND:
MESSAGE:
```

7.2 FORCE INTERACTION

The user must interact with FORCE when establishing application/program specifications within the Dictionary. FORCE facilitates this communication with the same consistent, interactive methods employed by its generated programs.

A simple menu system enables the user to quickly determine and select the desired function. FORCE program names reflect the required menu choices that must be selected to arrive at the function (see 7.2.1). Comment, Command and Message lines establish a standardized method of communication between FORCE and its users (see 7.2.2). User proficiency is enhanced by uniformity of interactive techniques (menus, comprehensible displays, help screens).

User-application specifications undergo extensive edits and validations before FORCE registers the information in the Dictionary. For example, when linking a string subdivision (see 1.3.4) FORCE checks to ensure that the specified subdivision size (when added to previously-specified subdivisions) does not exceed the total length of the string. Such validations are performed automatically by FORCE, and an explicit message notifies the user of an invalid specification.

7.2.1 MENU SELECTION PROGRAMS

The FORCE user is directed to the required function by selections from various menu programs. The initial system menu, the Master System Control Executive, is shown in Section 7.1. Input of the number that corresponds to a menu choice invokes a submenu which provides selections in more specific functional areas. Submenu choices are selected until the required function is invoked. FORCE programs are never nested more than three menu selections deep (except for Report Definition functions - see 4.2).

Program names, displayed in the upper right corner of each program screen, reflect the menu choices required to select the function. This convention parallels the manual's section naming technique, as described in the Preface. The numbers following the initial five characters of a program name represent required menu choices from the Master System Control Executive. For example, the program to Design or Modify a Report Header is named FS106432. As indicated by the last three numbers in the program name, this function is invoked by selecting 4 from the Master System Control Exective Menu, then 3, then 2 from the following submenus.

Menu selection 0 or the ESCAPE key may be used to back out of the current menu to the menu from which control was previously transferred.

To expedite the back-out process, two additional commands are available. Entry of a # symbol at the Command Line of any menu exits the FORCE System and returns the user to SCOPE. Entry of a * symbol at the Command Line of any menu transfers control to the Master System Control Executive Menu.

Upon completion of most FORCE functions, control is transferred to the previous submenu. FORCE functions which typically require repetition are re-invoked after completion. An example of a function which is re-invoked is Add Data Elements to a Record Layout (1.4.2).

When the cursor is positioned at the first entry field of a function, the ESCAPE key may be used to back out to the previous submenu.

7.2.2 COMMENT/COMMAND/MESSAGE LINES

FORCE programs display Comment, Command and Message lines on the bottom three lines of each screen. These lines serve as a communication center between FORCE and the user. They are used in the same manner in FORCE-generated data entry and menu programs.

The Comment Line provides the appropriate system prompt to the user. These prompts may assist execution of a function, ask for information, or advise the user of a system condition. Three sample Comment Line prompts are shown below:

```
ENTER THE NUMBER OF THE FUNCTION YOU WISH TO EXECUTE

DO YOU HAVE 8 1/2 X 11 PAPER IN THE PRINTER?

FILE INITIALIZATION IN PROGRESS, DO NOT DISTURB!
```

Command Line entries are reserved for brief user responses to Comment Line prompts. Typical Command Line entries are menu selections and Y or N (Yes or No) responses to system prompts.

FORCE System error messages are given on the Message Line. These messages are explicit, and often provide the information required to rectify the error. Message Line errors are accompanied by a single beep from the terminal.

7.2.3 SYSTEM PROMPTS AND INPUTS

This section provides general information regarding FORCE data entry.

DATA INPUT

FORCE input screens are designed to facilitate data entry. Input fields are underscored to the maximum length of a valid entry. Some functions contain help modules, and all function inputs are explained in the corresponding manual section (see Sections 1-4).

After entering data at an input field, a RETURN registers the entry and moves the cursor to the next field. The ESCAPE key may be used to back up to the previous entry field, at which point the current entry may be overtyped. This enables corrections of previous inputs.

When all entry fields on a screen have been completed, the cursor positions at the Command Line and FORCE asks if all above inputs are correct. Entry of (Y)es signals completion of data entry. A (N)o positions the cursor at the last entry field. New information may then be entered, or the ESCAPE key may be used to back up to any field for modification.

Data entries are not registered in the Dictionary until the (Y)es response is given to the final prompt.

A RETURN at an entry field, without any accompanying input, sets a default value as an entry (if allowed). For example, the default Disk File Name when establishing a file (see 1.4.1) is the previous Data File Name entry. If a field has no default value, FORCE generates a message advising that the data must be entered.

If an input is invalid, FORCE generates an error message. FORCE error messages often contain specific corrective responses to the invalid entry.

RETRIEVED/DISPLAYED DATA

Many fields on FORCE input screens are automatically displayed once the user enters associated information. These automatic displays are performed by accessing the Dictionary for appropriate information. For example, the descriptions for systems, screens, files, etc. are automatically displayed once the associated name is input.

Some FORCE entry screens are automatically adjusted according to initial input specifcations. For instance, when establishing a data element (see 1.3.1), entry of a (M)atrix element type invokes an input field for specification of matrix coordinates. This field is not displayed for other data element types.

When a program is in modify mode, a keyed entry invokes display of all associated input field information. This information may be modified by typing over the current entry, or it may be left intact by pressing RETURN. A RETURN advances the cursor to the succeeding input field, and an ESCAPE backs up to the previous field. Using these keys, each input field may be reviewed and/or modified.

PROMPTS

FORCE prompts, which guide the user in data entry, are available from a number of sources. The Comment Line often displays information to assist in data entry. Underscores at any data entry field represent the maximum valid length of input. Most error messages, displayed on the Message Line, provide corrective information for valid entries.

Standard FORCE input fields do not accept lowercase letters. Entries should be in uppercase characters. (Screen and field formatting characters sometimes deviate from this rule.)

A FORCE prompt which encloses the initial letter of a word in parentheses indicates that the single letter should be entered, not the entire word. For example, the prompt

```
    ENTER (Y)ES OR (N)O
```

means that the user should enter Y or N.

DO NOT DISTURB MESSAGES

Any DO NOT DISTURB message from the FORCE System should be carefully observed. Failure to heed a DO NOT DISTURB message causes an interruption of normal FORCE operation, with unpredictable results.

7.2.4 SPECIAL FUNCTION KEYS

ESCAPE Key

In FORCE, the ESCAPE key is used:

- to exit a menu or a program (when at the first input field of a screen) and return to the menu from which control was previously transferred
- to back up to the previous entry field on an input screen
- to exit FORCE from the Master System Control Executive Menu
- to proceed from the FORCE logo to the Master System Control Executive Menu
- to change modes of operation in Screen Display Formatting and Output Formatting

RETURN Key

In FORCE, the RETURN key is used:

- to signal completed entry of data and move to the next input field
- to signal entry of the default value of an input field (when applicable)
- while performing screen or output formatting, to update cursor position status (first RETURN), and to move cursor to the first position of the next vertical line (second RETURN)

* Sign

In FORCE, the * sign is used:

- when entered at the Command Line of any submenu, to return to the Master System Control Executive Menu
- to specify a multiplication operation to be performed in the calculation of a synonym
- to indicate completion of break output specifications and delete any succeeding entries

# Sign

In FORCE, the # sign is used:

- when entered at the Command Line of any submenu, to exit FORCE and transfer control to SCOPE.

### @ Sign

In FORCE, the @ sign is used:

- during output formatting, to indicate a starting location for the printing of data
- in format masking, to indicate a location within the mask for each character of the data

7.3 HARDWARE/SOFTWARE REQUIREMENTS

HARDWARE

- POINT 4-type computer
- 5MB disc storage (minimum)
- Printer(s)
- CRT terminal(s)
  required features:
  - cursor addressing
  - clear to end of line
  - clear screen
  - dim intensity

SOFTWARE

- IRIS Operating System (R8 with FORCE-coded R8 Pico-N)
- POINT 4 Business BASIC
- $TERMS enabled with cursor tracking

Note that although the operation of FORCE 1.A requires IRIS R8, the source code generated by FORCE can run under earlier IRIS releases. Generated menu and data entry programs run on IRIS 7.4 (or later). Generated report programs run under IRIS 7.5 (or later).

The FORCE Automatic Programming System comprises approximately 190 program modules, which require 2600 blocks on disc. (These figures are subject to change, due to continuing product enhancement.) The size of the Dictionary is dependent upon user specifications.

Section 8
PROGRAMMING WITH FORCE

Application generation with the FORCE Automatic Programming System minimizes four important developmental problems:

- Time of development - data entry, report and menu programs are generated automatically
- Application maintainability - labeled source code, standardized logic and documentation all facilitate program maintenance
- Uniformity of program/operator interaction - generated programs employ consistent interactive techniques and functional operations
- Documentation - generated source code is fully-commented, and reports on system components are available from the Dictionary

Effective utilization of FORCE ensures the fullest advantage in these critical areas. Section 8 explores developmental concepts and techniques for using FORCE to construct an application. This section contains specific methods for developing data entry and report programs. The primary subsections of Section 8 are:

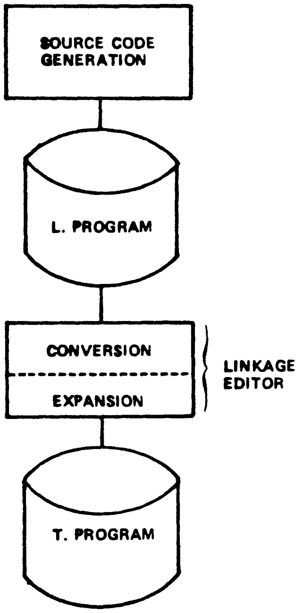
8.1 - LABELED SOURCE CODE
8.2 - APPLICATION DEVELOPMENT
8.3 - DATA ENTRY PROGRAM DEVELOPMENT
8.4 - REPORT PROGRAM DEVELOPMENT

8.1 LABELED SOURCE CODE

FORCE generates labeled source code for menu, data entry and report programs. Program entry points are referenced by descriptive labels; other program statements do not require labels or line numbers.

Labeled source code, when processed by the Linkage Editor (see Section 2.1), is expanded into IRIS Business BASIC source code which may then be loaded and executed. This feature dramatically improves productivity by eliminating the need to program using line numbers. FORCE-generated programs can be maintained in labeled source code, then converted to Business BASIC. Additionally, programs written manually in labeled source code may be processed by the Linkage Editor for conversion to Business BASIC.

The flowchart below illustrates the progression from generated labeled source code to IRIS Business BASIC.



The advantages of programming with labels include:

- programmers don't have to anticipate branching line numbers
- entry points are referenced by English-language labels
- labels are easier to remember than line numbers
- descriptive labels enhance the readability of the code
- code that is labeled is easier to maintain and manipulate
- unlabeled lines cannot inadvertently be used as entry points
- no renumbering of copied/inserted source code modules
- the possibility of duplicate line numbers is eliminated
- labeled text doesn't have to be renumbered

FORCE-generated programs are now prefixed with L. to denote a labeled text file. Subsequently, the Linkage Editor processes labeled files and generates Business BASIC text files with a T. prefix. For example, a program defined to FORCE as AEI17 is generated as labeled source code in a file called L.AEI17. After conversion by the Linkage Editor, a corresponding BASIC text file is created as T.AEI17 on disk.

All maintenance and macro insertion may be performed in labeled program source code. The Linkage Editor incorporates edits and expands macros when the L. file is converted. Maintenance and macro insertion may also be performed in the converted T. source code file. However, in order to avoid discrepancies between corresponding L. and T. files, all development should be limited to either labeled or BASIC text files.

Examine the two source code segments below. They are portions of the same program source code, in labeled and BASIC text.

LABELED SOURCE CODE

