SC21-7744-2

File No. S38-40

IBM System/38

IBM System/38 Programmer's/User's Work Station Guide

Program Number 5714-SS1

SC21-7744-2

File No. S38-40

IBM System/38

IBM System/38 Programmer's/User's Work Station Guide

Program Number 5714-SS1

Third Edition (December 1981)

This is a major revision of, and obsoletes, SC21-7744-1 and Technical Newsletter SN21-8192. Changes or additions to the text and illustrations are indicated by a vertical line to the left of the change or addition.

This edition applies to release 3, modification 0 of the IBM System/38 Control Program Facility (Program 5714-SS1), the Interactive Data Base Utilities (Program 5714-UT1), and to all subsequent releases until otherwise indicated in new editions or technical newsletters. Changes are periodically made to the information herein; these changes will be reported in technical newsletters or in new editions of this publication.

This publication contains examples of data and reports used in daily business operations. To illustrate them as completely as possible, the examples include the names of individuals, companies, brands and products. All of these names are fictitious and any similarity to the names and addresses used by an actual business enterprise is entirely coincidental.

It is possible that this material may contain reference to, or information about, IBM products (machines and programs), programming, or services that are not announced in your country. Such references or information must not be construed to mean that IBM intends to announce such IBM products, programming or services in your country.

Use this publication only for the purposes stated in *About This Manual*.

Publications are not stocked at the address below. Requests for copies of IBM publications and for technical information about the system should be made to your IBM representative or to the branch office serving your locality.

This publication could contain technical inaccuracies or typographical errors. Use the Reader's Comment Form at the back of this publication to make comments about this publication. If the form has been removed, address your comments to IBM Corporation, Publications, Department 245, Rochester, Minnesota 55901. IBM may use and distribute any of the information you supply in any way it believes appropriate without incurring any obligation whatsoever. You may, of course, continue to use the information you supply.

Contents

About This Manual

PURPOSE OF THIS MANUAL

This publication describes how to use an interactive work station to communicate with System/38, in particular using the functions available through the Control Program Facility (CPF). It also gives a complete description of how to use programmer and user call menus, command prompting, and system request menu.

ORGANIZATION OF THIS MANUAL

The publication is divided into three parts:

- *Part I. Basic Work Station Operations* is intended for all work station users, especially those who create their own interactive job on the system by signing on. This part describes how to:
  - Sign on
  - Use the work station keyboard and displays on the work station screen to enter information into the system
  - Use the message handling facilities provided by CPF

- *Part II. Basic CPF Operations* is intended for work station users who request application programs by using the program call menu. This part describes how to use the menu to:
  - Call a program
  - Display messages
  - Send messages to the system operator
  - Sign off

- *Part III. Operations Using CPF* is intended for programmers and others who request system functions by entering control language (CL) commands at a work station. This part describes how to enter commands, using the command entry display and the prompting facilities provided by CPF. It then describes how to use commands to:
  - Display and send messages
  - Request functions of the Interactive Data Base Utilities (IDU) or call particular application programs
  - Sign off

Three appendixes provide supplementary information:

- Command function key assignments for licensed programs available from IBM (Appendix A)

- What to do if you suspect a problem with your work station (Appendix B)

- Glossary of Terms and Abbreviations

Note: Both 12- and 24-line display screens are used in this manual. If you are using a 12-line display screen, you can see the additional information if available, by using either the Roll key or the Enter/Rec Adv key, whichever is applicable.

WHAT YOU SHOULD KNOW

To use this manual effectively, you must know how to operate the 5251/5252 display stations. The following manuals contain the information you need:

- *IBM 5251 Display Station Models 1 and 11, IBM 5252 Dual Display Station Operator's Guide,* GA21-9248

- *IBM 5251 Display Station Models 2 and 12 Operator's Guide,* GA21-9323

IF YOU NEED MORE INFORMATION

To correct a problem you may need to refer to another IBM publication for a specific type of information.

Messages

- *IBM System/38 Messages Guide: CPF, RPG III and IDU,* SC21-7736
  - Contains all messages other than COBOL.
  - Contains related message information.

- *IBM System/38 Messages Guide: COBOL,* SC21-7823
  - Contains COBOL messages.
  - Contains related message information.

CPF Commands and Functions

- *IBM System/38 Control Language Reference Manual,* SC21-7731
  - Contains commands and parameters that are used for the various CPF functions.

- *IBM System/38 Control Program Facility Programmer's Guide,* SC21-7730
  - Explains how to use control language commands.

- *IBM System/38 Operators's Guide,* SC21-7735
  - Describes the operation of the system console and differences between the system console and the 5251/5252 display stations.

IDU

- *IBM System/38 Data File Utility Reference Manual and User's Guide,* SC21-7714
  - Describes the function and use of the Data File Utility.

- *IBM System/38 Source Entry Utility Reference Manual and User's Guide,* SC21-7722
  - Describes how to create and update source records using Source Entry Utility.

- *IBM System/38 Query Utility Reference Manual and User's Guide,* SC21-7724
  - Describes query for programmers and work station users who want to analyze information selected from their System/38 data base.

- *IBM System/38 Screen Design Aid Reference Manual and User's Guide,* SC21-7755
  - Describes the function and use of the Screen Design Aid Utility.

Problem Determination

- *IBM System/38 Problem Determination Guide,* SC21-7876
  - Contains procedures for resolving system problems that are indicated by error messages, operator/service panel lights, interactive/batch jobs, spooling functions, or devices that do not work as expected.

General

- *IBM System/38 Application Example I,* SC21-7881
  - Describes the use of the Programmer Menu in association with IDU utilities.
  - Describes the use of the Programmer Menu to create programs and files.
  - Gives examples of using the Program Call Menu.
  - Gives examples of creating and using files and programs from a work station.

- *IBM 5250 Information Display System Introduction,* GA21-9246
  - Describes the general features of the 5251/5252 work stations and the 5256 work station printer.

- *IBM 5250 Information Display System Functions Reference Manual,* SA21-9247
  - Describes the programming requirements for communicating with the 5250 Information Display System.

- *IBM System/38 Introduction,* GC21-7728
  - Contains System/38 configurations and an overview of the programming system.

- *IBM System/38 Glossary and Master Index,* GC21-7727
  - Contains a glossary of all terms used in System/38 customer publications, and combines index entries from the indexes of frequently used System/38 publications.

- *IBM System/38 Keyboard Template,* GX21-7756
  - Indicates which command function keys are used by a particular licensed program.

This page is intentionally left blank.

Work Stations and System/38

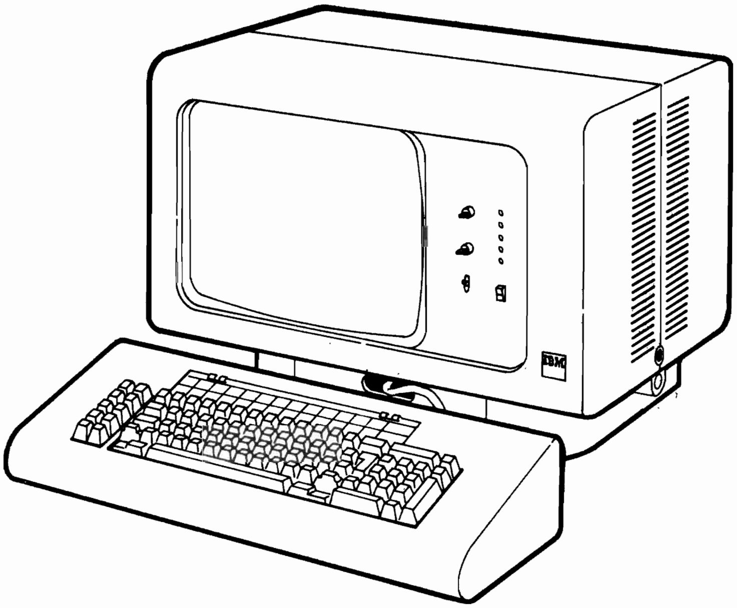
The IBM System/38 is designed so that a variety of users performing different tasks can communicate with it at the same time. You communicate with System/38 by using a work station.

WHAT IS A WORK STATION?

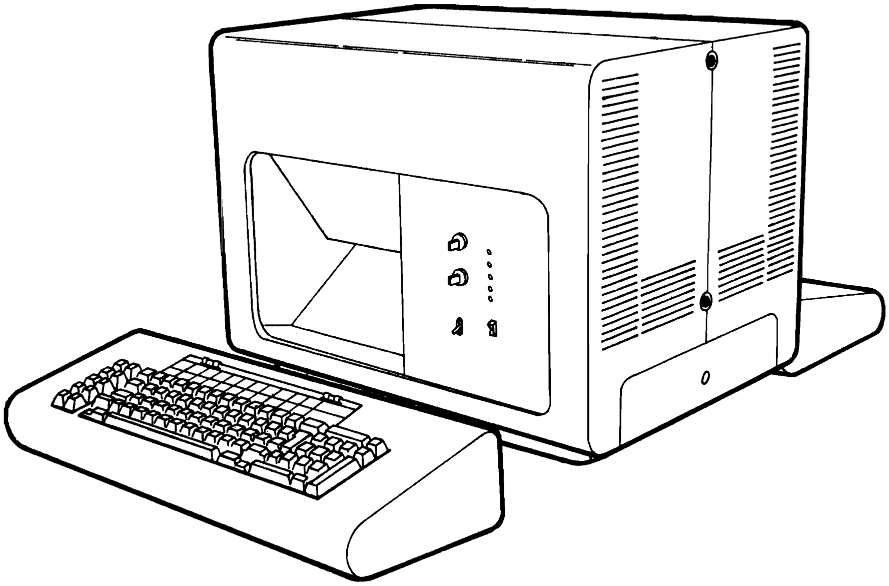
The term *work station*, as used in this publication, refers to an interactive terminal device containing a keyboard and display screen, through which you make requests of the system and receive responses from it. Interactive means that you receive a separate response to each request, as if you were conversing with the system.

The primary work stations used with the System/38 are:

- IBM 5251 Display Station



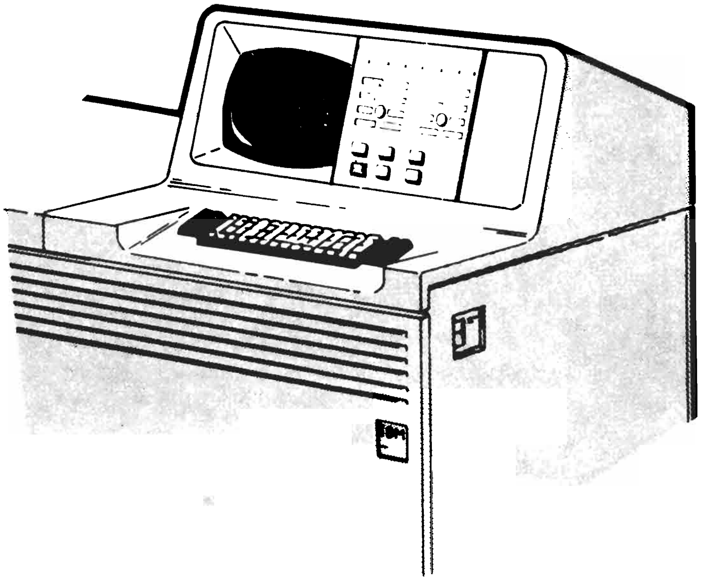
- IBM 5252 Dual Display Station



The 5251 can have either a 1920- or 960-character display screen. The 5252 has back-to-back 960-character displays that are connected to separate keyboards. The same keyboards are used on both the 5251 and 5252. Once you have mastered the use of one display station, you can easily switch to the other.

For general information on the 5251 and 5252, see the *IBM 5250 Information Display System Introduction*. For details on the screen indicators and the keyboards, see the *IBM 5251 Display Station Models 1 and 11, IBM 5252 Dual Display Station Operator's Guide*, or the *IBM 5251 Display Station Models 2 and 12 Operator's Guide.*

The system console, located on the System/38 processing unit, is intended primarily for use by the system operator, but it can also be used as a work station.



Although its screen indicators and keyboard are somewhat different from the 5251 and 5252, its operation as a work station is similar. However, some of the features of the 5251/5252 work stations are not available on the system console. If such features are used by your programs, they may not operate the same way on the system console as on the 5251/5252 work stations. For details of the system console, see the *IBM System/38 Operator's Guide*.

The order numbers for all referenced publications are given in *About This Manual*.

HOW TO USE THIS PUBLICATION

This publication is arranged in three parts, beginning with the basic work station functions in Part I and progressing to the more advanced functions using Control Program Facility (CPF) commands in Part III. The parts of the publication that apply to you depend on the type of functions you will be using at the work station. When you begin your job at the work station, you receive a display that is appropriate to the functions you will be using. This initial display, therefore, provides a simple guide to which parts of the publication you should read. The following discussion summarizes the types of displays you could receive when you begin your work station job and how these displays relate to this publication. Note that this text contains keys that point to the diagram that follows the discussion.

1 *No Sign-On*: If you receive a working display when you set the Power switch to the On position, your work station is under control of a specific application program that has already created a working session for you.

Information Source: Refer to the document describing the specific application for information on how to use the work station. You may also want to read Chapter 1 of this publication, which describes the use of the work station keyboards and displays in communicating with the system.

2 *Sign-On Required:* If you receive the sign-on display, you must sign on to create your own interactive job on the system.

Information Source: Refer to Chapter 2 of this publication for the information you need to sign on, including the basic procedure for entering information into the system using the work station keyboard and displays.

After your sign-on is completed, you receive one of four types of displays.

A *Initial Application Display*: If your system is set up for you to use a specific application, such as order entry, control of your work station is transferred to the application programming and you receive a display that is designed for the application.

Information Source: For information on how to use the application at the work station, refer to the instructions or document provided for the application. You should also read all of Part I of this publication.

B *Routing Display*: For some types of sign-on, the sign-on display is followed by a *routing* display that contains an input area in which you are to enter additional information. The information you provide then determines which display you receive next.

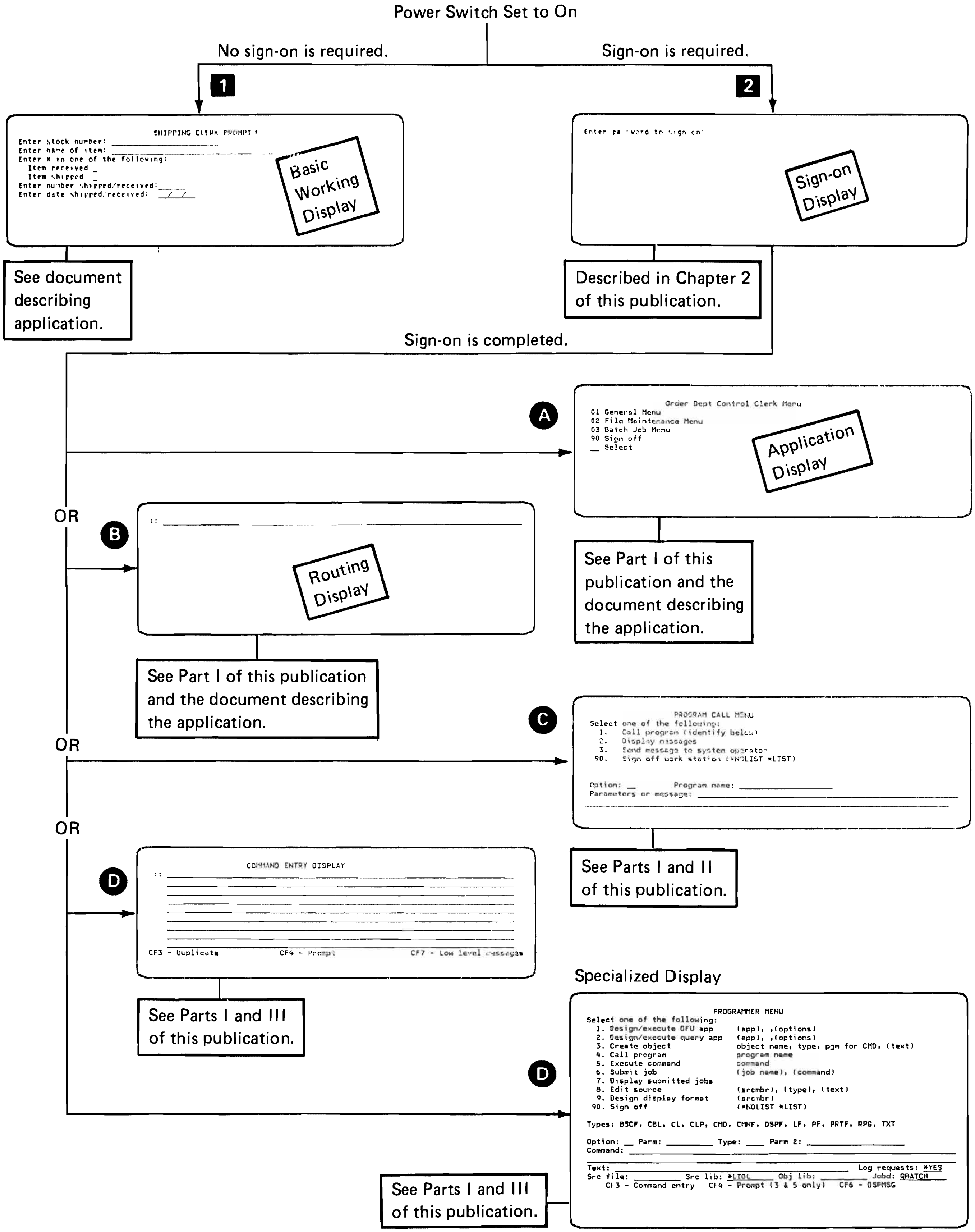
Information Source: Refer to the instructions or document provided for the application. You should also read Part I of this publication.

Ⓒ *Program Call Menu*: If your system is set up for you to be given access to various application programs, you receive a basic CPF display known as the *program call menu*. Using this menu, you can call a program tailored for a specific type of work you want to do, as well as perform other basic CPF functions. You could also receive the program call menu if you use a default sign-on as described in Chapter 2 of this publication.

Information Source: For information on how to use the program call menu, refer to Part II of this publication. You should read Part I before going on to Part II.

Ⓓ *Command Handling Displays*: If your system is set up for you to enter CPF control language commands, you receive a *command entry display or a specialized menu* on which you can enter or select individual command functions.

Information Source: You should read, or at least review, Part I of this publication and then, for information on using command entry facilities or the programmer menu at a work station, refer to Part III of this publication.



```
                   SHIPPING CLERK PROMPT
Enter stock number: ________________
Enter name of item: ________________________
Enter X in one of the following:
  Item received _
  Item shipped  _
Enter number shipped/received:_____
Enter date shipped/received:  __/__/__
```

See document describing application.

```
Enter password to sign on
```

Described in Chapter 2 of this publication.

```
              Order Dept Control Clerk Menu
01 General Menu
02 File Maintenance Menu
03 Batch Job Menu
90 Sign off
__ Select
```

See Part I of this publication and the document describing the application.

```
:: ____________________________________________
```

See Part I of this publication and the document describing the application.

```
                 PROGRAM CALL MENU
Select one of the following:
  1.   Call program (identify below)
  2.   Display messages
  3.   Send message to system operator
 90.   Sign off work station (*NOLIST *LIST)

Option: __        Program name: ____________
Parameters or message: ______________________________
```

See Parts I and II of this publication.

```
                  COMMAND ENTRY DISPLAY
:: ______________________________________________

CF3 - Duplicate        CF4 - Prompt        CF7 - Low level messages
```

See Parts I and III of this publication.

```
                       PROGRAMMER MENU
Select one of the following:
  1. Design/execute DFU app    (app), ,(options)
  2. Design/execute query app  (app), ,(options)
  3. Create object             object name, type, pgm for CMD, (text)
  4. Call program              program name
  5. Execute command           command
  6. Submit job                (job name), (command)
  7. Display submitted jobs
  8. Edit source               (srcmbr), (type), (text)
  9. Design display format     (srcmbr)
 90. Sign off                  (*NOLIST *LIST)

Types: BSCF, CBL, CL, CLP, CMD, CMNF, DSPF, LF, PF, PRTF, RPG, TXT

Option: __ Parm: ________ Type: ___ Parm 2: ______________
Command: _______________________________________________________
________________________________________________________________
Text: _______________________________________ Log requests: *YES
Src file: ________ Src lib: *LIBL____ Obj lib: ________ Jobd: QBATCH___
    CF3 - Command entry   CF4 - Prompt (3 & 5 only)   CF6 - DSPMSG
```

See Parts I and III of this publication.

Part I

Basic Work Station Operations

Chapter 1. Using a Work Station

HOW TO ENTER INFORMATION

To enter information into the system through a work station, you use the work station keyboard as you would a typewriter or keypunch, depending on whether the work station has the typewriter-like keyboard (Figure 1-1) or one of the data-entry keyboards (Figure 1-2).

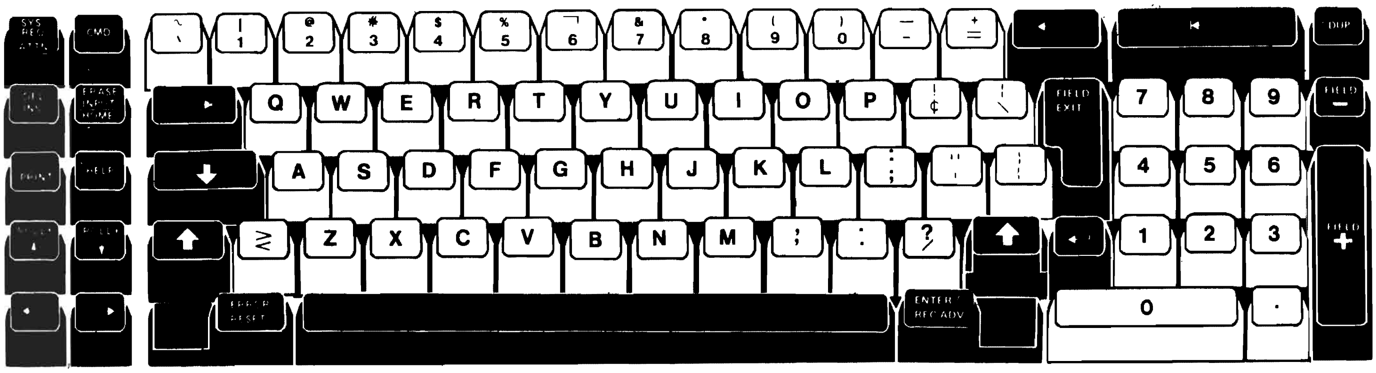


Figure 1-1. 5251/5252 Typewriter-Like Keyboard

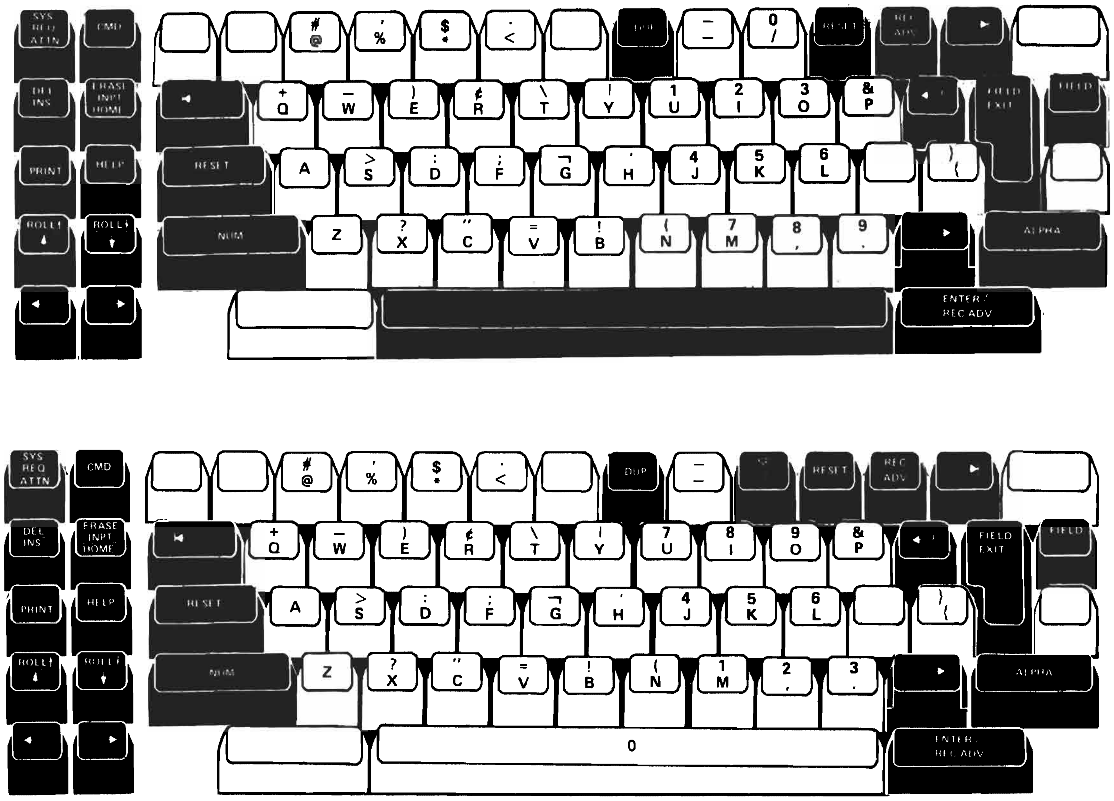


Figure 1-2. 5251/5252 Data-Entry Keyboards

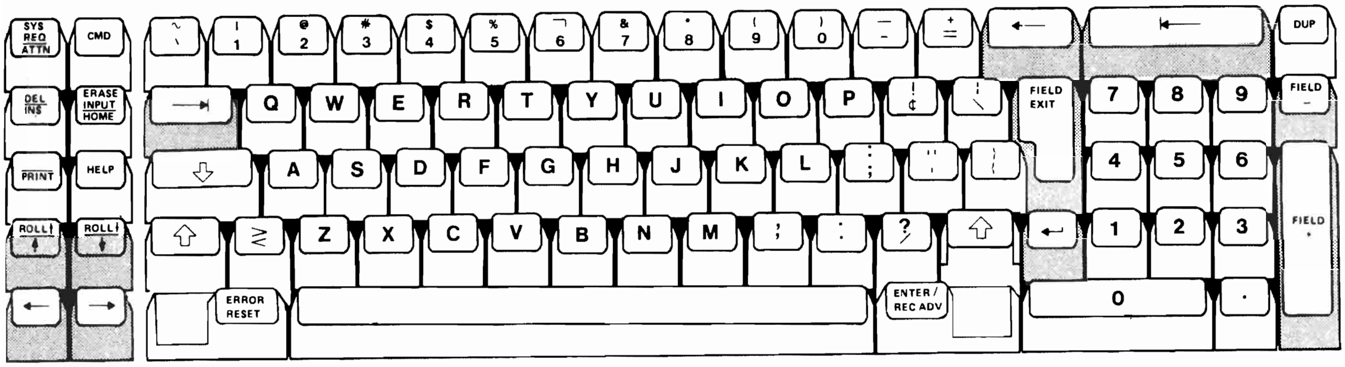
Data Keys and Function Keys

The data keys (light keys in Figures 1-1 and 1-2) are used to enter individual characters (letters, numbers, and symbols). The function keys (dark keys) are used as needed to position the information and request special functions. (For a detailed description of the function keys, see the 5251/5252 operator's guides.)

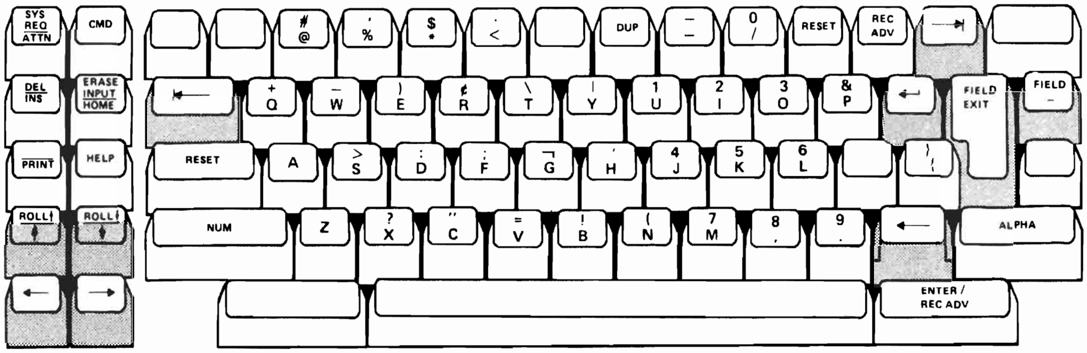
Keying in the Information

As you press each data key, the indicated character appears on the work station screen. This procedure, which is similar to using a typewriter, is referred to as *keying in* the information. A cursor, which looks like a bright underscore, moves along the screen to indicate where the next character will appear.

You can use the cursor movement keys (shaded keys in Figure 1-3) to change the position of the cursor, and thus change the location where the next character will appear.



Typewriter-Like Keyboard



Data-Entry Keyboard

Figure 1-3. Cursor Movement Keys

Using the Shift Keys

As on a typewriter, you obtain uppercase letters or the characters indicated on the upper half of the keys by using an upper shift, that is:

On a typewriter-like keyboard, press the ⇧ (Upper Shift) key.

When you press the ⇩ (Shift Lock) key on the typewriter-like keyboard, the keyboard locks in upper shift. You can return the keyboard to lower shift by pressing and releasing either of the Upper Shift keys. There is no Shift Lock key on the data entry keyboard.

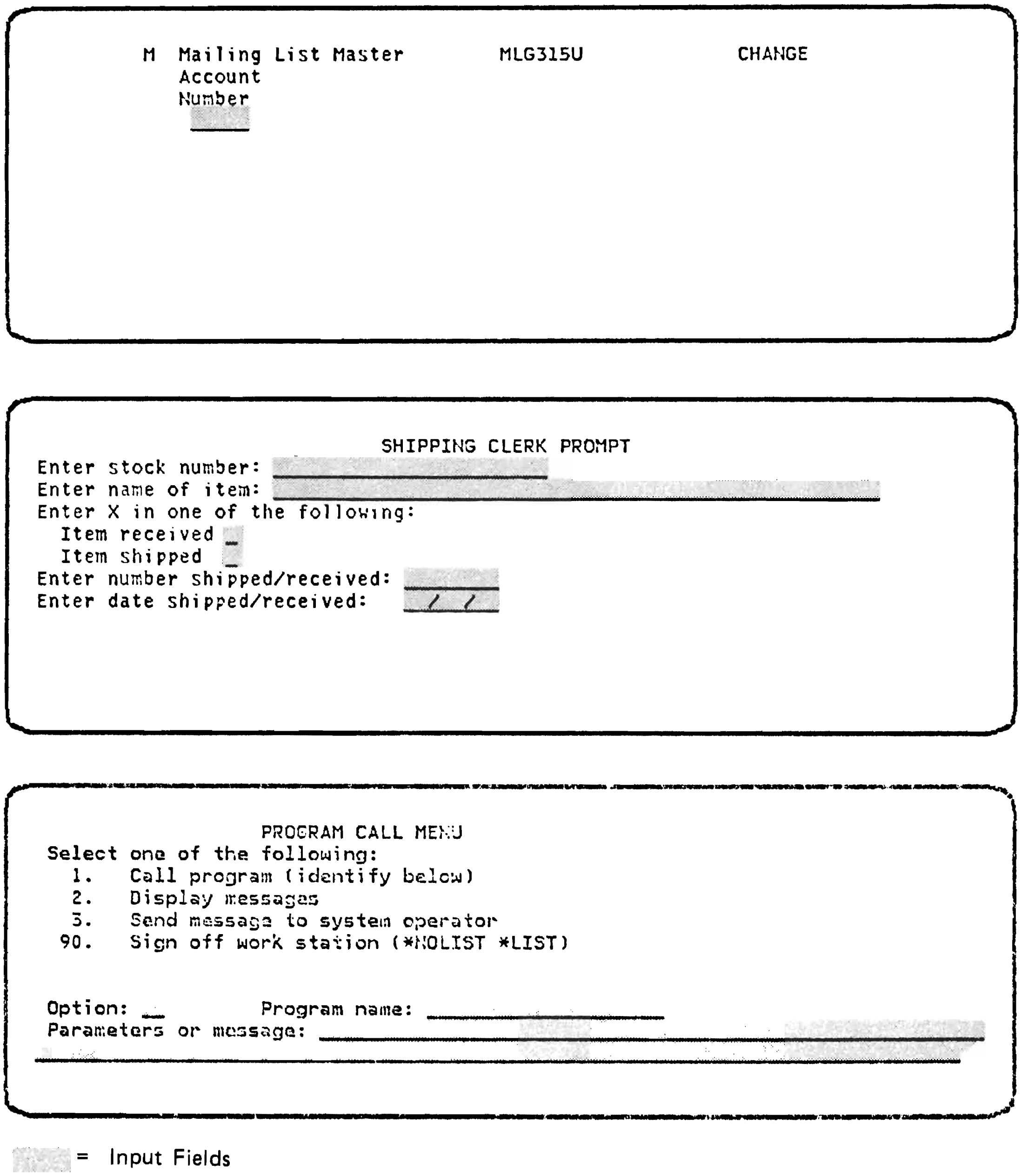
In the remainder of this publication, the term Shift key will be used to mean either of these keys.

Note that on the data entry keyboards, you press the NUM (Numeric Shift) key to enter characters or request functions indicated on the top half of the keys. The data entry keyboards also contain an ALPHA (Alpha Shift) key. You press this key when you want to key in alphabetic characters in a numeric field.

Displays and Input Fields

In communicating with the system, you will make use of a variety of displays on the work station screen. Most of the display examples used in this publication are shown using 12-line screens, such as a 5251 Model 1 or 11 or a 5252.

The displays you use allow you to enter information only in certain areas of the display (Figure 1-4). Those areas are called input fields. On most CPF displays, each input field is indicated by a continuous underscore. The exceptions are displays containing nondisplay fields, such as the sign-on display (see Chapter 2), where the information being entered is not to be displayed. Displays developed specially for your system may indicate input fields differently.



```
           M  Mailing List Master       MLG315U             CHANGE
              Account
              Number
               _____
```

```
                               SHIPPING CLERK PROMPT
 Enter stock number: _______________
 Enter name of item: _________________________________
 Enter X in one of the following:
   Item received _
   Item shipped  _
 Enter number shipped/received: _____
 Enter date shipped/received:  _/_/_
```

```
                   PROGRAM CALL MENU
  Select one of the following:
    1.   Call program (identify below)
    2.   Display messages
    3.   Send message to system operator
   90.   Sign off work station (*NOLIST *LIST)

  Option: __        Program name: _____________
  Parameters or message: _____________________________________
  ____________________________________________________________
```

▨ = Input Fields

Figure 1-4. Examples of Input Fields

When a display containing more than one input field appears on your screen, the cursor is generally located at the first available position of the first input field. When your entry fills an input field, the system automatically repositions the cursor at the beginning of the next input field. If your entry does not fill an input field, you can advance the cursor to the next input field by pressing the FIELD EXIT (Field Exit) key or, if you only want to advance the cursor to the next field without clearing the remainder of your current input field, press the ⇥ (Field Advance) key.

If you are using a display that is formatted for entering numeric data, you can also advance the cursor to the next input field by pressing the FIELD + (Field +) key or the FIELD - (Field -) key.

The input fields of some displays already contain information when those displays appear on your screen. You can change the information in the displays by keying in the new words or characters on top of the old ones.

As indicated in Figure 1-4, a single input field can extend to more than one line of a display. When entering information into such a field, you key in your entry as if you were typing it on one continuous line. When you reach the end of one line, the cursor automatically moves to the beginning of the next line and the characters you are keying in appear on that subsequent line. If you choose, you can break your text at a convenient point before reaching the end of a line. In that case, you press the ↵ (New Line) key to move the cursor to the next line.

Entering the Information

When you have completed all additions or changes to the display currently on your screen, you send the information on the display to the system for processing by pressing the ENTER/ REC ADV (Enter/Rec Adv) key or in some cases one of the command function keys (see *How to Use Command Function Keys* in this chapter). The information is not actually acted on by the system until you press the Enter/Rec Adv key or a command function key. Therefore, you *enter* information into the system using a 2-step procedure:

1. Key in the information; that is, press the appropriate keys so that the characters to be entered appear on the work station screen.

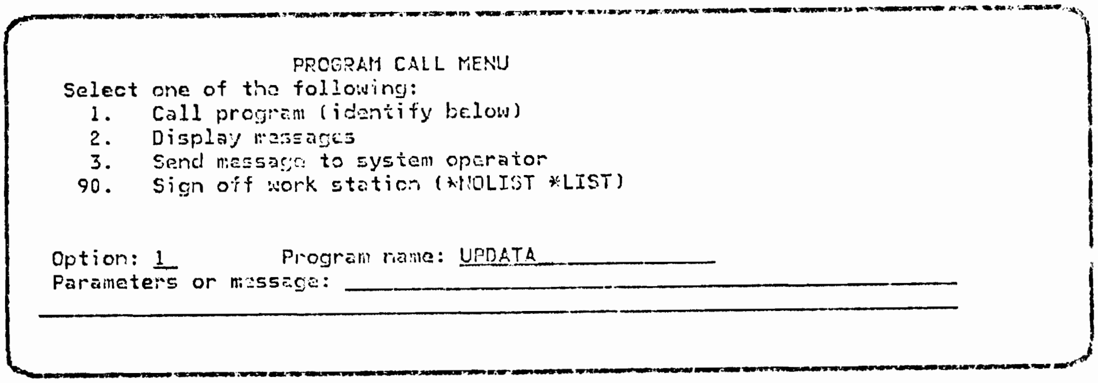
2. Press the Enter/Rec Adv key (or a command function key where so indicated).

HOW THE SYSTEM RESPONDS

When the system processes the information you entered, the response could be indicated on your screen in different ways.

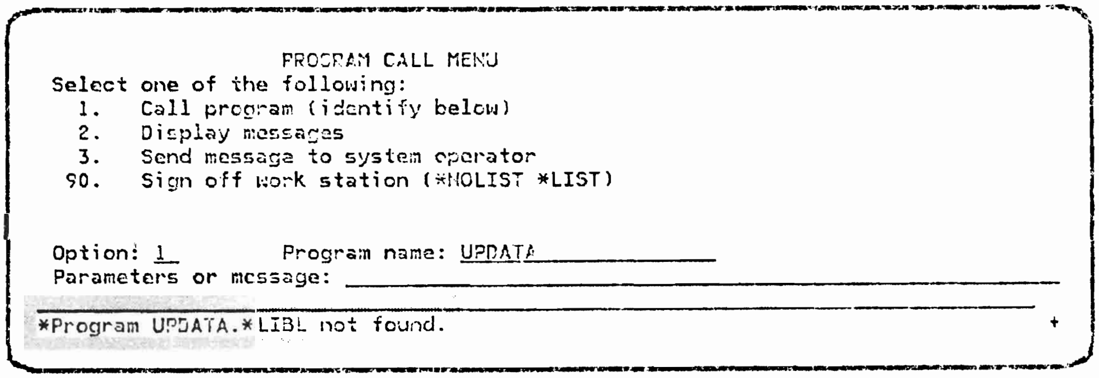
Your current display could be modified to include information related to your request (Figure 1-5), or your current display could be replaced by an entirely new display providing information you requested (Figure 1-6).

A display with your entries.



```
                     PROGRAM CALL MENU
 Select one of the following:
   1.   Call program (identify below)
   2.   Display messages
   3.   Send message to system operator
  90.   Sign off work station (*NOLIST *LIST)


Option: 1_        Program name: UPDATA____________
Parameters or message: ___________________________________________
_____________________________________________________________________
```

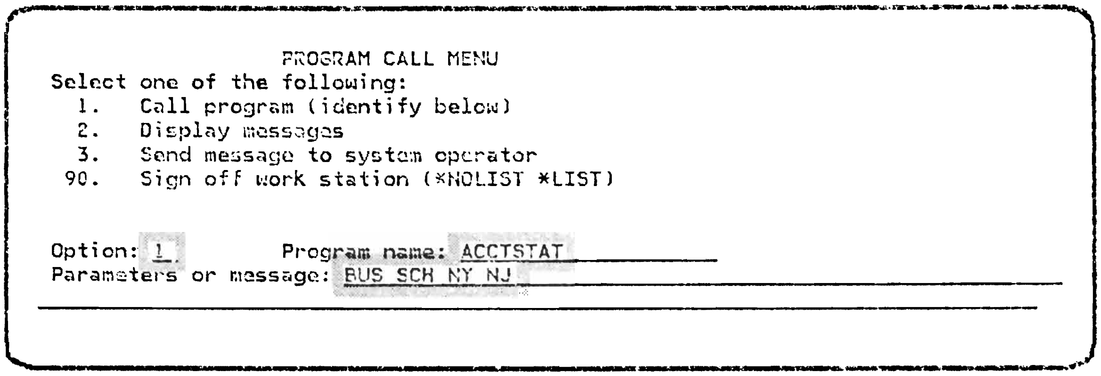
A modified display after system response.



```
                     PROGRAM CALL MENU
 Select one of the following:
   1.   Call program (identify below)
   2.   Display messages
   3.   Send message to system operator
  90.   Sign off work station (*NOLIST *LIST)


 Option: 1_        Program name: UPDATA____________
 Parameters or message: ___________________________________________
_____________________________________________________________________
 *Program UPDATA.*LIBL not found.                                  +
```

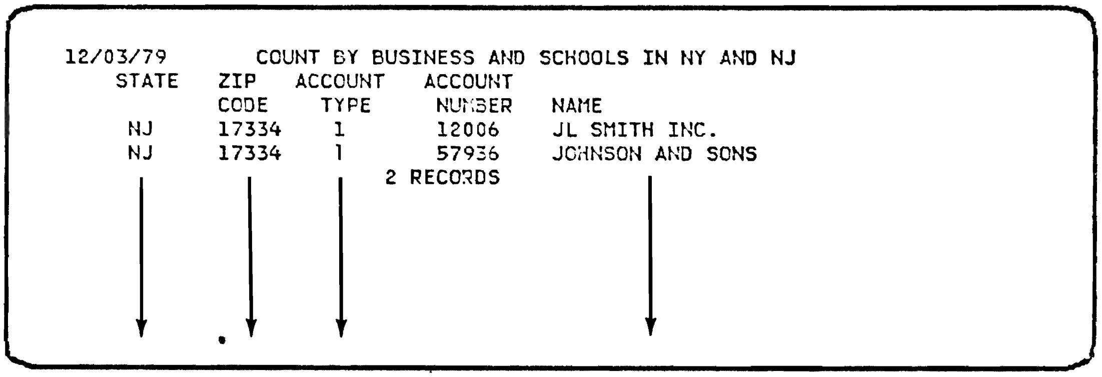
Figure 1-5. Example of Modified Display as Response

An original display with your entries.



```
                    PROGRAM CALL MENU
Select one of the following:
  1.   Call program (identify below)
  2.   Display messages
  3.   Send message to system operator
 90.   Sign off work station (*NOLIST *LIST)


Option: 1         Program name: ACCTSTAT
Parameters or message: BUS SCH NY NJ
```

A new display after system response.



```
12/03/79       COUNT BY BUSINESS AND SCHOOLS IN NY AND NJ
    STATE    ZIP   ACCOUNT   ACCOUNT
             CODE    TYPE     NUMBER   NAME
     NJ      17334   1        12006    JL SMITH INC.
     NJ      17334   1        57936    JOHNSON AND SONS
                           2 RECORDS
```

Figure 1-6. Example of New Display as Response

The sample displays shown here have a format that is typical of displays provided in IBM programs. Displays provided by programs developed specially for your system may have a different format.

HOW TO USE COMMAND FUNCTION KEYS

When you press the Cmd key followed by one of the keys on the top row of the keyboard, either with or without an upper shift, the top row key becomes a command function (CF) key. A CF key alerts the program currently in control of the work station to take an action, and the program may or may not be passed data from the display on the screen depending on which CF key was pressed.

The process of pressing the Cmd key followed by a top row key is referred to simply as pressing a CF key. For example, you might be told to press the CF1 key. You would then press the Cmd key followed by the key on the top row that is associated with CF1.

The relationship between the CF keys and top row keys on the 5251/5252 keyboards is summarized in Figure 1-7.

To obtain command functions 1 through 12 (CF1 through CF12):

1. Press and release the Cmd key.

2. Press the top row key matching the number of the desired CF key as shown in Figure 1-7.

To obtain command functions 13 through 24 (CF13 through CF24):

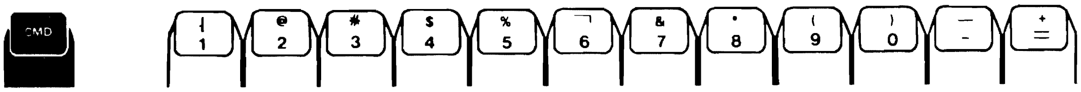
1. Press and release the Cmd key.

2. Press and hold the Shift key.

3. Press the top row key matching the number of the desired CF key as shown in Figure 1-7.

Typewriter-Like Keyboard Top Row of Keys

Upper Shift	CF13	CF14	CF15	CF16	CF17	CF18	CF19	CF20	CF21	CF22	CF23	CF24
Lower Shift	CF1	CF2	CF3	CF4	CF5	CF6	CF7	CF8	CF9	CF10	CF11	CF12



Data-Entry Keyboards Top Row of Keys

Upper Shift	CF13	CF14	CF15	CF16	CF17	CF18	CF19	CF20	CF21	CF22	CF23	CF24
Lower Shift	CF1	CF2	CF3	CF4	CF5	CF6	CF7	CF8	CF9	CF10	CF11	CF12

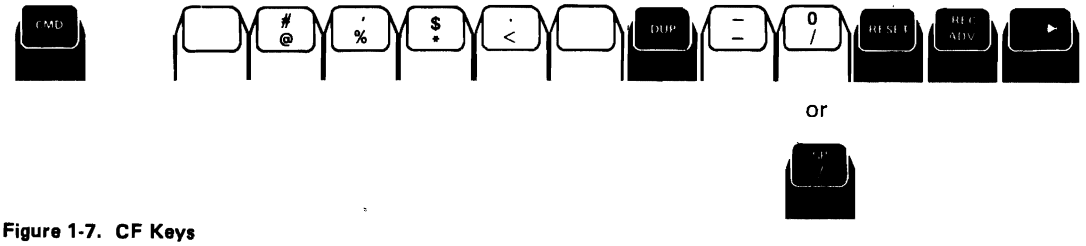


Figure 1-7. CF Keys

The use of most CF keys is defined in each program and, therefore, can vary from one program (or set of displays) to another. The function of some CF keys is indicated directly on the display you are using. In addition, templates are provided that indicate which keys you press to obtain the command functions available for specific programs (Figure 1-8).

Each template is divided into blocks that represent individual keys on the top row of the keyboard. When this template is placed in the depression above the keyboard, each block representing a specific command function is positioned above the appropriate top row key. To obtain the function indicated in the block, you press the key below it. If the block is in the upper row of the template, you must also press the Shift key when you press the top row key below the block. As an additional reference, the number in the upper left corner of each block indicates the CF key number as shown in Figure 1-7. For example, the block containing the number 13 is the CF13 key shown in Figure 1-7.

Specific uses of CF keys are discussed where appropriate in subsequent chapters. The command function assignments for IBM-supplied programs are summarized in Appendix A.

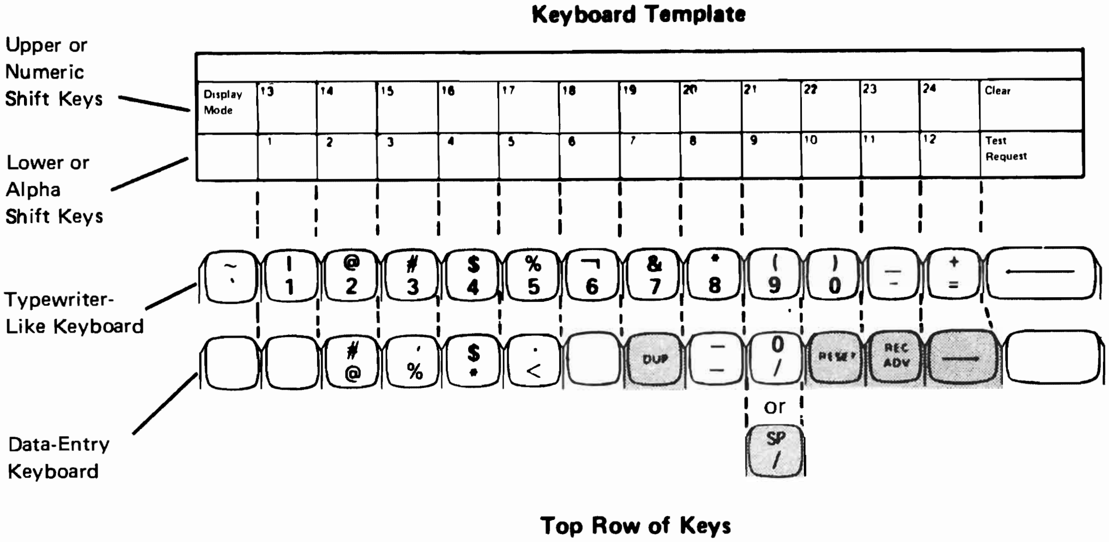


Figure 1-8. Command Function Template

SPECIAL USE OF KEYS FOR SYSTEM/38

Function keys on the 5251/5252 keyboards (dark keys in Figures 1-1 and 1-2) generally perform the functions described in the separate 5251/5252 operator's guides (see *About This Manual*). However, several of these keys have different capabilities when used with System/38 CPF programming. The following summarizes those differences.

System Request (Sys Req) Key: You can use the Sys Req key to interrupt the job you are currently working on. Then you can perform several options, including one which allows you to display the messages on your work station message queue, and another which allows you to create a second interactive job on the system at the same work station, and then transfer back and forth between the original job and the second job. The procedure is described in Chapter 5.

Attention (Attn) Key: The Attn key has no function in IBM-supplied CPF programming. The interrupt or escape function frequently associated with the Attn key is performed by the Sys Req key, as described in Chapter 5.

Clear Key: The function performed by the Clear key varies, depending on the program in control of the work station. For additional information about the Clear key, see the *CPF Programmer's Guide.*

Home Key: The Home key serves as a record backspace key; that is, pressing the Home key returns the cursor to the first position on the screen where you can enter information, or to another home position specified for the display on your screen. If the cursor is already in the home position when you press this key, the program currently in control of your work station determines what happens.

Help Key: The Help key is used to display additional information about a message that you have received. To do this, position the cursor anywhere under the message and press the Help key. If you are using the IDU, and there is no message on the display, you can use the Help key to request help text for your current display.

Roll Keys: The roll keys are used to view additional information not shown on the current display. If there is more information that can be viewed on one display at a time, a plus sign (+) appears in the lower right corner of the area that can be rolled.

Delete Key: The Delete key is used to remove characters from an input field. When the cursor is positioned under a character and the Delete key is pressed, the character above the cursor is deleted and the following characters are shifted one space to the left.

Insert Key: The Insert key is used to insert characters in an input field. If the input field is not filled and the Insert key is pressed, the data after the cursor location shifts one space to the right when the next data key is pressed and that data character is inserted.

Print Key: The Print key is used to print the information currently displayed on the screen of your work station. The way in which the information is printed can be defined in one of several ways for your current display and device. Depending on what was specified, one of the following could occur when you press the Print key:

- The program in control determines the use of the key.
- The displayed information is printed immediately on a printer associated with your work station.
- The information is placed on a queue to be printed along with other information at a later time.
- The printer device file may be spooled.
- Nothing is printed because there is no printer associated with your work station or because the printer associated with your work station is currently not available. You receive a message at your work station indicating either that the print request completed successfully to the default printer device file or that the print request could not be completed. Your job log indicates any problems encountered while the print request is being processed.
- Nothing is printed because the hardware feature on a remote 5250 system is damaged or not available.

Note: The print functions obtained by pressing the Print key on a 5251 or 5252 display station are obtained on the system console by pressing the CF24 key.

Other special uses of function keys are described where appropriate in later chapters of this publication and other System/38 publications.

VALIDITY CHECKING

As you enter information at a work station, the system makes various tests of your input (called validity checking) to help ensure that the information can be processed as you intended. The system will generally detect errors such as:

- Attempting to enter data that is in the wrong format or is the wrong type for a particular input field.

- Requesting a system action on information that is not known to the system. (You may, for example, have misspelled the name of the data to be acted on.)

If an error is detected, the system displays an error code or message on your work station screen. Error messages and other types of messages are described in Chapter 3.

If your input passes the validity tests, the system processes the information as you entered it.

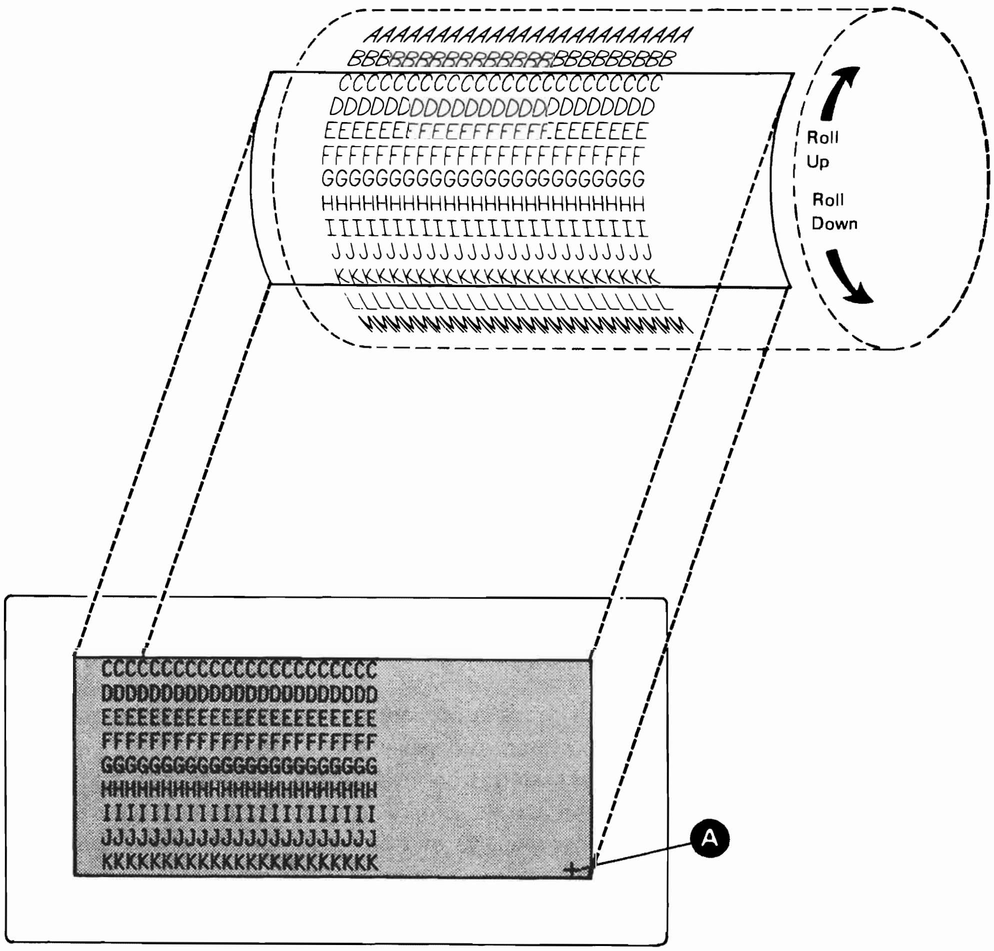
HOW TO HANDLE ROLL CAPABILITY AND MULTIPLE DISPLAYS

Your work station job may involve interacting with a single display or a group of displays. One or more of those displays each may contain many lines of information that can be viewed a few lines at a time on your work station screen. Other types of information may be contained on a series of displays that you can view one after another on your screen. You select the additional lines of information or displays by pressing specific keys or entering a specific request.

The program in control of your work station determines the type of displays you use and how you obtain additional displayed information, if at all. The following discussion describes the procedures for handling CPF displays. If your work station job involves a different type of display or displays, such as those provided for a special application, refer to the instructions for the application. The instructions should indicate whether you handle the displays as described here or in a different manner.

Displays with Roll Capability (+)

Displays, such as those described in Parts II and III of this publication, may contain an area that can be rolled. With this area, new lines of information can be moved onto the screen and old lines moved off. You might imagine the information for this part of the display as being on the outside surface of a drum or cylinder that you can roll forward or backward to view a specific part on your work station screen.



If there is more information than can be viewed on one display at one time, a plus sign (+) appears Ⓐ in the lower right corner of the area that can be rolled. When the + disappears, or if there is no +, there is no more information to be viewed.

You adjust the information by using the roll keys. To move the lines of information forward (up), press the Roll Up key, that is:

1. Press and hold the Shift key.

2. Press the ROLL↑ key.

To move the lines of information backward (down), press the Roll Down key, that is:

1. Press and hold the Shift key.

2. Press the ROLL↓ key.

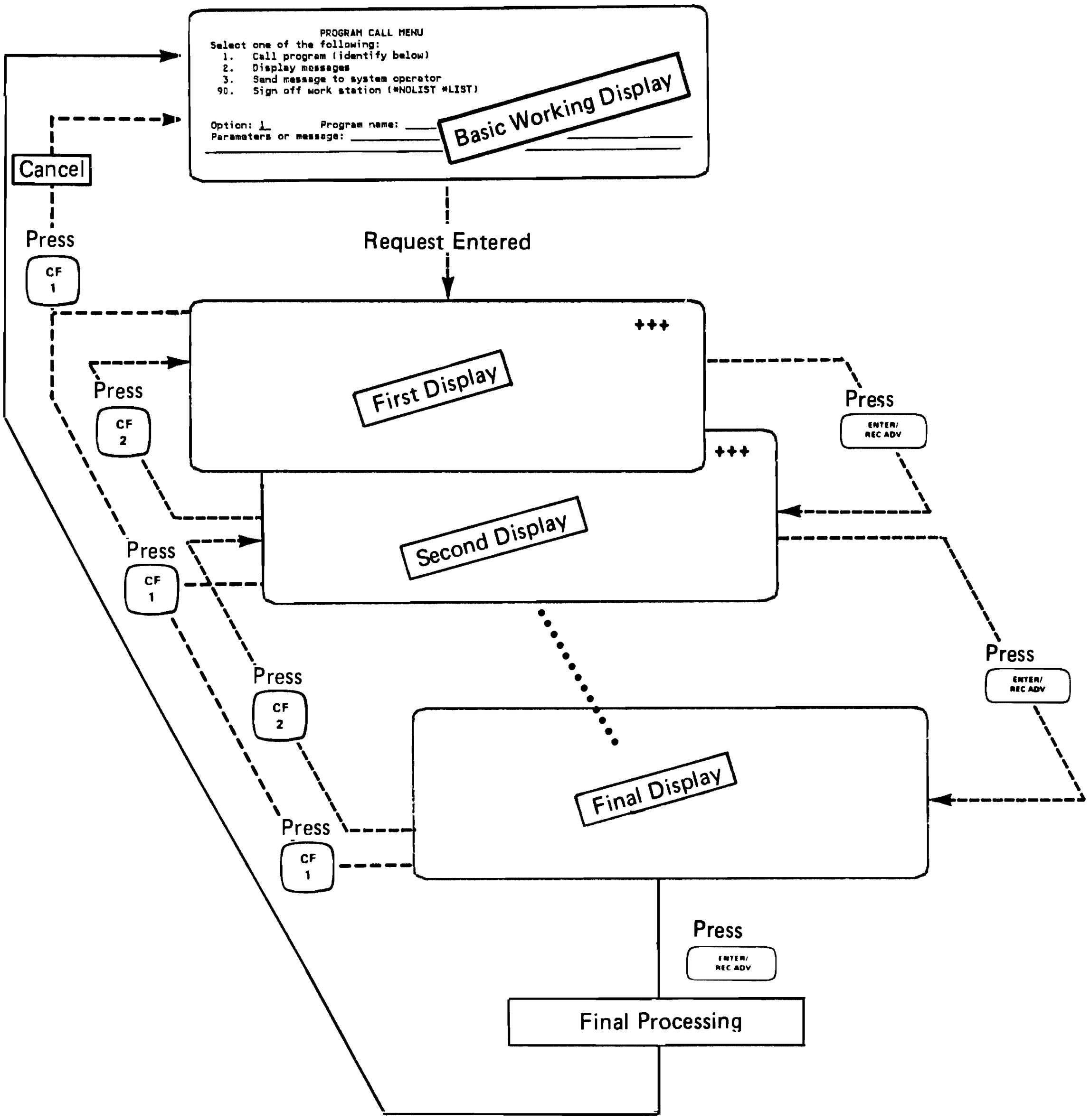
When you reach the end of the information, pressing the Roll Up key has no effect. Similarly, when you are at the beginning of the information, pressing the Roll Down key has no effect. You receive a message at the bottom of your screen informing you of this condition.

Multiple System Displays (+++)

During your work station job you may receive a series of separate system displays that you can view one after another on the work station screen. If this happens, a series of three pluses (+++) appears in the upper right corner of each display, except the last, to indicate that there is another display to follow. You advance to the next display in the series by pressing the Enter/Rec Adv key. When the +++ disappears, you know you are on the final display. You can also back up to the previous display in the series by pressing the CF2 key (see *How to Use Command Function Keys* earlier in this chapter).

If the final display in the series is on your screen when you press the Enter/Rec Adv key, the system processes the information on the entire set of displays (if any final processing is defined for the displays), and you exit the displays; that is, you return to the original display from which you requested the series of displays. For displays that provide information only, pressing the Enter/Rec Adv key merely returns you to your original display. For displays containing information on which the system is to take an action, pressing the Enter/Rec Adv key results in the action being performed as well as returning you to your original display.

Processing Multiple Displays



```
                PROGRAM CALL MENU
Select one of the following:
  1.   Call program (identify below)
  2.   Display messages
  3.   Send message to system operator
 90.   Sign off work station (*NOLIST *LIST)

Option: 1       Program name: ____
Parameters or message: __________
```

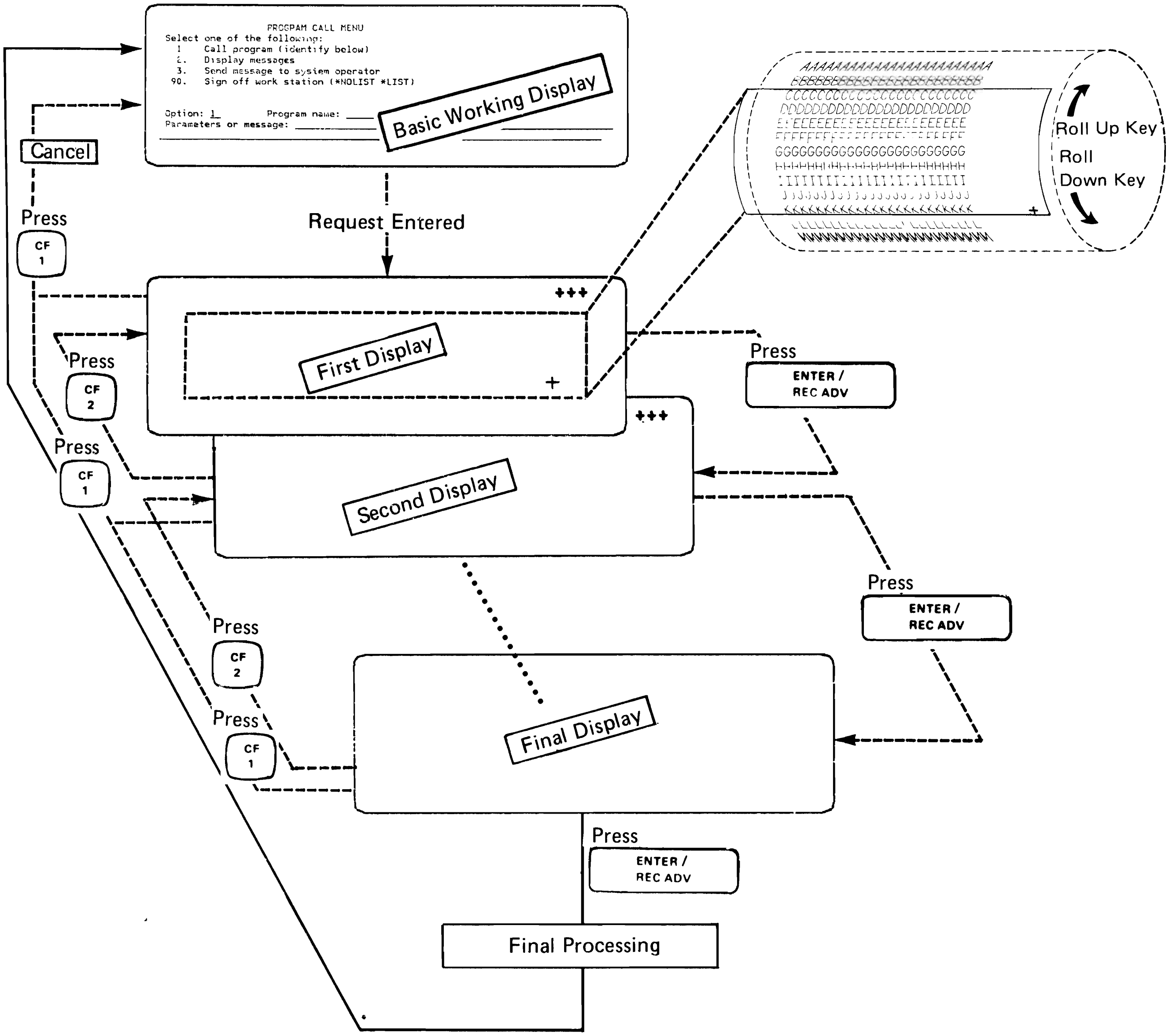
Whether you are viewing the first, the last, or an intermediate display in a series, you can return immediately to your original display by pressing the CF1 key. When you press the CF1 key, you are, in effect, canceling the function, so that no processing is done even if you are on the final display.

Multiple CPF Displays with Roll Capability

Some or all of the individual displays in a series could also have the capability for rolling additional lines onto the screen as described earlier. Thus, you could have a combination of displays that you could:

- Advance by using the Enter/Rec Adv key.
- Back up by using the CF2 key.
- Roll forward or backward using the roll keys.
- Exit by using the Enter/Rec Adv key when on the final display. Any processing requested will be done.
- Exit without processing by using the CF1 key any time.

All of these procedures are summarized in Figure 1-9.



```
                    PROGPAM CALL MENU
Select one of the following:
  1    Call program (identify below)
  2.   Display messages
  3.   Send message to system operator
 90.   Sign off work station (*NOLIST *LIST)

Option: 1       Program name: ____
Parameters or message: ______________
```

Figure 1-9. Procedures for Handling Displays

Requesting Displays from Other Displays

The displays for some display commands allow you to request other displays from them. This may be done by a CF key (which is indicated at the bottom of the display) or the keying of a display option such as 1 or 2 next to an entry that you want to display. In these situations, the CF1 key serves as an exit from all requested displays, returning you to your basic working display from which you requested the primary display. If you want to return to the primary display rather than all the way back to the basic working display, you should press the Enter/Rec Adv key. This will step you through one or more displays of the secondary display functions and return you to the primary display after all requested secondary displays have been shown. If multiple display requests are requested, all of them are satisfied sequentially prior to returning to the primary display.

This procedure is summarized in Figure 1-10.

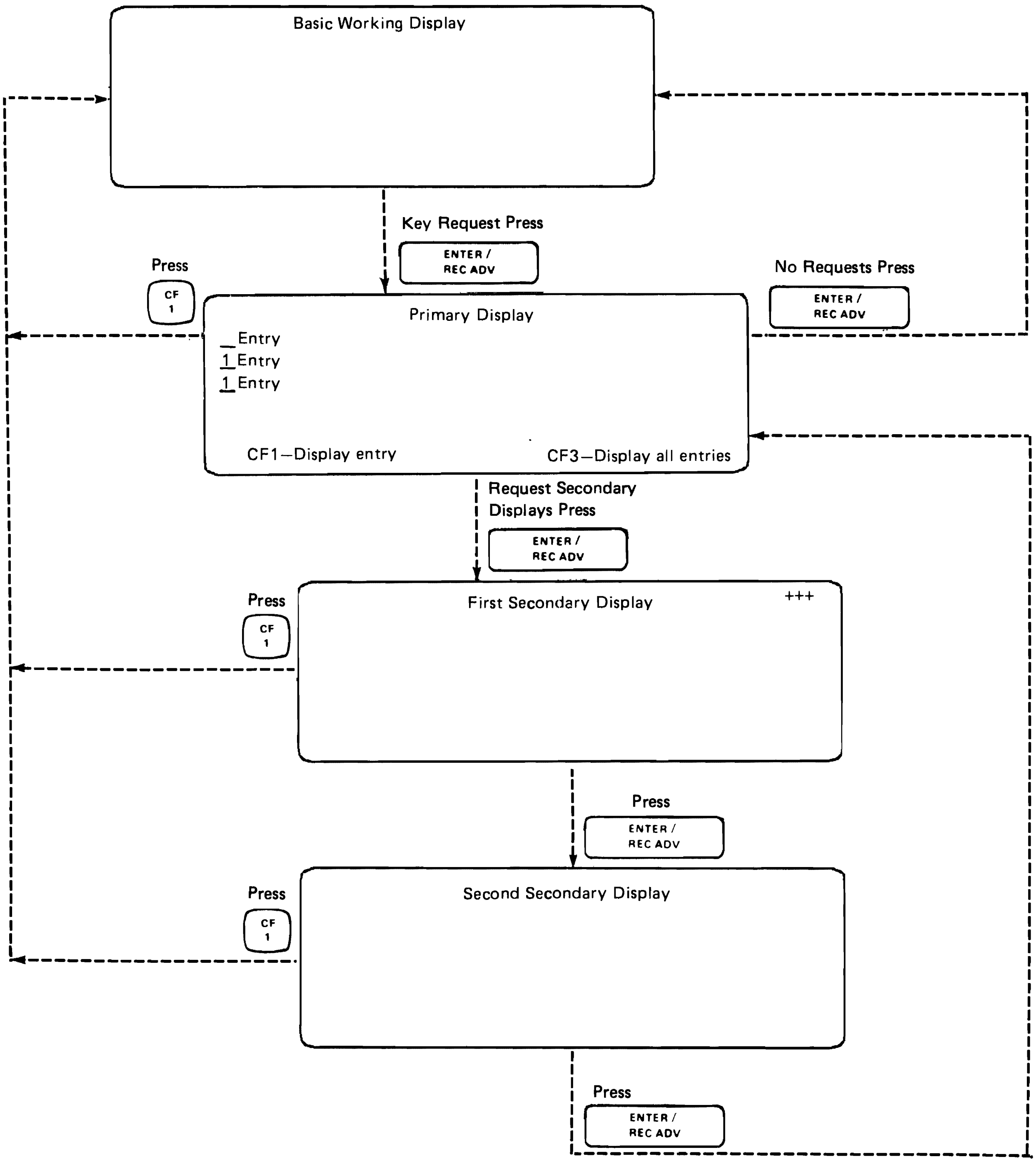


Figure 1-10. Requesting Displays from Other Displays

For more details on using these displays, see the discussion of the appropriate command in the *System/38 Operator's Guide*, or the *CL Reference Manual*.

WHAT TO DO ABOUT A PROBLEM

If your work station does not appear to be working properly, or if you receive no response from the system, refer to the general problem determination guidelines in Appendix B.

Chapter 2. Signing On and Off

When you sign on the system at a work station, you create your own interactive job on the system. The job remains active at the work station until the job ends, such as by signing off. This chapter describes how to sign on and directs you to the proper procedure for signing off.

If your work station is equipped with a Keylock and/or a Magnetic Stripe Reader, see *Security Features* in the *IBM 5251 Display Station Models 2 and 12 Operator's Guide*.

HOW TO TURN ON AN INACTIVE WORK STATION

If the work station you want to use is inactive, that is, the power is off and the screen is completely blank, you can turn the power on by using the control panel beside the screen (Figure 2-1). Note that the control panel is on only one side of the dual-display 5252 and controls both displays.

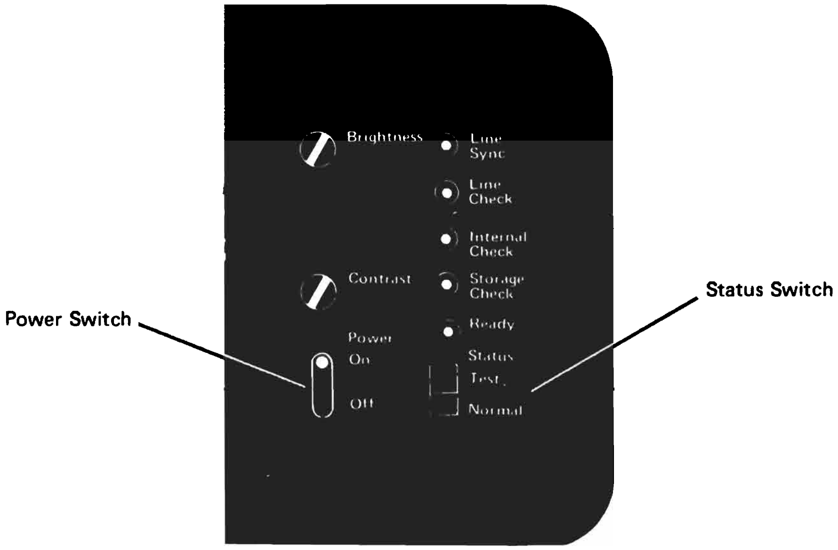
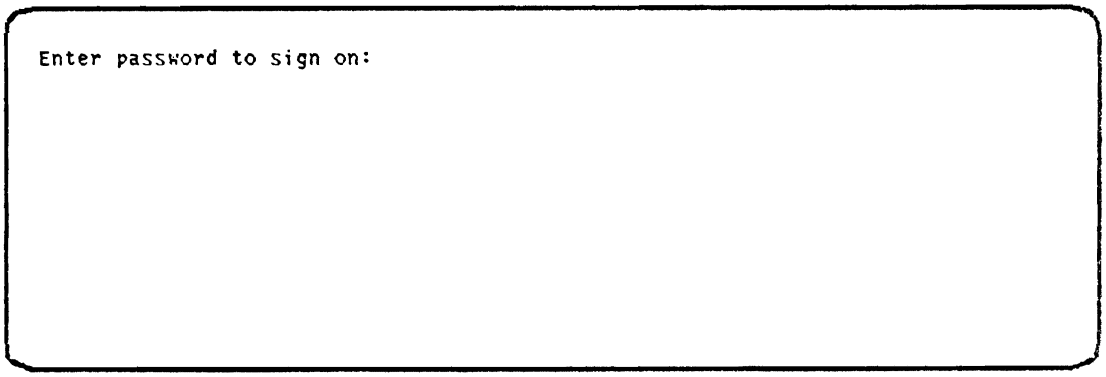


Figure 2-1. Work Station Control Panel

Move the Power switch to the On position, making sure that the Status switch is in the Normal position. The Line Sync and Ready light on the work station control panel will turn on. If your system is equipped with the Keylock feature, turn the key clockwise approximately one-half turn (until it stops). When the cursor is displayed, the display station is unlocked. In about 30 seconds, you should see the sign-on prompt display (Figure 2-2) if the system is available for your use from the work station or you may be routed directly to a program.

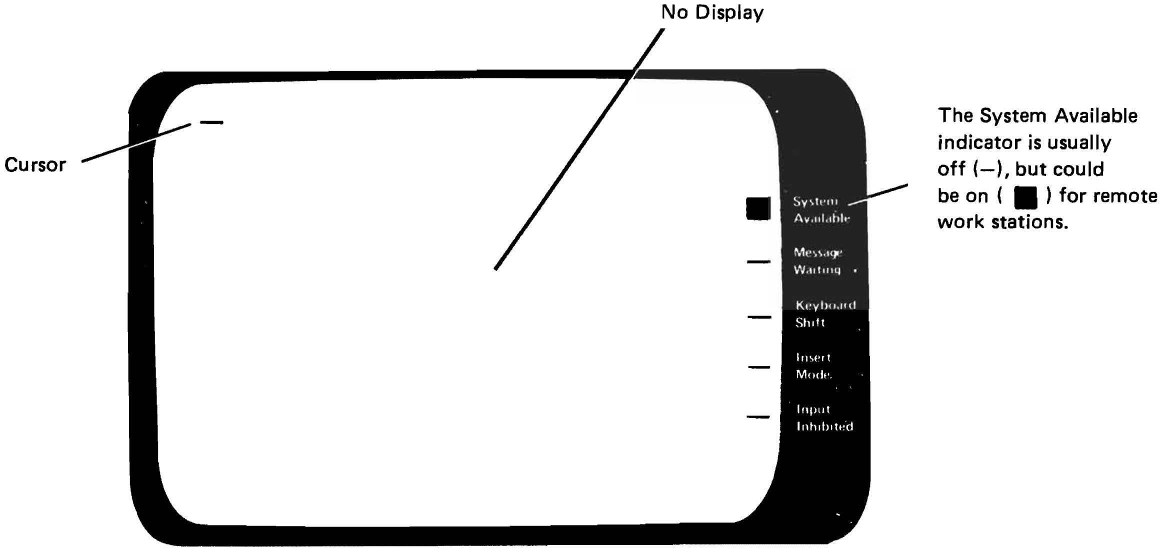
If the work station is connected to the system through a remote communications line, you have to establish communications with the system before you receive the sign-on prompt. The procedure is described in the *IBM 5251 Display Station Models 2 and 12 Operator's Guide.*

If the work station screen remains blank, except for the display indicators and the cursor (Figure 2-3), you will not be able to use the system from that work station. The system may be up but a subsystem may not have allocated the device. If the System Available indicator is on (■), the work station is communicating normally with its control unit. If the System Available indicator is off (–), the work station is not communicating with its control unit and cannot be used.



```
Enter password to sign on:
```

Figure 2-2. Sign-On Prompt–Indicates the System is Available



The System Available indicator is usually off (—), but could be on (■) for remote work stations.

Figure 2-3. Indications that the System is Not Available

SIGNING ON

Sign-On Procedure

Once you have obtained the sign-on prompt (Figure 2-2), you sign on by entering your password, that is:

1. Key in your password. (The cursor is automatically positioned at the beginning of the input field.)

2. Press the Enter/Rec Adv key.

Note that your password does not appear on the screen as you key it in. The input area of the sign-on prompt is a nondisplay field, which means that any entries into the field are not displayed. This ensures that your password cannot be seen by someone else as you enter it.

If your password is valid, the system responds with your first working display. If your password is not valid, you receive an error message telling you so.

What Your Password Tells the System

When you enter your password, it is matched to a user profile stored in the system. This user profile tells the system what you are authorized and not authorized to do.

Based on your user profile and other information stored in the system, the system provides you with a specific display from which you begin your system activities. This could be the program call menu (Chapter 4), or the command entry display (Chapter 5), or the routing display (see *How to Use the Routing Display* later in this chapter), or a display especially designed for your type of work.

Default Sign-On

Depending on the security restrictions for your system, you may be able to sign on the system without entering a password on the sign-on prompt. This is known as default sign-on.

If default sign-on is allowed for the work station you are using, you merely press the Enter/Rec Adv key when you see the sign-on prompt. The system then selects a default user profile that has been defined for the work station.

Generally, you should use default sign-on only if you do not have your own password. If you have a password, you should enter it when you sign on. This ensures that you receive a display that is appropriate to the system functions you will be using.

If Sign-On Fails

If your sign-on is not accepted by the system, you receive an error message explaining the reason for the failure, and the sign-on prompt is displayed for you to try again.

The most common cause of sign-on failure is an invalid password. Make sure you enter your password correctly and that the password you are using is the one currently accepted by the system.

SIGNING OFF

When you sign off, you end the interactive job that you created at the work station when you signed on. The procedure you use to sign off depends on which display or displays you are using at the work station.

- If you received a specific application display when you signed on, that display or a resulting display provides the means for signing off. The procedure is indicated on the display or in the instructions for using the application.

- If you received the program call menu when you signed on, you sign off by selecting an option on the menu. The procedure is described in Chapter 4.

- If you received a display that allows you to enter a command when you signed on, you sign off by entering the SIGNOFF command in the appropriate input area on the display.

- If you did not receive a display that provides the means for signing off or that allows you to enter a command, you sign off by pressing the Sys Req key. The system request menu is displayed. Then you select option 90 (SIGNOFF) and press the Enter/Rec Adv key. The procedure is described in Chapter 5.

If the system is still available after you sign off, the sign-on prompt is displayed again for the next work station user.