```
+
. ******************************************
. * INPUT AND EDIT OF THE DESIRED FUNCTION *
. ******************************************
.
K001
         PRINT @09,21;'CL';
         PRINT "ENTER THE NUMBER OF THE FUNCTION YOU WISH TO EXECUTE";
*
K001A
         PRINT @09,22;'CL';
         INPUT @09,22;I$
         PRINT @09,23;'CL';
         IF I$<>"" GOTO K001B
         PRINT @09,23;'CLRB';
         PRINT "THE FUNCTION DESIRED MUST BE ENTERED, TRY AGAIN !";
         GOTO K001A
```

BUSINESS BASIC SOURCE CODE

```
110 REM
110 REM ******************************************
110 REM * INPUT AND EDIT OF THE DESIRED FUNCTION *
110 REM ******************************************
110 REM
110 REM ---> ENTRY POINT: K001
110 REM
110 PRINT @09,21;'CL';
120 PRINT "ENTER THE NUMBER OF THE FUNCTION YOU WISH TO EXECUTE";
130 REM
130 REM ---> ENTRY POINT: K001A
130 REM
130 PRINT @09,22;'CL';
140 INPUT @09,22;I$
150 PRINT @09,23;'CL';
160 IF I$<>"" GOTO 200
170 PRINT @09,23;'CLRB';
180 PRINT "THE FUNCTION DESIRED MUST BE ENTERED, TRY AGAIN !";
190 GOTO 130
```

Statements in the sample source code segments contain the same essential components. Each uses an operator (the process to be performed by the statement) and operands (which determine how the statement will be executed). The difference is that each BASIC statement must be referenced or identified by a line number.

In L. files, labels are used as a reference point for transfer of control operators (GOTO, GOSUB, etc.). Only referenced statements require a label.

Note the line references in corresponding program statements. Note also that unreferenced lines in labeled source code require no labels.

After conversion, the BASIC source code is commented to denote each label as an entry point. This facilitates comparison of labeled and BASIC files. Comment lines are numbered so that the REM statements are overstored when the program is loaded.

Labels are alphanumeric, and may contain up to eight characters. They are printed beginning at the first position on a line. A label may appear on the same line as a program statement, as long as each begins at the correct line position.

Operators and subsequent operands are printed after a ten-space tab in the standard BASIC format. The Linkage Editor provides a facility for aligning an L. file. A line entry not encountered in the first character position on the line is tabbed over ten spaces. FORCE-generated programs are tabbed automatically during source code generation.

Generated L. files and expanded T. files exist on disk independently of each other. A regenerated program overwrites the existing L. program on disk. Each time a program text file (either L. or T.) is processed by the Linkage Editor, it overwrites the existing T. file.

The Linkage Editor verifies manually written labeled source code. If it encounters a statement that references a label not contained within the program, the expansion process is halted and an appropriate message is generated.

Three symbols (+ * .) are available to indicate a REM line. These symbols correspond closely to ANSI standards as print control characters and may be used to format the printed text file.

A + symbol initiates a form feed during listing. The * symbol forces a top of form if encountered within the last ten lines of the page (to ensure that a section of code is not split between pages). A standard comment line is denoted by the . symbol.

8.2 APPLICATION DEVELOPMENT

Prior to development of an application with FORCE, the user should understand its immediate capabilities as well as its potential. Although complete menu, data entry and report programs can be generated automatically, sophisticated applications may require additional manual programming.

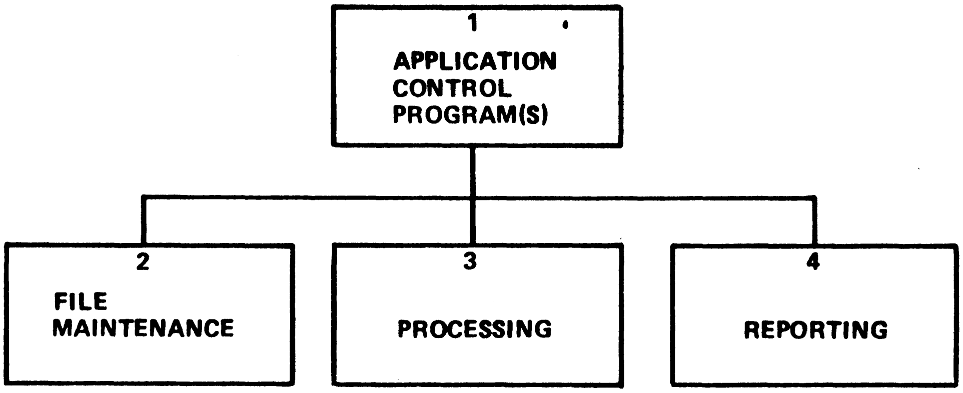
Section 8.2 is organized as follows:

8.2.1 - System Analysis Using FORCE
8.2.2 - Recommended Conventions

In most applications, nearly 80% of the actual source code performs standard interaction with the operating system (opening channels, writing to files, etc.). FORCE virtually eliminates this segment of programming by generating the required source code from Dictionary specifications. Depending upon application requirements, the remaining 20% may be generated completely or partially, or it may be developed through a combination of macro expansion and manual coding.

The developmental power of FORCE is enhanced by a well-conceived application design. This design should be completed before FORCE is ever invoked. Consideration must be given to the programs which can be generated automatically, and those which require development with macros and manual coding.

Below is an illustration of a typical application. Each functional area is numbered to show how FORCE-generated source code can accommodate the task.



1. Generated menu selection programs
2. Generated data entry programs
3. Macro expansion
4. Generated report programs

8.2.1 SYSTEM ANALYSIS USING FORCE

The following scenario describes how FORCE can be beneficial during the initial phases of application design.

One individual, acting as a system analyst, meets with the user to determine the application requirements. The system analyst then designs the system and reviews its performance requirements with the user. Once approved, the system analyst begins defining the application to the FORCE Dictionary.

Portions of this process (screen design, report output formatting) may be accomplished interactively with the user. FORCE lends itself to user involvement during application development. Dictionary definition functions are user-friendly and can be performed quickly. Modifications are easily applied to existing Dictionary specifications.

Documentation from the FORCE Dictionary provides hard-copy reports on system components for the user to review. Affording the user immediate evaluation of such things as screens and synonyms may provide insights beneficial for both the programmer and the eventual operators. The analyst can actually use system documentation to obtain sign-offs from the customer.

Subsequent application development is then performed with the assurance that essential system requirements have been verified. Through effective definition and maintenance of Dictionary specifications, all generated source code will reflect these established requirements.

At this point, substantial development of Dictionary specifications has been accomplished in the application design phase.

8.2.2 RECOMMENDED CONVENTIONS

In order to achieve a high degree of efficiency, naming conventions should be established for program names, file names and user macro text files. This section suggests conventions for these system components, as well as a numbering convention for screen displays. Additionally, a discussion of system names and prefixes examines fundamental guidelines for application development.

Note that the following recommendations are not imposed as restrictions by FORCE. The user is not required to adopt or adhere to any constricting naming procedures, and invalid data (system names, file names, etc.) is identified as such during Dictionary definition. Conventions in this section are offered as suggestions and incentive for conscientious programmming practice.

Naming conventions facilitate the acquisition of sorted reports and queries. They also complement the ease with which data may be shared between systems in the Dictionary by establishing a common method of reference. Finally, effective naming conventions make it easy for programmers to understand the design of unfamiliar applications, thereby facilitating maintenance.

SYSTEM NAMES AND PREFIXES

A system name must accurately reflect its function. Execution of most Dictionary definition functions requires prior association with a system name. A system name is proliferated throughout FORCE, in Dictionary documentation and in the application itself. The choice of a meaningful system name facilitates subsequent development within the Dictionary, and operation of the finished application.

After determining a system name, construct a system prefix. A system prefix is an abbreviated version of the system name (two or three characters) that is affixed to the beginning of program names, files, etc. The system prefix immediately identifies the system to which application components are related.

DESCRIPTIONS

FORCE requires the user to assign a description for most application components. A meaningful description for screens, programs, etc. assists the user when querying or reporting on system specifications. Additionally, FORCE uses descriptions of such things as data elements and index files in the generation of error messages. Descriptions should be chosen with these things in mind.

Specifically, the description entered for an index file should describe the key as a composite. This description is used by FORCE to construct an error message for successful/unsuccessful indexed file checks. For example, consider a customer information data entry program with a directory described as CUSTOMER NUMBER. If a matching key was not located for a customer number input of 829, the following message would be displayed:

```
CUSTOMER NUMBER 829 WAS NOT FOUND
```

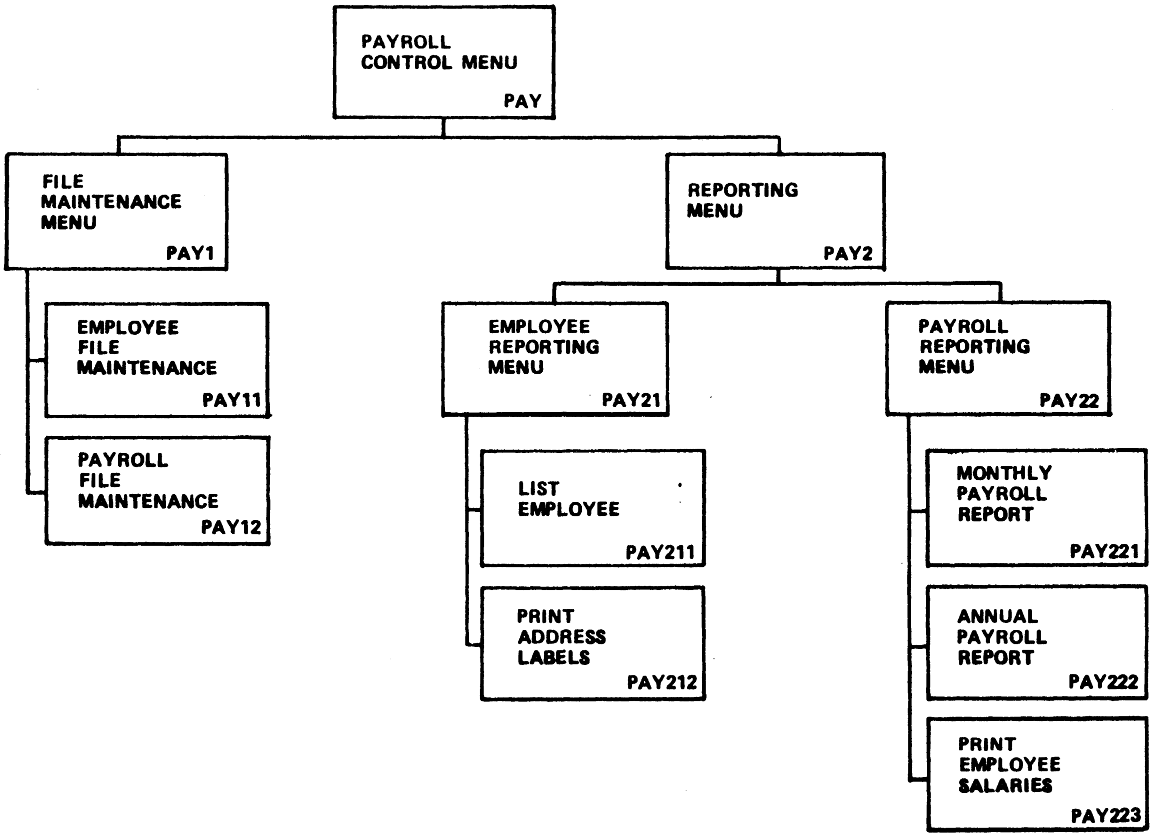
However, if the index file's description was KEY CUST. SYSTEM INDEX, the following, less explicit message would be generated:

```
KEY CUST. SYSTEM INDEX 829 WAS NOT FOUND
```

PROGRAM NAMES

Program names should begin with the system prefix, followed by a number which represents the program's position in the hierarchical flow of the system. For example, consider the following system, named PAYROLL. The system prefix is PAY. Program names are noted in the bottom right corner of each function. Note the numbering system used in the program names.

(To avoid unnecessary complexity of design, it is recommended that programs not be nested more than three levels from the master menu.)

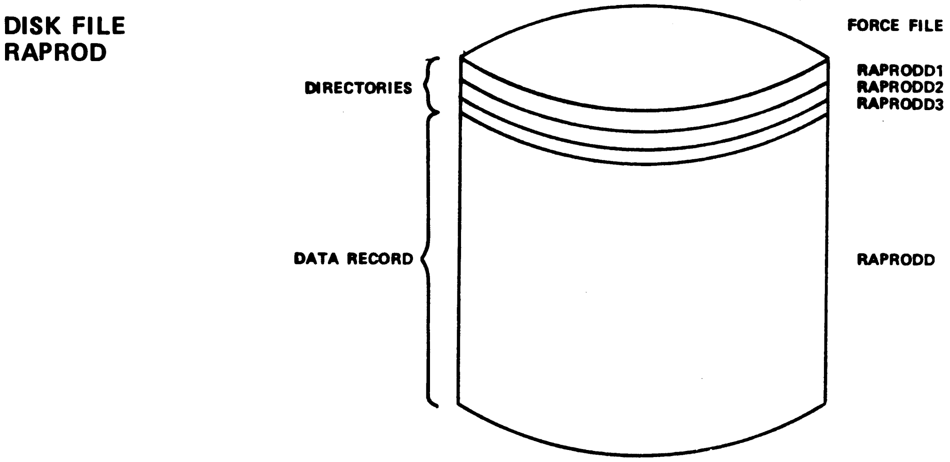


FILE NAMES

Disk file names are composed of the system prefix, followed by a set of alpha identifiers (three or four characters) which describe the file contents. (This makes it easy to determine file contents by referencing system documentation.)

See Section 1.4.1 for more on disk files.

Data files which represent the data record of an indexed contiguous file should use the disk file name with a D (for Data) appended to the end. For the first directory of an indexed contiguous file, use the disk file name with a D1 (for Directory 1) appended to the end. Other directories follow the same convention, using D followed by a directory number. The graphic below illustrates this convention.



SCREEN DISPLAY NUMBERS

To facilitate reference, assign screen display numbers that represent the program which invokes the screen. Because screen display numbers may be three characters long, the number should be right-justified and left zero-filled. For example, the screen invoked by program PAY11 is number 011. The screen number for program PAY211 is 211, and the screen number for PAY3 is 003.

Use the numbers 998 and 999 for the master menu and the system logo (if required).

USER MACRO TEXT FILE NAMES

Text files on disk that are to be implemented as User Macros should be named with a common prefix, just as FORCE-generated programs use T. and L. prefixes to facilitate reference. Begin User Macro text file names with an S. to denote subroutine. This convention makes the purpose of these source code segments easily recognizable.

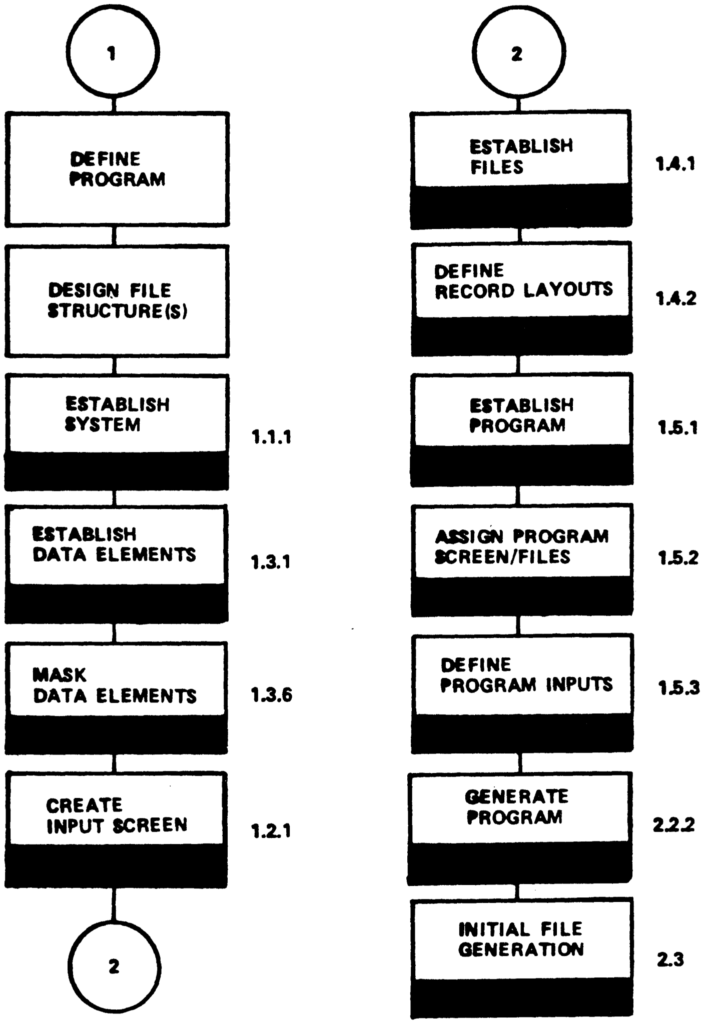
8.3 DATA ENTRY PROGRAM DEVELOPMENT

Generated data entry programs offer a complete set of file maintenance functions. A generated data entry program enables the user to add, modify, delete and query records. Keys may be automatically inserted in up to nine directories, and indexed file checks with specified continuation logic may be implemented for any or all inputs. Masks, range checks and default values may be established for program inputs.

Section 8.3 is organized as follows:

8.3.1 - Masking
8.3.2 - Screen Formatting
8.3.3 - Files Structures and Access
8.3.4 - Continuation Logic

The flowchart below outlines recommended procedures for developing a single data entry program with FORCE. Steps that are performed in FORCE are denoted by a black bar. Each step's corresponding manual section is printed at the right of the function.



DATA ENTRY PROGRAM OPERATION

Upon invoking a generated data entry program, the following message appears on the Comment Line:

```
    (A)DD, (M)ODIFY, (D)ELETE OR (I)NQUIRE ?
```

Entry of A, M, D or I invokes the specified mode of operation. All four operations of a data entry program employ the same screen.

In Add mode, after each input the cursor positions for data entry at the next field. In Modify mode, entry of the key retrieves the record. Each RETURN displays current data for that field. At any field, the current information may be left intact by a RETURN, or it may be changed by entering new information.

In Add and Modify mode, the ESCAPE key may be used to back up to the previous field for modifications.

For Delete and Inquire modes, entry of the key retrieves the record. The key may be composed of the first screen input, or it may require input of multiple fields. Current record information is displayed for the entire screen. For Delete mode, the program prompts:

```
    DELETE THIS RECORD?
```

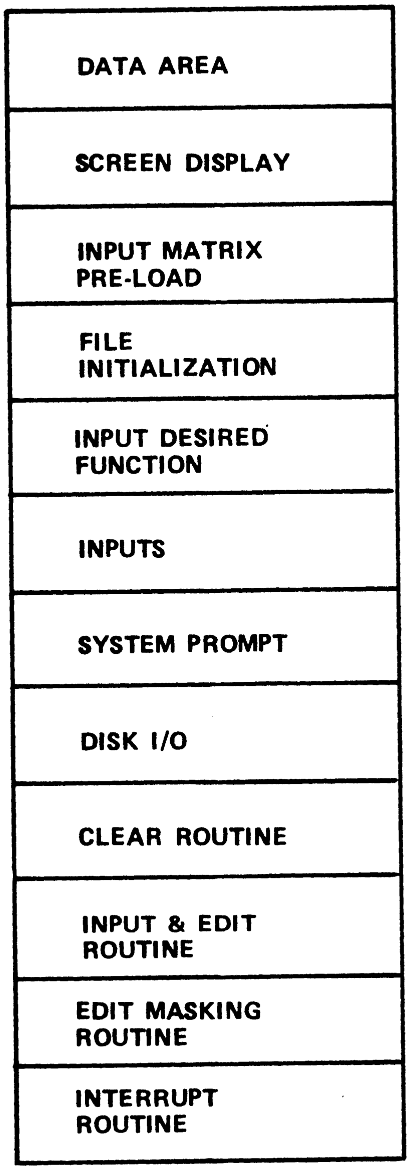
An ESCAPE at the first input field on the screen reinvokes the mode of operation prompt. This is the procedure to switch modes of operation (e.g., from Add mode to Modify mode).

An ESCAPE at the mode of operation prompt exits the program.

Generated data entry programs employ Comment, Command and Message lines (see Section 7.2.2) as a communication center at the bottom of the screen. This feature duplicates the technique employed by FORCE. Explicit, helpful error messages (based upon Dictionary specifications) are displayed when appropriate.

DATA ENTRY PROGRAM STRUCTURE

The graphic below illustrates the standard program structure for a generated data entry program.



8.3.1 EDIT MASKING

Format masks may be established for redisplay of any program inputs (see Section 1.3.6). Displayable keyboard characters and/or literals may be placed before, within, or after entered data.

Masking is an important feature to incorporate into data entry programs. Effective masking not only enhances the aesthetic quality of a program, but it makes data input more comprehensible, which may help avoid input of incorrect data.

Masking is for redisplay of data only; it does not change the way data is written to a file.

The @ symbol is used to designate each position within a mask where data is to be displayed. Other entries in the mask are printed literally at the established location. The \ symbol and the " symbol are not allowed in a mask.

When developing a data entry program, format masks should be established before creating an input screen. This procedure ensures that the user knows the actual length required for an input field when its mask is included. The proper amount of space can then be allotted between input fields on the data entry screen.

The chart below shows some sample format masks, data input, and the corresponding redisplay employing the mask.

Element Type	Mask	Input	Redisplay
Alphanumeric	@. @@@@@@@@	SJONES	S. JONES
Alphanumeric	<@@> - @@	4D5C	<4D> - 5C
Numeric	(@@@) @@@-@@@@	4086242225	(408) 624-2225
Numeric	@@/@@/19@@	010771	01/07/1971
Numeric	TOTAL - @,@@@	9387	TOTAL - 9,387

MASKING WITH DECIMALS

Valid numeric masks may employ a single decimal point. This important feature differs slightly from other masking operations. A single decimal point may be placed within a mask to separate the integer and fractional portion of the data element. Upon data entry, a decimal may then be included at the appropriate location.

If a decimal is not included within the mask, the input may not include a decimal. The number of @ signs on each side of the decimal point establishes a range to the furthest integer and fractional positions (see INVALID inputs below). Examples of format masks using a decimal point are shown below.

MASK - @@@@@.@@@@@		MASK - @@@,@@@.@@	
Input	Redisplay	Input	Redisplay
083	83.00000	4	4.00
9	9.00000	7392.5	7,392.50
.9	0.90000	93873	93,873.00
654.023	654.02300	.8	0.80
938473.4	INVALID	264.394	INVALID

After establishing a mask, FORCE allows further formatting of an input field. The formatting operations available depend on whether the input is alphanumeric or numeric.

ALPHANUMERIC FIELD FORMATTING

Alphanumeric fields may be right or left justified, and a filler character may be specified. Most alphanumeric inputs should be left justified. For example, at a input field, an entry preceded by blanks might look like this:

```
    DEPARTMENT NAME:      AUTOMATED SOFTWARE PRODUCTS
```

If left justification was used for the input, it would be redisplayed:

```
    DEPARTMENT NAME: AUTOMATED SOFTWARE PRODUCTS
```

If no justification is specified, a mask is redisplayed beginning at the first position of input.

A filler is a single keyboard character used to fill an entry to its maximum length when current input occupies only a portion. Consider a five-character sales code using the * symbol as a filler. Below are the redisplayed values for an input of S24.

Left Justified	Right Justified	No Justification
S24**	**S24	S24**

NUMERIC FIELD FORMATTING

Before displaying field formatting options for numerics, FORCE determines whether the established mask reflects valid numeric format. A valid numeric format contains any series of numbers from 0 to 9, with or without a leading minus sign and/or a single decimal point.

For valid numeric masks, a filler and a float character may be established. If a mask deviates from valid numeric format, a filler is the only field formatting option.

Numerics are always right justified.

A filler character is used for numerics in the same manner as alphanumerics, except that it is always displayed to the left of the input. Numeric inputs often employ a filler character of 0 (zero).

If selected, a float character is always printed immediately before numeric data. The $ sign is often used as a float character. For a numeric input field with a floating $ sign, an entry of 382928 is displayed as:

```
    $382928
```

For the same input, an entry of 16 is displayed as:

```
    $16
```

8.3.2 SCREEN FORMATTING

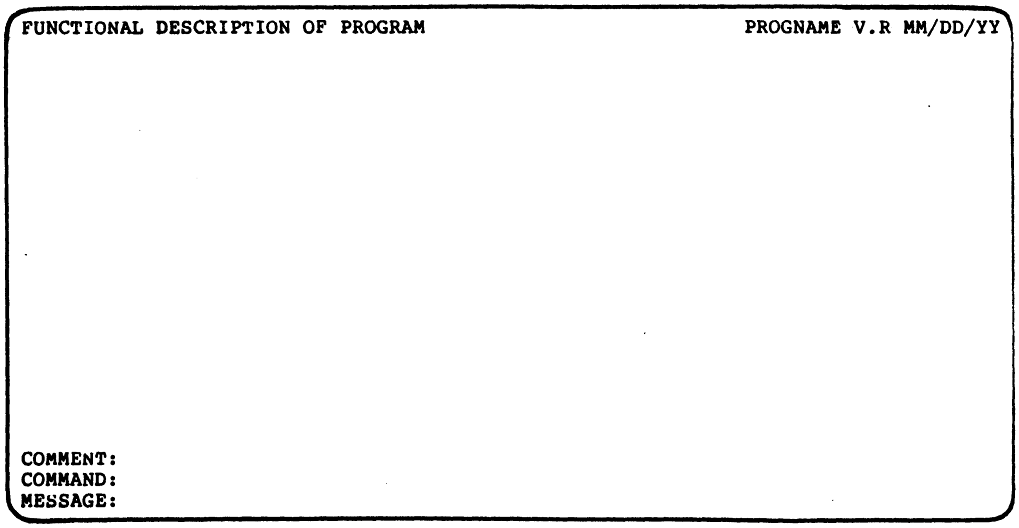
System screen displays should be consistent in format. They are the focal point of interaction with operators who often have limited computer background. A well-organized, aesthetically pleasing screen display facilitates data entry.

In order to achieve consistency, all FORCE-generated screens employ Comment, Command and Message lines (see Section 7.2.2). FORCE automatically generates menu screens, centering selections horizontally and vertically on the screen. Screens for report programs are standardized, and consist of a simple prompt on the Comment Line.

A single, user-designed data entry screen is employed for the add, modify, delete and inquiry operations.

For many FORCE-generated applications, the data entry screen is the only screen display that the user must design. The top line of a data entry screen should provide information about the invoked program.

The screen below illustrates the recommended convention for data entry screen headings. In the upper left corner is a functional description of the program. The program name, version/revision, and release date are shown in the upper right corner. This provides a means of referencing an executing program for maintenance or debugging.



```
FUNCTIONAL DESCRIPTION OF PROGRAM                PROGNAME V.R MM/DD/YY











COMMENT:
COMMAND:
MESSAGE:
```

When formatting a data entry screen, do not save underscores at input fields in the final screen format. FORCE automatically places underscores in a data entry screen for the maximum length of each input. Saving additional underscores in the screen format wastes program space.

When formatting screens, use FORCE's screen reproduction function (see Section 3.2.1) to obtain hardcopy reports. Some suggested guidelines for screen formatting are:

- Establish data element masks prior to designing screens.
- Use underscores to gauge the maximum length of an input (be sure to include the mask in this measurement), then erase the underscores from the screen.
- Leave at least five blank spaces between the maximum masked length of an input and the next entry field on the line.
- Use only even-numbered vertical lines when creating a screen, to allow insertion of additional material if subsequent modification is required.
- Learn and make use of the formatting commands, as well as the screen copy function (see Section 1.2.4).

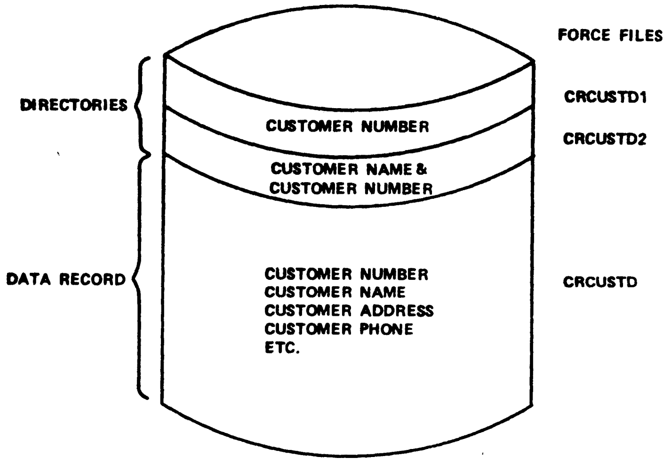
8.3.3 FILE STRUCTURES AND ACCESS

File structures and accessing requirements are an essential consideration in any application. When developing data entry programs, subsequent report programs should be anticipated so that files can be structured to accommodate reporting requirements (see Section 1.4.1).

FORCE-generated programs can access any IRIS file type. FORCE, through Initial File Generation (see Section 2.3), generates contiguous (with data and/or indexed records), formatted and text files.

For contiguous files with multiple file structures (indexed and data records, differing data record layouts) each record layout is defined to FORCE as a separate file. Record layout information must be maintained in the Dictionary so that FORCE can generate file I/O statements. FORCE files that are part of one contiguous file are interrelated by association with the same disk file name.

The graphic below illustrates an indexed contiguous file for a data entry program that maintains customer information. The file names adhere to the convention suggested in Section 8.2.2.



When initializing a contiguous file with multiple record layouts, process the FORCE file with the longest record layout. This ensures sufficient record length allocation for I/O statements. Upon entry of the FORCE file to be processed, the associated disk file name is redisplayed because that is the file being generated.

INDEX FILES

Index files are established within the Dictionary (see Section 1.4.1) as (I)ndexed, and initialized as (C)ontiguous. Keys are constructed and concatenated by adding elements to an index file's record layout (see Section 1.4.2).

FORCE-generated programs may automatically perform up to nine key inserts, and any number of indexed file checks (SEARCH for a matching key). Directories into which keys are inserted are specified when assigning a program's screen and files (see Section 1.5.2). Key inserts may be used for input verifications or file access in reporting.

Indexed file checks may be performed at any or all program inputs. They may employ valid files from any system in the Dictionary. Indexed file checks search a specified directory for a key that matches the input(s). These checks, which are used to validate entered data, are established when defining program inputs (see Section 1.5.3).

8.3.4 CONTINUATION LOGIC

When defining program inputs (see Section 1.5.3), continuation logic must be specified for each indexed file check. Continuation logic determines whether data entry should proceed if an indexed file check locates a matching key. This logic must be determined for both the add and modify operations of a generated data entry program. Continuation logic is not applicable if the data entry program doesn't employ an indexed file check.

Continuation logic specifications enable FORCE to generate effective SEARCH statements in data entry programs.

For each field that incurs an indexed file check, a (Y)es or (N)o entry must be specified at the Continue If Found prompt.

Use the following question to determine program continuation logic:

> FOR THIS INPUT, SHOULD DATA ENTRY CONTINUE TO
> NEXT FIELD IF THE RECORD IS ON FILE?

For example, when adding a customer to a file, should the program continue if the customer entered is already on file? No, because there shouldn't be duplicate customers on file. When modifying a customer record, should the program continue if the customer entered is already on file? Yes, because the customer must be on file in order to modify the record.

8.4 REPORT PROGRAM DEVELOPMENT

FORCE generates report programs which may access up to nine files per report. Synonyms are used to reference data elements in the report generation process. Report outputs are created much in the same manner as screen displays, by designing the required format on the CRT screen and saving it in the Dictionary. Sophisticated calculations and statistical functions may be defined for report data. Record selection and break criteria may be specified for each file access.

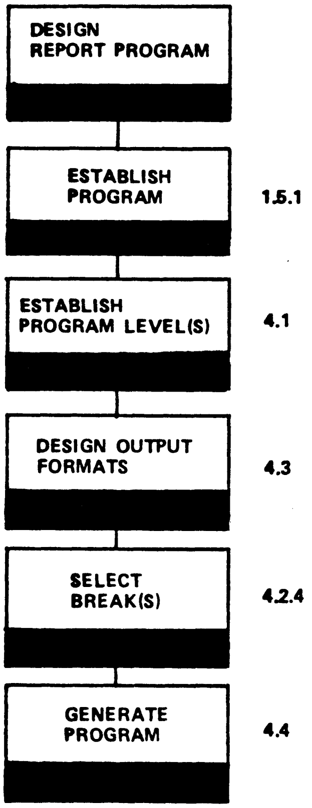
Section 8.4 is organized as follows:

8.4.1 - Report Levels
8.4.2 - Report Definition
8.4.3 - Output Formatting

A report program must be established in the Dictionary in the usual manner (see Section 1.5.1). Subsequent report program development is performed through Report Preparation functions.

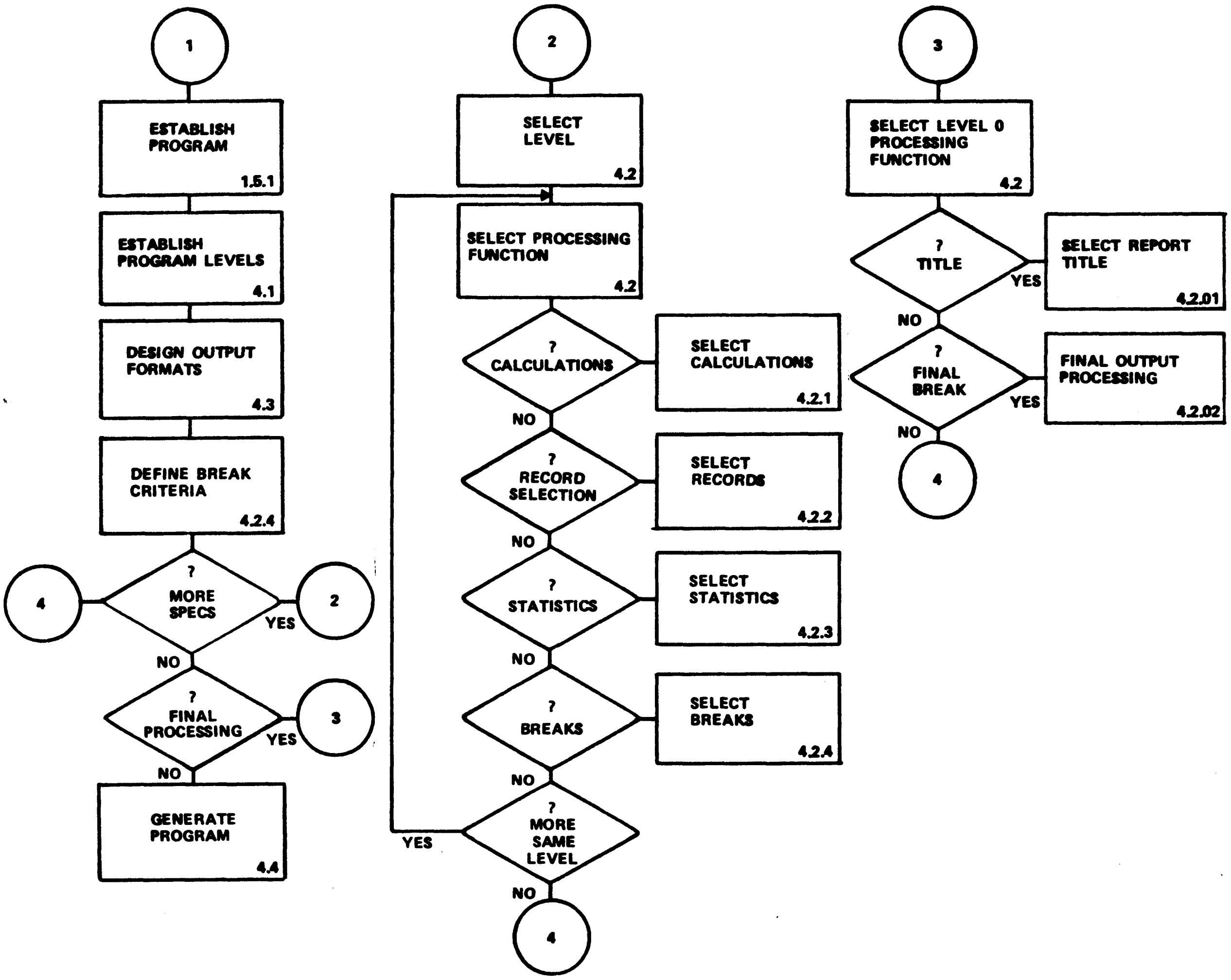
To qualify as a report, a program must perform at least one file access and one print of accumulated data. This means that a minimum specification set for a report program contains one level, an output format, and a break.

The flowchart below illustrates the minimum developmental requirements for a report program. Steps that are performed in FORCE are denoted by a black bar, and the corresponding manual section is printed to the right of the function.



Report Preparation procedures are outlined in the flowchart below. These procedures presuppose that the file(s) to be accessed and the synonyms employed are defined to the Dictionary.

Steps in column 1 are mandatory. Column 2 procedures are level-dependent and optional. Final processing operations in column 3 are also optional. The manual section which corresponds to each step is printed in the lower right corner of the function.



Before any report development functions are invoked, the following documentation should be obtained. This documentation will assist in determining report program specifications.

Documentation	Manual Section
List system data elements	3.1.3
Print element attributes	3.2.3
Print format masks*	3.3.5
List synonyms	4.5.4
Print synonym cross-reference	4.5.5
List system files	3.1.4
Reproduce record layouts	3.2.5

*These are the masks used for data entry programs. They must be re-established during output formatting if they are to be used in a report program.

In addition to this documentation, a mock-up of the report should be sketched. The mock-up should illustrate the required report format and data. A flowchart illustrating report levels is also helpful in level definition.

REPORT PROGRAM OPERATION

FORCE automatically creates a screen for a generated report program. The program's description appears in the upper left corner, and its name and date of generation are printed in the top right corner. Comment, Command and Message lines at the bottom of the screen are used for program/user interaction.

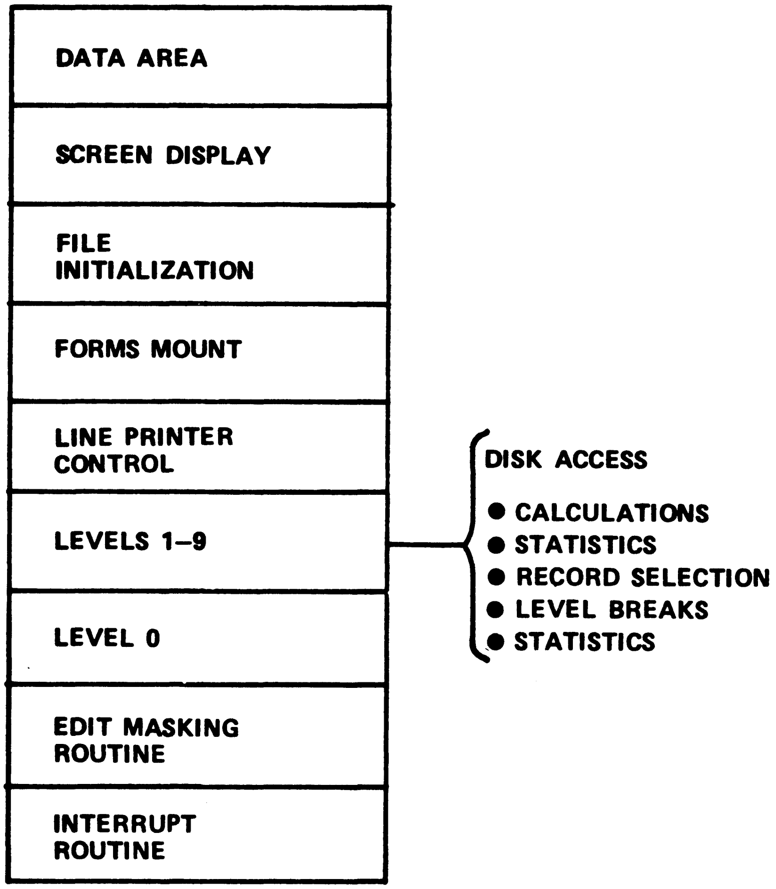
Upon invoking a generated report program, the following prompt is displayed on the Comment Line:

```
DO YOU HAVE PAPER IN THE PRINTER AND IS IT ON-LINE ?
```

A (Y)es initiates execution of the report program. A (N)o or an ESCAPE entry exits the program.

REPORT PROGRAM STRUCTURE

The illustration below represents the standard structure of a generated report program.



8.4.1 REPORT LEVELS

Each file to be accessed by a report program constitutes a level. Access to a directory and the data portion of an indexed contiguous file is regarded as a two-level access. A report program may access up to nine levels.

A level involves a disk access to a designated file, using a specified mode. Numerous processing activities may then be conducted. Prior to execution of subsequent levels, all specified operations of the current level must be performed (unless a reject or transfer is invoked).

After I/O at each level, any of the following operations may be performed. Although any of the operations may be excluded, they are executed in the order in which they are listed below:

- Calculations
- Statistical Computations (before Record Selection)
- Record Selection
- Level Breaks
- Statistical Computations (after Record Selection)

Level 0 refers to final report processing. Level 0 processing does not involve file access and is not included in the nine available report levels.

Files to be accessed by a FORCE-generated program must be defined to the Dictionary (i.e., record layouts established).

When developing a report program, the first step under Report Preparation involves specification of report levels (see Section 4.1). The order in which these files are specified represents the logical order for access during program execution.

For each indexed access, the user may specify a synonym to store the value of the retrieved record number. For data file access, the same entry directs access on the specified value. By defaulting from this specification, FORCE assigns a data element to maintain the value. Use the default assignment whenever a retrieved pointer is used to access the next level. The default assignment is cleared after a file-read and may be used to access subsequent report levels.

Consider the following simple report to list employees in employee number sequence, with their department and supervisor.

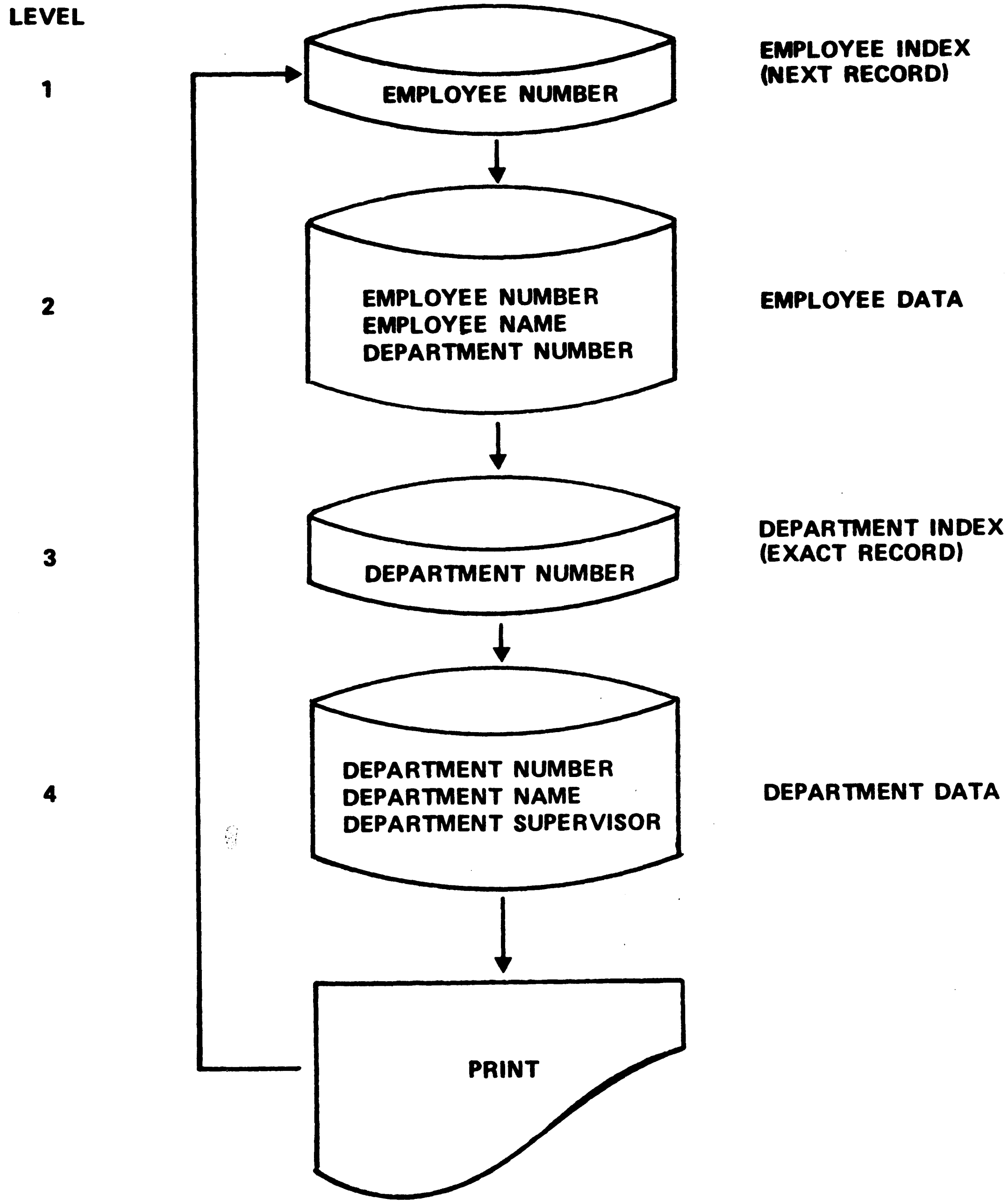
A four-level program is designed to access the employee master file through an index built on the employee number. Employee records in the master file contain information which includes employee name and department number.

The employee index file is read to retrieve the location of the first employee record. The employee name and department number are then read from the master file. At Level 3, the program reads the department index on the department number retrieved from the employee master file. The department master file is then read, to obtain the department name and supervisor.

The program then breaks and prints accumulated data: employee number, employee name, department name and supervisor.

A get NEXT mode is used to read the employee index, and the department index is read in an EXACT mode. The retrieved record location from both indexes is maintained in a system default element.

The flowchart below illustrates the report program logic.



TRANSFER

Transfer means to change control by directing the report program to execute instructions at a specified level. In report programs, transfer of control is invoked when an indexed access results in V2<>0. This condition occurs when an access in a NEXT or AS IS mode reaches the end of file, or an EXACT mode fails to locate a matching key.

In a generated report program, transfer of control to Level 0 is automatically established. After final processing at Level 0, the program terminates. If there is no Level 0 processing, a report program initiates a normal termination after a transfer to Level 0.

The transfer level in a report program may be set at each program level.

A transfer instruction at a report level is executed after the indexed access I/O. Therefore, a transfer should be set at the level preceding the level at which the transfer is required.

Report programs which employ indexed access at multiple levels may require that a transfer of control operator be selected. Transfer of control criteria is specified under Record Selection (see Section 4.2.4). A set transfer remains in effect throughout the program unless reset at another level.

REJECT

A report reject, like a transfer, changes control by directing the program to execute instructions at a specified level. The reject function is invoked by record selection operations, and is also used for program continuation logic after a record has been processed by all report levels. Therefore, a set reject changes program control after a record is rejected, and after a record completes report processing.

The Select Records function (see Section 4.2.2) is used to set program reject levels.

In a generated program, the reject level is automatically established for Level 1. If a report is designed to access another level after a record has completed processing, the appropriate reject must be set.

A set reject remains in effect throughout the program unless reset at another level.

8.4.2 REPORT DEFINITION

Execution of Report Definition functions, available for each program level, must be preceded by File Selection (see Section 4.1). Upon invoking the Report Definition function, the user specifies the program to be defined and the level for which definition is to be specified. The program then displays either of two menus.

If Level 0 definition is chosen, the menu invoked offers title selection and final output processing. Level 0 definition functions are unique to a report and not level-dependent:

- Report Title Selection
- Final Output Processing

The menu for definition of levels 1-9 provides for specification of calculations, record selection, statistics and breaks at the selected level. An entry point to the synonym maintenance facility is also provided. Any or all definition functions may be executed at each level:

- Select Calculations
- Select Records
- Select Statistics
- Select Breaks
- Synonym Maintenance

TITLES

Only one title may be selected for a report. A selected title is printed at the top of every report page. If a title is selected for a report, it is printed before any report data. Title Selection also provides an opportunity to specify report form length and a header to be printed automatically beneath the selected title.

FINAL OUTPUT PROCESSING

Final Output Processing is used to print selected output formats after all report levels have been executed. This end-of-job specification is used to print such things as accumulated report statistics (statistic formats).

CALCULATIONS

The Select Calculations function (see Section 4.2.1) is a program directive to perform a mathematical operation at a selected level. This operation must have previously been defined as a synonym value (see Section 4.5.3). A calculation must be selected at the appropriate level before it can be printed in a report output, or used in break or record selection criteria.

Components of a calculation must be available at the selected report level. Consider the synonym NET, defined as:

```
    NET=GROSS-DEDUCTIONS
```

In order to select this calculation, the synonyms GROSS and DEDUCTIONS must be available (accessed and/or computed) at or before the current report level.

A calculation may not use the same synonym on both sides of the = sign. FORCE disallows this type of circular calculation. For example, the calculation

```
    PAYRATE=PAYRATE*PERCENTAGE
```

is invalid. If this type of calculation is required, a program statistic which yields the same value may be defined. If a statistic cannot meet the requirement, a circular calculation can be avoided by establishing one of the duplicate values as a separate synonym.

It is important to distinguish between a calculation and a statistic. A calculation is a selected equation which uses report synonyms and/or user-specified values. A synonym is assigned to the result of the mathematical operation. A statistic is a predetermined mathematical operation (average, count, high or low value, total) to be maintained on report data.

Calculated synonyms are denoted in queries and Dictionary documentation by an asterisk.

RECORD SELECTION

Record selection, the ability to selectively accept or reject records, is a key function in many reports. Record selection criteria is specified in conditional sentences (see Section 4.2.2). Each sentence represents an executable, high-level program instruction.

Record selection criteria may be specified at each program level, and may employ any number of selection statements. The three record selection directives are:

- ACCEPT - retain the current record and continue evaluation against other record selection statements.
- REJECT - reject the current record; set control to the current reject level.
- SELECT - select the current record, discontinue evaluation at the current level and proceed to the next step.

The Select Records function may also be used to set a program reject or transfer level (see 8.4.1 for transfer level). A reject level is the level to which program control is transferred after a record is rejected.

Program reject levels are automatically set to Level 1 unless otherwise specified. A set reject level remains in effect throughout the program unless reset at another level.

Reject instructions are executed in the order in which they are specified among record selection criteria. Multiple rejects may be set within a report, or within a report level.

STATISTICS

The Select Statistics function (see Select 4.2.3) is employed only to maintain synonym values to be used but not printed in a report (in record selection or break criteria). Otherwise, the Select Statistics function should not be used.

Report statistics are established and selected by specifying the print of a statistics format (for a break or final output processing). Specification to print a statistics output format also serves as a directive (implied specification) to maintain the related statistics values.

BREAKS

Report data is printed by specifying level break criteria. A break is an interrupt of the file-read process, and a directive to print an output format (or multiple formats). A report must have at least one break.

Breaks are selected (see Section 4.2.4) at the report level(s) where accumulated data is to be printed. Header, detail and statistics formats may be printed at each break. A level may employ multiple break statements, and differing output formats may be printed for each. After execution of a break instruction, the program continues to the next report level.

Many programs employ an unconditional break statement to force a break at a report level. Unconditional break criteria use the logic BREAK IF SYNONYM=ITSELF. For example:

```
    BREAK IF EMPNUM EQUALS EMPNUM
```

An unconditional break invokes print of the specified format(s) after selection of each record.

Break statements are executed by a report program in the order in which they are specified.

In the following report example, employees are listed by department number. An unconditional break prints a detail format for each employee. Each time the department number changes, a new header is printed to show that the list now contains employees in a different department. The program's break statements are shown below:

```
    BREAK IF DEPARTMENT CHANGES
    BREAK IF EMPNUM EQUALS EMPNUM
```

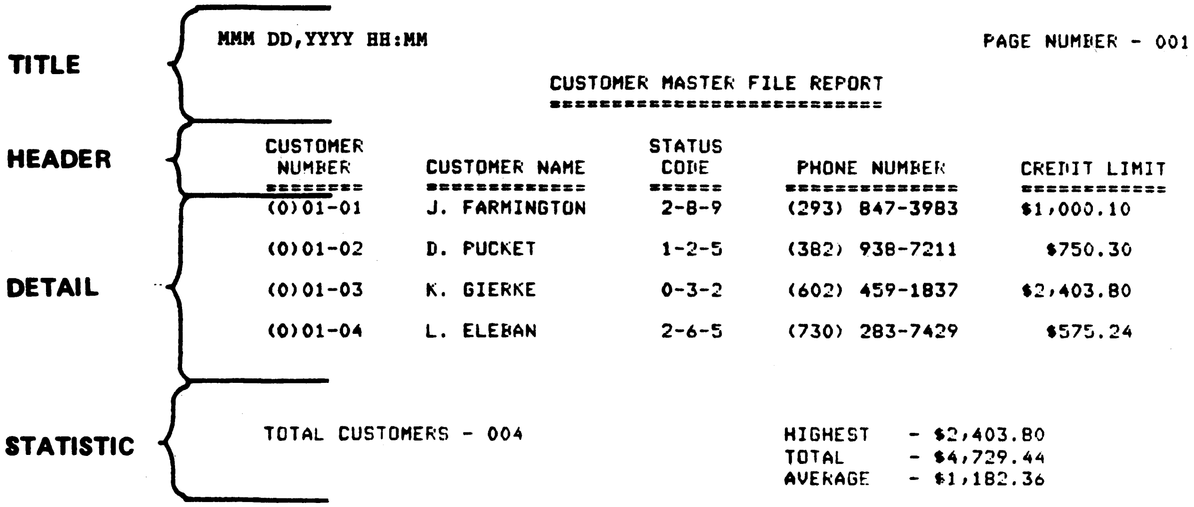
The unconditional break (IF EMPNUM EQUALS EMPNUM) must follow the other break to ensure that employee records are printed with the proper header.

If the order of these break specifications were reversed, the first record of a new department would be printed with the previous department's records. This would occur because the break which prints the employee detail record is executed before the break which prints the new department header.

8.4.3 OUTPUT FORMATTING

All information to be printed in a report must be included in an output format. Report data is assigned to various output formats, which are specified for printing at report breaks.

There are four types of output formats: Title, Header, Detail and Statistics. Each type of output format is demonstrated by the sample report below.


TITLE
HEADER
DETAIL
STATISTIC

```
MMM DD,YYYY HH:MM                                            PAGE NUMBER - 001
                                 CUSTOMER MASTER FILE REPORT
                                 ===========================
    CUSTOMER                            STATUS
     NUMBER     CUSTOMER NAME            CODE     PHONE NUMBER     CREDIT LIMIT
    ========    =============           ======   ==============    ============
    (0)01-01    J. FARMINGTON            2-8-9   (293) 847-3983    $1,000.10

    (0)01-02    D. PUCKET                1-2-5   (382) 938-7211      $750.30

    (0)01-03    K. GIERKE                0-3-2   (602) 459-1837    $2,403.80

    (0)01-04    L. ELEBAN                2-6-5   (730) 283-7429      $575.24


    TOTAL CUSTOMERS - 004                        HIGHEST   - $2,403.80
                                                 TOTAL     - $4,729.44
                                                 AVERAGE   - $1,182.36
```

Each output format contains different types of information. The chart below shows valid contents for each output format type:

Output Formats	Title	Header	Detail	Statistics
Page/Date Functions	X			
Literals	X	X		
Data (Synonyms)	X		X	
Statistics				X
Masks	X		X	X
Round/Truncate Data	X		X	X
Macros Available		X	X	

MASKING

In report programs, masks for printed data are established during output formatting. The same guidelines apply for masking data in reports as in data entry programs (see Section 8.3.1), except that rounding and truncation may be specified as part of the field formatting operation.

A synonym may use a different mask each time it is included in an output format.

APPENDICES

Appendix A
GLOSSARY

The definitions in this glossary describe terms as they pertain to the FORCE Automatic Programming System.

Break - in a report, a directive to interrupt normal program logic and to print data (in assigned output formats)

Byte - a single keyboard character or blank space

Continuation Logic - in a data entry program, a specification that determines whether program input continues if an indexed file check locates a matching key

Conversion - a process performed by the Linkage Editor, which creates a BASIC source code text file from a labeled source code text file

Cursor Tracking - the ability to move the cursor horizontally and vertically on the CRT screen for screen display design

Cursor Addressing - (see Cursor Tracking)

Data Base - (see Data Dictionary)

Data Dictionary - a set of files in which FORCE stores and maintains system/application specifications

Data Element - an IRIS Business BASIC variable name used in a FORCE application

Data Entry Program - a FORCE-generated file maintenance program which allows the user to: add, modify, delete and query records; verify inputs; insert keys; perform indexed file checks and mask displayed data

Detail (report) - the portion of a printed report containing retrieved data records, usually representing units of a record classification

Default - to press RETURN at an input field without entering data, automatically invoking a predetermined input

Dictionary - (see Data Dictionary)

Disk File - an IRIS file generated on disk; the file name used in a program for OPEN statements

Entry Program - (see Data Entry Program)

Expansion - the process by which the Linkage Editor generates source code from macro command lines

Filler Character - a character to be printed or displayed with data when the current data value occupies only a portion of the specified length

FORCE File - a pseudo file defined to FORCE for the purposes of record layout specifications and subsequent system documentation

Format Mask - a specified arrangement of characters to be printed or displayed with a data element

Function (report) - a statistical operation to be performed on synonym data: Average, Count, High Value, Low Value and Total; a statistic

Generation - the process by which FORCE creates source code from application specifications stored in the Dictionary

Header (report) - identifying, descriptive information, usually printed above columnal data in a report

Horizontal Position - a column position on a screen or output format

Justify - to laterally adjust data for print or display (right or left justify)

Key Construct - a data element, or an arrangement of concatenated data elements in an index directory

Label - a descriptive word or notation, used (instead of a line number) to reference an entry point in a program/routine

Labeled Source Code - BASIC source code, without line numbers, which uses labels to reference program entry points

Level - 1:(in Report Generation) a specified file for access in the acquisition of report data 2:(in FORCE Packaging) developmental/sales package of the FORCE system

Level 0 - processing that is independent of report levels 1 through 9; Final Processing and Title Selection

Macro - a modified REM statement (REM + MACRO) that directs the Linkage Editor to generate source code based upon macro command line specifications

Mask - (see Format Mask)

Master System Control Executive - the FORCE master menu

Menu Program - a FORCE-generated, interactive program which provides the user selective access to a group of programs

Purge - to completely remove all components of a system from the Dictionary

Range Check - input validation by ensuring that data falls between two specified ASCII characters

Record - a set of consecutive fields relating to a common subject

Record Layout - a sequential arrangement of fields within a file record

Regeneration - generation of a program, incorporating updated Dictionary specifications, over a program of the same name which had been previously generated

Reindex - a manager function which re-inserts record pointers in the Dictionary index files to ensure the integrity of records

Reorganization - a manager function which analyzes and restructures Dictionary data files to increase accessing efficiency and validate records

Report Program - a FORCE-generated program that prints a report according to user specifications

Source Code - a group of high-level language statements which may comprise a program

Statistic - (also "function" in Report Preparation) a statistical value to be printed or maintained in a report; Average, Count, High Value, Low Value and Total

Sub-function (report) - a statistical value, calculated on a specified portion of data, whose value is reset to zero after it is printed

Submenu - a menu that is invoked from a previous menu selection

Synonym - a descriptive word that references a data element

System - 1:(general FORCE use) the name of an application, or partial application, used to reference components (files, screens, etc.) in the Dictionary 2:(in manager functions) the computer system

Text File - a disk file that contains string data, usually in the form of program source code

Title (report) - information printed at the top of each report page

Transfer - in a report program, to change control by directing the program to execute instructions at a specified level

Vertical Position - a row position on a screen or output format

COMMENT SHEET

MANUAL TITLE FORCE Automatic Programming System User Manual

PUBLICATION NO. AM-140-0023 REVISION B

FROM: NAME/COMPANY: ____________________

BUSINESS ADDRESS: ____________________

CITY/STATE/ZIP: ____________________

COMMENTS: Your evaluation of this manual will be appreciated by POINT 4 Data Corporation. Notation of any errors, suggested additions or deletions, or general comments may be made below. Please include page number references where appropriate.

NO POSTAGE NECESSARY IF MAILED IN U.S.A.
Fold on Dotted Lines and Tape

NO POSTAGE
NECESSARY
IF MAILED IN
UNITED STATES

BUSINESS REPLY MAIL

FIRST CLASS PERMIT NO. 5755 SANTA ANA, CA

POSTAGE WILL BE PAID BY ADDRESSEE

POINT 4 Data Corporation
PUBLICATIONS DEPARTMENT
2569 McCabe Way
Irvine, CA 92714

CUT ON THIS LINE

POINT 4 DATA CORPORATION
2569 McCabe Way / Irvine, California 92714 / (714) 754-4114