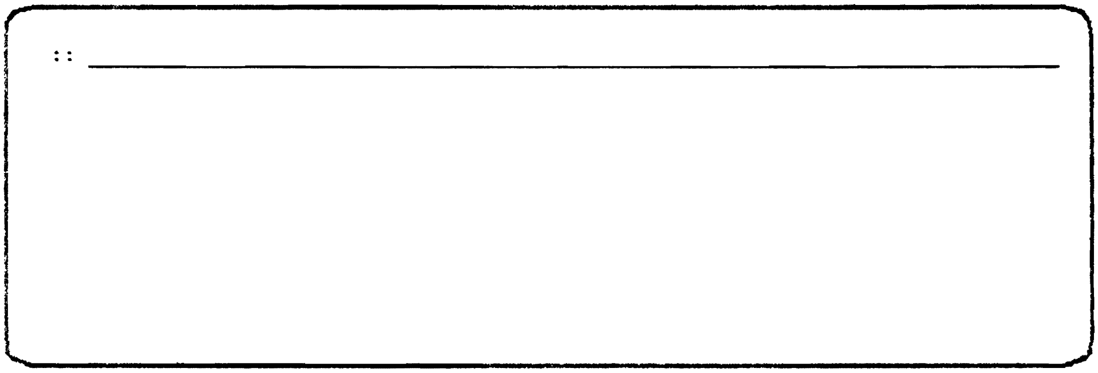
If you want to power off the work station after receiving the sign-on prompt, move the Power switch to the Off position. If you have the Keylock feature, turn the key counter-clockwise.

HOW TO USE THE ROUTING DISPLAY

After you have signed on you may receive this routing display (Figure 2-4).

You should be given instructions on what information you need to enter to proceed. The information you enter will determine which display you receive next.

To sign off you must request a display that supports sign off (see *Signing Off* earlier in this chapter). Note that the system request function is not available from the routing display. The system request menu is described in more detail in Chapter 5 of this publication. If your installation includes the use of system request in its procedures, refer to Chapter 5.



```
: : ________________________________________________________________
```

Figure 2-4. Routing Display

Chapter 3. Message Handling

WHAT IS A MESSAGE?

A message is a communication sent from one point in the system to another. Messages are:

- Words or phrases that are sent to another system user by entering them at your work station.

- Words or phrases that appear on your work station screen when sent to you by the system or another person using the system. These messages could be a *response* to your entry at the work station or could be part of a display of messages in a *message queue*.

It is likely that you *receive* at least some kinds of messages regardless of which display you are using. However, only specific displays provide the capability to *send* messages.

HOW TO HANDLE RESPONSE MESSAGES

When you enter information into the system by keying the information into one or more input fields of a display and pressing the Enter/Rec Adv key, the system might respond with a message. This response message is displayed automatically on the screen of your work station. For most displays, the response message appears on the bottom line, which is known as the *message line*. Unless you are using the command entry display (described in Part III of this publication), the response message you see most likely is an *error message*.

If you make a keying error that is detected by the work station device as you are entering information, you receive a numeric *error code* rather than an error message on the message line of your display. You can, however, request a message that describes the meaning of the error code. You handle this message the same way that you handle other error messages.

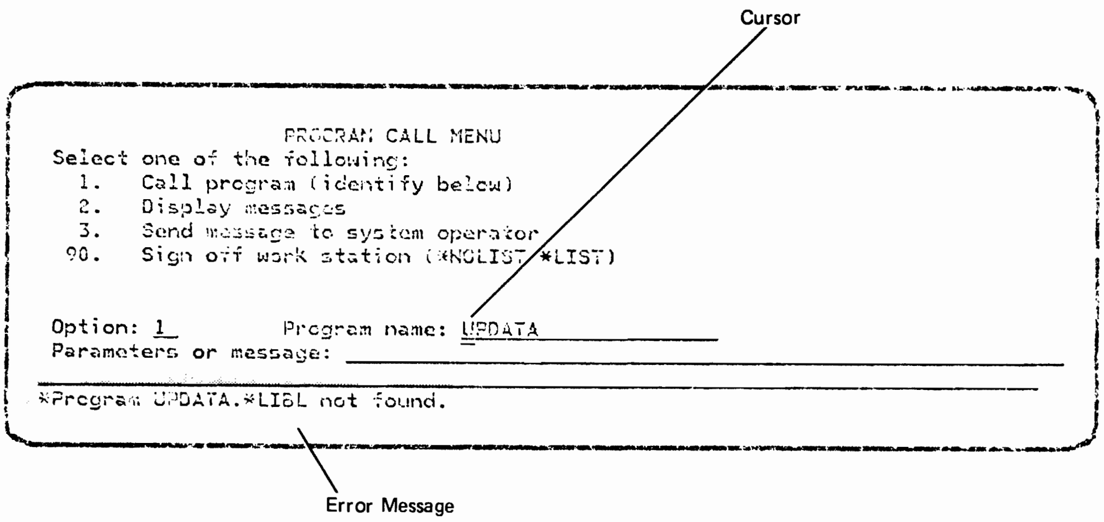
The following sections describe how to handle both error messages and error codes, as well as system responses in the form of inquiries that request information from you.

Error Messages Related to Your Input

If you enter information that the system cannot process because of an error, the cursor is positioned at the input field containing the error, and an error message is shown on the message line of your display (Figure 3-1).

On some displays, the input field containing the error is shown in reverse image (dark characters on a bright background). Depending on the type of error and the program in control, the keyboard may also be locked. You unlock the keyboard by pressing the Error Reset key; that is:

- On the typewriter-like keyboard, press the Error Reset key.
- On the data-entry keyboard, press the Reset key.



```
                    PROGRAM CALL MENU
Select one of the following:
  1.   Call program (identify below)
  2.   Display messages
  3.   Send message to system operator
 90.   Sign off work station (*NOLIST *LIST)


Option: 1_         Program name: UPDATA__________
Parameters or message: _______________________________________________

*Program UPDATA.*LIBL not found.
```

Figure 3-1. Example of Error Message on a Display

What You Can Do About the Error Message

When an error message appears on your display, you can respond to the message in one of three ways:

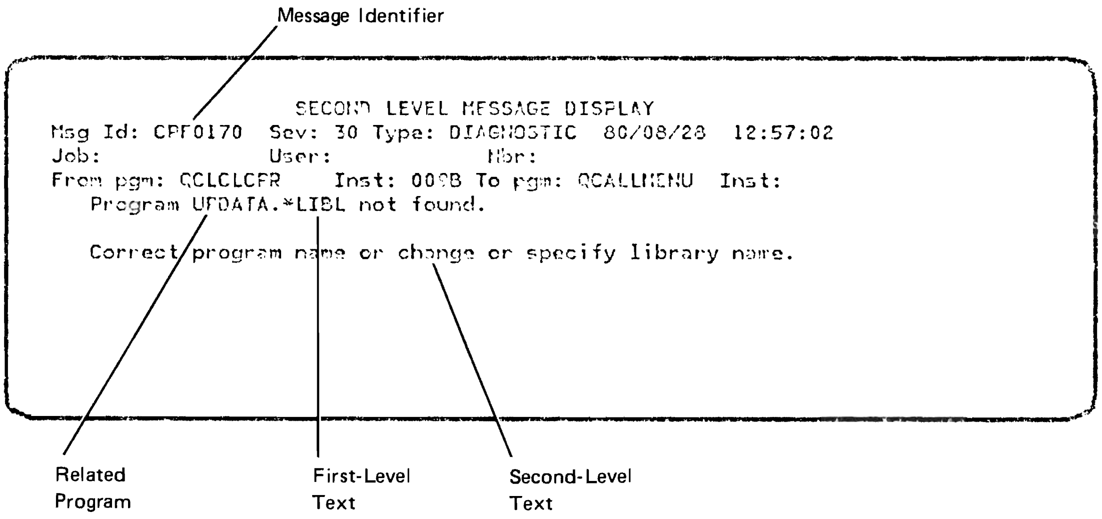
- Press the Error Reset key, if necessary; correct the error (by keying the correct information into the input field), and continue.

- If the Input Inhibited indicator is on, press the Help key to obtain additional information about the error without first pressing the Error Reset key. If the second-level message text does not fit on the display, you can use the Display Message Description (DSPMSGD) command to display the entire text of the message. Specify the message identifier, the proper message file, and OUTPUT(*). Refer to the *CL Reference Manual.*

- If the Input Inhibited indicator is not on, position the cursor under the message. Then press the Help key to obtain additional information about the error.

If you press the Help key, you receive a second-level message display that shows both the original message (called *first-level text*) and an expanded description of the error (called *second-level text*) (Figure 3-2). The display also contains other basic information about the message such as the message identifier which you can use to look up the message in the *Messages Guide: CPF, RPG III,* and *IDU* or the *Messages Guide: COBOL.*

After examining the second-level message display, you return to your original display by pressing the Enter/Rec Adv key.

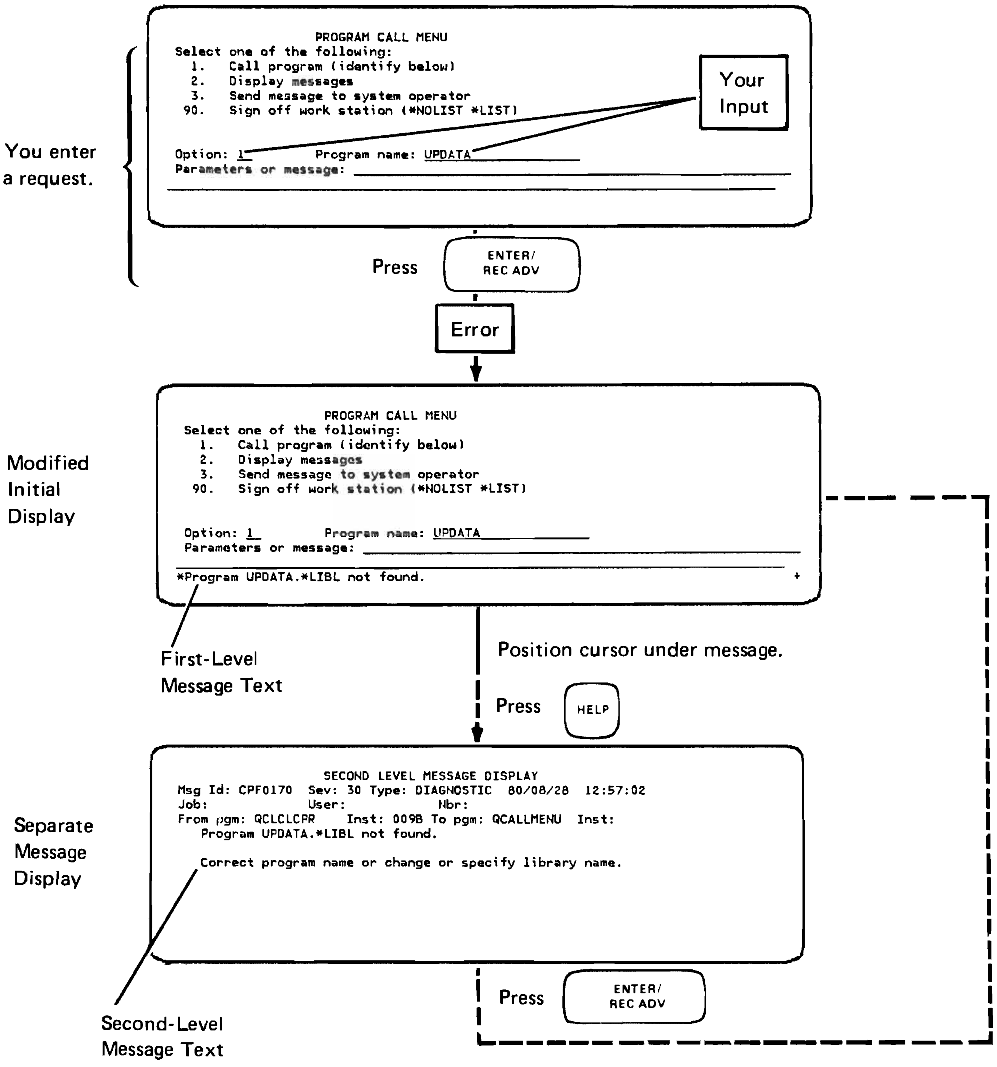


```
                        SECOND LEVEL MESSAGE DISPLAY
Msg Id: CPF0170  Sev: 30 Type: DIAGNOSTIC  80/08/28  12:57:02
Job:                User:              Nbr:
From pgm: QCLCLCPR     Inst: 009B To pgm: QCALLMENU  Inst:
   Program UPDATA.*LIBL not found.

   Correct program name or change or specify library name.
```

Figure 3-2. Second-Level Message Display for Error Message

These procedures are summarized in Figure 3-3.

If you are still not sure how to correct the error after examining the second-level message display, refer to the entry for the error message in the *Messages Guide: CPF, RPG III, and IDU* or the *Messages Guide: COBOL*. Information in either messages guide is arranged in numerical order according to the message identifier. You can get the message identifier from the second-level message display (Figure 3-2). When you correct the error, the message is cleared from your display.



```
                    PROGRAM CALL MENU
Select one of the following:
  1.   Call program (identify below)
  2.   Display messages
  3.   Send message to system operator
 90.   Sign off work station (*NOLIST *LIST)


Option: 1_        Program name: UPDATA__________
Parameters or message: ______________________________
```

```
                    PROGRAM CALL MENU
Select one of the following:
  1.   Call program (identify below)
  2.   Display messages
  3.   Send message to system operator
 90.   Sign off work station (*NOLIST *LIST)


Option: 1_        Program name: UPDATA__________
Parameters or message: ______________________________
*Program UPDATA.*LIBL not found.                                 +
```

```
                    SECOND LEVEL MESSAGE DISPLAY
Msg Id: CPF0170  Sev: 30 Type: DIAGNOSTIC  80/08/28  12:57:02
Job:                 User:              Nbr:
From pgm: QCLCLCPR    Inst: 009B To pgm: QCALLMENU  Inst:
   Program UPDATA.*LIBL not found.

   Correct program name or change or specify library name.
```

Figure 3-3. Handling Error Messages

Multiple Error Messages

It is possible that you could receive more than one error message at a time. If this occurs, and there are more messages than can be shown on the screen at one time, a plus (+) is shown to the right of the last message currently displayed, indicating that there are more messages to be displayed.

You obtain the additional messages by positioning the cursor on one of the message lines and using the roll keys (see *How to Handle Roll Capability and Multiple Displays* in Chapter 1). A new message or set of messages appears in the message area of your display each time you press the Roll Up key, until all of the messages have been displayed. You can redisplay a previous message by pressing the Roll Down key.

For each of the error messages, there is a second-level message display, such as Figure 3-2. To obtain the second-level display for a particular message, you position the cursor on the message and press the Help key while that message is shown on your screen. After viewing the second-level display for a message, you can press the Enter/Rec Adv key to return to your original display, or press a roll key to display another message, and then press the Help key to obtain the second-level display for that message.

On a menu when you correct the error that caused the messages and press the Enter/Rec Adv key, all of the previous messages are cleared from your display.

The displays with multiple options (such as DSPOUTQ, DSPJOBQ, and DSPSBS), may issue error messages, but the keyboard will not be locked. These error messages indicate which requests were performed and which were not performed. If you request a new function, all the error messages disappear, and only those messages for the new request are displayed.

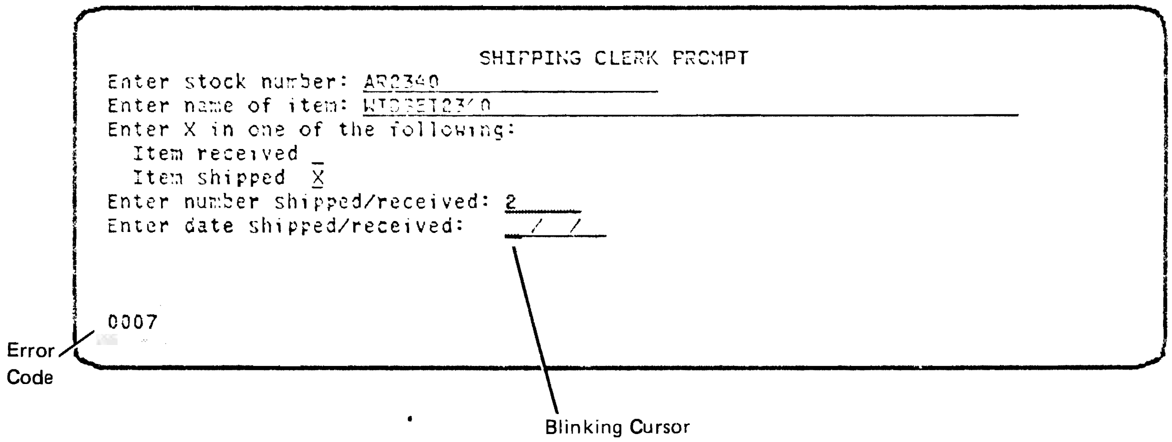
Error Codes and Messages for Keying Errors

If an error occurs that is detected at a 5251/5252 work station while you are keying in information, the keyboard locks to prevent further use, and a blinking cursor indicates the location of the error. However, instead of an error message, only a 4-digit error code is shown on the message line of your display (Figure 3-4).

If you do not know what the code means, press the Help key. The code is replaced by an error message (Figure 3-5).

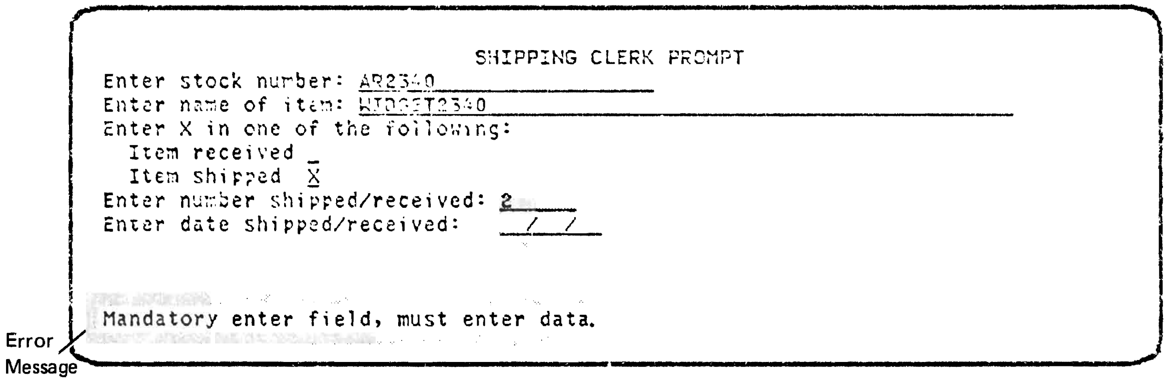
Once the error message is on your display, you follow the same procedures as those just described for handling error messages (see Figure 3-3). If you do not know how to correct the error, you can press the Help key to obtain a second-level display that contains second-level text describing what to do (Figure 3-6).

Note that you must press the Enter/Rec Adv key to return to your original display (Figure 3-5) before doing what is indicated in the second-level text. All keying errors cause the keyboard to lock, so you always have to press the Error Reset key before you can correct the error.



```
                              SHIPPING CLERK PROMPT
Enter stock number: AR2340______________
Enter name of item: WIDGET2340______________________________________
Enter X in one of the following:
  Item received _
  Item shipped  X
Enter number shipped/received: 2___
Enter date shipped/received:   __/__/__


0007
```

Figure 3-4. Indications of Keying Error



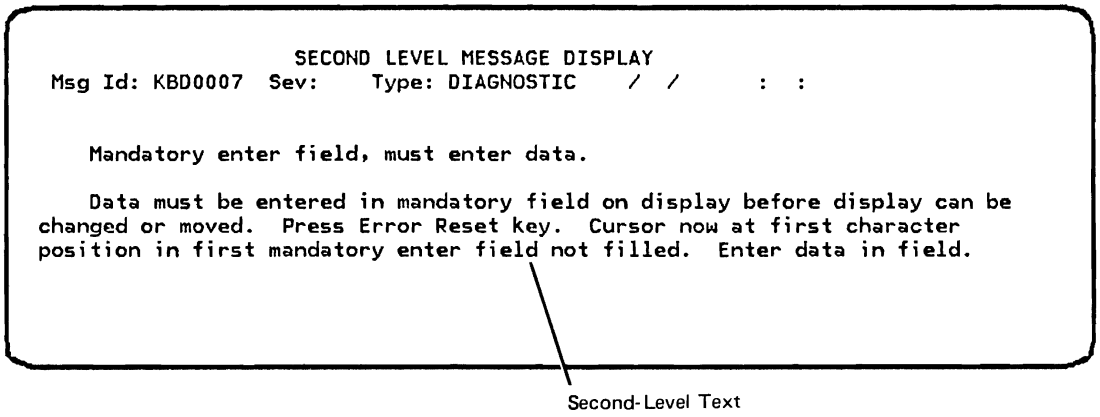
```
                              SHIPPING CLERK PROMPT
Enter stock number: AR2340______________
Enter name of item: WIDGET2340______________________________________
Enter X in one of the following:
  Item received _
  Item shipped  X
Enter number shipped/received: 2___
Enter date shipped/received:   __/__/__


Mandatory enter field, must enter data.
```

Figure 3-5. Typical Error Message for Keying Error

Messages from an Active Program

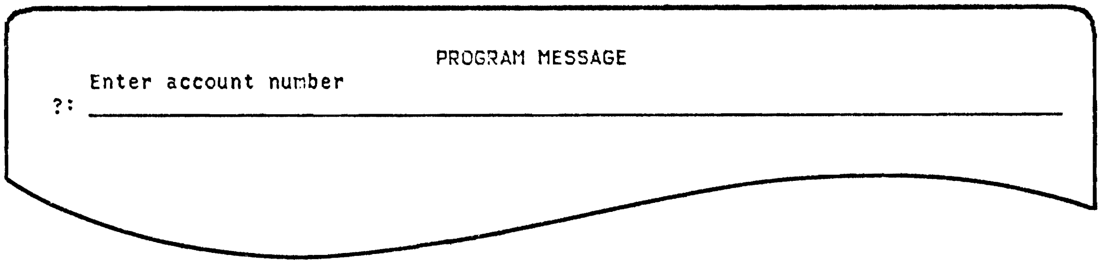
If you request a function for which the program in control sends you a message informing you of its status or requesting additional information, your current display is replaced by a separate *program message display*. If the message does not require a reply, only the message is shown. If you want the program to continue executing after you view the message, press the Enter/Rec Adv key. If you receive an inquiry message, it is shown followed by an input field (Figure 3-7). The symbol ?: to the left of the input field indicates that this input field has been provided for your reply to the message.

When you key in your reply and press the Enter/Rec Adv key, the program continues execution and the program message display is replaced by your previous display or the next display for your job. However, if your reply did not provide the proper information, you receive an error message.



```
                       SECOND LEVEL MESSAGE DISPLAY
 Msg Id: KBD0007  Sev:    Type: DIAGNOSTIC    / /       : :


    Mandatory enter field, must enter data.

    Data must be entered in mandatory field on display before display can be
 changed or moved.  Press Error Reset key.  Cursor now at first character
 position in first mandatory enter field not filled.  Enter data in field.
```

Figure 3-6. Display Showing How to Correct Keying Error



```
                                PROGRAM MESSAGE
    Enter account number
 ?: ____________________________________________________________________________
```

Figure 3-7. Example of Inquiry Message as Response

HOW TO HANDLE MESSAGES FROM A MESSAGE QUEUE

In addition to messages you receive in response to your entries at your work station, you may also receive messages that were sent to a *message queue* associated with your work station. A message queue is like a mailbox for messages; it accumulates and holds messages for you to display later. There is at least one message queue associated with each work station.

The messages stored in a message queue could have been sent by the system or by another person at a work station. Depending on the type of displays you are using, you may also be able to send a message to another user at a work station or to the system operator. If you do send such a message, it goes to the message queue associated with the work station of the person receiving it. These concepts are summarized in Figure 3-8.

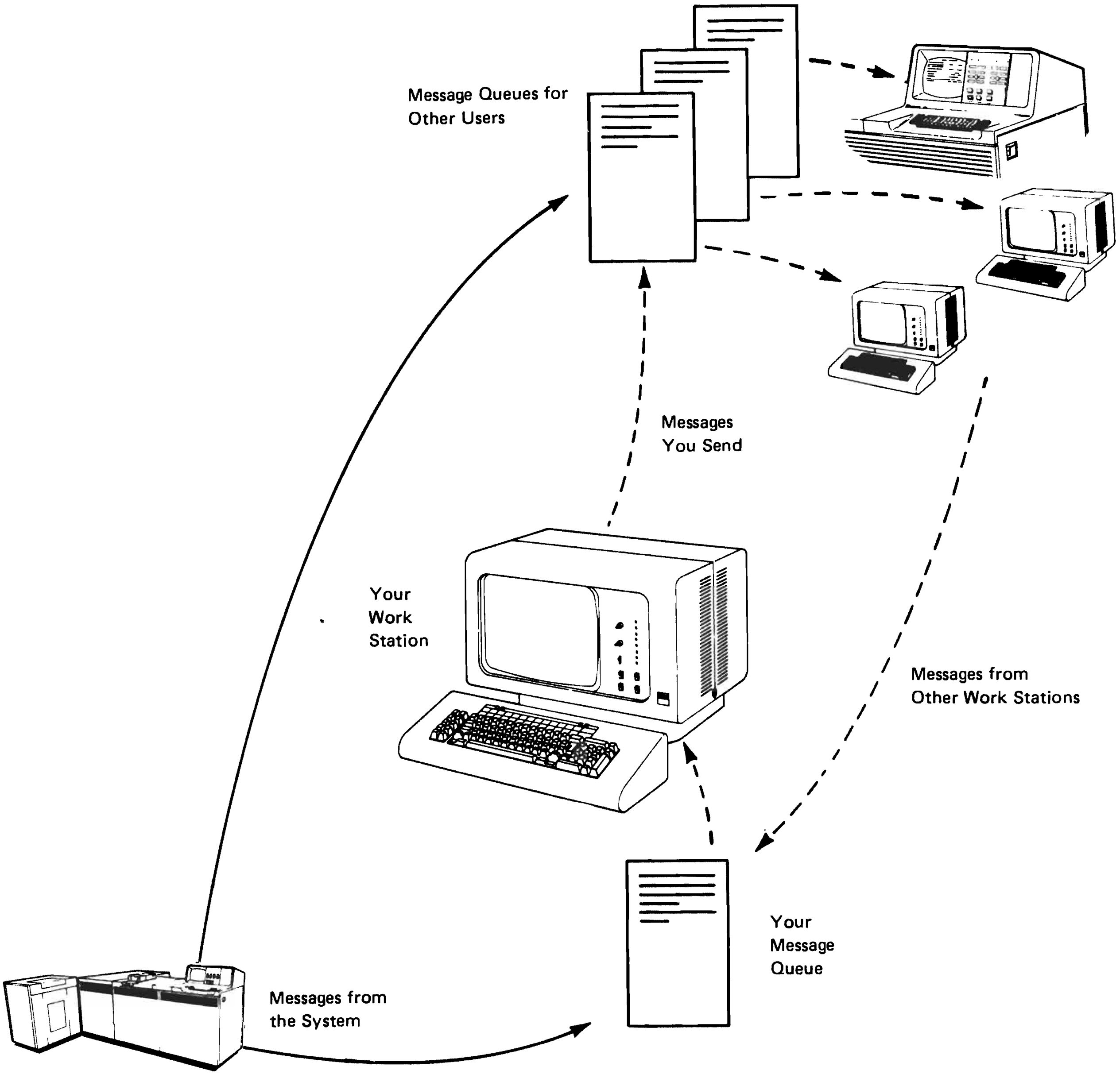


Figure 3-8. Messages and Message Queues

Displaying the Messages

There are two ways you can receive messages from a message queue:

- The messages are displayed automatically without any action by you.
- The messages are displayed only when you ask for them.

Usually, only a specific type of message, known as a break message, is displayed automatically from a message queue. The system operator sends break messages to inform work station users of important system events that could affect them. If you are able to use the command entry facilities described in Part III of this publication, you can change your message queue so that *all* messages sent to the queue are displayed automatically or so the Message Waiting indicator and keyboard buzzer notifies you of the arrival of a message (see *How to Set the Message Delivery Mode* in Chapter 6).

If the information to be displayed contains any undisplayable hexadecimal characters, they will be converted to a hex '1F' and will appear on the screen as either a blank or a reverse image blank (■).

Automatic Display of Messages

Whenever a message is sent to a queue and is displayed automatically (break message), the following occurs:

- A keyboard buzzer sounds briefly.
- Your current display is saved and you receive a display containing the message.

After you have examined the message, you return to your original display by pressing the Enter/Rec Adv key. You can then resume your work at the point where you were interrupted.

Asking for a Display of Messages

If a program in control of your work station allows you to display messages in a message queue, the following occurs:

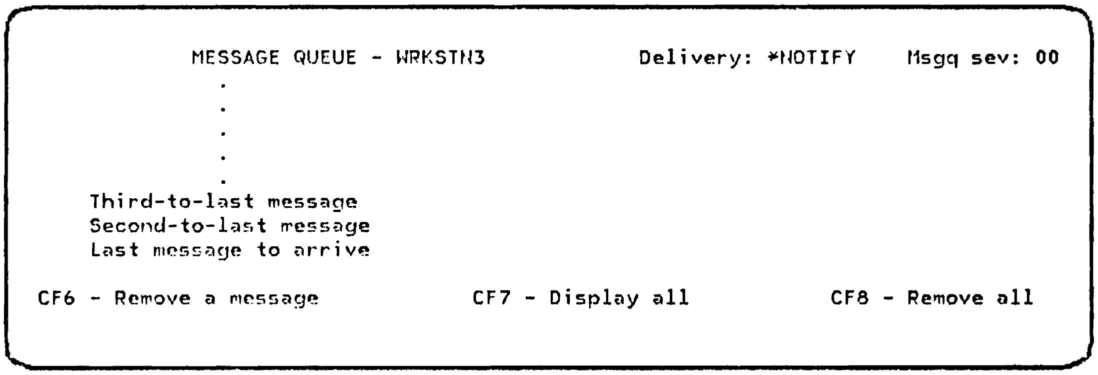
- The Message Waiting indicator comes on.
- The keyboard buzzer sounds to notify you that there is a message to be displayed (unless you have changed the delivery mode of the message queue by using the procedure discussed in Chapter 6). If the Message Waiting indicator is already on, the keyboard buzzer does not sound.

At this time, or at a later time if you choose, you can request a display of the messages on your work station message queue. You can do this in one of the following ways depending on the program in control of your work station:

- If the program in control is an application program, refer to the instructions for the program to determine what procedures you use.
- If the program call menu is in control, select the display messages option (see *How To Display Messages* in Chapter 4).
- If the command entry display is available, use the Display Messages (DSPMSG) command (see *How To Display Messages From a Message Queue* in Chapter 6).
- Use the Sys Req key and select the display messages option from the system request menu or enter the option number in the vertical bars at the bottom of the display (see *The System Request Menu* in Chapter 5).

Using Message Displays

Whether the messages from a message queue are displayed automatically or you request them (with the Display Messages [DSPMSG] command), you receive a separate message display. The last message in the queue is shown first (at the bottom of the display), followed by the second-to-last message to arrive, the third-to-last to arrive and so on.



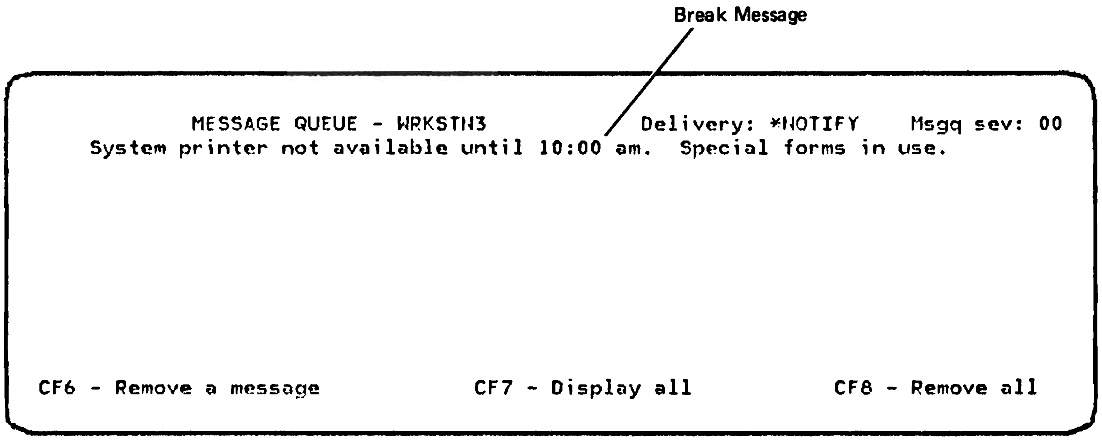
```
              MESSAGE QUEUE - WRKSTN3          Delivery: *NOTIFY    Msgq sev: 00
                     .
                     .
                     .
                     .
                     .
       Third-to-last message
       Second-to-last message
       Last message to arrive

   CF6 - Remove a message          CF7 - Display all          CF8 - Remove all
```

If the last message to arrive is at the bottom of the display because the display is filled, then there may be more messages on the queue than are on the display. To see the messages that arrived earlier, press the Roll Down key. (No + appears in the lower right corner of the display.)

To see any messages that arrived later than those on the display, press the Roll Up key.

Each message is displayed on a separate line. As much of the message that can be displayed on one line is shown. In some cases the entire message is not shown in the first-level text. There is a separate second-level message display for each message shown (see *Obtaining the Second-Level Message Display* later in this chapter).

If you received a break message, the message display is adjusted so that the break message appears at the top of the display area. This means that the display will usually show only the break message because the message is displayed as soon as it arrives.



```
          MESSAGE QUEUE - WRKSTN3          Delivery: *NOTIFY    Msgq sev: 00
     System printer not available until 10:00 am.  Special forms in use.






  CF6 - Remove a message            CF7 - Display all           CF8 - Remove all
```

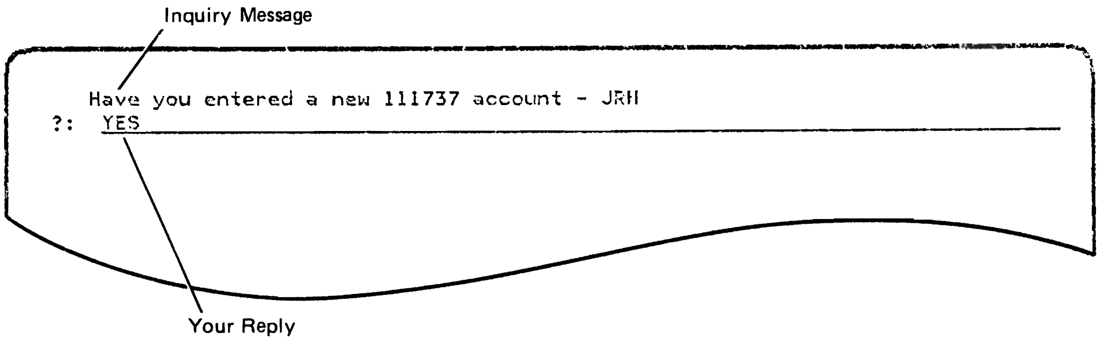
Should another message, which is displayed automatically, arrive in the queue after the break message but before you receive the display, the additional message is shown on the line below the break message. If another message arrives in the message queue when you are displaying a break message, you receive the next message if you press the Roll Up key or the Enter/Rec Adv key.

Obtaining Messages Not On the Screen

If there are more messages on the display than can be shown on your screen at one time (as indicated by a plus sign (+) in the lower right corner of the message area), you can obtain the additional messages by using the Roll keys (see *How to Handle Roll Capability and Multiple Displays* in Chapter 1).

Handling Inquiry Messages

If any of the messages is an inquiry message that requires your reply, an input field preceded by the symbol ?: is provided on the next line below the inquiry message for you to enter your reply. However, entering your reply may require shifting to uppercase. Key your reply into the input field and press the Enter/Rec Adv key. For example:



```
    Have you entered a new 111737 account - JRH
?:  YES
```

Your reply is then sent to the message queue of the person (or program) who sent the inquiry message. If you want to know which work station the message was issued from, press the Help key. The inquiry message and your reply remain in the queue after the reply is sent. You can review the message and your reply at a later time by displaying the messages in the queue.

If you do not want to reply to the inquiry message immediately, press the CF1 key to exit.

Deleting Messages

You can delete a message from the message queue by the following procedure:

1. Position the cursor under the message.

2. Press the CF6 key.

If you delete an inquiry message that you have not answered, a default reply is sent to the sender of the message and the inquiry and the default reply are deleted. If you delete an inquiry message that you have already answered, both the message and your reply are deleted. You can delete all messages from the message queue by pressing the CF8 key.

After you review a message and determine that it is no longer needed, you should delete it from the message queue. Deleting obsolete messages makes it easier to review current messages and frees up system storage.

Obtaining Messages that were Excluded from the Display

Display message requests can be entered so that only some of the messages in the queue (for example, messages of a high severity) are shown by using the roll keys while on the message display. You can cause all messages to be displayed by pressing the CF7 key from the message display. If there are any messages in the queue that were not displayed, they are added to the message display.

Obtaining the Second-Level Message Display

To obtain the second-level message display for a particular message:

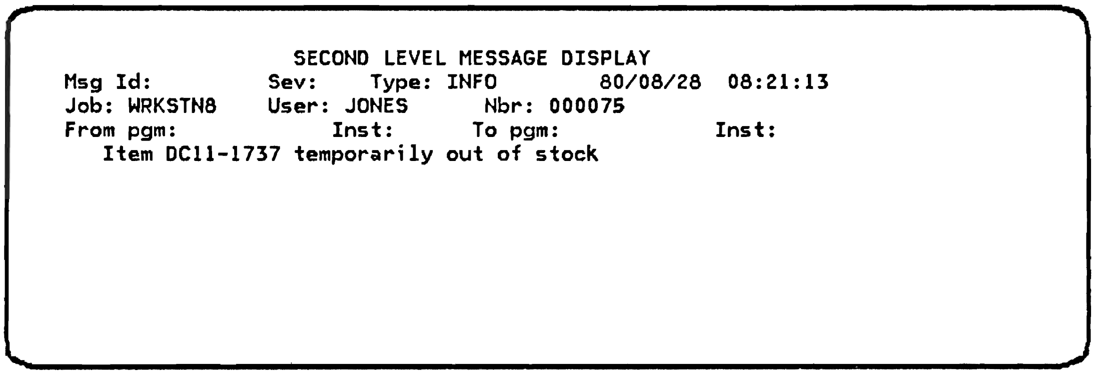
1. Position the cursor on the line containing the message.

2. Press the Help key.

The display you receive has the same form as the second-level message display for an error message (see Figure 3-2). However, the information shown on the display varies, depending on the type of message and on the attributes of the message queue being displayed. For example, a message sent by another person at a work station does not have second-level text. The information on other second-level message displays depends on how the message queue was created. The SENDER parameter on the CRTMSGQ command determines what information is in the messages sent to the message queue. For example, if you specify *JOB for the SENDER parameter you receive the job name that sent the message. If you specify *PGM, you receive the program name that sent the message and if you specify *DTS you receive the date and time. Work station message queues are created with *JOB and *DTS.

The amount of first- and second-level message text depends on the message identifier and how that message identifer was added to the message file.

If the message is sent from the system, such as an error message (see Figure 3-2), the Job field on the second-level message display shows the name of your own interactive job; that is, the name of the work station you are using. Similarly, the User field on the display shows the name of your own user profile. If the message is sent from another person at another work station, the Job field shows the name of the work station from which the message was sent and the User field shows the user profile associated with the person who sent the message. For example, if you position the cursor on the same line as the last message and press the Help key, you might receive a second-level message display of the form:



```
                    SECOND LEVEL MESSAGE DISPLAY
Msg Id:          Sev:    Type: INFO        80/08/28  08:21:13
Job: WRKSTN8     User: JONES      Nbr: 000075
From pgm:             Inst:       To pgm:            Inst:
   Item DC11-1737 temporarily out of stock
```

This display indicates that the message was sent by a person with the user profile Jones from the work station identified as WRKSTN8 at 08:21:13 system time. If you want to send a message to the person who sent the displayed message, you send your message to the WRKSTN8 message queue.

This page is intentionally left blank.

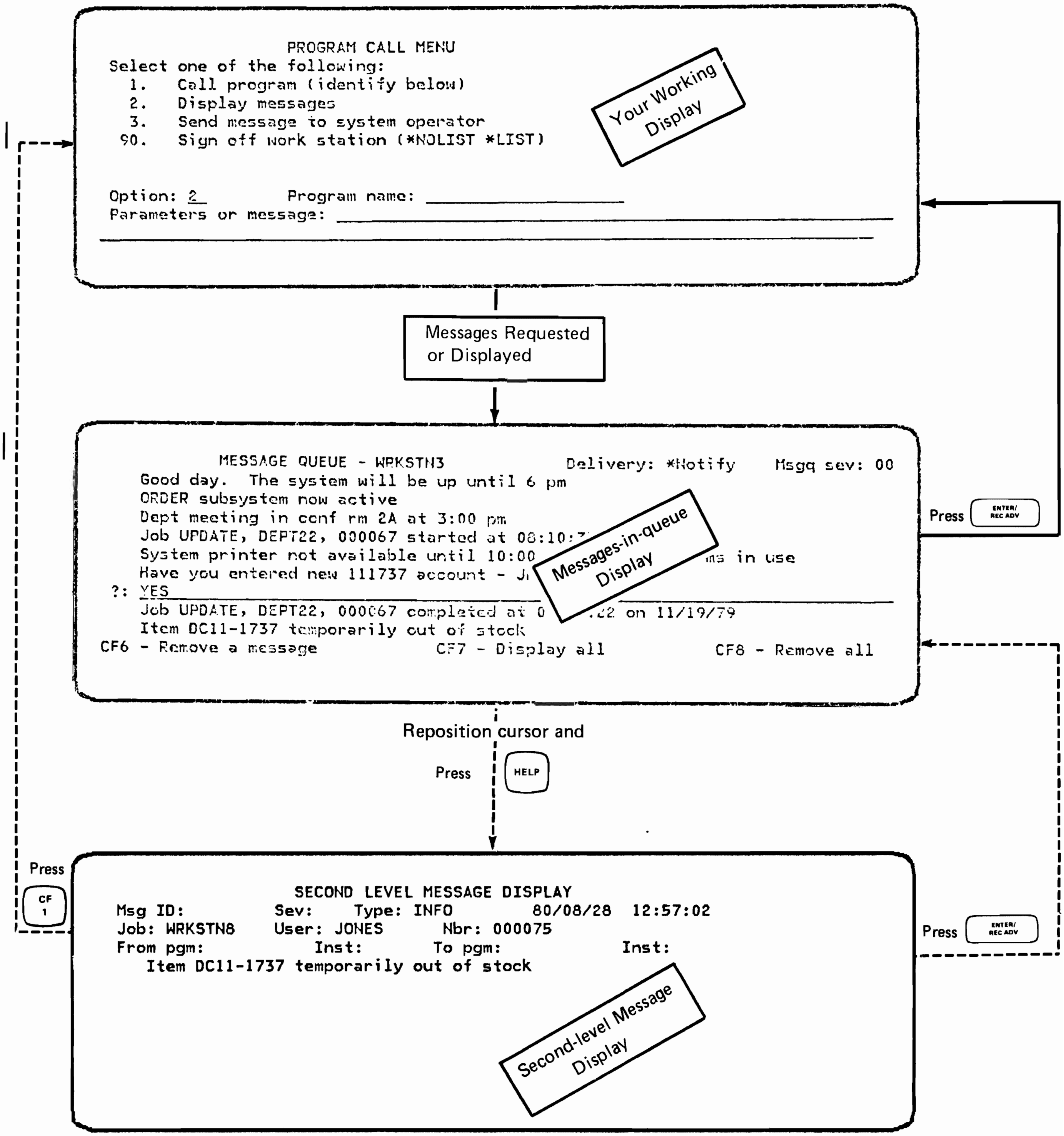
Returning to Your Working Display

After you have examined the messages-in-queue display, you can exit the display and return to your working display by pressing the Enter/Rec Adv key. If you have requested a second-level message display, you must press the Enter/Rec Adv key once to return to the messages-in-queue display and then press the Enter/Rec Adv key again to return to your working display or press the CF1 key in the second-level message display to return you to your working display. These procedures are summarized in Figure 3-9.

SENDING MESSAGES FROM A WORK STATION

Some displays provide the capability for sending messages from a work station. The procedure for sending messages by means of the program call menu or the command entry facilities is described in Parts II and III of this publication. If you are using a different type of display as your basic working display, refer to the instructions for that display to determine if you can send messages and what procedure you use.

If your basic working display does not provide the capability for sending messages, you can press the Sys Req key and use the system request menu to send a message (see *How to Use System Request* in Chapter 5).



```
                    PROGRAM CALL MENU
Select one of the following:
  1.    Call program (identify below)
  2.    Display messages
  3.    Send message to system operator
 90.    Sign off work station (*NOLIST *LIST)

Option: 2_         Program name: ___________________
Parameters or message: _____________________________________________________
```

```
        MESSAGE QUEUE - WRKSTN3              Delivery: *Notify    Msgq sev: 00
   Good day.  The system will be up until 6 pm
   ORDER subsystem now active
   Dept meeting in conf rm 2A at 3:00 pm
   Job UPDATE, DEPT22, 000067 started at 08:10:7
   System printer not available until 10:00              ms in use
   Have you entered new 111737 account - J
?: YES
   Job UPDATE, DEPT22, 000067 completed at 0      .22 on 11/19/79
   Item DC11-1737 temporarily out of stock
CF6 - Remove a message              CF7 - Display all            CF8 - Remove all
```

```
                      SECOND LEVEL MESSAGE DISPLAY
Msg ID:          Sev:     Type: INFO          80/08/28   12:57:02
Job: WRKSTN8     User: JONES        Nbr: 000075
From pgm:              Inst:        To pgm:              Inst:
   Item DC11-1737 temporarily out of stock
```

Figure 3-9. Obtaining and Exiting Message Displays

Part II

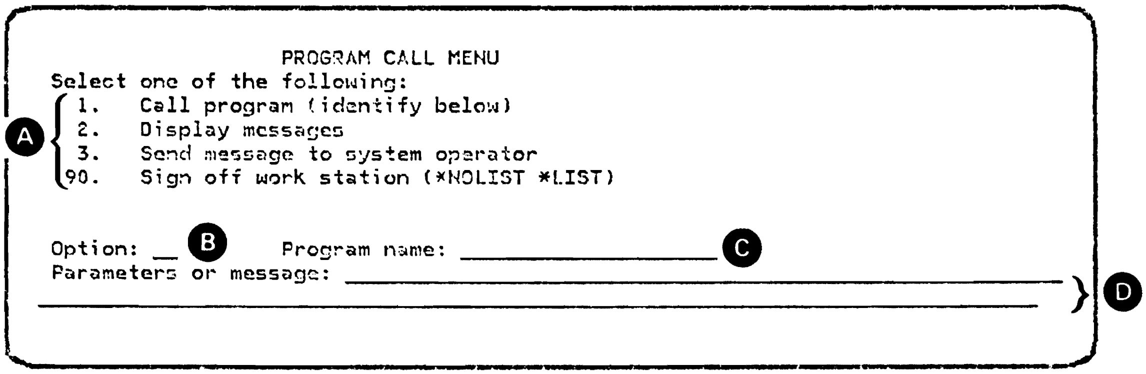
Basic CPF Operations

Chapter 4. Using the Program Call Menu

WHAT IS THE PROGRAM CALL MENU?

The program call menu (Figure 4-1) is a display that allows you to perform tasks at a work station without knowing the System/38 control language. Using the menu, you can call a program from your work station, display messages sent to your work station, send a message to the system operator, or sign off your work station.

When you sign on, your user profile and other information stored in the system determine which working display you receive. If you receive the program call menu, you should read this chapter.



```
                    PROGRAM CALL MENU
Select one of the following:
  1.   Call program (identify below)
  2.   Display messages
  3.   Send message to system operator
 90.   Sign off work station (*NOLIST *LIST)

Option: __         Program name: ________________
Parameters or message: ___________________________________________
__________________________________________________________________
```

Figure 4-1. Program Call Menu

HOW YOU INTERACT WITH THE MENU

The program call menu provides four options Ⓐ. To select one of the options, you enter the number of that option in the Option input field Ⓑ. (The cursor is automatically positioned at Ⓑ when the menu is first displayed.) Depending on which option you choose, you may or may not be required to enter information in the remaining fields Ⓒ and Ⓓ. The options from the program call menu are explained in more detail later in this chapter.

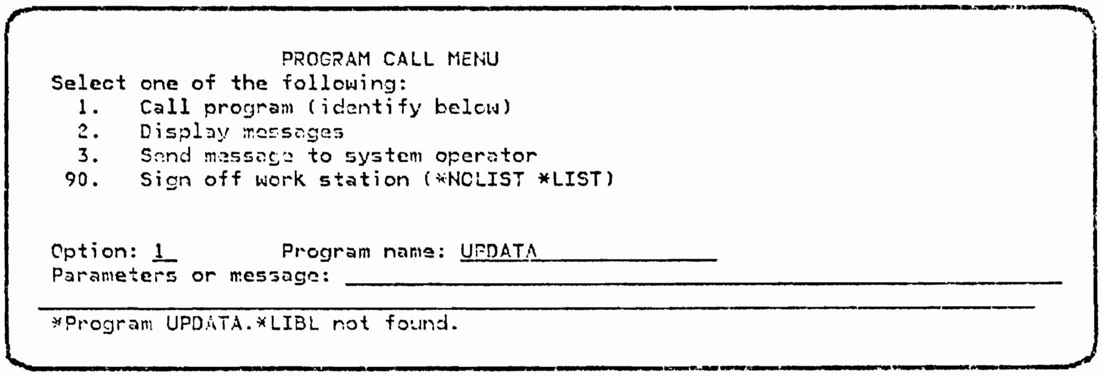
After you make an entry into the Option field Ⓑ, the cursor automatically moves to the Program name field Ⓒ. You can move the cursor from field Ⓒ to the first line of the Parameters or message field Ⓓ by pressing the Field Exit key or the ⇥ (Field Advance) key.

When you have completed all entries on the menu, press the Enter/Rec Adv key to send the information to the system for processing.

After your request is processed, the information you entered on the menu is cleared, and the system is ready for your next input.

HOW TO HANDLE ERRORS

If you make an error in the information you enter on the menu, you receive an error message on the bottom line of the menu (Figure 4-2).



```
                      PROGRAM CALL MENU
Select one of the following:
  1.   Call program (identify below)
  2.   Display messages
  3.   Send message to system operator
 90.   Sign off work station (*NOLIST *LIST)


Option: 1          Program name: UPDATA
Parameters or message:
-----------------------------------------------------------
*Program UPDATA.*LIBL not found.
```

Figure 4-2. Typical Error Message on Program Call Menu

Typical errors you might make that result in an error message are:

- Entering an option number other than 1, 2, 3, or 90 in the Option field
- Entering an incorrect program name in the Program name field, so that the system cannot find the program
- Entering the name of a program that you are not authorized to use
- Entering incorrect parameters in the Parameters or message field

If you enter a program name incorrectly or specify a program that you are not authorized to use, you receive an error message of the type shown in Figure 4-2.

You should also be careful that you do not enter information in the wrong input field. The result is either an error message or incorrect processing of the information.

As described in Chapter 3 (see *How to Handle Response Messages* in that chapter), you can position the cursor under the message and press the Help key to obtain additional information about an error message. If the keyboard is locked, you have to press the Error Reset key before you can correct the error.

For most errors, the correction needed is obvious, such as:

- Keying in a valid option number in place of the incorrect number
- Keying in a correct program name or parameter value in place of a misspelled name or value

If you received an error message and have verified that you entered the information correctly, it is likely that the information itself is incorrect and should be checked.

HOW TO CALL A PROGRAM (OPTION 1)

To call a program, select option 1 on the program call menu and enter the name of the program on the menu, along with any appropriate parameters. Your work station is then placed under control of the program to perform a function or type of work for which the program is designed.

For example, you might want to update the inventory using a program named INVENTORY. By specifying option 1 and the name INVENTORY on the menu, you start the INVENTORY program and it begins executing the functions it was designed to do. Those functions may include communicating with you at the work station by showing a display on the work station screen and requesting you to enter information on that display.

The following summarizes the procedure for calling a program by means of the program call menu.

1. Select option 1 by keying a 1 into the Option field (B in Figure 4-1).

   The cursor then moves to the Program name field (C in Figure 4-1).

2. Key in the name of the program to be called.

3. Press the Field Exit key to move the cursor to the first line of the Parameters or message field (D in Figure 4-1).

4. Key in any parameter values to be used by the program when it is called. If you enter more than one value, separate the values by a space; that is, after keying in one value, press the Spacebar before keying in the next value.

5. Press the Enter/Rec Adv key.

Steps 3 and 4 are optional. If you have no parameters to specify, you press the Enter/Rec Adv key after step 2.

A program name can be entered as a simple name or in the form of a *qualified name*. A qualified program name consists of the program name followed by the name of the library in which the program is located. The two names are connected by a period.

```
Program     Library
Name        Name
⏞           ⏞
INVENTORY.RECORDS
         └(connector)
```

You must not put a space between either of the names and the period.

```
INVENTORY.RECORDS
         V
      (no space allowed here)
```

If there is a space before or after the period, the system interprets the program name and library name as two separate names, which results in an error.

The library name contained in the qualified program name tells the system where to find the program you are calling. If you enter only the program name without the library name, the system searches through a specified library list, identified as *LIBL, until it finds the program.

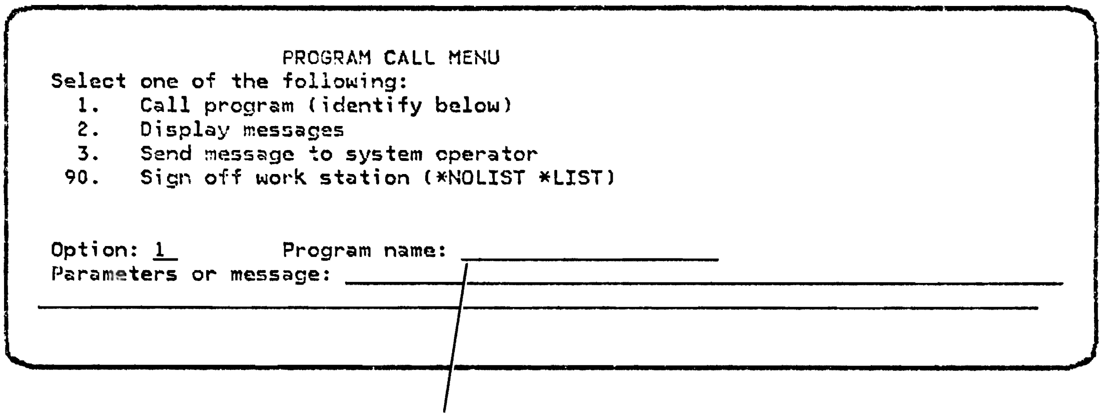
You should always enter the program name and parameter values in the exact form that they are given to you. This ensures that the name and values are identified correctly to the system. If you are entering multiple parameter values, remember to put spaces between them, so that the system interprets them as separate values, not as a single value.

When you have entered your information correctly, you receive the first display (or the results) of the program you called.

Examples:

1. To call the program named INVPRT that prints the weekly inventory:

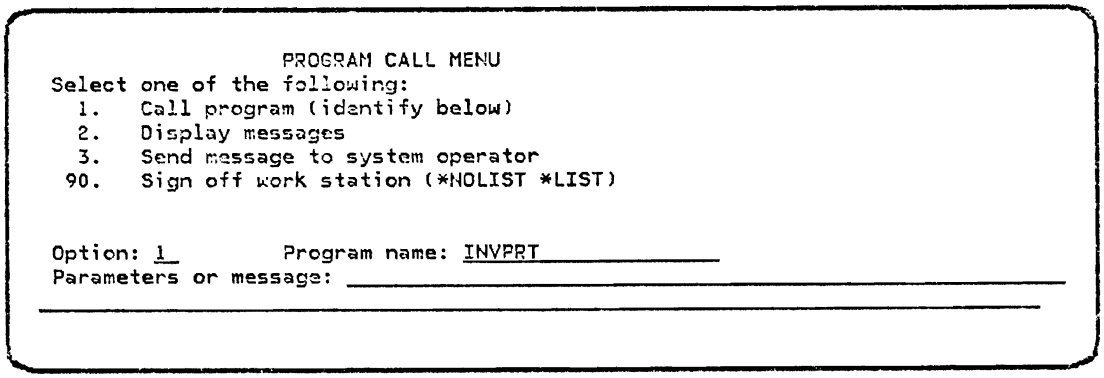
   a. Key in the number 1.



```
                         PROGRAM CALL MENU
Select one of the following:
  1.   Call program (identify below)
  2.   Display messages
  3.   Send message to system operator
 90.   Sign off work station (*NOLIST *LIST)


Option: 1_        Program name: ____________________
Parameters or message: _________________________________________________________
______________________________________________________________________________
```

Cursor moves to here.

   b. Key in INVPRT.

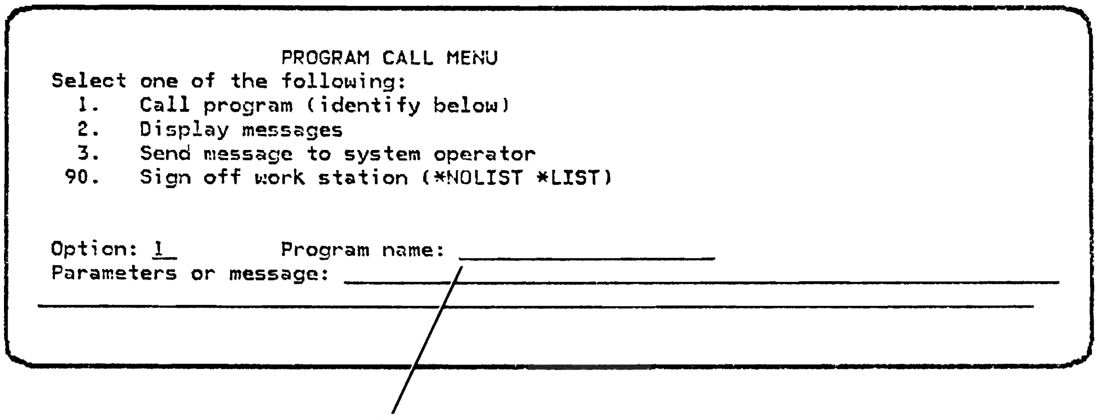


```
                         PROGRAM CALL MENU
Select one of the following:
  1.   Call program (identify below)
  2.   Display messages
  3.   Send message to system operator
 90.   Sign off work station (*NOLIST *LIST)


Option: 1_        Program name: INVPRT______________
Parameters or message: _________________________________________________________
______________________________________________________________________________
```

   c. Press the Enter/Rec Adv key.

2. To call the program STOCK and place the parameters NEW, CURRENT, and SHIP under its control:

   a. Key in the number 1.

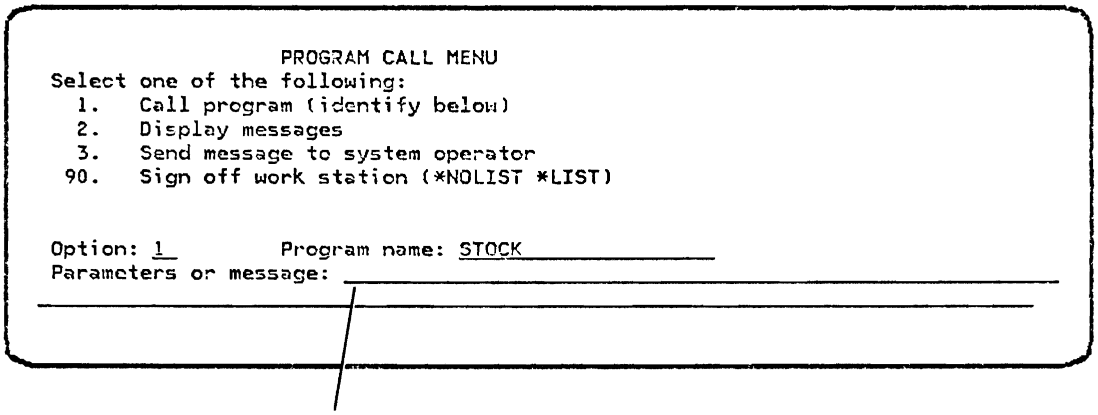


```
                    PROGRAM CALL MENU
Select one of the following:
  1.   Call program (identify below)
  2.   Display messages
  3.   Send message to system operator
 90.   Sign off work station (*NOLIST *LIST)


Option: 1_        Program name: ___________________
Parameters or message: ____________________________________________________
___________________________________________________________________________
```

Cursor moves to here.

   b. Key in STOCK.

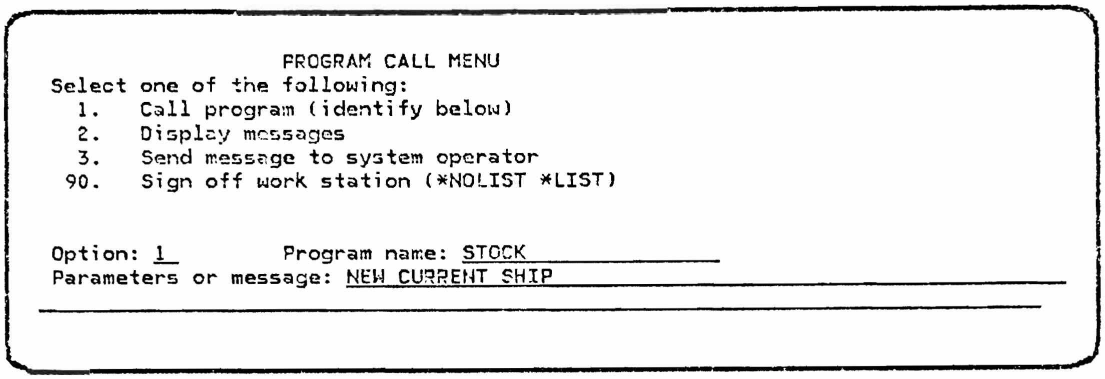
   c. Press the Field Exit key.



```
                    PROGRAM CALL MENU
Select one of the following:
  1.   Call program (identify below)
  2.   Display messages
  3.   Send message to system operator
 90.   Sign off work station (*NOLIST *LIST)


Option: 1_        Program name: STOCK______________
Parameters or message: ____________________________________________________
___________________________________________________________________________
```

Cursor moves to here.

d. **Key in NEW, CURRENT, and SHIP, putting a space between the words.**



```
                    PROGRAM CALL MENU
Select one of the following:
  1.   Call program (identify below)
  2.   Display messages
  3.   Send message to system operator
 90.   Sign off work station (*NOLIST *LIST)


Option: 1           Program name: STOCK
Parameters or message: NEW CURRENT SHIP
```

e. **Press the Enter/Rec Adv key.**

HOW TO DISPLAY MESSAGES (OPTION 2)

From time to time, messages are sent to your work station by other users of the system, by the system operator who controls the system, or by programs running on the system. Those messages are delivered to a message queue for your work station.

When a message arrives in your work station message queue, you are signaled in the following manner:

- The Message Waiting indicator on the side of your display screen changes from (-) to (■).

- The work station keyboard buzzer sounds briefly.

To display the messages, use option 2 on the program call menu.

The procedure is:

1. Select option 2 by keying a 2 into the Option field (B in Figure 4-1).

2. Press the Enter/Rec Adv key.

You receive a separate display that lists all messages on the work station message queue.

For a description of message displays, see *How to Handle Messages from a Message Queue* in Chapter 3.

HOW TO SEND MESSAGES TO THE SYSTEM OPERATOR (OPTION 3)

There may be occasions when you need to contact the system operator to obtain information or request assistance. You can do this by using option 3 of the program call menu.

The procedure is:

1. Select option 3 by keying a 3 into the Option field (Ⓑ in Figure 4-1).

   The cursor then moves to the Program name line (Ⓒ in Figure 4-1).

2. Press the Field Exit key to move the cursor to the first line of the Parameters or message field (Ⓓ in Figure 4-1).

3. Key in your message. If the text is too long to fit on the first line, it automatically continues on the second line as you key it in. The message can be up to 132 characters long.

4. Press the Enter/Rec Adv key.

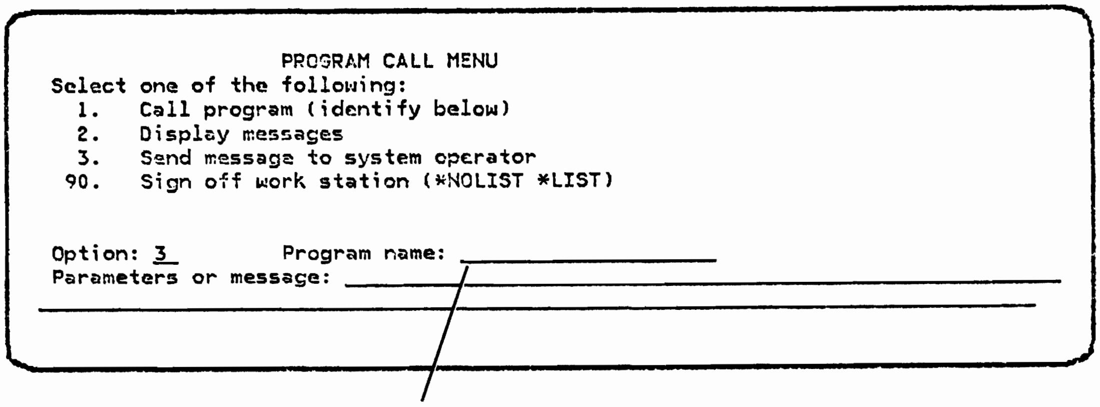
Example:

You want to send the message

There will be a department meeting today at 3:00 p.m.

to the system operator from the program call menu. Therefore, you do the following:

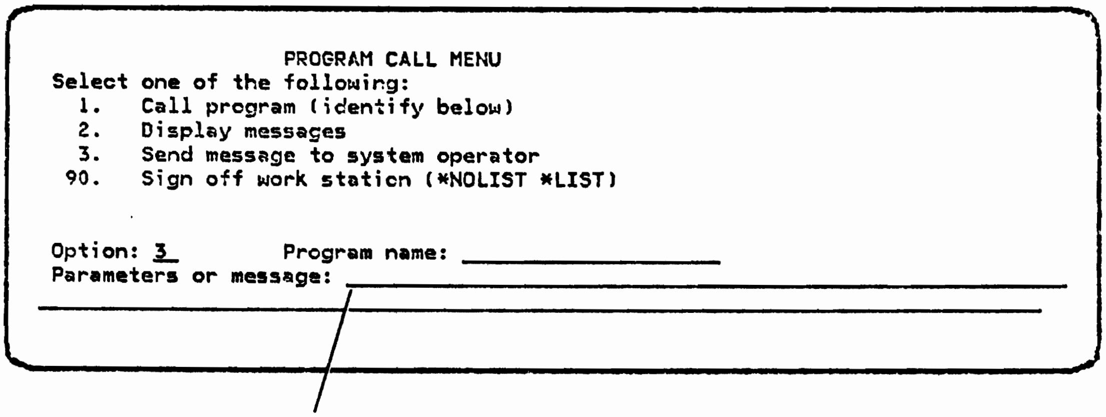
1. Key in the number 3.



```
                    PROGRAM CALL MENU
Select one of the following:
  1.   Call program (identify below)
  2.   Display messages
  3.   Send message to system operator
 90.   Sign off work station (*NOLIST *LIST)

Option: 3_         Program name: __________________
Parameters or message: ___________________________________________________
___________________________________________________________________________
```

Cursor moves to here.

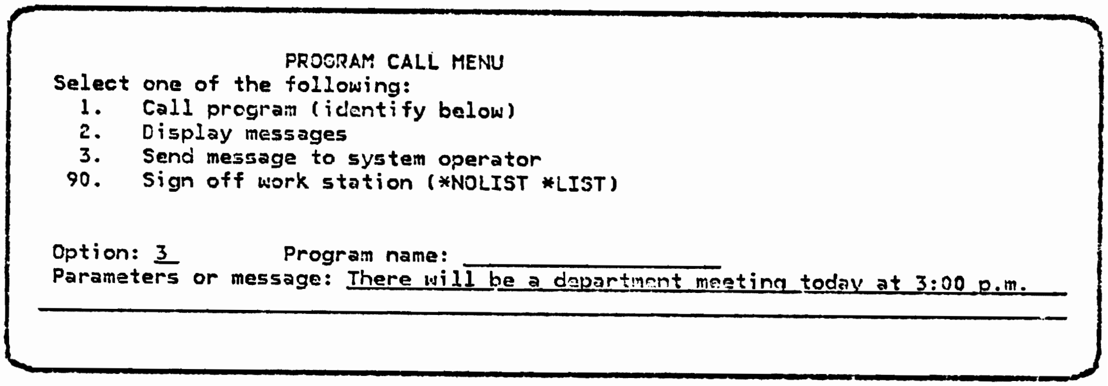
2. **Press the Field Exit key.**



```
                    PROGRAM CALL MENU
Select one of the following:
  1.   Call program (identify below)
  2.   Display messages
  3.   Send message to system operator
 90.   Sign off work station (*NOLIST *LIST)

Option: 3_        Program name: ____________________
Parameters or message: ________________________________________________________
______________________________________________________________________________
```

Cursor moves to here.

3. **Key in the message.**



```
                    PROGRAM CALL MENU
Select one of the following:
  1.   Call program (identify below)
  2.   Display messages
  3.   Send message to system operator
 90.   Sign off work station (*NOLIST *LIST)

Option: 3_        Program name: ____________________
Parameters or message: There will be a department meeting today at 3:00 p.m.___
______________________________________________________________________________
```

4. **Press the Enter/Rec Adv key.**

HOW TO SIGN OFF (OPTION 90)

Option 90 of the program call menu allows you to sign off your work station. When you sign off, your job at the work station is ended and the work station becomes available for someone else to use.

The procedure is:

1. Select option 90 by keying a 90 into the Option field (Ⓑ in Figure 4-1).

   The cursor then moves to the Program name field (Ⓒ in Figure 4-1).

2. If you want the job log, press the Field Exit key to move the cursor to the first line of the Parameters or message field (Ⓓ in Figure 4-1) and then key in *LIST.

3. Press the Enter/Rec Adv key.

HOW YOU RETURN TO THE PROGRAM CALL MENU

When you call a program using option 1 of the program call menu, the menu may be replaced on your work station screen by a different display or set of displays that are under control of the called program. Generally, the display, or the primary display in a set of displays, indicates how you exit the program. When you take the indicated action, such as pressing a specific function key, the program call menu returns to your screen.

If you cannot determine how to exit the program, contact your supervisor or the data processing manager for your system.

When you use option 2 of the program call menu to display messages, you receive a separate message display as described in Chapter 3. After you have viewed all of the messages, you can return to the program call menu by pressing the Enter/Rec Adv key or the CF1 key.

Generally, if you have received a display or set of displays, such as the message display, as a result of invoking a CPF function from the program call menu, you can immediately exit the function and return to the menu by pressing the CF1 key (see *How to Use Command Function Keys* in Chapter 1).

Part III

Operations Using CPF Commands

Chapter 5. Using CPF Control Language

WHAT IS CPF CONTROL LANGUAGE?

The CPF control language (CL) is an extensive set of individual commands that are used to control the system or information in the system. Each command performs a single function.

In its most commonly used form, a CL command consists of a *command name* followed by one or more *parameters* for which you supply specific *values*. A typical command and the rules for entering it are shown in Figure 5-1. A summary of command syntax follows. For a complete description, refer to the *CL Reference Manual* (see *About This Manual*).

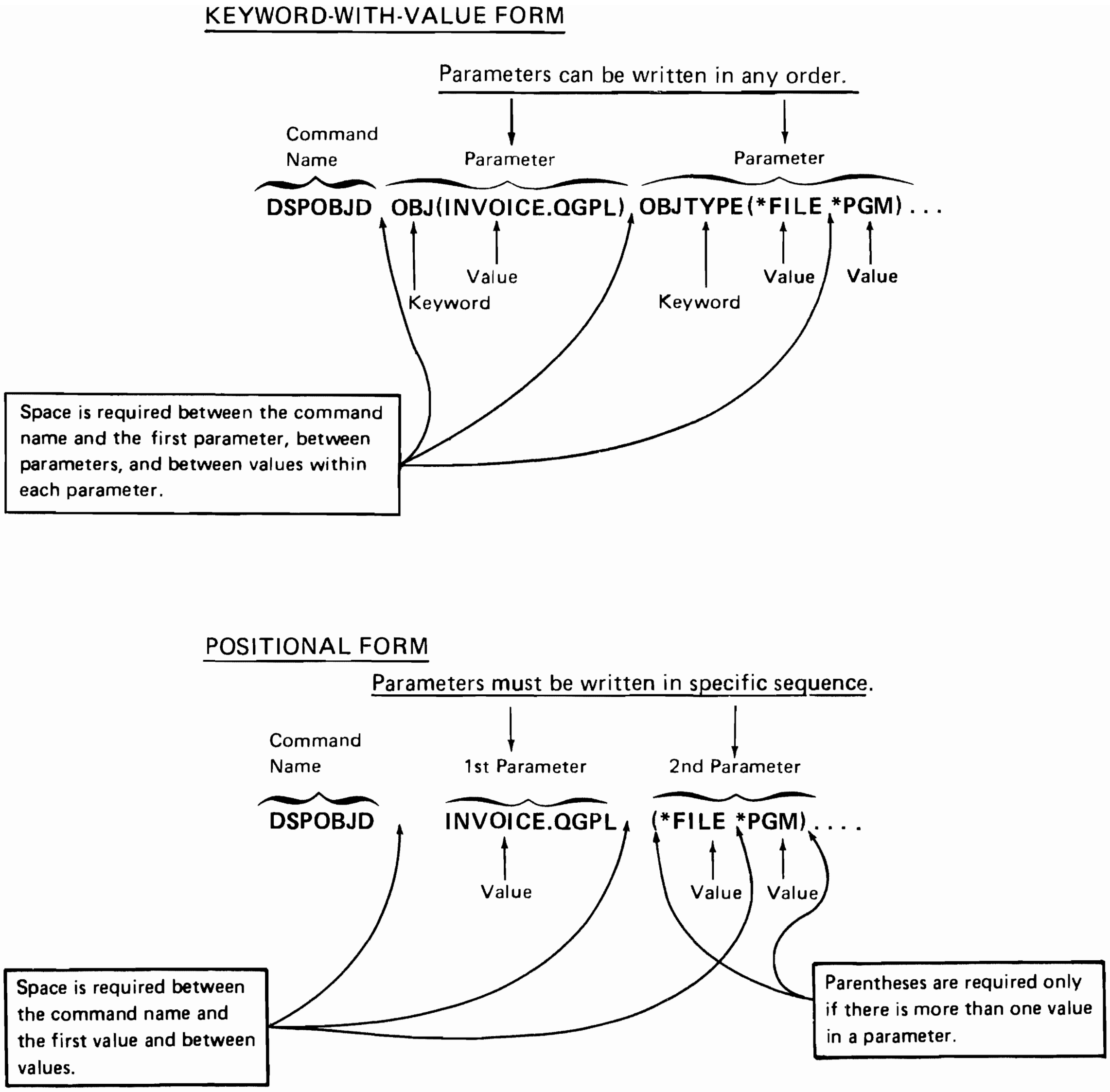

KEYWORD-WITH-VALUE FORM
Parameters can be written in any order.
Command Name
Parameter
Parameter
DSPOBJD OBJ(INVOICE.QGPL) OBJTYPE(*FILE *PGM) . . .
Value
Keyword
Keyword
Value
Value
Space is required between the command name and the first parameter, between parameters, and between values within each parameter.
POSITIONAL FORM
Parameters must be written in specific sequence.
Command Name
1st Parameter
2nd Parameter
DSPOBJD INVOICE.QGPL (*FILE *PGM). . . .
Value
Value
Value
Space is required between the command name and the first value and between values.
Parentheses are required only if there is more than one value in a parameter.

Figure 5-1. Command Coding Format

COMMAND SYNTAX

The general form of a CL command is:

```
[label:] command-name[.library-name] [parameter-set]
```

where the brackets indicate optional items. The general form of the command shown in Figure 5-1 might look like:

```
LOOP1: DSPOBJD.QSYS OBJ(INVOICE.QGPL) OBJTYPE(*FILE *PGM)
```

Label

The command label is used when needed to provide a tag, or identifier, for a command that is part of a program composed of CL commands. A command label is generally not needed for interactive execution of commands. If you use a label, it must be followed by a colon (:).

Command Name

The command name indicates what the command does. Every command must at least have a command name; a few commands consist of only a command name.

The command name is composed of abbreviations of one or more words that describe the function to be performed and, usually, the object of the action. Examples are:

CRTLIB (Create Library)

DSPMSG (Display Messages)

MOVOBJ (Move Object)

Usually, the abbreviation for each word making up the command name is three characters long as in the examples just shown. For some command names, the abbreviation of the final word or words is shortened to one or two characters such as:

CPYF (Copy File)

CRTPF (Create Physical File)

DSPOBJD (Display Object Description)

A few command names are composed of a complete word that describes the action to be taken, such as:

CALL (Call a program)

SIGNOFF (Sign Off)

The abbreviation for a particular word is the same in all command names that contain the word. For example, the command names for all display commands begin with the abbreviation DSP, and the command names for all commands that act on a library contain the abbreviation LIB. Once you are familiar with the abbreviations, you can determine what a particular command does just from its command name.

Library Qualifier

A library qualifier following a command name indicates the library in which the command description is to be found by the system. It is connected to the last character of the command name by a period, such as

```
DSPOBJD.QSYS
```

If you do not specify a library qualifier for the command name when you enter a command, the system searches through the list of libraries (*LIBL) for the command name until it finds the command definition that tells it how to respond to the command.

Parameter Set

A command may have one or more parameters, or no parameters at all. For some commands, specific parameters are required and must be included when the command is used. Some commands have both required and optional parameters. The optional parameters may or may not be used as your needs dictate. Other commands have only optional parameters.

Each parameter contains one or more values that provide the details necessary for processing a command. These values may, for example, specify where data is to be found or placed, indicate what the data is to be called, identify a device to be used, or provide the text of a message to be sent. When you enter a command, you must determine the values to be used in each parameter of that command.

Although a command may have many parameters, you can usually execute the command by entering only a few of those parameters. In each case, you must enter the command name and any required parameters, usually one or two. However, you can leave out any optional parameters, allowing the system to use a preassigned default value for each missing parameter. By not specifying an optional parameter, you are, in effect, entering the default value of that parameter.

A command with its associated set of parameters can be entered in either of two ways (see Figure 5-1):

- *Keyword-with-Value Form*: Each parameter consists of a keyword followed immediately by one or more values in parentheses. The parameters can be entered in any order following the command name:

  ```
  DSPOBJD OBJ(INVOICE.QGPL) OBJTYPE(*FILE *PGM)

  DSPOBJD OBJTYPE(*FILE *PGM) OBJ(INVOICE.QGPL)
  ```

- *Positional Form*: The keyword is not specified and the position of the values after the command name defines which parameter they relate to. In effect, the value or values of a parameter become the parameter itself. The number of parameters that can be specified in positional form may be limited. This limit is shown in the syntax diagram for the command in the *CL Reference Manual*. The values must be entered in a specific sequence:

  ```
  DSPOBJD INVOICE.QGPL (*FILE *PGM)
  ```

  If you enter the parameters in the wrong sequence, such as

  ```
  DSPOBJD (*FILE *PGM) INVOICE.QGPL
  ```

  the system associates the values with the wrong parameter, or does not process the command at all.

  If the values for certain parameters in the sequence are not included, the absence of each parameter must be indicated by a null value symbol, *N:

  ```
  DSPOBJD INVOICE.QGPL *FILE *N *LIST
  ```

  The null symbol (*N) indicates that the default value is to be used for the parameter in that position.

  However, no indication is needed for parameters that follow the last parameter actually used in the sequence:

  DSPOBJD INVOICE.QGPL (*FILE *PGM) ~~*N~~ ~~*N~~

  The correct sequence for entering each command in the positional form and in the positional limit is shown in the syntax diagram for the command in the *CL Reference Manual.*

A command can be entered with some of its parameters in the positional form and others in the keyword form. If it is entered that way, however, the positional parameters must appear first:

```
DSPOBJD INVOICE.QGPL *FILE DETAIL(*FULL) OUTPUT(*LIST)
```

This combination of the two forms would be useful, for example, if you enter the first few parameters of a command and then skip several parameters before entering another parameter. If you enter the entire command in positional form, you have to enter an *N for each missing parameter up to the last parameter used.

Multiple Parameter Values

Some types of parameters can contain more than one value at a time. These multiple values are known as a *list*. For example, the OBJTYPE parameter in Figure 5-1 contains a list of two values, *FILE and *PGM. When you enter multiple values in a parameter, you indicate that they go together by enclosing them in parentheses whether using the keyword-with-value form

```
OBJTYPE(*FILE *PGM)
```

or the positional form

```
(*FILE *PGM)
```

The values *FILE and *PGM in this example are typical of a simple list, which is the most common type of list used in CL commands. A simple list consists of values that are all the same type. That is, if one of the values in a simple list is an object name, all of the other values in the list are object names; if one of the values is an object type, all of the other values are object types.

A few commands contain parameters that can have mixed lists. A mixed list consists of a set of values that are a different type. For example, in the INCREL parameter of the Copy File (CPYF) command, the mixed list

```
(*IF REC *EQ 96)
```

contains a relation (*IF), a field name (REC), an operator (*EQ), and a number (96).

When you enter a parameter containing one or more mixed lists, each mixed list must be enclosed in parentheses. This means that you must use double parentheses whether entering a single mixed list

```
INCREL((*IF REC *EQ 96))
```

or multiple mixed lists

```
INCREL((*IF REC *EQ 96) (*AND VAL *EQ 50))
```

in the keyword-with-value form or entering a single mixed list

```
((*IF REC *EQ 96))
```

or multiple mixed lists

```
((*IF REC *EQ 96) (*AND VAL *EQ 50))
```

in the positional form.

Delimiters

When you write or type a sentence, you put a space between the words to indicate that they are separate elements. Similarly, when you enter a command, you insert a space, or blank, between the command name, parameters, and parameter values to indicate that they are separate elements. The blank serves as the basic delimiter in CL commands (see Figure 5-1).

For specific rules on the use of blanks and other delimiters in CL commands, refer to the *CPF Programmer's Guide*.

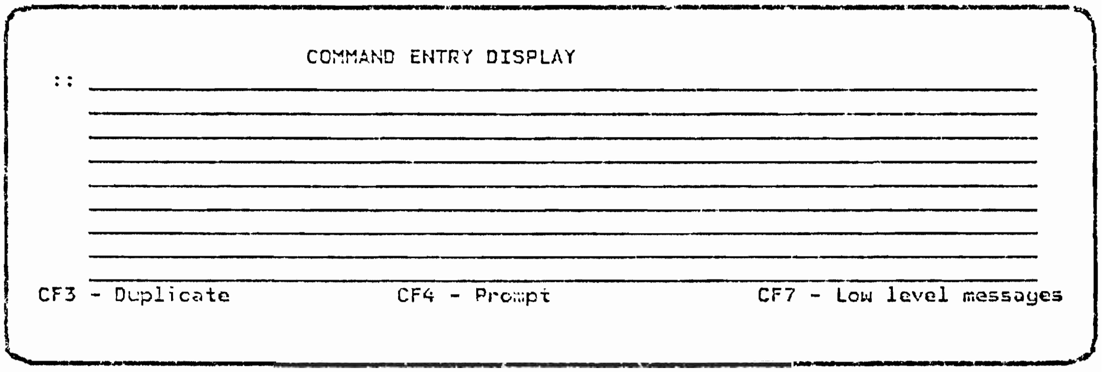
HOW TO USE GENERAL COMMAND ENTRY FACILITIES

If you know what command to use and what parameter values to specify for that command, you can enter the complete command into the system using a display known as the command entry display. If you are not sure what commands are available or what values to use, however, you can obtain assistance by requesting menu and prompt displays. If the information to be displayed contains any undisplayable hexadecimal characters, they will be converted to a hex '1F' when displayed and will appear on the screen as either a blank or a reverse image blank (■).

Command Entry Display

If your work involves entering CL commands, after you sign on, the next display you see or that you can request from a menu is the command entry display (Figure 5-2). The display provides, in effect, a continuous writing pad for entering commands. You enter one command at a time; the system processes the command and indicates its response before you enter the next command.

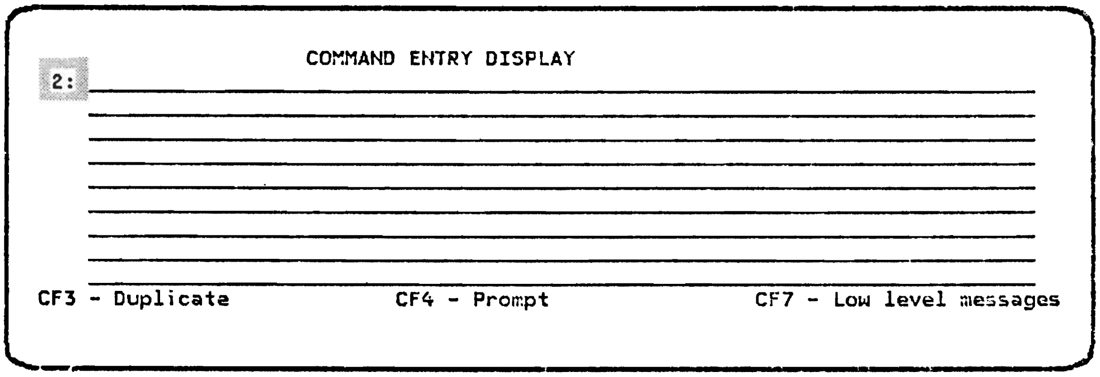
When the command entry display is the first command-entry-capable display you receive, the entry area on the display begins with the symbol :: and is said to be at the first level of nesting (Figure 5-2).



```
                          COMMAND ENTRY DISPLAY
  ::  ________________________________________________________________
      ________________________________________________________________
      ________________________________________________________________
      ________________________________________________________________
      ________________________________________________________________
      ________________________________________________________________
      ________________________________________________________________
      ________________________________________________________________
      ________________________________________________________________
CF3 - Duplicate            CF4 - Prompt              CF7 - Low level messages
```

Figure 5-2. Command Entry Display

The command entry display can also appear in a somewhat different form; the :: symbol might be replaced by the symbol 2: (or some other number depending on the level of nesting).



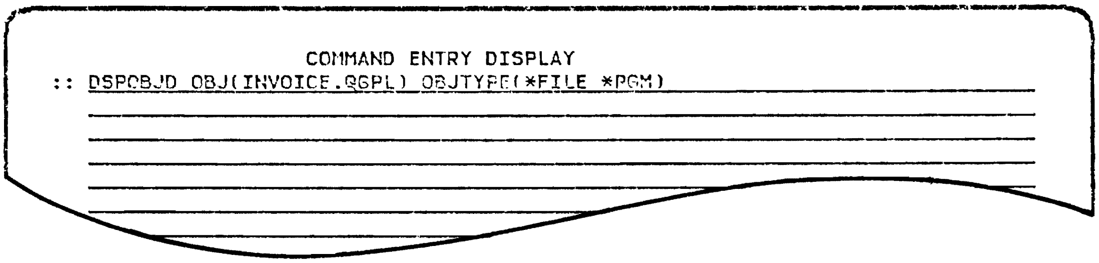
```
                     COMMAND ENTRY DISPLAY
2: ________________________________________________________________

CF3 - Duplicate          CF4 - Prompt          CF7 - Low level messages
```

The 2: symbol indicates that the program (QCL) controlling the command entry display is a second-level nested function that is executing with another program. The function of the command entry display is the same.

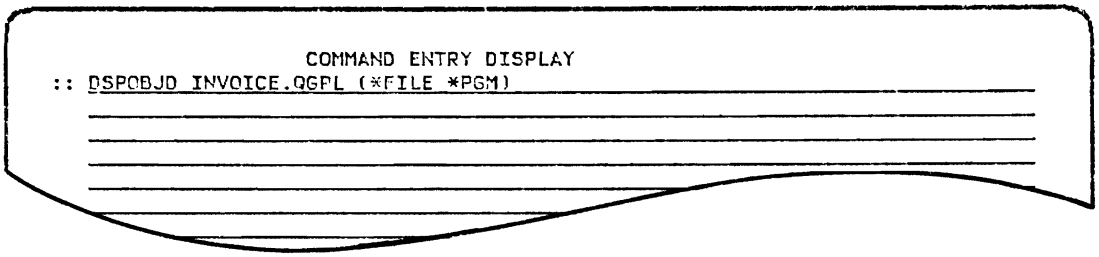
Pressing the CF1 key or executing a RETURN command from the command entry display returns you to the previous level, and processing is resumed where you selected command entry from. All commands you enter are at the same level of nesting until you press the CF1 key or request another lower level of nesting.

How the Display Operates

The symbols :: or n: followed by an underlined area show where you enter your input. The cursor is automatically placed at the first character position of the input area. Commands can be entered in either the keyword-with-value or positional form:



```
                        COMMAND ENTRY DISPLAY
:: DSPOBJD OBJ(INVOICE.QGPL) OBJTYPE(*FILE *PGM)
```



```
                        COMMAND ENTRY DISPLAY
:: DSPOBJD INVOICE.QGPL (*FILE *PGM)
```

If the command is too long to fit on one line, you can complete it on subsequent lines. When you reach the end of a line, the cursor automatically moves to the beginning of the next line. To complete your input, continue keying it in as if you were typing the entire input on a single line; that is, if you would normally put a space after the last character of the first line, put that space at the beginning of the next line. (Continuation characters that indicate continuation on the next line are neither required nor allowed.) If you choose, you can break the command at a convenient point where blanks are allowed (such as between two parameters) before reaching the end of a line. In that case, you press the ↵ (New Line) key to move the cursor to the next line.

If you need assistance in entering a command, see *Command Selection and Prompting* later in this chapter.

When you press the Enter/Rec Adv key, the system acts on the command you have keyed in. Default values are used for optional parameters you did not enter. Any system messages in response to the command you entered appear on the next line or lines below your entry. If the command is successfully executed, you could receive no messages at all or one or more messages giving details of the completion. If there is an error, you receive one or more error messages. The underlined input area is displayed again on the next line below your last entry, or below the last message in response to your last entry. To show where to make your next entry, the input area is preceded by the entry symbol ::. The underline under your last command disappears to indicate that the line containing the command is no longer an input field.

As you continue to enter commands and receive responses, the underlined input area continues to appear on the next available line below your last request or message.

If a command you enter results in a new display or displays, you can return to the command entry display by:

- Pressing the Enter/Rec Adv key when on the final display in a series
- Pressing the CF1 key regardless of which display you are viewing

Handling Errors

If you enter a command that is in error, you receive one or more error messages on the following line or lines, and the entry symbol :: appears on the line below the last message. You must enter the command again in the correct form if you want it executed. Note that these error messages are the same type of messages you receive on the bottom line of most other displays. The messages displayed are the messages resulting from the function you selected.

There may be additional messages generated by lower level programs that are not normally displayed. If those lower level messages are needed for debugging, you can request them by pressing the CF7 key after you receive the message or messages normally displayed in response to your entered command. The additional messages are then displayed in the proper chronological sequence before or after the normal response messages they are associated with. If you want to return to normal command entry after viewing the lower level messages, press the CF1 key or the Enter/Rec Adv key.

The lower level messages begin with those for the last command you received on your display or with the message you identified by positioning the cursor under it when the CF7 key was pressed. You can then roll back and forth through the lower level messages using the roll keys. If you need additional information to figure out what a message means, position the cursor under the message in question and press the Help key to obtain additional details of the error.

If your work involves using specific application displays, you may see a function check message. This message indicates that an unexpected message was sent to the function you requested. The function check message identifies the unexpected low level message by a message ID, such as:

```
Function check. CPF3335 unmonitored by PAYPGM at X'0017'.
```

If you are using the command entry display, you can press the CF7 key to display a low level message, identified as CPF3335 in this example. If you are using a display other than the command entry display when you receive a function check message, refer to the *Messages Guide: CPF, RPG III, and IDU* or the *Messages Guide: COBOL* for additional information about the message.

Correcting A Command

There are two ways you can correct a command. You can:

1. Key in the entire command again in the correct form. For more information see *Command Selection and Prompting* later in this chapter.

2. Press the Enter/Rec Adv key.

Or you can:

1. Press the CF3 key, which duplicates your previously entered incorrect command into the current input area.

2. Key in new characters on top of those that are incorrect.

3. Press the Enter/Rec Adv key.

Duplicating Previously Entered Commands

Any commands you have entered that you can still view on the screen, including commands you can roll onto the screen using the roll keys, can be duplicated into the input entry area. To duplicate a command:

1. Position the cursor on the line containing the command.

2. Press the CF3 key.

The command then appears in the underlined input field. You can press the Enter/Rec Adv key to execute that same command again, or you can first modify it by keying in new characters over some of those shown and press the Enter/Rec Adv key to execute the modified command. You might use this procedure, for example, to enter a command that is only slightly different from one you had entered previously.

Erasing Your Current Input

Any command you have keyed or duplicated into the current input area can be erased completely by using the Erase Input key; that is:

1. Press and hold the Shift key.

2. Press the ERASE INPT HOME key.

Commands previously entered are not erased because they are no longer in the input area.

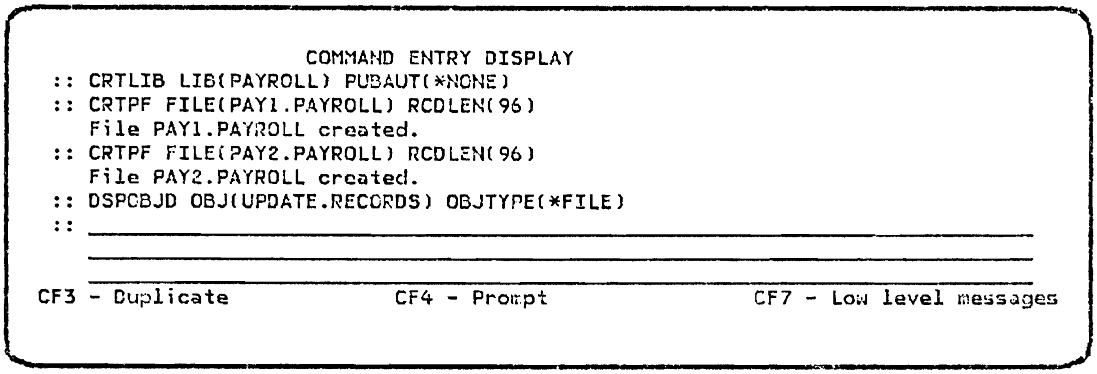
Effects of Job Logging Level

A particular logging level is specified for your interactive job that determines what information is stored in the job message queue for the job in case you should request a job log. See Chapter 7, *Signing Off Through a Command.* This logging level also affects the contents of the command entry display on your work station screen. You can change the logging level of your interactive job by using the LOG parameter of the Change Job (CHGJOB) command (see the CHGJOB command in the *CL Reference Manual*).

If the logging level is such that previously entered commands and their associated response messages are not retained in the job message queue, the information on the display is shifted up after each command is executed so that you see only the most recently entered command and its associated messages on the screen at any time. The remaining lines of the screen display an input field for your next entry.

If the previously entered commands and their associated messages are retained in the job message queue, they remain on the display. Therefore, the screen is gradually filled with your entries and system responses as you continue to enter commands (Figure 5-3).

When you approach the bottom of the screen, the display is automatically shifted up to allow space for more entries. Thus, old lines are continually moving off the top of the screen and new lines onto the bottom.



```
                          COMMAND ENTRY DISPLAY
 :: CRTLIB LIB(PAYROLL) PUBAUT(*NONE)
 :: CRTPF FILE(PAY1.PAYROLL) RCDLEN(96)
    File PAY1.PAYROLL created.
 :: CRTPF FILE(PAY2.PAYROLL) RCDLEN(96)
    File PAY2.PAYROLL created.
 :: DSPCBJD OBJ(UPDATE.RECORDS) OBJTYPE(*FILE)
 :: ________________________________________________________________________
    ________________________________________________________________________
    ________________________________________________________________________
CF3 - Duplicate               CF4 - Prompt                  CF7 - Low level messages
```

Figure 5-3. Command Entry Display after Several Entries

This page is intentionally left blank.

Adjusting the Display

As needed, you can move the display up or down using the roll keys (see *How to Handle Roll Capability and Multiple Displays* in Chapter 1). Any lines that have been rolled off the top of your screen, and are retained in your job message queue, can be returned to the screen for review by pressing the Roll Down key.

Regardless of how far you have rolled the display, you can return immediately to the input entry area by pressing the Enter/Rec Adv key.

Command Selection and Prompting

When you are using the command entry display, you can request command prompting in selecting and entering commands. That assistance is in the form of menu and prompt displays. A menu guides you to the proper command for performing a specific function. Once you know what command to use, a prompt display guides you in entering suitable parameter values that lead to execution of the command.

To obtain assistance, you make a prompt request by pressing the CF4 key. This key is called the *prompt* key. (A ? placed in front of a command will result in the same prompt request, although the ? is intended primarily for use within CL programs.)

The relationship of the command entry display to menus and prompts obtained through a prompt request is summarized in Figure 5-4. Note that:

- If you press the prompt (CF4) key without entering a command name, you receive a menu of command categories to assist you in selecting the proper command.

- If you key in a valid command name and then press the prompt (CF4) key, you receive a prompt for the parameters of that command.

- If you key in a valid command name plus one or more parameters with the valid values and then press the prompt (CF4) key, you receive a prompt for the parameters of the command that fills in the values you specified.

- If you key in a command name, or a command name plus parameters, that contains invalid syntax and press the prompt (CF4) key, the command entry display is redisplayed with messages describing the errors.

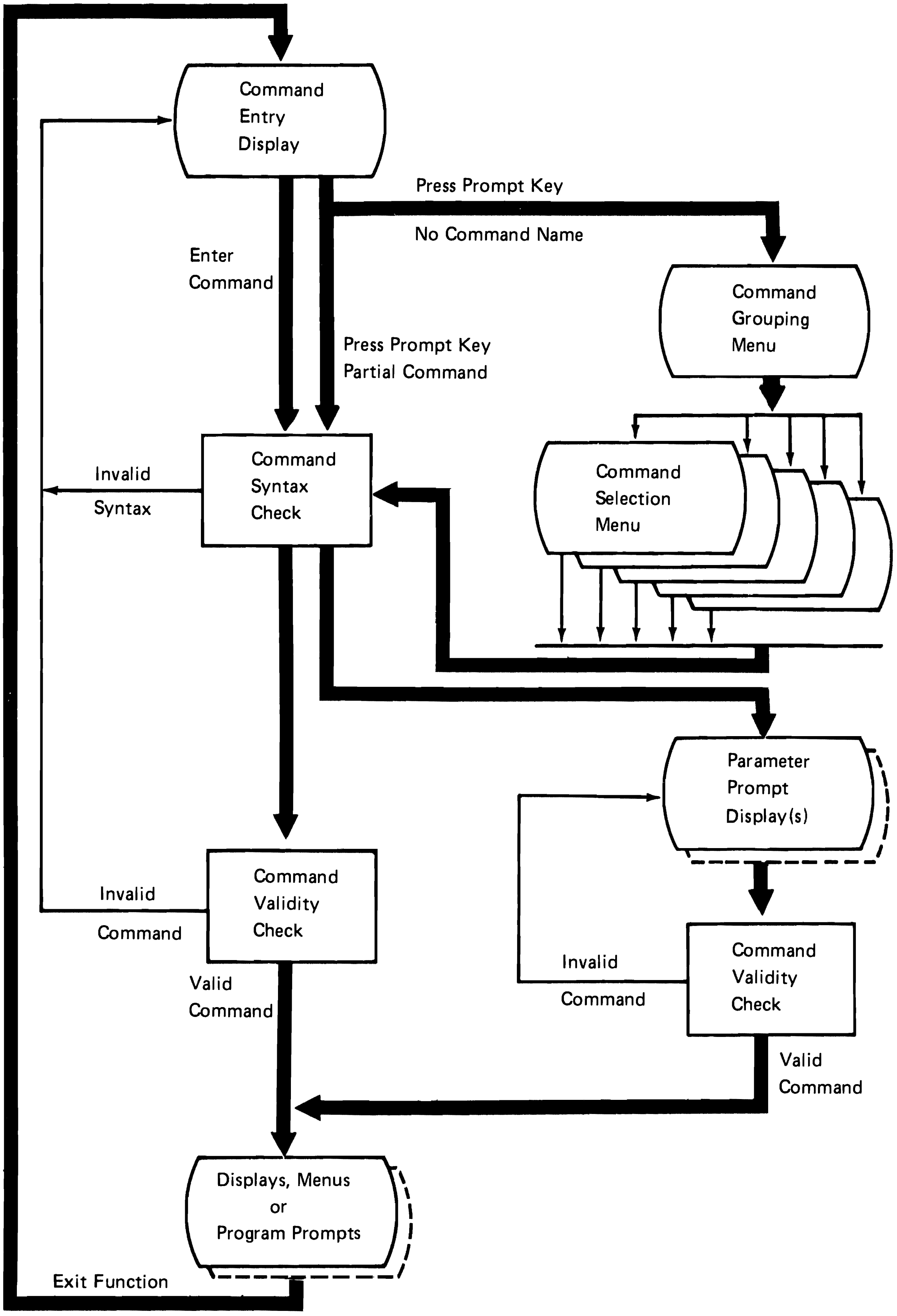
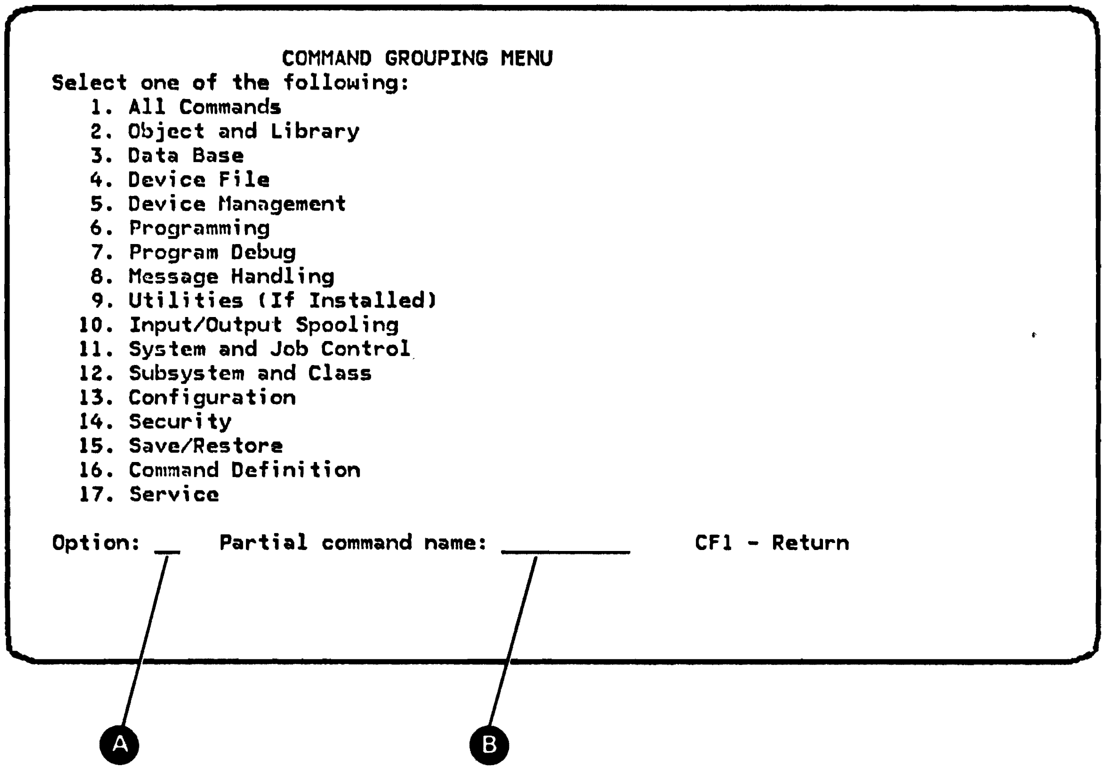


Figure 5-4. Relationship of Displays

Command Selection

There may be occasions when you are uncertain of what commands are available to execute specific functions. Menus provide the means for you to find a needed command.

If you press the prompt (CF4) key without making an entry on the command entry display, the system displays a menu (Figure 5-5) that lists functional categories of commands.



```
                  COMMAND GROUPING MENU
Select one of the following:
   1. All Commands
   2. Object and Library
   3. Data Base
   4. Device File
   5. Device Management
   6. Programming
   7. Program Debug
   8. Message Handling
   9. Utilities (If Installed)
  10. Input/Output Spooling
  11. System and Job Control
  12. Subsystem and Class
  13. Configuration
  14. Security
  15. Save/Restore
  16. Command Definition
  17. Service

Option: __   Partial command name: __________     CF1 - Return
```

Figure 5-5. Command Grouping Menu

To make a selection from this menu:

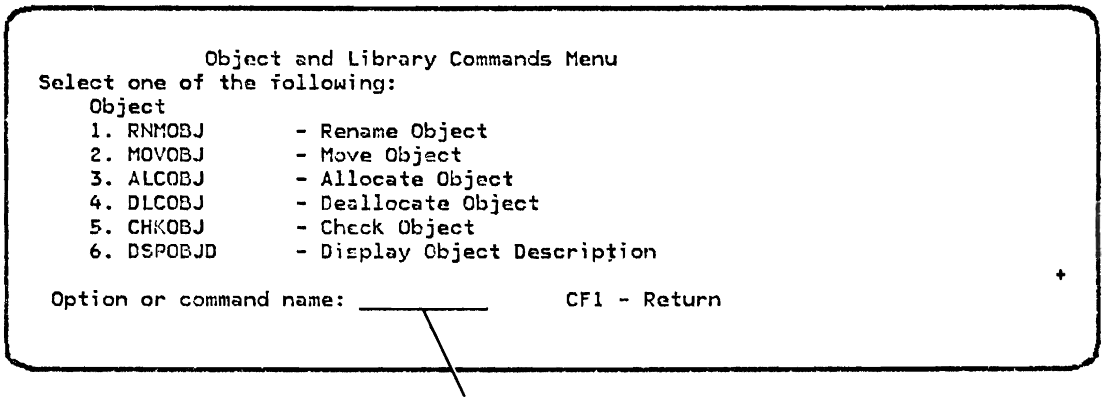
1. Key the number of one of the 17 command groups into the Option field Ⓐ.

2. If you want to restrict the commands to only those that begin with certain characters within the group you selected in step 1, key in a partial command name (such as DSP for display) into the Partial command name field Ⓑ. (This step is optional.)

3. Press either the Enter/Rec Adv key or the CF4 key.

   Note: The Option field of the command grouping menu is a 2-character input field. Therefore, if you select one of the first nine options and key in only a 1-digit number (1-9), the cursor does not automatically move to the Partial command name field; you must move the cursor yourself by pressing the Field Exit key.

The system responds with a command selection menu. This menu provides a list of commands that is determined by what you entered on the command grouping menu. If you selected one of the 17 command categories, but made no entry in the Partial command field, the resulting command selection menu displays all of the commands in that category, such as shown in Figure 5-6. If you also entered characters in the Partial command name field, the resulting menu displays only the commands of the selected group whose name contains the characters.



```
             Object and Library Commands Menu
Select one of the following:
    Object
    1. RNMOBJ        - Rename Object
    2. MOVOBJ        - Move Object
    3. ALCOBJ        - Allocate Object
    4. DLCOBJ        - Deallocate Object
    5. CHKOBJ        - Check Object
    6. DSPOBJD       - Display Object Description
                                                                +
 Option or command name: __________       CF1 - Return
```

Enter selected option number or name of command here.

Figure 5-6. Typical Command Selection Menu

It may be necessary to use the roll keys to display additional commands in the group. A + is shown in the lower right corner of the command display if more commands are available. To make a selection:

1. Key the selected option number or command name into the input field at the bottom of the menu.

2. Press either the Enter/Rec Adv key or the CF4 key.

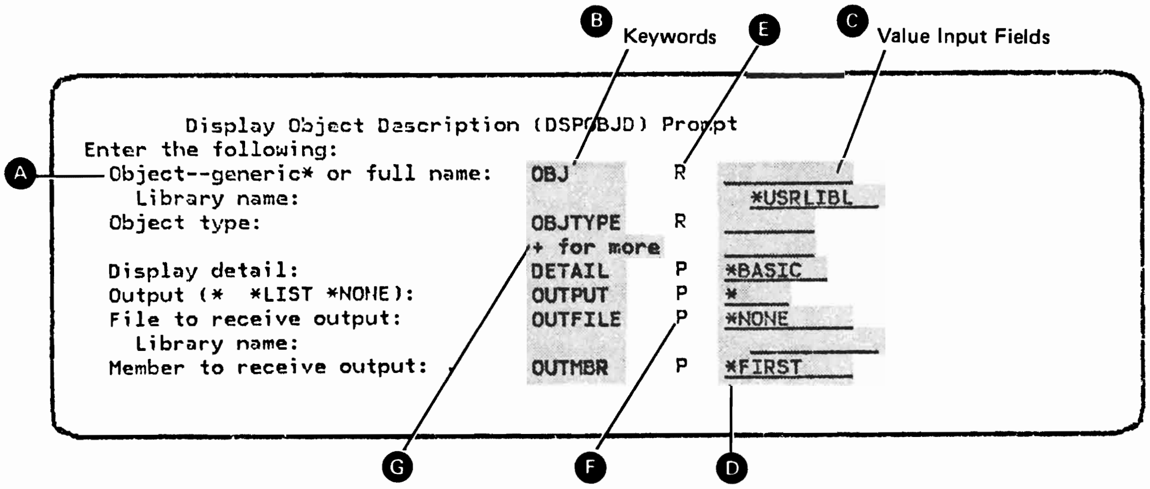
If you enter an invalid option number or leave the field blank, an error message will be displayed at the bottom of the menu.

The system then displays the parameter prompt display for the command you selected.

If you are not sure of the parameters for a certain command, you can request prompting assistance from the system. The system then provides a prompt display that lists all the parameters and default values of the command (Figure 5-7). There is a prompt for every CL command.

For each parameter on the command, the following is shown, (see Figure 5-7).

- Text description Ⓐ
- Keyword name Ⓑ
- One or more input fields to enter the values for the parameter Ⓒ



```
        Display Object Description (DSPOBJD) Prompt
   Enter the following:
     Object--generic* or full name:   OBJ          R   ________
       Library name:                                    *USRLIBL
     Object type:                     OBJTYPE      R   ________
                                      + for more       ________
     Display detail:                  DETAIL       P   *BASIC
     Output (*  *LIST *NONE):         OUTPUT       P   *
     File to receive output:          OUTFILE      P   *NONE
       Library name:                                   ________
     Member to receive output:        OUTMBR       P   *FIRST
```

Figure 5-7. Typical Prompt Display

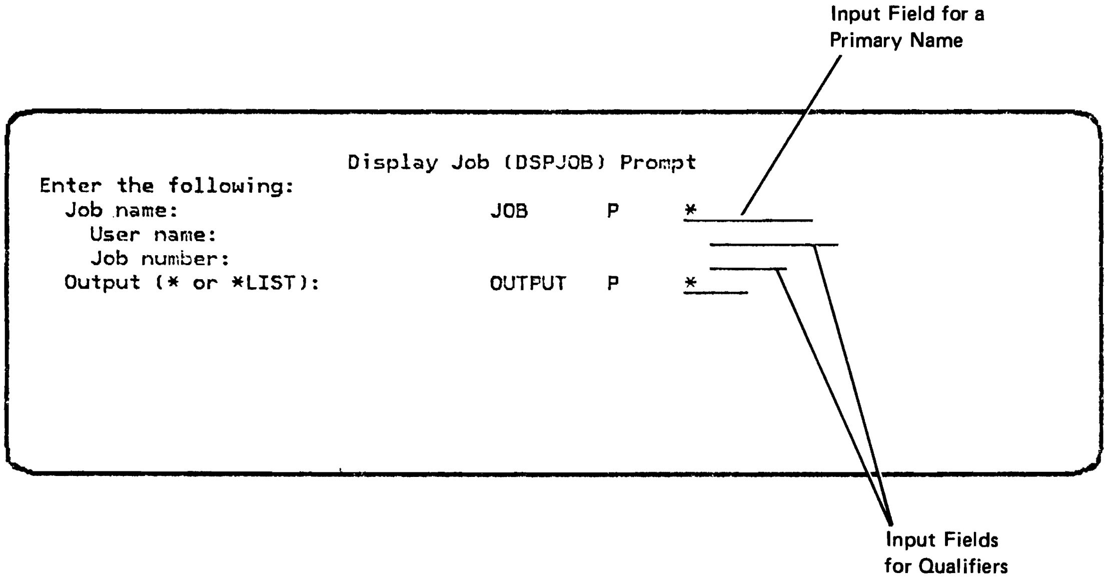
As discussed in the previous section, a prompt is displayed when you enter a command name on a command selection menu. However, you can also request a prompt directly from the command entry display by the following procedure:

1. Enter a partial command, just the command name, or the command name and one or more parameters.
2. Press the prompt (CF4) key.

When you fill in the prompt(s) to meet your needs and press the Enter/Rec Adv key, the command is executed as if you entered it directly on the command entry display. When you are using a prompt, follow these guidelines (also refer to Figure 5-7).

- There is a separate input field for each value of each parameter. Each input field is displayed on a separate line; an underline indicates its location. You can move to the field below by using the Field Exit key, or the ⇥ (Field Advance) key, or the ↵ (New Line) key; you can move to the field above by using the ⇤ (Field Backspace) key.

- Various commands allow you to enter qualified object names such as OBJ1.LIB1, or qualified job names such as JOB1.USER1.000123. On the prompt for a qualified object name, the input field of the library qualifier is indented below the input field for the object (see the OBJ keyword in Figure 5-7). Similarly, on a prompt for a command that allows a qualified job name, the input fields for the user name and job number are indented:



```
                        Display Job (DSPJOB) Prompt
Enter the following:
  Job name:                          JOB      P     *_______
    User name:                                         ________
    Job number:                                        ______
  Output (* or *LIST):               OUTPUT   P     *___
```

- If a default value, Ⓓ (Figure 5-7), is shown in an input field, you can accept it as is or change it by keying in another value over the one shown.

- A letter R next to an input field Ⓔ (Figure 5-7) means that the parameter is required and you must enter a value for that parameter.

- A letter P next to an input field Ⓕ (Figure 5-7), means that the parameter may be specified in positional form. A reverse image blank next to an input field Ⓕ means that the parameter was specified in positional form and should not have been. The reverse image blank will only be displayed the first time the parameter is displayed.

- If the phrase

  + for more

  appears next to an input field G (Figure 5-7) indicating that the parameter will accept more values, entering a + in that field results in a separate list prompt that allows you to enter additional values. (See the discussion of *Special Prompting Functions* earlier in this chapter.) Only some parameters accept multiple values.

- A symbol +++ in the upper right corner of a prompt display indicates there is another prompt display to follow containing additional parameters. This occurs where there are too many parameters to be shown on your screen at one time. Use the Enter/Rec Adv key to obtain the next display; use the CF2 key to return to the previous display if needed. Note, however, that pressing the Enter/Rec Adv key when the last display of the prompt is on your screen (as indicated by the absence of the +++ symbol) results in the command being accepted as complete.

- After you key in values on a prompt, you press the Enter/Rec Adv key to enter those values into the system. The values you keyed in are then validity checked and if they are valid, the next prompt is shown. You can continue this process until the last prompt display is shown (identified by the absence of +++ in the upper right corner of the display). This time when you press the Enter/Rec Adv key, all values, including those previously entered, will be validity checked for conflicts between parameters. If there are no conflicts, the command will be executed normally. Note that you can also press the CF16 key at any time to request that the command be entered. If you press the CF16 key before you have keyed in all values on every prompt, default values will be assumed for the other input fields.

- When one or more of your input fields containing the values in error are shown in reverse image (except on the system console), an error message is shown on the bottom line of your display. If more than one field fails validity checking, multiple messages are generated, but only one error message appears at the bottom of your display. A + at the end of the message (in the lower right corner of the screen) indicates that there are more messages to be seen. You can obtain the additional messages by positioning the cursor on the message line and pressing the Roll Up key. If you press the Help key while any of these messages are shown, you receive a second-level message display that gives an expanded description of the error message.

- If you choose to ignore an error detected by the validity checker and want to continue on to the next display, you can leave your current display by pressing the Enter/Rec Adv key, provided that you do not key in more information that causes other errors. The command cannot be executed, however, until you have corrected all errors on all displays. The error messages will continue to be shown on every display that has errors until you correct the errors and press the Enter/Rec Adv key again. Errors pertaining to parameter relationship conflicts are shown on each display after final validity checking. They remain on the displays until you modify any parameter values for the command.

- If the command is entered and passes validity checking and an error occurs while the command is being executed (such as the system not finding an object you specified), you return to the display from which you requested prompting (for example, the programmer menu or command entry display). The error messages are shown on that display, not the prompt display. If you were using the command entry display, the command you entered through prompting is shown as well as the error messages (see Example 3 in *Examples of Command Entry* later in this chapter).

Canceling Prompting

At any time before you have successfully completed the prompting process and pressed the Enter/Rec Adv key or the CF16 key, you can cancel prompting by pressing the CF1 key. You then return to the basic working display (for example, the programmer menu or the command entry display) and you can start over and enter another request.

Special Prompting Functions

When you are using a prompt for the parameters of a command, there are a variety of functions you can use to assist you in completing the prompt. Some of these functions have special applications for parameters that allow multiple values (a list). How they are used depends on the type of list.

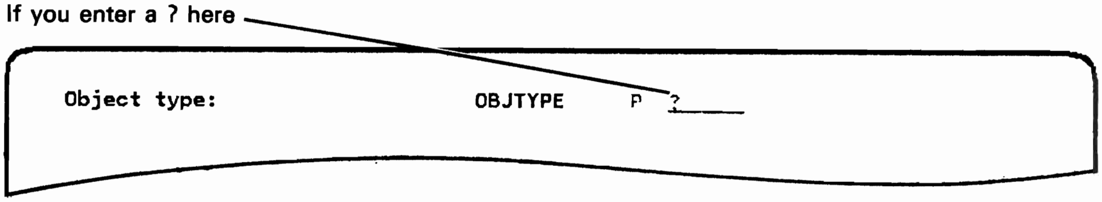
Permissible Values for an Input Field (?): If you want to know what values can be entered in an input field, you can obtain a display of the permissible values by the following procedure:

1. Key in a ? at the beginning of the input field and at least one blank if a value is present.

2. Press the Enter/Rec Adv key.

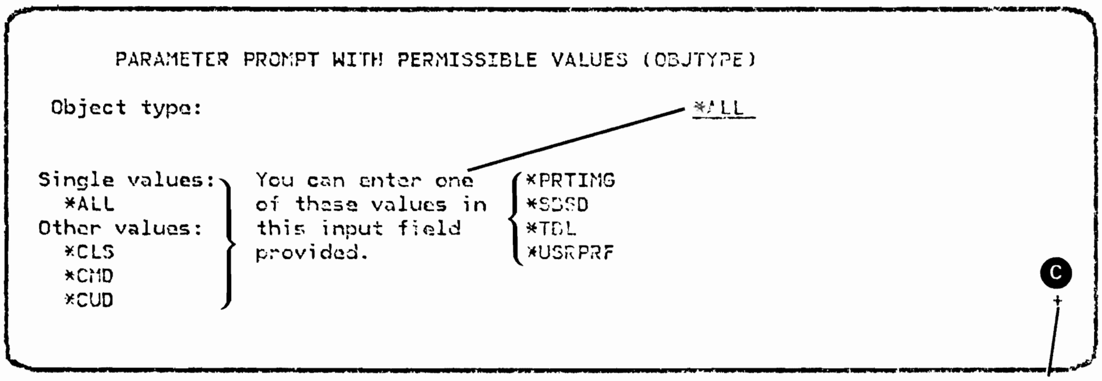
The ? function is valid anywhere a parameter value may be entered.

If you enter a ? here



```
Object type:                    OBJTYPE       P  ?____
```

you receive this display.



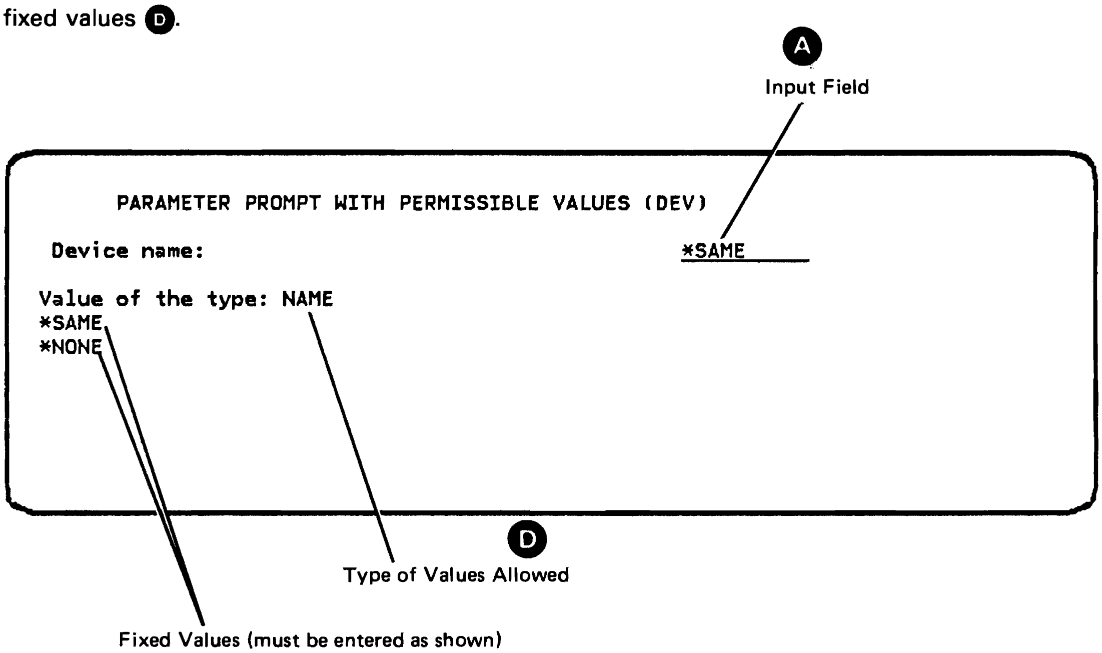
```
     PARAMETER PROMPT WITH PERMISSIBLE VALUES (OBJTYPE)
 Object type:                                         *ALL

Single values:    *PRTIMG
  *ALL            *SBSD
Other values:     *TBL
  *CLS            *USRPRF
  *CMD
  *CUD                                                          +
```

You can enter one of these values in this input field provided.

More Values Indicator

If there are a limited number of fixed values, but too many to fit within the display area allowed, a + appears in the lower right corner of the display area Ⓒ. When the cursor is positioned anywhere within that display area, you can use the roll keys to adjust the display to view all of the permissible values.

Some parameters will accept any value of a certain type (such as an object name or a job number), rather than (or in addition to) specific fixed values. When you request the permissible values for such a parameter, you receive an indication of the type of value allowed instead of (or in addition to) the list of fixed values Ⓓ.

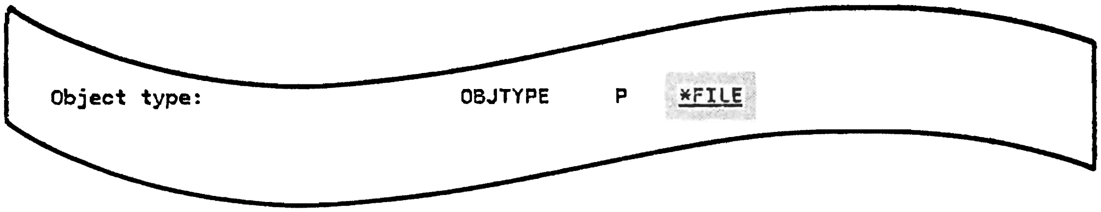


```
      PARAMETER PROMPT WITH PERMISSIBLE VALUES (DEV)

 Device name:                                        *SAME

Value of the type: NAME
*SAME
*NONE
```

If you enter a value in the input field Ⓐ provided on the permissible values display, that value is included in the original parameter prompt when you return to it. You return to the parameter prompt by pressing the Enter/Rec Adv key.

Default Value Request (blanks): The default values are shown for parameters allowing them when you first receive a prompt display. If you key in other values, those defaults are replaced. By entering blanks in place of a parameter value, you request that the value be replaced by the original default value for the parameter. Similarly, blanking all the fields for a mixed list (see *Mixed List* later in this chapter) makes the entire mixed list take on its original default value or values. You do not have to know what the original default values were.

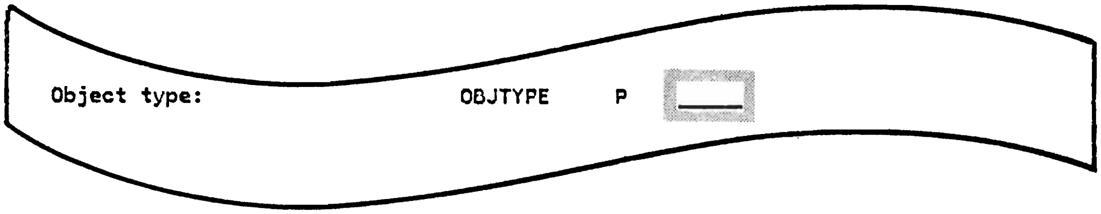
For example, to replace the value *FILE



```
Object type:                    OBJTYPE      P    *FILE
```

by its default value without knowing what the default value is, enter blanks in the input field; that is:

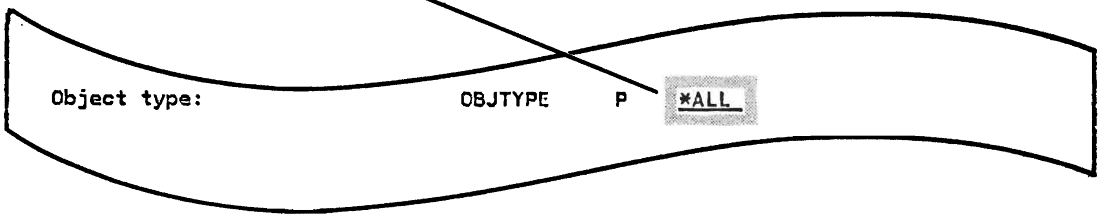
1. Position the cursor at the beginning of the input field and press the Field Exit key or the Field + key:



```
Object type:                    OBJTYPE      P    _____
```

2. Press the Enter/Rec Adv key.

The default value then appears.

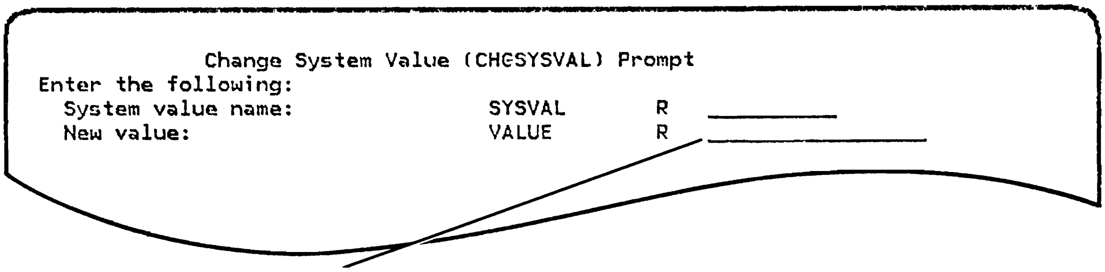


```
Object type:                    OBJTYPE      P    *ALL
```

Expanded Input Field (&): If a particular input field is not large enough for the parameter value or values you want to enter, you can expand the field by the following procedure:

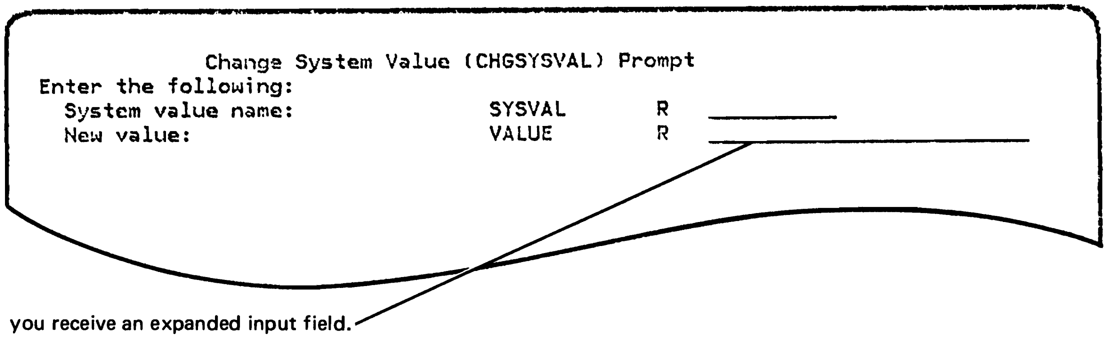
1. Key in an ampersand (&) at the beginning of the input field.

2. Press the Enter/Rec Adv key.



```
              Change System Value (CHGSYSVAL) Prompt
Enter the following:
  System value name:                SYSVAL      R  __________
  New value:                        VALUE       R  _________________
```

If you enter & here and press the Enter/Rec Adv key



```
              Change System Value (CHGSYSVAL) Prompt
Enter the following:
  System value name:                SYSVAL      R  __________
  New value:                        VALUE       R  _________________________
```

you receive an expanded input field.

The field will expand to the next larger length of either 11, 17, 25, 32, 50, 80, 132, 256, or 512 characters. That is, if the field was originally 4 characters long, it expands to a length that accepts 11 characters; if it was 50 characters long, it expands to a length that accepts 80 characters. For fields that are indented qualifiers, the field will expand to the next larger of 11, 25, or 80. Input fields are generally shown large enough to accommodate the largest value that can be entered in that field.

The & function would most likely be used to expand an input field to:

- Enter a character string enclosed in quotes (apostrophes) in an input field that allows a character string but is not large enough for you to enter the apostrophes along with the characters.

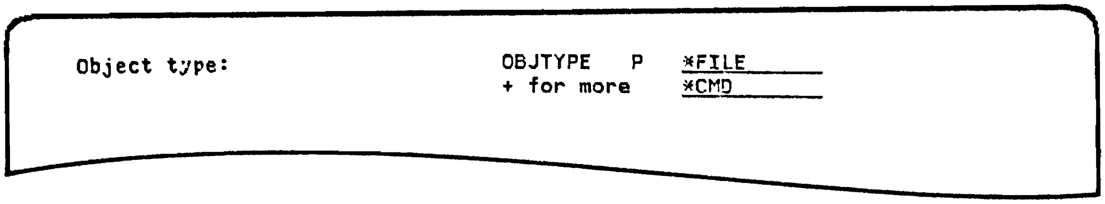
- Enter a value that contains embedded quotes or a value in hexadecimal form. The field is initially set to the character length of the value for normal entry.

- Enter a parameter value that can have a variable length and type, such as in the Change System Value command.

Simple List: The most common type of list used in commands is a simple list consisting of two or more values of the same type. In an OBJTYPE parameter, for example, several object types can be specified at the same time. The OBJTYPE parameter entered with a list of two object types, *FILE and *CMD (command), would look like this:

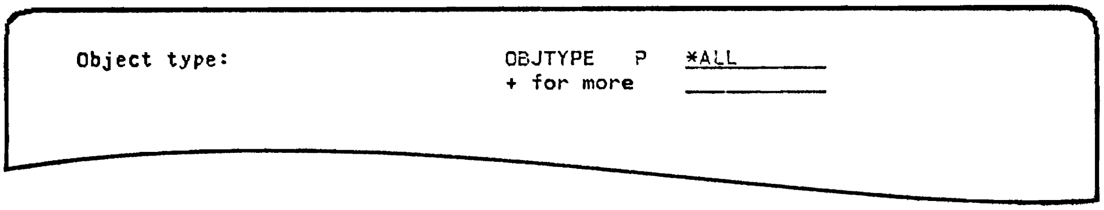
```
OBJTYPE(*FILE *CMD)
```

On a prompt, the same two values would be entered like this:



```
Object type:                       OBJTYPE    P    *FILE
                                   + for more      *CMD
```

For each parameter that can contain a simple list of values, at least two input fields are provided on separate lines of the prompt, and the phrase + *for more* appears beside the last line to indicate that more input fields can be obtained. If there is a single default value that can be used for the parameter instead of the list of values, the default appears in the first input field for the parameter. For example, the section of a prompt containing the OBJTYPE parameter, whose default is *ALL, would look like this:

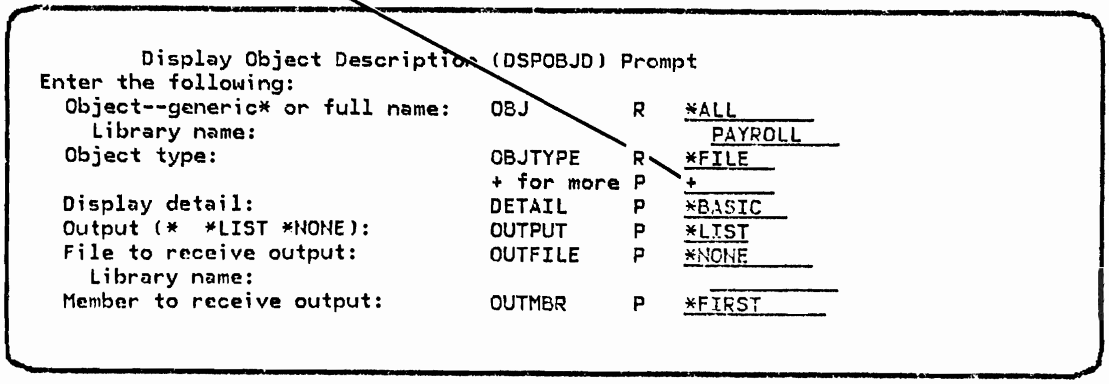


```
Object type:                       OBJTYPE    P    *ALL
                                   + for more      __________
```

Add After (+): When + *for more* is displayed beside an input field for a simple list, entering a + in the field results in a separate list prompt on which you can enter individual values.

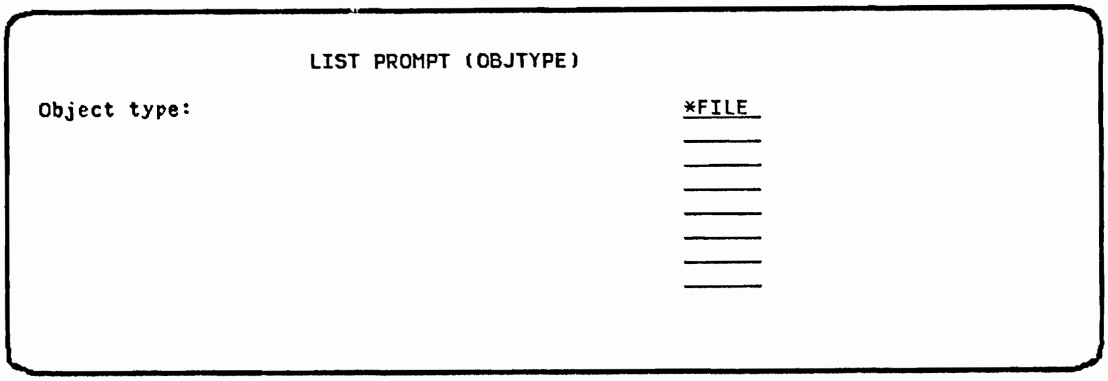
If you enter a + in the input field for one value in a simple list of values, you receive a list prompt formatted for entering additional values of the same type.

If you enter a + here



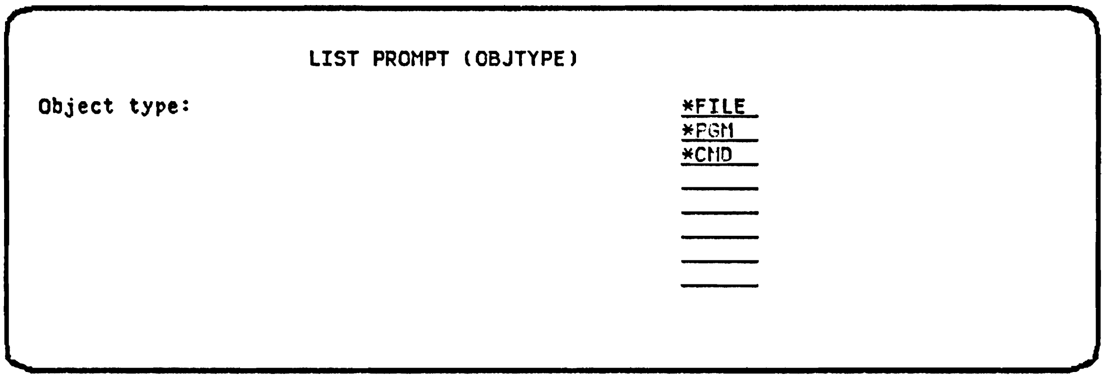
```
       Display Object Description (DSPOBJD) Prompt
Enter the following:
  Object--generic* or full name:   OBJ          R   *ALL______
    Library name:                                     PAYROLL___
  Object type:                     OBJTYPE      R   *FILE__
                                   + for more   P   +______
  Display detail:                  DETAIL       P   *BASIC_
  Output (*  *LIST *NONE):         OUTPUT       P   *LIST
  File to receive output:          OUTFILE      P   *NONE_____
    Library name:                                     __________
  Member to receive output:        OUTMBR       P   *FIRST____
```

you receive this display.



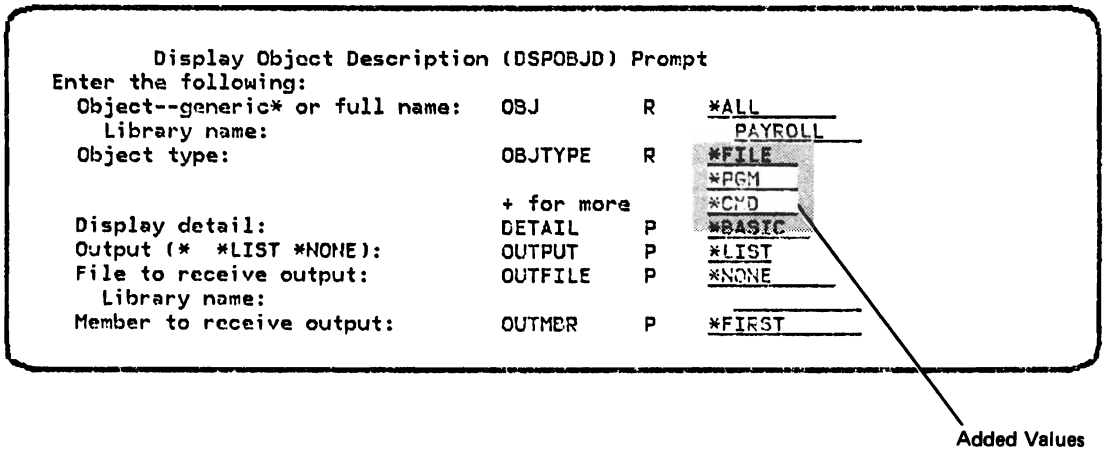
```
                         LIST PROMPT (OBJTYPE)

Object type:                                        *FILE_
                                                    ______
                                                    ______
                                                    ______
                                                    ______
                                                    ______
                                                    ______
                                                    ______
```

If you key in the values *PGM and *CMD



```
                      LIST PROMPT (OBJTYPE)

Object type:                                          *FILE
                                                      *PGM
                                                      *CMD
```

and press the Enter/Rec Adv key, you receive this display.



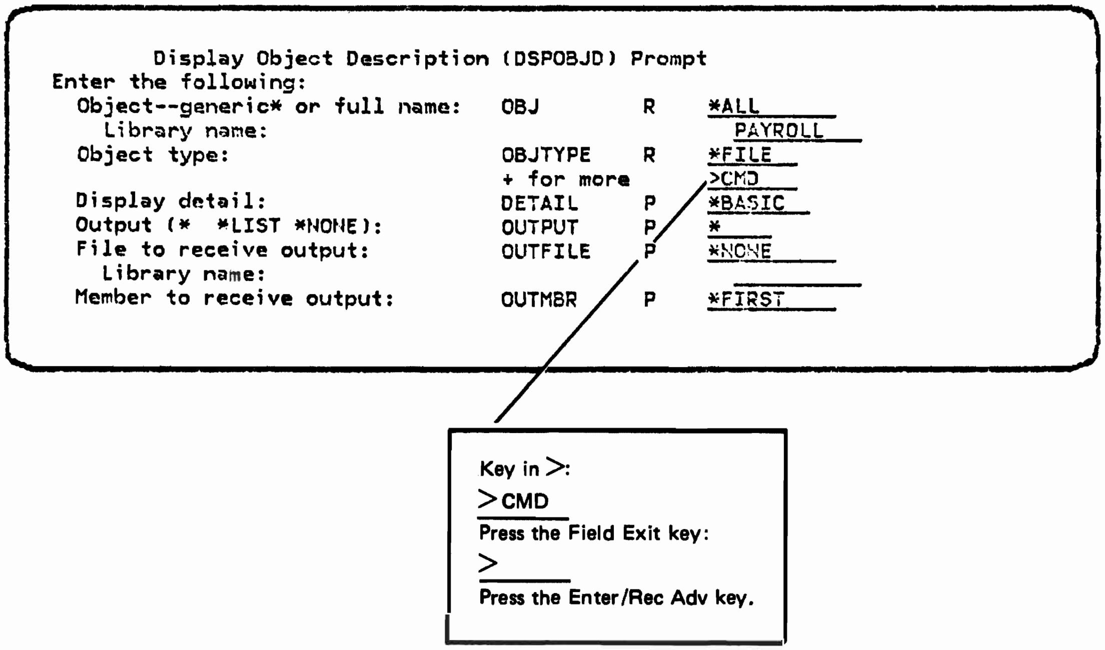
```
         Display Object Description (DSPOBJD) Prompt
 Enter the following:
   Object--generic* or full name:   OBJ        R    *ALL
     Library name:                                    PAYROLL
   Object type:                     OBJTYPE    R    *FILE
                                                    *PGM
                                    + for more      *CMD
   Display detail:                  DETAIL     P    *BASIC
   Output (*  *LIST *NONE):         OUTPUT     P    *LIST
   File to receive output:          OUTFILE    P    *NONE
     Library name:
   Member to receive output:        OUTMBR     P    *FIRST
```

If you want to add even more values in the example shown, you would enter a + (followed by at least one blank) over the * in *CMD and press the Enter/Rec Adv key to obtain a list prompt again. After you obtain and key any new values onto the list prompt and press the Enter/Rec Adv key, the new values are added after those already shown.

Insert (>): The > function allows you to insert additional values ahead of a value in a list. (In contrast, you use the + function to add values at the end of a list.)

Entering a > in the input field tells the system you want to enter additional values *ahead* of that value. If you are entering a > in an input field that already contains a value, key in the > over the first character of the value and enter at least one blank or press the Field Exit key to remove the remaining characters; then press the Enter/Rec Adv key. You receive the same kind of list prompt as you receive when you use the + function.

If you enter > in place of the * in *CMD



```
        Display Object Description (DSPOBJD) Prompt
Enter the following:
  Object--generic* or full name:   OBJ        R    *ALL
    Library name:                                   PAYROLL
  Object type:                     OBJTYPE    R    *FILE
                                   + for more      >CMD
  Display detail:                  DETAIL     P    *BASIC
  Output (*  *LIST *NONE):         OUTPUT     P    *
  File to receive output:          OUTFILE    P    *NONE
    Library name:
  Member to receive output:        OUTMBR     P    *FIRST
```

Key in >:

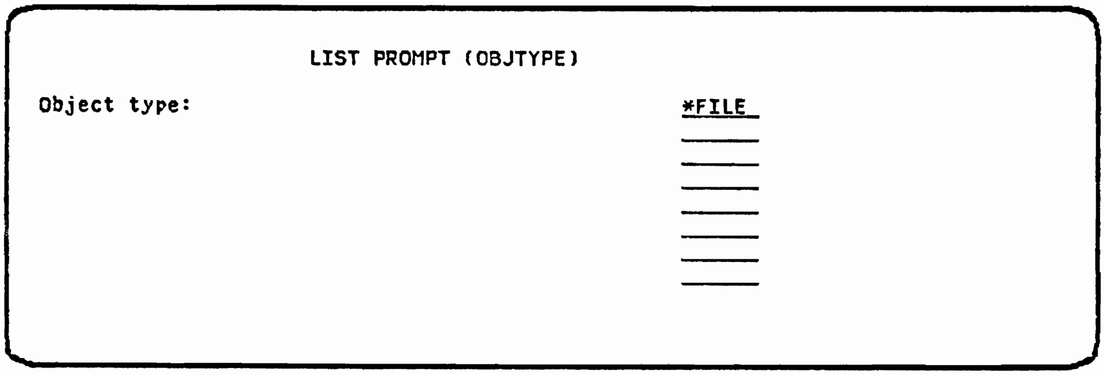
>CMD

Press the Field Exit key:

>

Press the Enter/Rec Adv key.

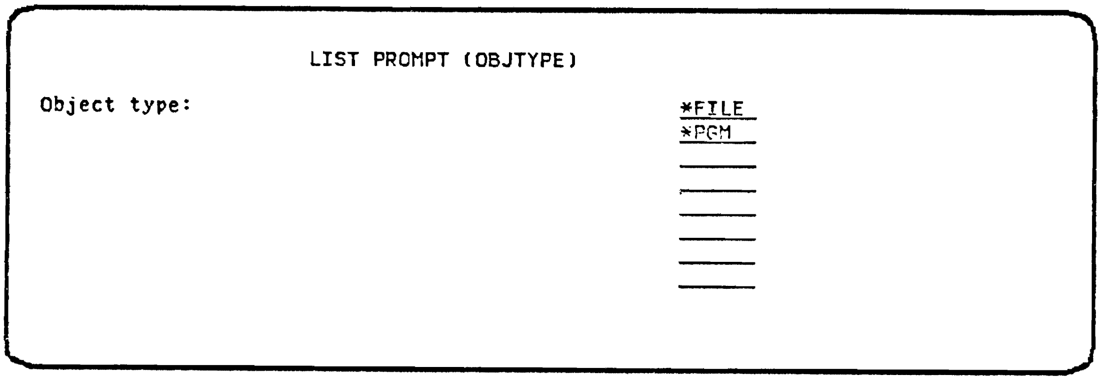
you receive this display.



```
                    LIST PROMPT (OBJTYPE)

Object type:                                    *FILE
```

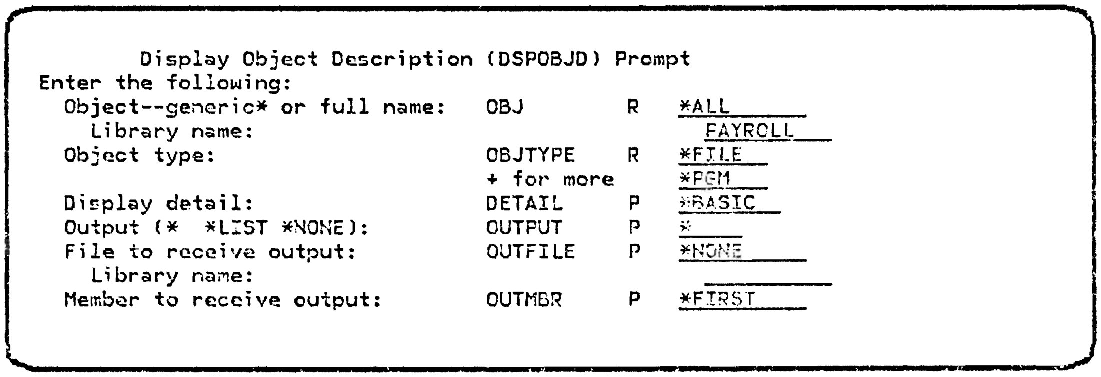
The value in the input field where you entered the > is not lost. When you key in the additional values on the list prompt and press the Enter/Rec Adv key, the replaced value reappears following the inserted values.

If you enter *PGM and press the Enter/Rec Adv key



```
                  LIST PROMPT (OBJTYPE)

Object type:                                          *FILE__
                                                      *PGM___
                                                      _______
                                                      _______
                                                      _______
                                                      _______
                                                      _______
                                                      _______
```

you receive this modified display.

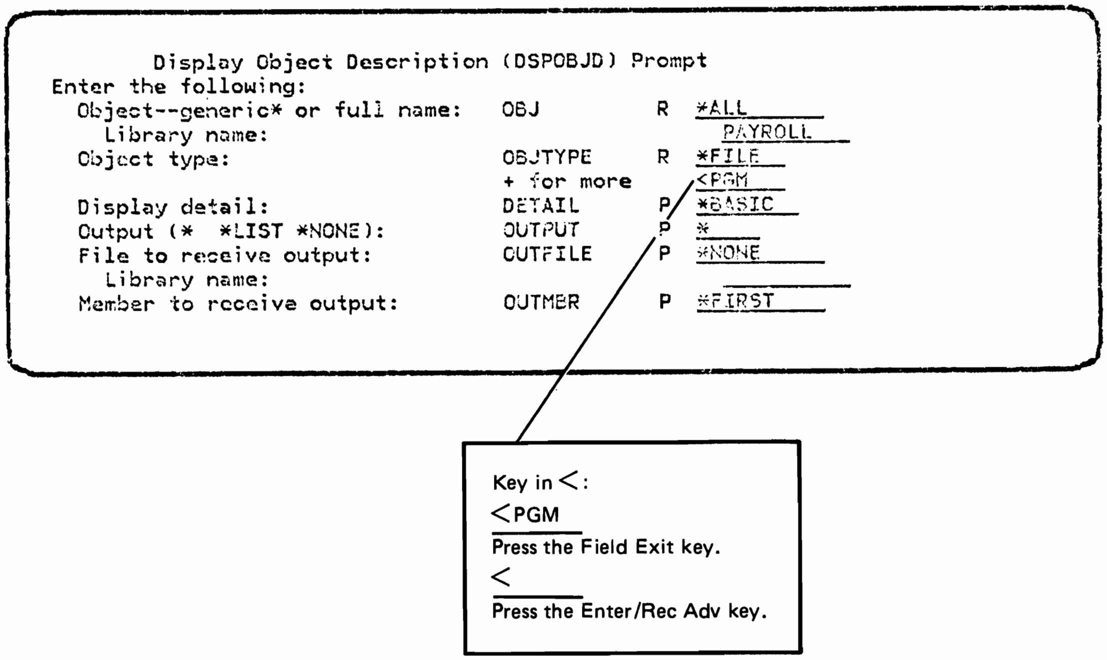


```
        Display Object Description (DSPOBJD) Prompt
Enter the following:
  Object--generic* or full name:   OBJ        R   *ALL______
    Library name:                                   FAYROLL___
  Object type:                     OBJTYPE    R   *FILE__
                                   + for more     *PGM___
  Display detail:                  DETAIL     P   *BASIC__
  Output (*  *LIST *NONE):         OUTPUT     P   *_____
  File to receive output:          OUTFILE    P   *NONE_____
    Library name:                                   __________
  Member to receive output:        OUTMBR     P   *FIRST____
```

Note that if you entered a + instead of a > on the line containing the second value, you would receive a list prompt formatted for you to enter new values after the second value instead of before the second value.

Delete (<): Using the < function, you can delete a value from a list. If you enter a < (followed by at least one blank) in place of the first character of a value, that value is deleted.

If you enter < in place of the * in *PGM



```
        Display Object Description (DSPOBJD) Prompt
Enter the following:
  Object--generic* or full name:    OBJ          R  *ALL
    Library name:                                     PAYROLL
  Object type:                      OBJTYPE      R  *FILE
                                    + for more      <PGM
  Display detail:                   DETAIL       P  *BASIC
  Output (*  *LIST *NONE):          OUTPUT       P  *
  File to receive output:           OUTFILE      P  *NONE
    Library name:
  Member to receive output:         OUTMBR       P  *FIRST
```

Key in <:

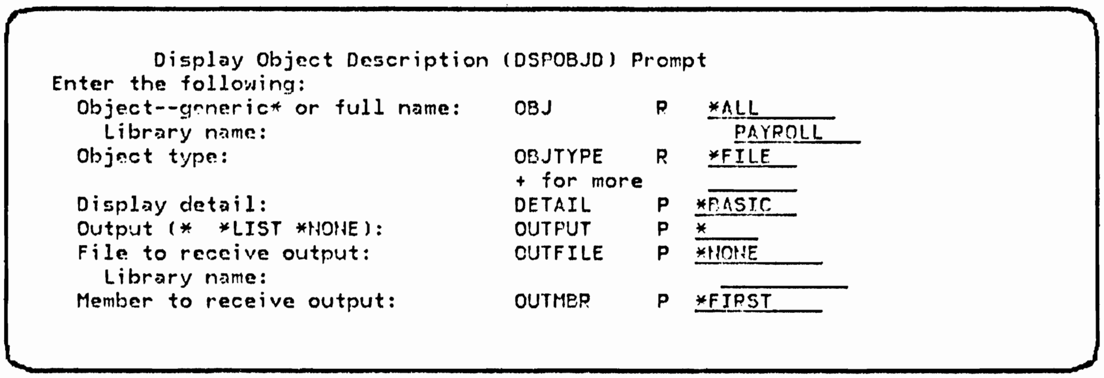
<PGM

Press the Field Exit key.

<

Press the Enter/Rec Adv key.

the value is deleted.

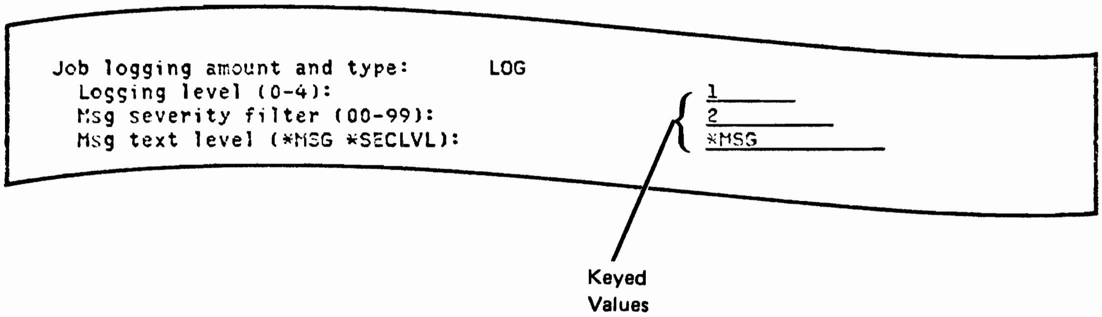


```
        Display Object Description (DSPOBJD) Prompt
Enter the following:
  Object--generic* or full name:    OBJ          R  *ALL
    Library name:                                     PAYROLL
  Object type:                      OBJTYPE      R  *FILE
                                    + for more
  Display detail:                   DETAIL       P  *BASIC
  Output (*  *LIST *NONE):          OUTPUT       P  *
  File to receive output:           OUTFILE      P  *NONE
    Library name:
  Member to receive output:         OUTMBR       P  *FIRST
```

Mixed List: A few commands contain parameters that can have multiple values of different types (a mixed list). For example, the LOG parameter of the Change Job command accepts a mixed list consisting of a number indicating a message-level like 1, a number specifying a message-severity like 2, and a character value like *MSG indicating what should be logged for each message. When entered in the command, the parameter would look like this:

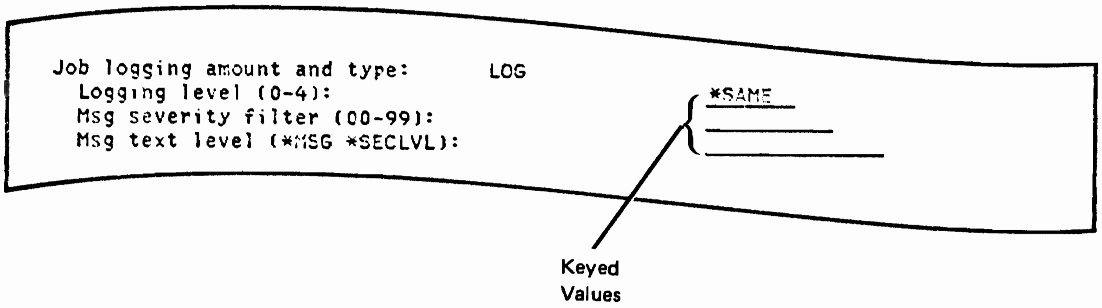
```
LOG(1 2 *MSG)
```

On a prompt, the same values for the LOG parameter would be entered like this:



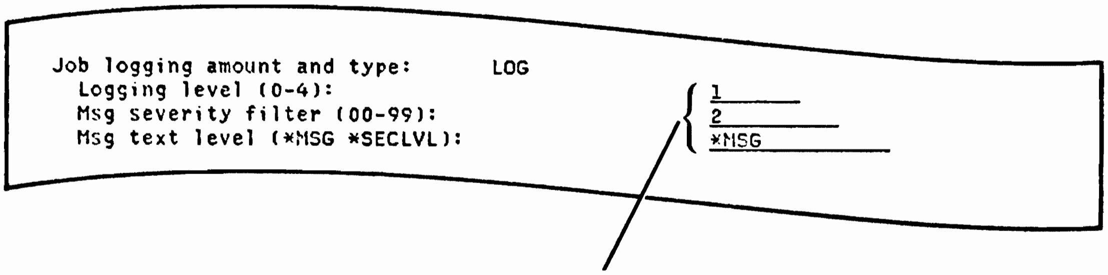
```
Job logging amount and type:       LOG
  Logging level (0-4):                             1
  Msg severity filter (00-99):                     2
  Msg text level (*MSG *SECLVL):                   *MSG
```

Some parameters that accept a mixed list can have a single default value for the entire list. Such a value appears in the first input field for the separate values.



```
Job logging amount and type:       LOG
  Logging level (0-4):                             *SAME
  Msg severity filter (00-99):
  Msg text level (*MSG *SECLVL):
```

If you enter one or more values in the input fields for the mixed list, those values replace the single value.



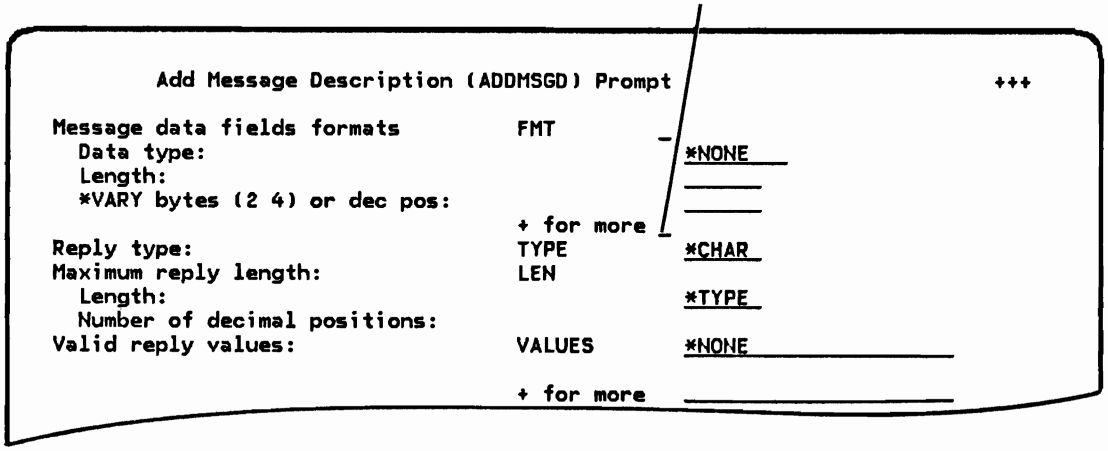
```
Job logging amount and type:      LOG
  Logging level (0-4):                                1
  Msg severity filter (00-99):                        2
  Msg text level (*MSG *SECLVL):                      *MSG
```

Keyed
Values

If the single value is entered in the first input field and other values are also entered for the list, the other values will be ignored.

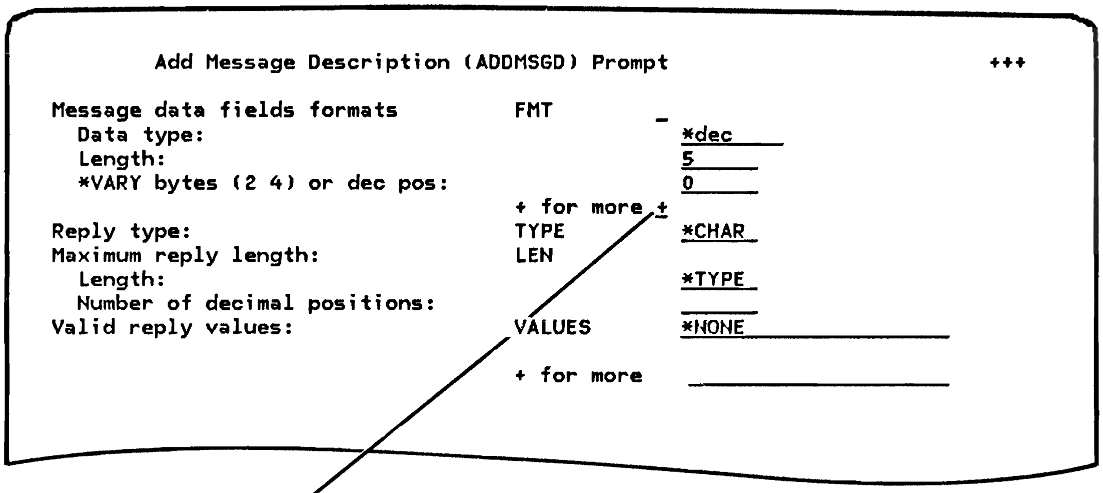
Simple List of Mixed Lists: If a parameter can have a simple list of mixed lists, an operator field is provided preceding each mixed list where delete and insert operators can be entered. At the end of the set of prompts for the parameter, a final operator field is provided where a + can be entered to obtain more mixed lists. The phrase + *for more* appears next to this final operator field.

Final Operator Field



```
        Add Message Description (ADDMSGD) Prompt                         +++

Message data fields formats        FMT          _
  Data type:                                     *NONE
  Length:                                        _____
  *VARY bytes (2 4) or dec pos:                  _____
                                   + for more  _
Reply type:                        TYPE          *CHAR
Maximum reply length:              LEN
  Length:                                        *TYPE
  Number of decimal positions:
Valid reply values:                VALUES        *NONE

                                   + for more    ____________________
```

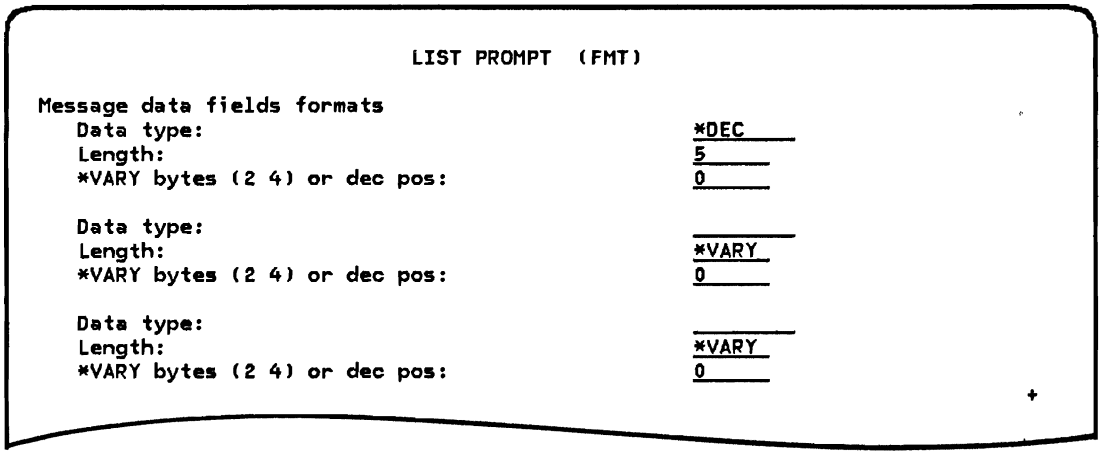
Add After (+): In the same manner as for a simple list, if you enter a + in the operator field at the end of a mixed list, you receive a list prompt formatted for you to enter additional sets of the mixed list.



```
          Add Message Description (ADDMSGD) Prompt                             +++

 Message data fields formats           FMT          _
   Data type:                                         *dec____
   Length:                                            5_____
   *VARY bytes (2 4) or dec pos:                      0_____
                                       + for more  +
 Reply type:                           TYPE           *CHAR_
 Maximum reply length:                 LEN
   Length:                                            *TYPE_
   Number of decimal positions:                       ______
 Valid reply values:                   VALUES         *NONE________________

                                       + for more     _____________________
```

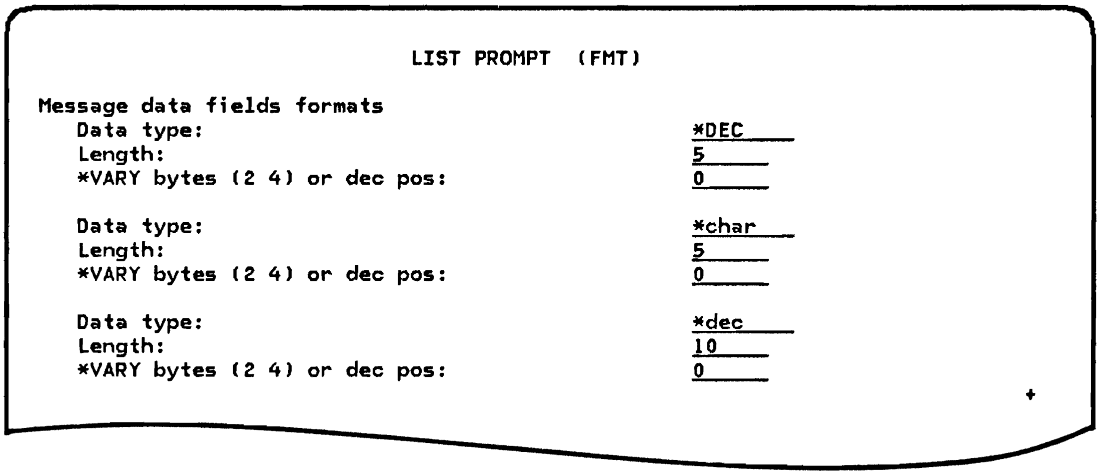
If you enter a + here and press the Enter/Rec Adv key

you receive this display.



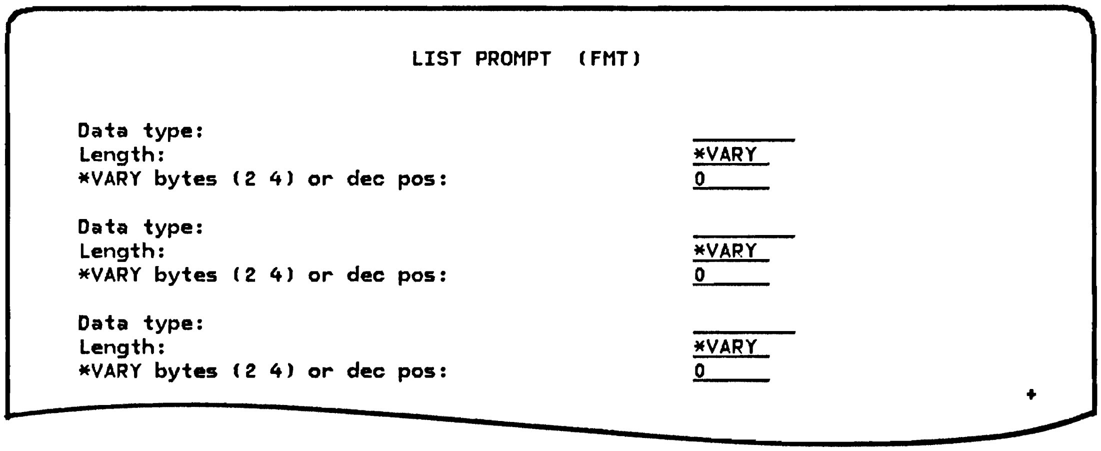
```
                               LIST PROMPT  (FMT)

Message data fields formats
   Data type:                                          *DEC___
   Length:                                             5_____
   *VARY bytes (2 4) or dec pos:                       0_____

   Data type:                                          _______
   Length:                                             *VARY_
   *VARY bytes (2 4) or dec pos:                       0_____

   Data type:                                          _______
   Length:                                             *VARY_
   *VARY bytes (2 4) or dec pos:                       0_____
                                                                          +
```

After you enter data in the displayed input fields, your display looks like this.



```
                          LIST PROMPT  (FMT)

Message data fields formats
   Data type:                                          *DEC
   Length:                                             5
   *VARY bytes (2 4) or dec pos:                       0

   Data type:                                          *char
   Length:                                             5
   *VARY bytes (2 4) or dec pos:                       0

   Data type:                                          *dec
   Length:                                             10
   *VARY bytes (2 4) or dec pos:                       0
                                                                    +
```

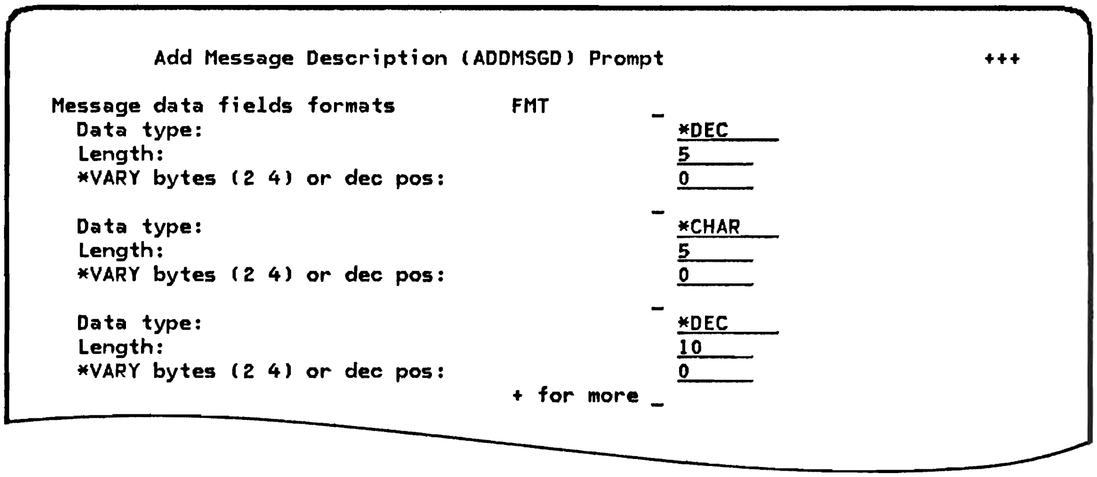
If you have additional values to enter and you press the Roll Up key, you receive this display.



```
                          LIST PROMPT  (FMT)


   Data type:                                          ______
   Length:                                             *VARY
   *VARY bytes (2 4) or dec pos:                       0

   Data type:                                          ______
   Length:                                             *VARY
   *VARY bytes (2 4) or dec pos:                       0

   Data type:                                          ______
   Length:                                             *VARY
   *VARY bytes (2 4) or dec pos:                       0
                                                                    +
```

When you press the Enter/Rec Adv key after keying the additional values into the list prompt, you receive the original prompt again, but in a modified form that includes the additional values.

> For example, if you key in the three sets of values (*DEC, 5, 0, and *CHAR, 5, 0, and *DEC, 10, 0) in the previous display and press the Enter/Rec Adv key, you receive this parameter prompt display. This display shows the original set of values and the added values.



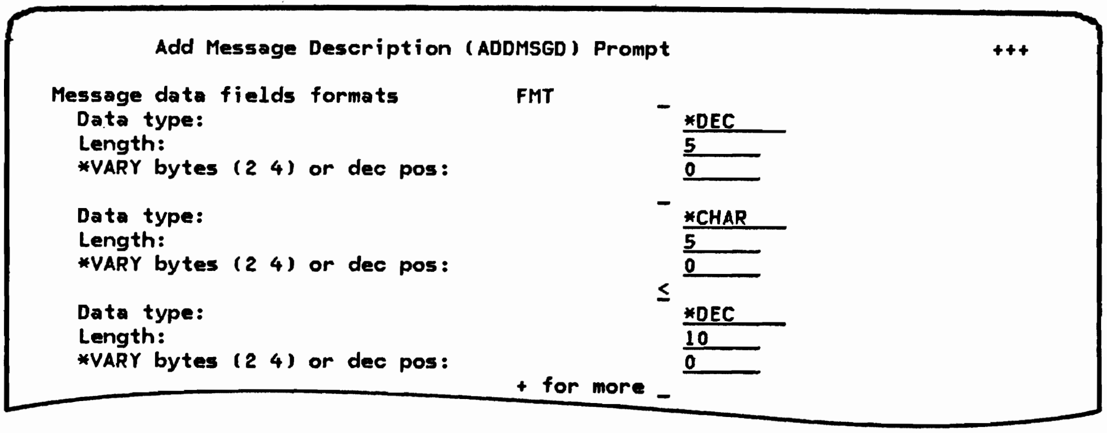
```
          Add Message Description (ADDMSGD) Prompt                          +++

  Message data fields formats         FMT          _
    Data type:                                      *DEC
    Length:                                         5
    *VARY bytes (2 4) or dec pos:                   0
                                                   _
    Data type:                                      *CHAR
    Length:                                         5
    *VARY bytes (2 4) or dec pos:                   0
                                                   _
    Data type:                                      *DEC
    Length:                                         10
    *VARY bytes (2 4) or dec pos:                   0
                                      + for more _
```

If you want to add even more mixed list values in the example shown, you would enter a + in the final operator field. After you obtain and complete the list prompt, any new mixed lists are added after those already shown.

Insert (>): If you enter a > in the operator field preceding a mixed list, you are telling the system that you want to insert a new set of values ahead of that mixed list. The list prompt would be displayed as in the Add After situation and, after keying the new values, the parameter prompt display is reshown with the inserted values.

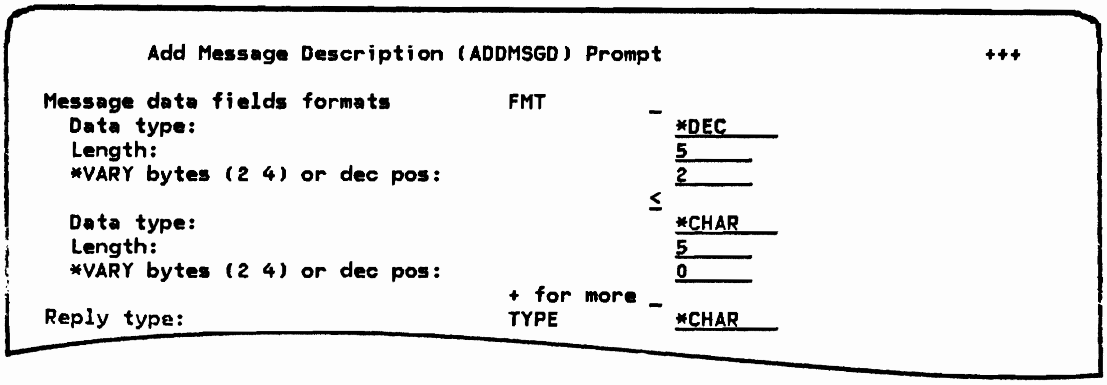
Delete (<): If you want to delete an entire occurrence of a mixed list, you enter a < in the operator field preceding the set of values to be deleted.

If you enter < in the operator field preceding the *DEC in this display



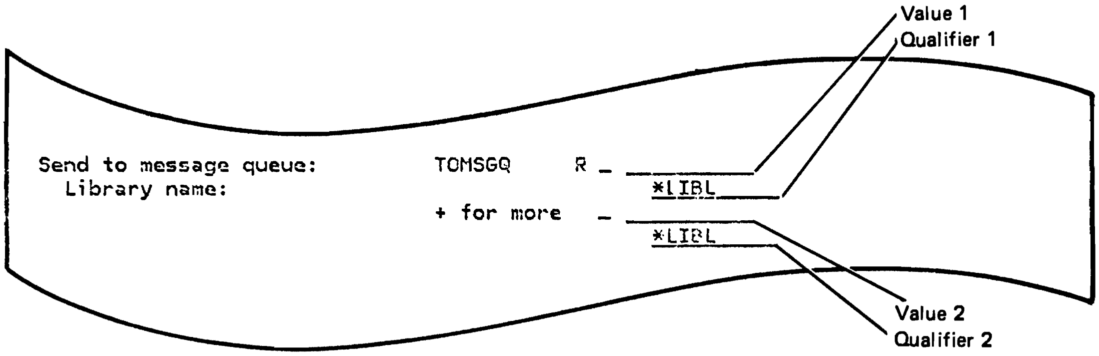
```
        Add Message Description (ADDMSGD) Prompt                    +++

Message data fields formats         FMT         _
  Data type:                                     *DEC
  Length:                                        5
  *VARY bytes (2 4) or dec pos:                  0
                                                _
  Data type:                                     *CHAR
  Length:                                        5
  *VARY bytes (2 4) or dec pos:                  0
                                                <
  Data type:                                     *DEC
  Length:                                        10
  *VARY bytes (2 4) or dec pos:                  0
                                    + for more _
```

all the values in the third mixed list are deleted. Only the first two mixed lists remain.



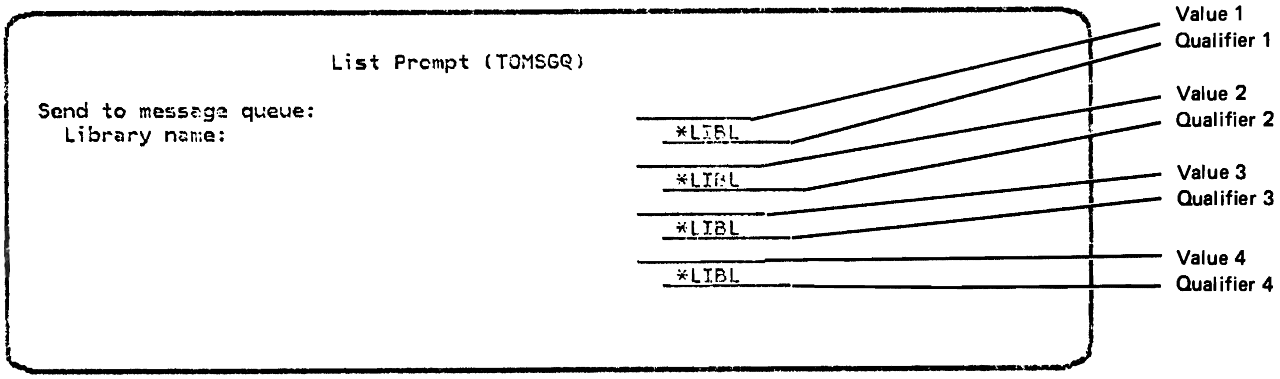
```
        Add Message Description (ADDMSGD) Prompt                    +++

Message data fields formats         FMT         _
  Data type:                                     *DEC
  Length:                                        5
  *VARY bytes (2 4) or dec pos:                  2
                                                <
  Data type:                                     *CHAR
  Length:                                        5
  *VARY bytes (2 4) or dec pos:                  0
                                    + for more _
Reply type:                         TYPE        _ *CHAR
```

Simple List of Qualified Names: Some commands, such as the Send Message command, allow you to enter multiple qualified values for a parameter. On the prompt for a command of this type, the input field for the single value shows where the first value can be entered. Indented below the first operator field is a set of input fields for the first of the multiple qualified values. The first field of the set is the input field where you enter the value itself. Indented below that is the input field where you enter the qualifier for that value.



```
Send to message queue:          TOMSGQ      R _ __________
  Library name:                                *LIBL______
                                + for more  _ __________
                                               *LIBL______
```

To enter additional values and their qualifiers, enter a + (followed by at least one blank) in any of the value fields. You then obtain a list prompt that displays sets of input fields for additional values and their qualifiers.



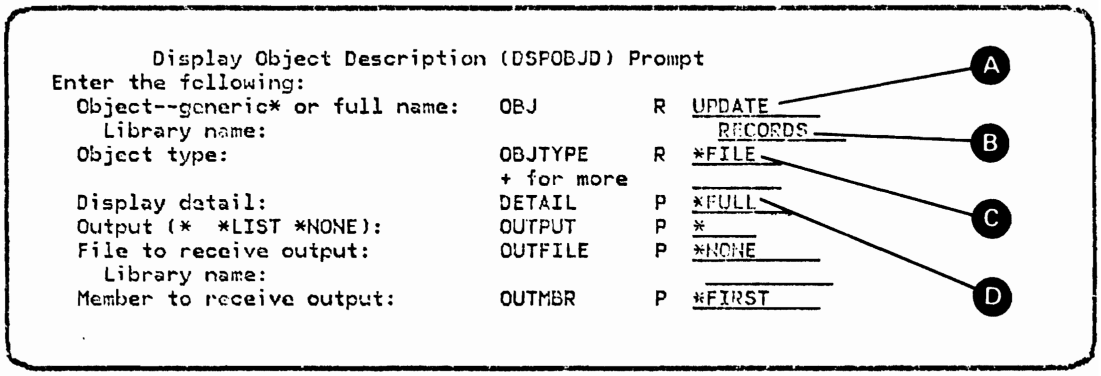
```
                         List Prompt (TOMSGQ)

Send to message queue:                          __________
  Library name:                                   *LIBL_____
                                                __________
                                                  *LIBL_____
                                                __________
                                                  *LIBL_____
                                                __________
                                                  *LIBL_____
```

Prompting for Character String Values

In CL commands, blanks, parentheses, and other special characters are used to separate multiple values. Therefore, if character strings containing any of those special characters are to be entered, the character string must be enclosed in quotes. The prompter only allows one value to be entered in a prompter input field. Therefore, to assist you in entering the correct values, if any special characters are detected by the prompter in character, date, or time input fields, the value is automatically quoted.

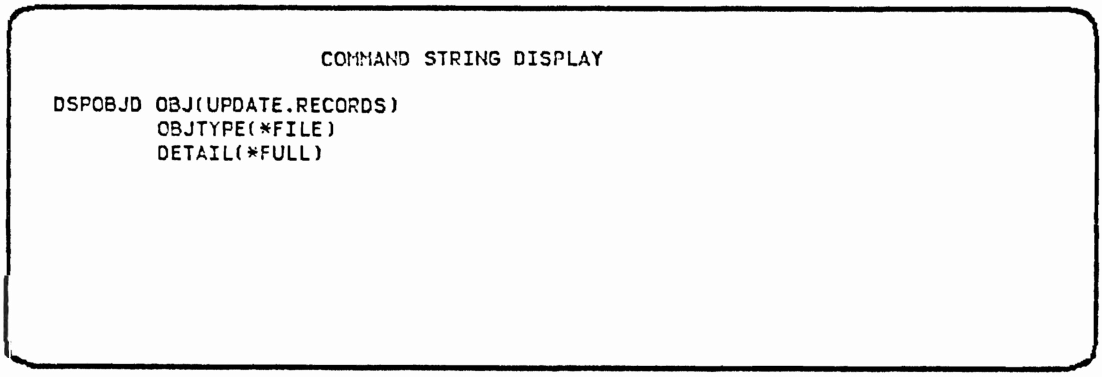
Display Command String (CF14): If you are entering values in a prompt display and want to see what the command would look like in string form as you would enter it on the command entry display, you can press the CF14 key to obtain a command string display. This display shows all parameters of the command built up from the values you have specified so far. It does not include any default values you have not changed. Note that each parameter following the first parameter is shown on a separate line.

If you enter values A, B, C, and D, then press the CF14 key



```
       Display Object Description (DSPOBJD) Prompt
Enter the following:
  Object--generic* or full name:    OBJ         R  UPDATE
    Library name:                                    RECORDS
  Object type:                      OBJTYPE     R  *FILE
                                    + for more
  Display detail:                   DETAIL      P  *FULL
  Output (*  *LIST *NONE):          OUTPUT      P  *
  File to receive output:           OUTFILE     P  *NONE
    Library name:
  Member to receive output:         OUTMBR      P  *FIRST
```

you receive this display.

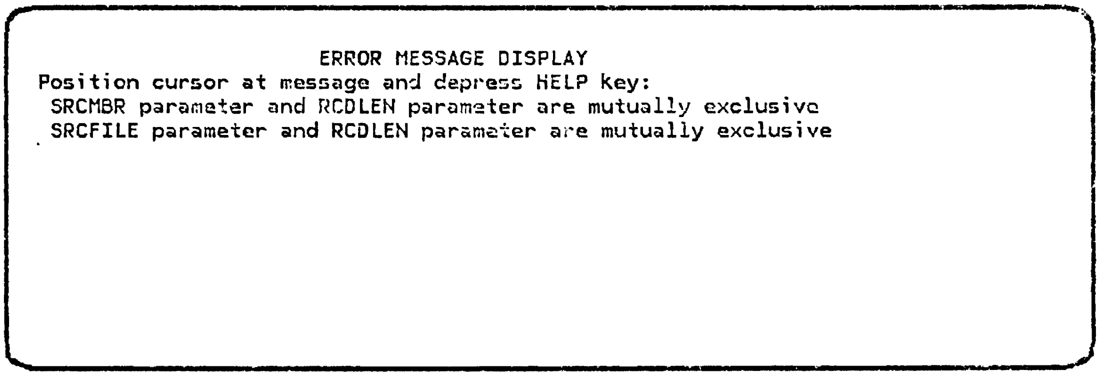


```
                        COMMAND STRING DISPLAY

DSPOBJD OBJ(UPDATE.RECORDS)
        OBJTYPE(*FILE)
        DETAIL(*FULL)
```

A + in the lower right corner of the screen indicates that there are more command lines to be seen. You can use the roll keys to review the entire command string.

Press the Enter/Rec Adv key to return to the prompt display.

Display Error Messages (CF15): By pressing the CF15 key, you can obtain an error message review display that shows all first-level error messages.



```
                          ERROR MESSAGE DISPLAY
Position cursor at message and depress HELP key:
 SRCMBR parameter and RCDLEN parameter are mutually exclusive
 SRCFILE parameter and RCDLEN parameter are mutually exclusive
```

You can also adjust the displayed lines to view additional messages by using the roll keys, or you can position the cursor at a particular message and press the Help key to obtain a second-level message display, as discussed in Chapter 3. To return to the display from which you requested the error message display, press the Enter/Rec Adv key.

Command Complete/Enter Command (CF16): When you complete the final display of a multiple-display parameter prompt and press the Enter/Rec Adv key, the command is executed if there are no errors on the command. If you press the Enter/Rec Adv key when you are not on the final display, however, you merely receive the next display of the prompt without the command being executed.

If you want to signal the system to process a command when you are not on the final display of a prompt, you press the CF16 key. For example:

- You could key in values only for the required parameters on the first display and press the CF16 key to execute the command immediately, using the default values for the parameters on the remaining displays.

- After keying in all of your entries, you could back up through the previous parameter prompt displays (by using the CF2 key) to check your entries and then press the CF16 key to execute the command.

Revert to Command Defaults (CF18): There may be occasions when you have entered values for a command that resulted in error messages you do not know how to resolve. By pressing the CF18 key, you can return to the beginning of the command with all the defaults and start again. All values you entered are removed, including those you may have entered as a partial command on the command entry display before requesting the prompt.

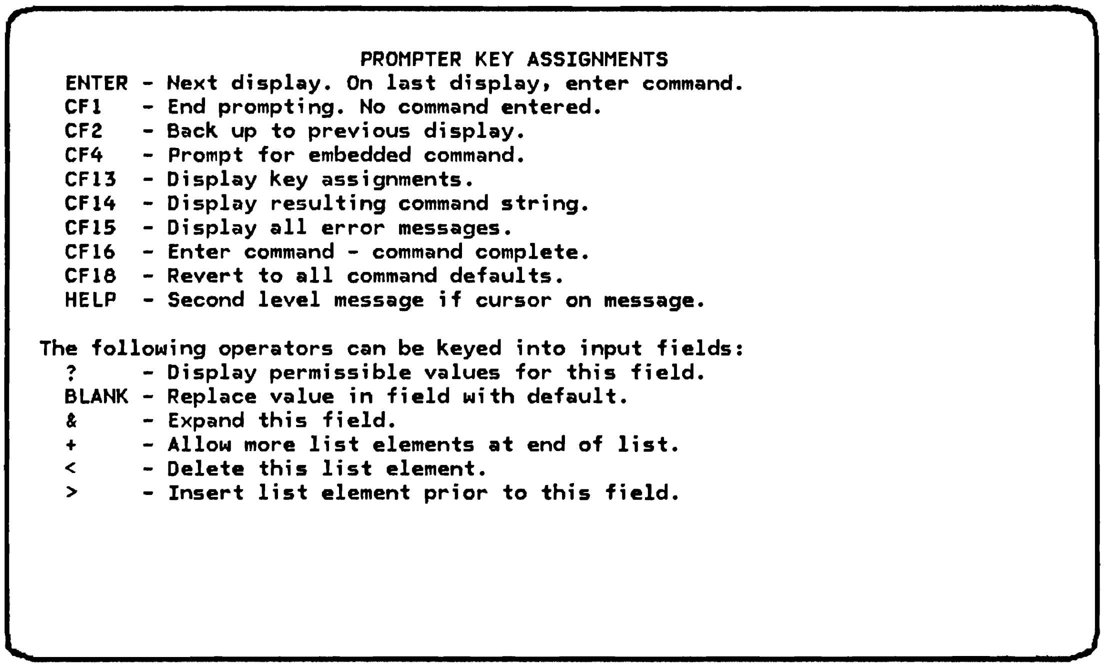
Return/End Prompting (CF1): At any point in the prompting process until you press the Enter/Rec Adv key or the CF16 key to have the command executed, you can stop command entry by pressing the CF1 key. When you press CF1, you return to the display from which you requested prompting. The command that is the result of prompting up to that time is returned so it can be changed and used in some other way.

Embedded Command Prompting (CF4): If you are using the prompt for an IF, ELSE, or MONMSG command that contains a parameter in which you can enter another complete command (an embedded command), you can request a prompt display for that embedded command by pressing the prompt key (CF4). The prompt for the embedded command appears the same as a regular prompt except that the heading at the top of the prompt contains a number that indicates the level to which the command is embedded. If the CF4 key is pressed with a partially entered embedded command, and no invalid values were entered for other parameters, prompting will be provided, if possible. Otherwise the embedded command diagnostics that prevent prompting will be immediately shown, as they are when ENTER is pressed.

The Enter/Rec Adv, command-complete/enter command (CF16), and end-prompting (CF1) keys have the same effect on an embedded command prompt as a regular parameter prompt, except that when the embedded command is completed, you return to the previous upper level prompt display rather than the display from which you requested prompting.

Error messages are displayed only for the current level command for which you have requested prompting. If your current level command contains an embedded command with errors, however, those errors may be identified by a general error message for that parameter at your current level.

Display CF Keys (CF13, Help): By pressing the CF13 key when prompting, you can obtain a display that describes the prompting function each assigned CF key performs and describes the valid operators. You also receive this CF key display if you press the Help key when the cursor is not positioned on the same line as an error message being displayed.. (If you press the Help key when the cursor is on the line containing an error message, you receive the second-level text for the message, as described in Chapter 3.)



```
                             PROMPTER KEY ASSIGNMENTS
  ENTER - Next display. On last display, enter command.
  CF1   - End prompting. No command entered.
  CF2   - Back up to previous display.
  CF4   - Prompt for embedded command.
  CF13  - Display key assignments.
  CF14  - Display resulting command string.
  CF15  - Display all error messages.
  CF16  - Enter command - command complete.
  CF18  - Revert to all command defaults.
  HELP  - Second level message if cursor on message.

The following operators can be keyed into input fields:
  ?     - Display permissible values for this field.
  BLANK - Replace value in field with default.
  &     - Expand this field.
  +     - Allow more list elements at end of list.
  <     - Delete this list element.
  >     - Insert list element prior to this field.
```

Note: On a 12- and 16-line screen, only the first 12 lines are shown.

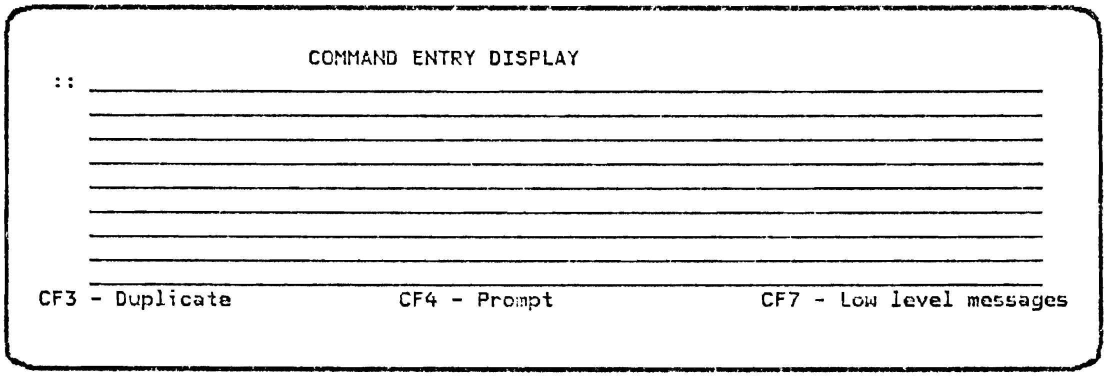
Pressing the Enter/Rec Adv key returns you to the display from which you requested the CF key review display.

Examples of Command Entry

The following three examples illustrate how you enter commands, both directly on the command entry display and with the assistance of menus and prompts.

Example 1 — Direct Command Entry

You have signed on a work station with a 12-line screen and you receive the command entry display (see *Displays and Input Fields* in Chapter 1).



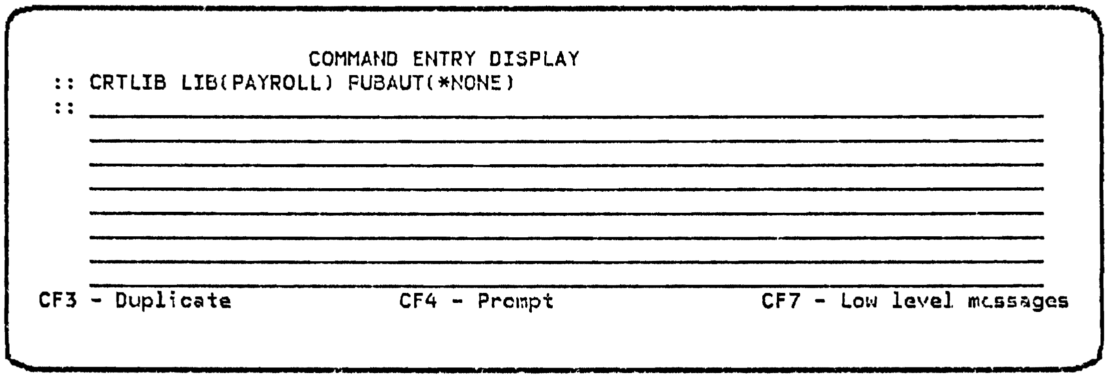
```
                         COMMAND ENTRY DISPLAY
 :: ______________________________________________________________
    ______________________________________________________________
    ______________________________________________________________
    ______________________________________________________________
    ______________________________________________________________
    ______________________________________________________________
    ______________________________________________________________
    ______________________________________________________________
    ______________________________________________________________
CF3 - Duplicate            CF4 - Prompt            CF7 - Low level messages
```

You want to create a library named PAYROLL and restrict it to only your use. You use the Create Library command:

1. Key in CRTLIB LIB(PAYROLL) PUBAUT(*NONE).

2. Press the Enter/Rec Adv key.

The system uses default values for the parameters you did not enter and positions the entry symbol :: for your next entry.



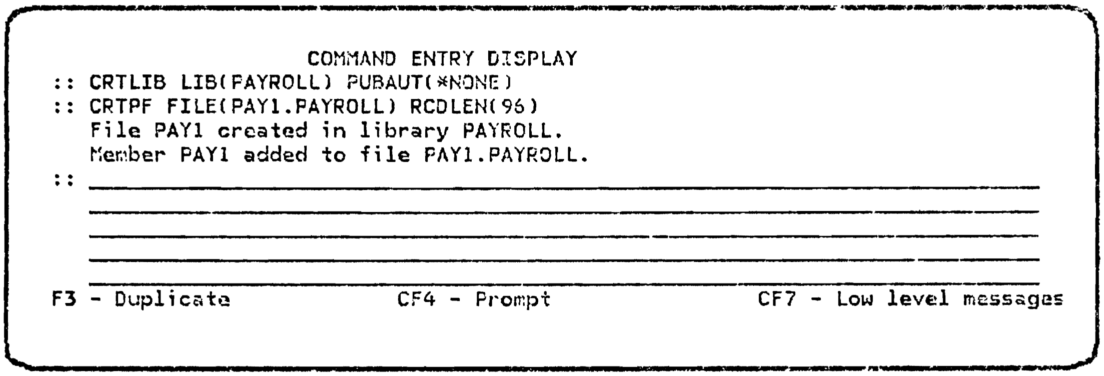
```
                         COMMAND ENTRY DISPLAY
 :: CRTLIB LIB(PAYROLL) PUBAUT(*NONE)
 :: ______________________________________________________________
    ______________________________________________________________
    ______________________________________________________________
    ______________________________________________________________
    ______________________________________________________________
    ______________________________________________________________
    ______________________________________________________________
    ______________________________________________________________
CF3 - Duplicate            CF4 - Prompt            CF7 - Low level messages
```

Now you want to establish a physical file in the library you just created. The file is to be named PAY1 and have a maximum record length that is the same as a 96-column card (96 bytes). You use the Create Physical File command:

1. Key in CRTPF FILE(PAY1.PAYROLL) RCDLEN(96).

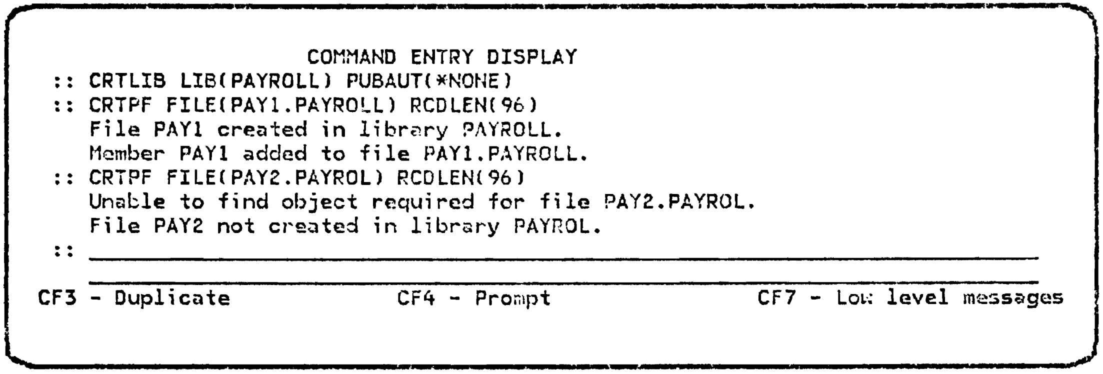
2. Press the Enter/Rec Adv key.

The system again uses default values for the parameters you did not enter and responds with a message indicating the file was created. The display now looks like this:



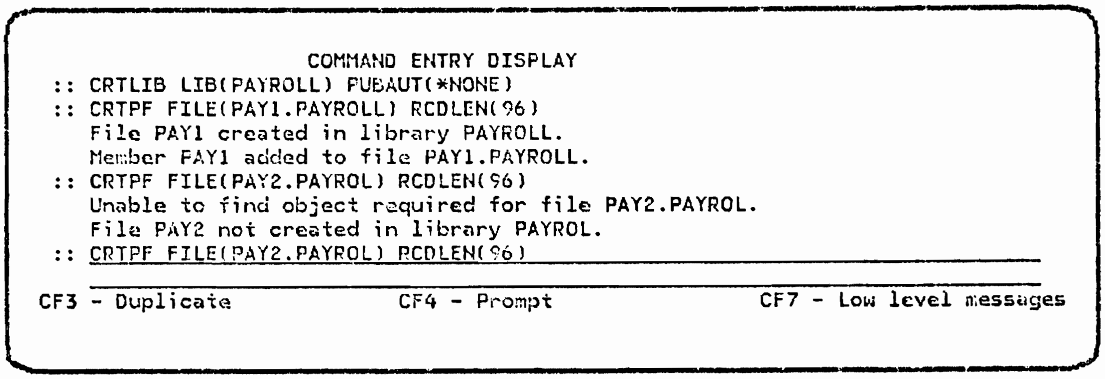
```
                         COMMAND ENTRY DISPLAY
:: CRTLIB LIB(PAYROLL) PUBAUT(*NONE)
:: CRTPF FILE(PAY1.PAYROLL) RCDLEN(96)
   File PAY1 created in library PAYROLL.
   Member PAY1 added to file PAY1.PAYROLL.
:: ________________________________________________________________
   ________________________________________________________________
   ________________________________________________________________
   ________________________________________________________________
   ________________________________________________________________
F3 - Duplicate           CF4 - Prompt              CF7 - Low level messages
```

You decide to create a second file named PAY2 that has similar characteristics, so you follow the same procedure you used to create PAY1. In entering the command, however, you misspell the library name. The system responds that the file could not be created:



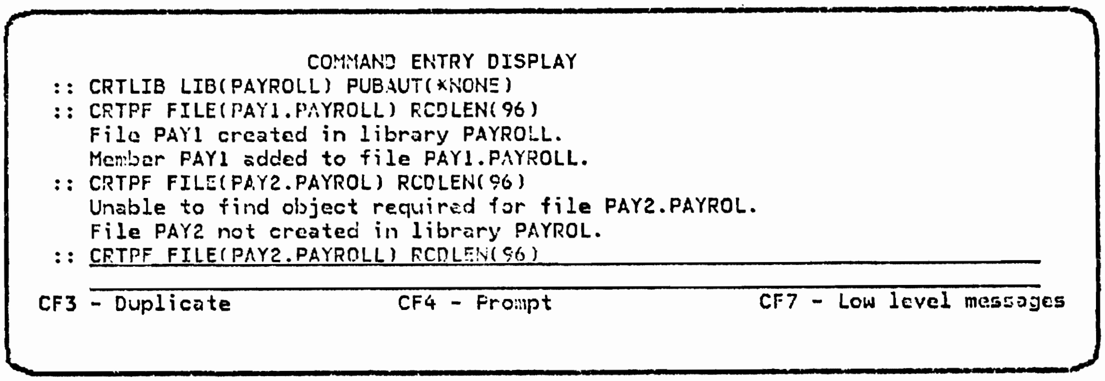
```
                         COMMAND ENTRY DISPLAY
:: CRTLIB LIB(PAYROLL) PUBAUT(*NONE)
:: CRTPF FILE(PAY1.PAYROLL) RCDLEN(96)
   File PAY1 created in library PAYROLL.
   Member PAY1 added to file PAY1.PAYROLL.
:: CRTPF FILE(PAY2.PAYROL) RCDLEN(96)
   Unable to find object required for file PAY2.PAYROL.
   File PAY2 not created in library PAYROL.
:: ________________________________________________________________
   ________________________________________________________________
CF3 - Duplicate          CF4 - Prompt              CF7 - Low level messages
```

You could reenter the command with the corrected value, but you use the duplication function of the CF3 key to simplify the task. You press the CF3 key and the incorrect command is duplicated into the current input area:



```
                         COMMAND ENTRY DISPLAY
 :: CRTLIB LIB(PAYROLL) FUBAUT(*NONE)
 :: CRTPF FILE(PAY1.PAYROLL) RCDLEN(96)
    File PAY1 created in library PAYROLL.
    Member FAY1 added to file PAY1.PAYROLL.
 :: CRTPF FILE(PAY2.PAYROL) RCDLEN(96)
    Unable to find object required for file PAY2.PAYROL.
    File PAY2 not created in library PAYROL.
 :: CRTPF FILE(PAY2.PAYROL) RCDLEN(96)

CF3 - Duplicate              CF4 - Prompt              CF7 - Low level messages
```

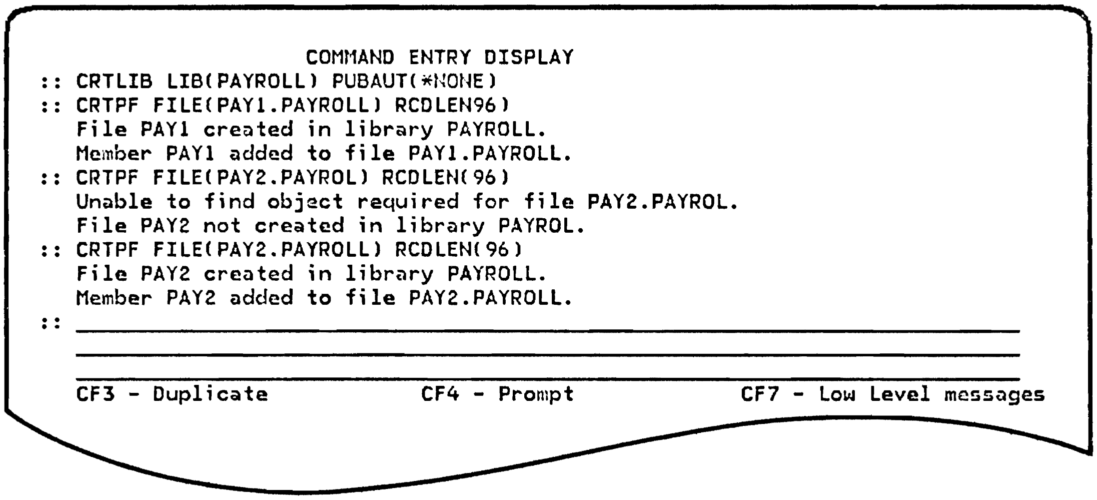
Now all you have to do is correct the part of the command that is wrong. You do this by keying in the remainder of the command from the point where it was incorrect.



```
                         COMMAND ENTRY DISPLAY
 :: CRTLIB LIB(PAYROLL) PUBAUT(*NONE)
 :: CRTPF FILE(PAY1.PAYROLL) RCDLEN(96)
    File PAY1 created in library PAYROLL.
    Member PAY1 added to file PAY1.PAYROLL.
 :: CRTPF FILE(PAY2.PAYROL) RCDLEN(96)
    Unable to find object required for file PAY2.PAYROL.
    File PAY2 not created in library PAYROL.
 :: CRTPF FILE(PAY2.PAYROLL) RCDLEN(96)

CF3 - Duplicate              CF4 - Prompt              CF7 - Low level messages
```

You could have simplified the task even more by using the previously entered command that created the file PAY1. If you had positioned the cursor under the line containing the command and pressed the CF3 key, that command instead of the incorrect command would have been duplicated into the input area. Then you would have needed only to change the 1 to a 2 and you would have the correct command.

When you press the Enter/Rec Adv key, the command is executed and the system responds with a message indicating that the file was created:



```
                        COMMAND ENTRY DISPLAY
:: CRTLIB LIB(PAYROLL) PUBAUT(*NONE)
:: CRTPF FILE(PAY1.PAYROLL) RCDLEN96)
   File PAY1 created in library PAYROLL.
   Member PAY1 added to file PAY1.PAYROLL.
:: CRTPF FILE(PAY2.PAYROL) RCDLEN(96)
   Unable to find object required for file PAY2.PAYROL.
   File PAY2 not created in library PAYROL.
:: CRTPF FILE(PAY2.PAYROLL) RCDLEN(96)
   File PAY2 created in library PAYROLL.
   Member PAY2 added to file PAY2.PAYROLL.
:: ____________________________________________________________________
   ____________________________________________________________________
   ____________________________________________________________________
   CF3 - Duplicate           CF4 - Prompt             CF7 - Low Level messages
```

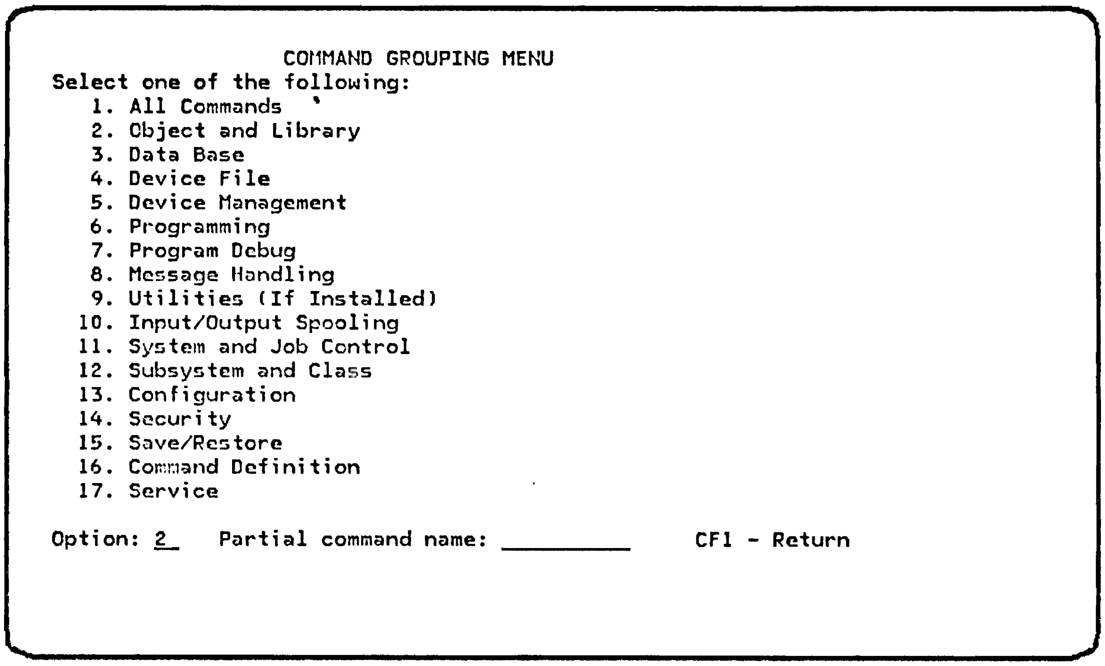
Example 2 — Finding and Executing an IBM-Supplied Command

You want to display the attributes of a file named UPDATE, but you are not sure what command to use. You need to find the correct command.

First, you make a prompt request by pressing the prompt (CF4) key without entering anything on the command entry display.

The system displays the command grouping menu. The cursor is automatically positioned for input in the Option field. You know that a file is an object, so you select option 2 (Object and Library); that is:

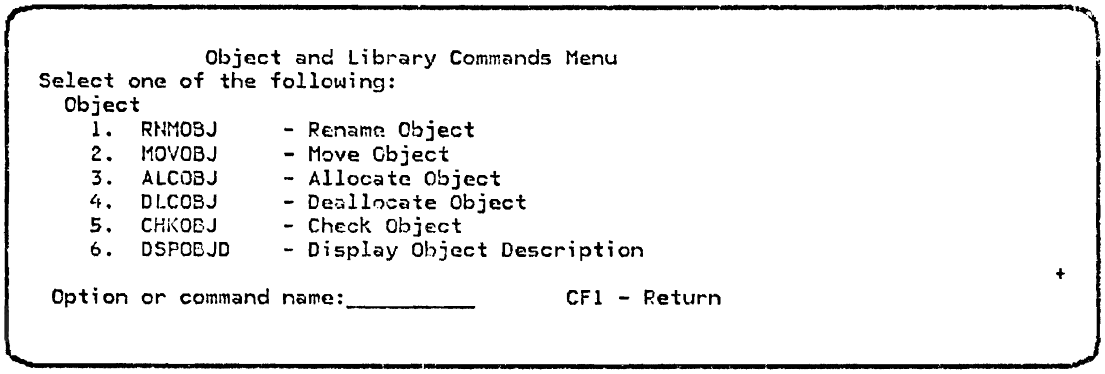
1. Key in the number 2.



```
                    COMMAND GROUPING MENU
Select one of the following:
   1. All Commands
   2. Object and Library
   3. Data Base
   4. Device File
   5. Device Management
   6. Programming
   7. Program Debug
   8. Message Handling
   9. Utilities (If Installed)
  10. Input/Output Spooling
  11. System and Job Control
  12. Subsystem and Class
  13. Configuration
  14. Security
  15. Save/Restore
  16. Command Definition
  17. Service

Option: 2_   Partial command name: __________     CF1 - Return
```

2. Press either the Enter/Rec Adv key or the CF4 key.

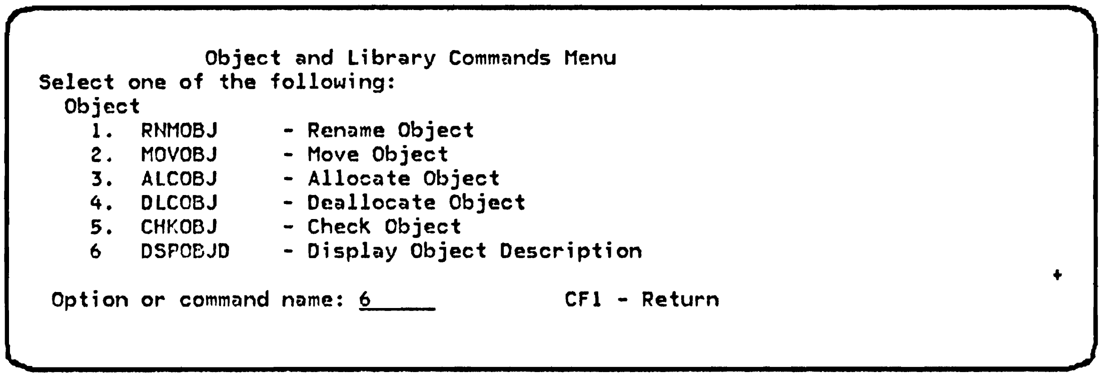
The system responds with a menu of object and library commands:



```
             Object and Library Commands Menu
Select one of the following:
  Object
    1.  RNMOBJ     - Rename Object
    2.  MOVOBJ     - Move Object
    3.  ALCOBJ     - Allocate Object
    4.  DLCOBJ     - Deallocate Object
    5.  CHKOBJ     - Check Object
    6.  DSPOBJD    - Display Object Description
                                                                +
 Option or command name:__________       CF1 - Return
```

You see that the option you want is number 6; therefore, you enter that option number or the command name in the input field at the bottom:

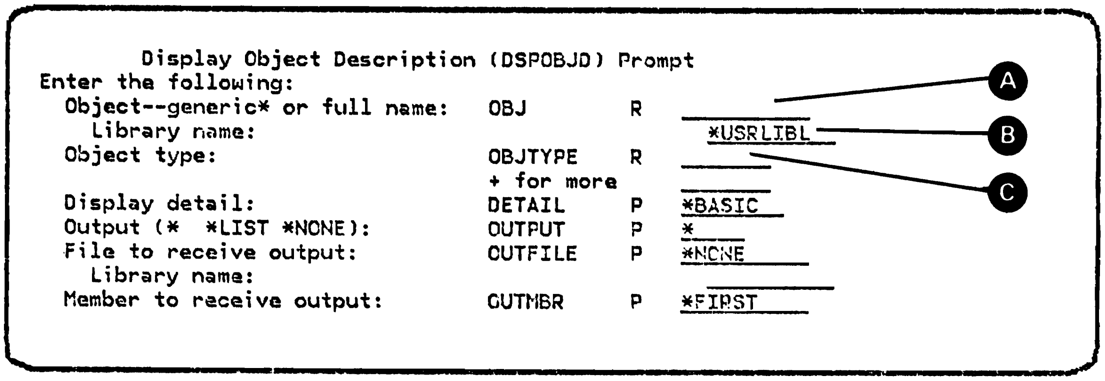
1. Key in 6 (indicating the option number) or DSPOBJD.



```
             Object and Library Commands Menu
 Select one of the following:
   Object
     1.  RNMOBJ      - Rename Object
     2.  MOVOBJ      - Move Object
     3.  ALCOBJ      - Allocate Object
     4.  DLCOBJ      - Deallocate Object
     5.  CHKOBJ      - Check Object
     6   DSPOBJD     - Display Object Description
                                                                +
  Option or command name: 6_____          CF1 - Return
```

2. Press either the Enter/Rec Adv key or the CF4 key.

The system responds with the display object description prompt, which lists all of the parameters of the DSPOBJD command:



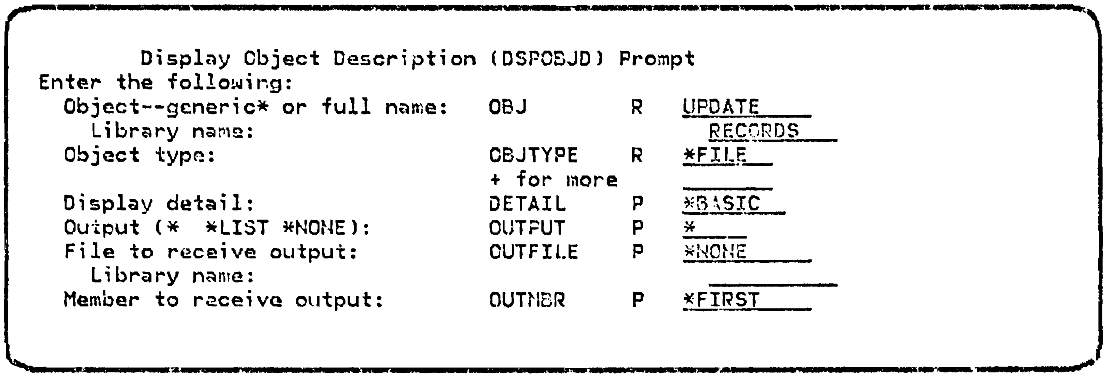
```
         Display Object Description (DSPOBJD) Prompt
 Enter the following:
   Object--generic* or full name:   OBJ        R   __________
     Library name:                                   *USRLIBL__
   Object type:                     OBJTYPE    R   _______
                                    + for more     _______
   Display detail:                  DETAIL     P   *BASIC__
   Output (*  *LIST *NONE):         OUTPUT     P   *____
   File to receive output:          OUTFILE    P   *NONE_____
     Library name:                                 __________
   Member to receive output:        OUTMBR     P   *FIRST____
```

The line for the OBJ and OBJTYPE parameter Ⓐ is blank and there is an R (meaning *Required*) next to the keyword. This indicates that you must enter the name of the object to be displayed. The remaining parameters have defaults, which you can either accept as shown or change.

In this case, you want to display the file named UPDATE that is located in the library RECORDS. Therefore, you:

1. Key UPDATE into the OBJ field Ⓐ.
2. Press the Field Exit key to move the cursor to the library qualifier field Ⓑ.
3. Key in RECORDS on top of the default value *USRLIBL, thus replacing the default.
4. Press the Field Exit key to move the cursor to the OBJTYPE field Ⓒ.
5. Change field Ⓒ to specify only files (which excludes other object types) by keying in *FILE.

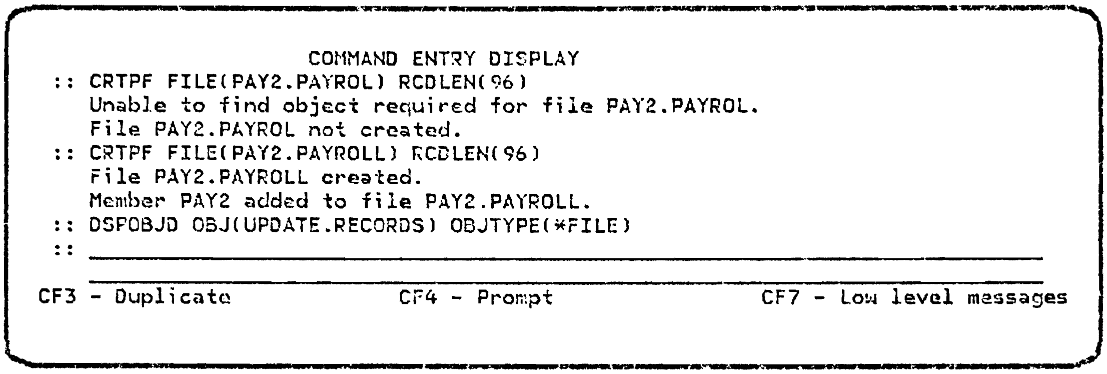
The prompt now looks like this:



```
       Display Object Description (DSPOBJD) Prompt
Enter the following:
  Object--generic* or full name:   OBJ        R   UPDATE
    Library name:                                  RECORDS
  Object type:                     OBJTYPE    R   *FILE
                                   + for more     ______
  Display detail:                  DETAIL     P   *BASIC
  Output (*  *LIST *NONE):         OUTPUT     P   *
  File to receive output:          OUTFILE    P   *NONE
    Library name:                                  __________
  Member to receive output:        OUTNBR     P   *FIRST
```

Now you have tailored the command to your needs. To execute it, you press the Enter/Rec Adv key as you did to execute a command you keyed into the command entry display.

When the command is executed, you receive a separate display from the DSPOBJD command. To return to the command entry display from the information display, you press the Enter/Rec Adv key. The command you entered through the prompt is incorporated into the command entry display. Note that the first two commands you entered have been rolled off the top of the screen to make room for more input area on the bottom.

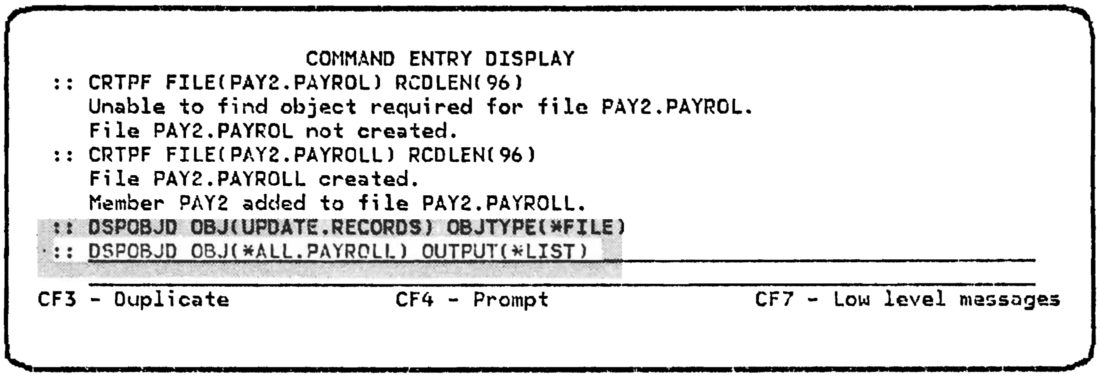


```
                       COMMAND ENTRY DISPLAY
 :: CRTPF FILE(PAY2.PAYROL) RCDLEN(96)
    Unable to find object required for file PAY2.PAYROL.
    File PAY2.PAYROL not created.
 :: CRTPF FILE(PAY2.PAYROLL) RCDLEN(96)
    File PAY2.PAYROLL created.
    Member PAY2 added to file PAY2.PAYROLL.
 :: DSPOBJD OBJ(UPDATE.RECORDS) OBJTYPE(*FILE)
 :: ____________________________________________________________________
    ____________________________________________________________________
CF3 - Duplicate              CF4 - Prompt                 CF7 - Low level messages
```

Example 3 – Executing a Command from Partial Entry

You want to use the DSPOBJD command to print the attributes of specific types of objects contained in the library PAYROLL. You know what to enter for some of the parameters, but you are not sure of the others. Therefore, you do the following:

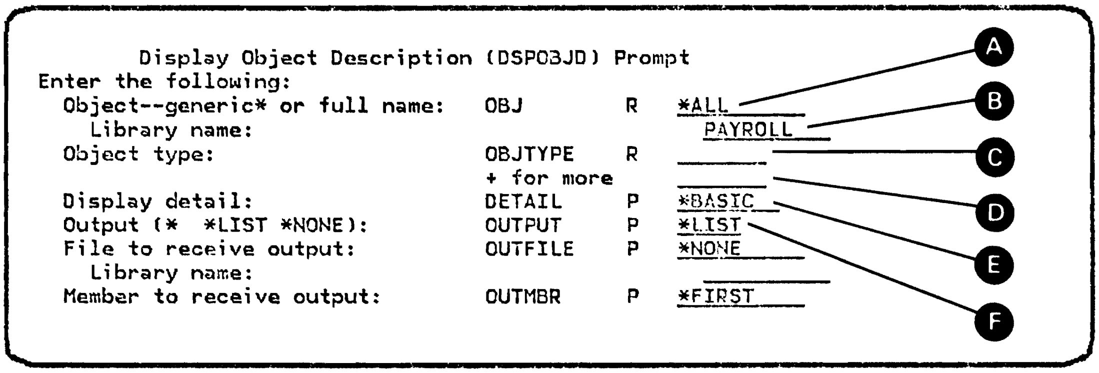
1. Key in the part of the command you know on the command entry display.



```
                        COMMAND ENTRY DISPLAY
 :: CRTPF FILE(PAY2.PAYROL) RCDLEN(96)
    Unable to find object required for file PAY2.PAYROL.
    File PAY2.PAYROL not created.
 :: CRTPF FILE(PAY2.PAYROLL) RCDLEN(96)
    File PAY2.PAYROLL created.
    Member PAY2 added to file PAY2.PAYROLL.
 :: DSPOBJD OBJ(UPDATE.RECORDS) OBJTYPE(*FILE)
 :: DSPOBJD OBJ(*ALL.PAYROLL) OUTPUT(*LIST)

CF3 - Duplicate           CF4 - Prompt             CF7 - Low level messages
```

2. Press the prompt (CF4) key.

The system responds with the display object description prompt as it did when you selected the DSPOBJD command from a menu in Example 2. Unlike the previous example, however, the parameter values you entered in the partial command appear on the prompt when it is displayed. The value *ALL appears in the OBJ field A, which is normally blank and the name PAYROLL appears in the library qualifier field B instead of the default value *USRLIBL; and *LIST appears in the OUTPUT field F instead of the default value *. The value *BASIC appears in the DETAIL field E because you did not specify the DETAIL parameter.



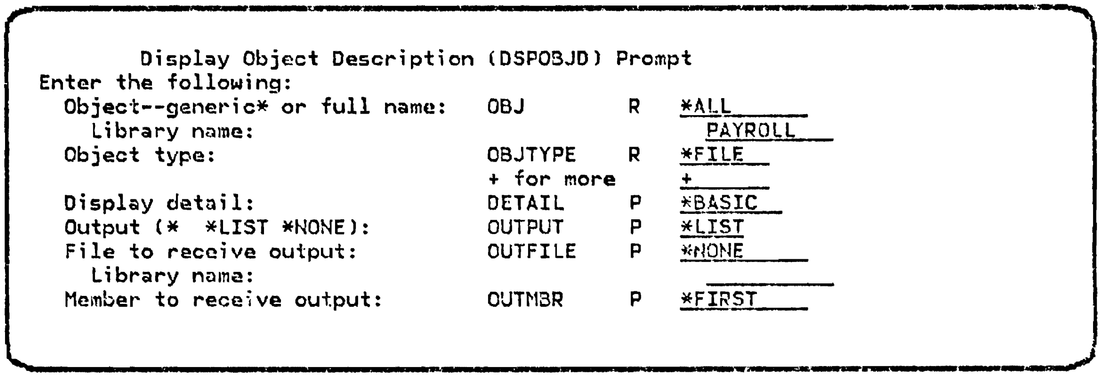
```
        Display Object Description (DSPOBJD) Prompt
Enter the following:
  Object--generic* or full name:   OBJ        R   *ALL_____
    Library name:                                   PAYROLL_
  Object type:                     OBJTYPE    R   ________
                                   + for more     ________
  Display detail:                  DETAIL     P   *BASIC__
  Output (* *LIST *NONE):          OUTPUT     P   *LIST___
  File to receive output:          OUTFILE    P   *NONE___
    Library name:                                   ________
  Member to receive output:        OUTMBR     P   *FIRST__
```

You want to specify the object types *FILE, *PGM (program), and *CMD (command), but the OBJTYPE parameter has only two input fields Ⓒ and Ⓓ. Therefore, you do the following:

1. Key *FILE into the first field Ⓒ.

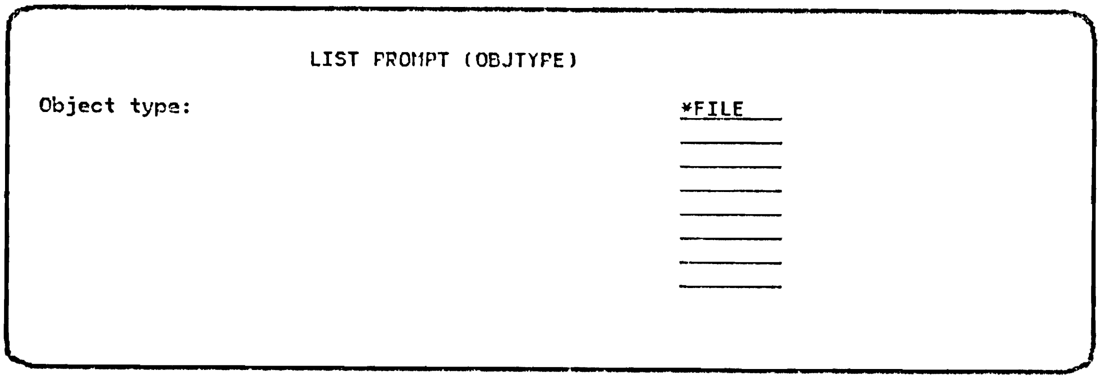
2. Key a + into the second field Ⓓ.

The prompt now looks like this:



```
       Display Object Description (DSPOBJD) Prompt
Enter the following:
  Object--generic* or full name:   OBJ        R   *ALL
    Library name:                                   PAYROLL
  Object type:                     OBJTYPE    R   *FILE
                                   + for more     +
  Display detail:                  DETAIL     P   *BASIC
  Output (*  *LIST *NONE):         OUTPUT     P   *LIST
  File to receive output:          OUTFILE    P   *NONE
    Library name:
  Member to receive output:        OUTMBR     P   *FIRST
```

When you press the Enter/Rec Adv key, the system responds with a separate list prompt on which you can enter additional object types.



```
                         LIST PROMPT (OBJTYPE)

Object type:                                          *FILE
```

Note that the value you previously entered appears as the first item of the list. The cursor is positioned on the first line, which is the first input field on the list prompt. Now you can add the additional object types:

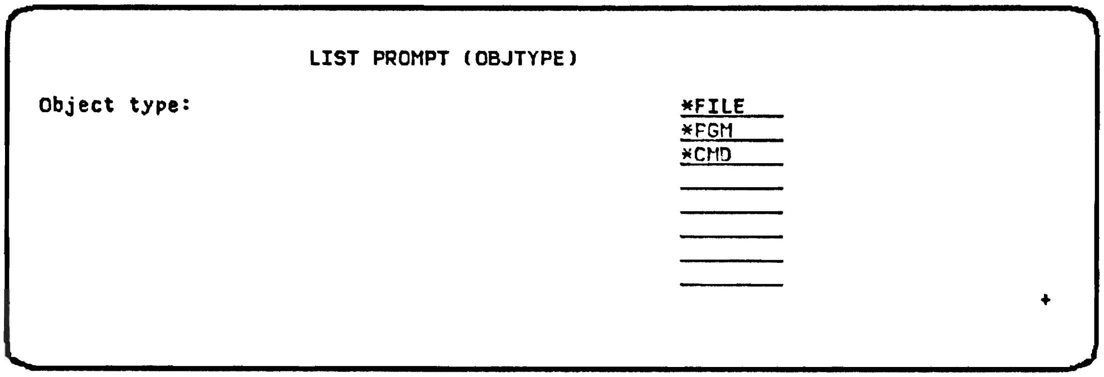
1. Press the ↵ (New Line) key to move the cursor to the next line (the second input field).

2. Key in *PGM.

3. Press the Field Exit key to move the cursor to the next line (the third input field).

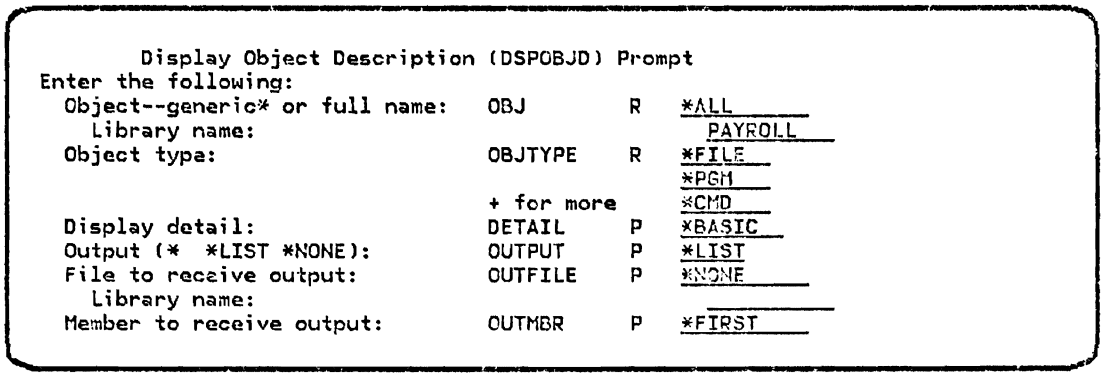
4. Key in *CMD.

The list prompt now looks like this:



```
                    LIST PROMPT (OBJTYPE)

Object type:                                         *FILE
                                                     *PGM
                                                     *CMD
                                                     ________
                                                     ________
                                                     ________
                                                     ________
                                                     ________
                                                                         +
```

When you press the Enter/Rec Adv key, the display object description prompt reappears on your screen, but in a modified form that includes the values you added:



```
        Display Object Description (DSPOBJD) Prompt
Enter the following:
  Object--generic* or full name:    OBJ          R    *ALL
    Library name:                                       PAYROLL
  Object type:                      OBJTYPE      R    *FILE
                                                      *PGM
                                    + for more        *CMD
  Display detail:                   DETAIL       P    *BASIC
  Output (*  *LIST *NONE):          OUTPUT       P    *LIST
  File to receive output:           OUTFILE      P    *NONE
    Library name:                                       ________
  Member to receive output:         OUTMBR       P    *FIRST
```

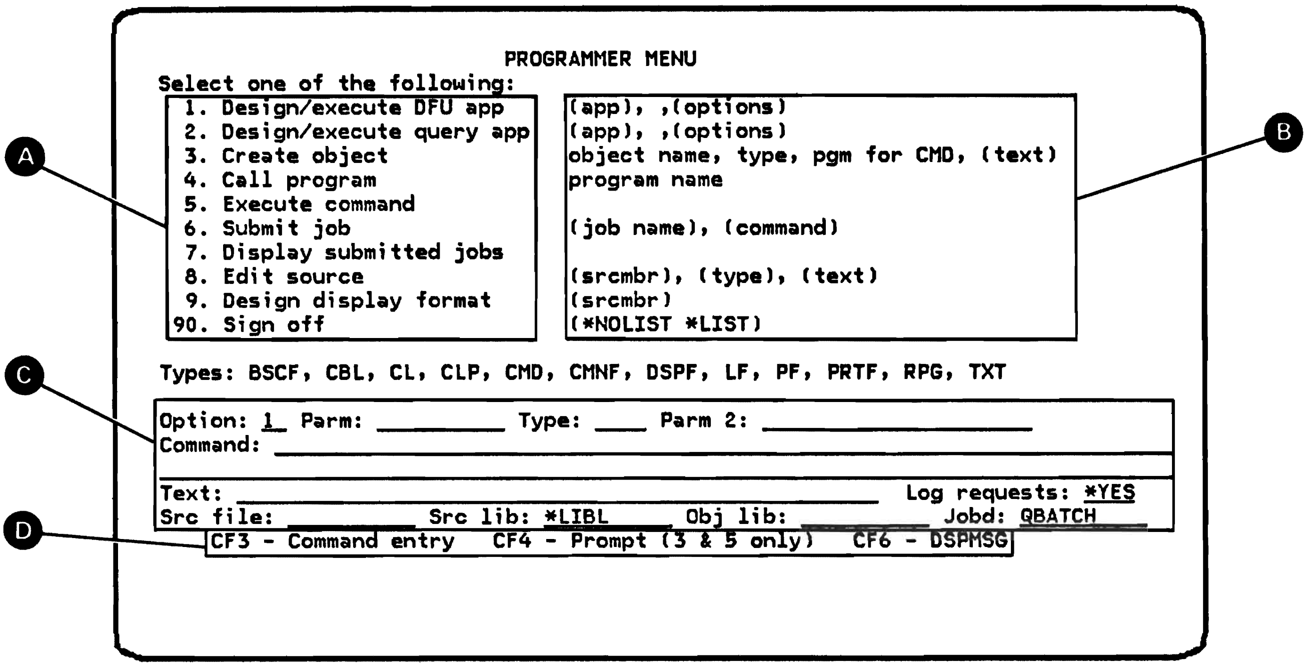
When you press the Enter/Rec Adv key, the requested information is placed on an output queue to be printed, and you receive the command entry display again with the command incorporated (as in Example 2).

How to Use Specialized Displays

THE PROGRAMMER MENU

The programmer menu can be your working display if you are using a work station with a 24-line screen (5251 Model 11 or 12). If you sign on to the system at a 5251 Model 11 or 12 using the QPGMR user profile, the programmer menu is automatically displayed. If you sign on with another profile, you can request the programmer menu (if you are authorized to it) by using the CALL command to call QPGMMENU. QPGMMENU is an IBM-supplied program that displays the programmer menu.

Figure 5-6 shows the programmer menu. It provides a convenient means of requesting 10 frequently used programmer functions. The menu itself includes the list of the functions Ⓐ, a list of required and optional parameters for each function (optional parameters are in parentheses) Ⓑ, an input area for requesting the function and specifying the appropriate parameters Ⓒ, and a short description of command keys that can be used from the menu Ⓓ.



```
                          PROGRAMMER MENU
Select one of the following:
  1. Design/execute DFU app      (app), ,(options)
  2. Design/execute query app    (app), ,(options)
  3. Create object               object name, type, pgm for CMD, (text)
  4. Call program                program name
  5. Execute command
  6. Submit job                  (job name), (command)
  7. Display submitted jobs
  8. Edit source                 (srcmbr), (type), (text)
  9. Design display format       (srcmbr)
 90. Sign off                    (*NOLIST *LIST)

Types: BSCF, CBL, CL, CLP, CMD, CMNF, DSPF, LF, PF, PRTF, RPG, TXT

Option: 1_ Parm: __________ Type: ____ Parm 2: _____________________
Command: ___________________________________________________________

Text: ________________________________________ Log requests: *YES
Src file: __________ Src lib: *LIBL_____ Obj lib: __________ Jobd: QBATCH____
    CF3 - Command entry   CF4 - Prompt (3 & 5 only)   CF6 - DSPMSG
```

Figure 5-6. Programmer Menu

The programmer menu simplifies many of the Control Program Facility (CPF), COBOL, RPG, and Interactive Data Base Utilities (IDU) tasks that the programmer frequently does. For example, you can use option 8 of the menu to request the Source Entry Utility (SEU) for editing an RPG source member. Then you can use option 3 to compile the RPG program from the source member. The compilation is automatically submitted as a batch job so that you can perform other functions while it compiles.

The programmer menu minimizes the amount of keying required because input values remain on the display from option to option. For example, entries that you make on the bottom line of the input area (source file, source library, object library, and job description) remain there until you change them.

If your system is organized with application libraries and you have source files in those libraries, you usually need to fill in the bottom line of the programmer menu once. This bottom line is then correct for all your work for that particular application.

Options 1, 2, 3, 4, 6, 7, 8, 9 and 90 generate command requests for you and, therefore, save keying time. These options, and the corresponding commands are summarized below:

Option	Use	Command
1	Requests DFU to create, change, delete, execute, or manage a DFU application	Design DFU Application (DSNDFUAPP)
2	Requests query to create, change, delete, execute, or manage a query application	Design Query Application (DSNQRYAPP)
3	Creates one of the following objects by submitting the generated command as a batch job:	
	Binary Synchronous Communications File	Create Binary Synchronous Communications File (CRTBSCF)
	COBOL program	Create COBOL Program (CRTCBLPGM)
	Communications File	Create Communications File (CRTCMNF)
	Control language program	Create Control Language Program (CRTCLPGM)
	User-defined command	Create Command (CRTCMD)
	DFU application	Create DFU Application (CRTDFUAPP)
	Display file	Create Display File (CRTDSPF)
	Logical file	Create Logical File (CRTLF)
	Physical file	Create Physical File (CRTPF)
	Printer device file	Create Printer File (CRTPRTF)
	Query application	Create Query Application (CRTQRYAPP)
	RPG program	Create RPG Program (CRTRPGPGM)
4	Calls a program	Call Program (CALL)
6	Submits a batch job to a job queue	Submit Job (SBMJOB)
7	Requests display of submitted jobs	Display Submitted Jobs (DSPSBMJOB)
8	Requests use of SEU to enter or update source in a source member	Edit Source (EDTSRC)
9	Requests use of screen design aid (SDA) to design, change, or test a display record format	Design Format (DSNFMT)
90	Sign off	Sign Off (SIGNOFF)

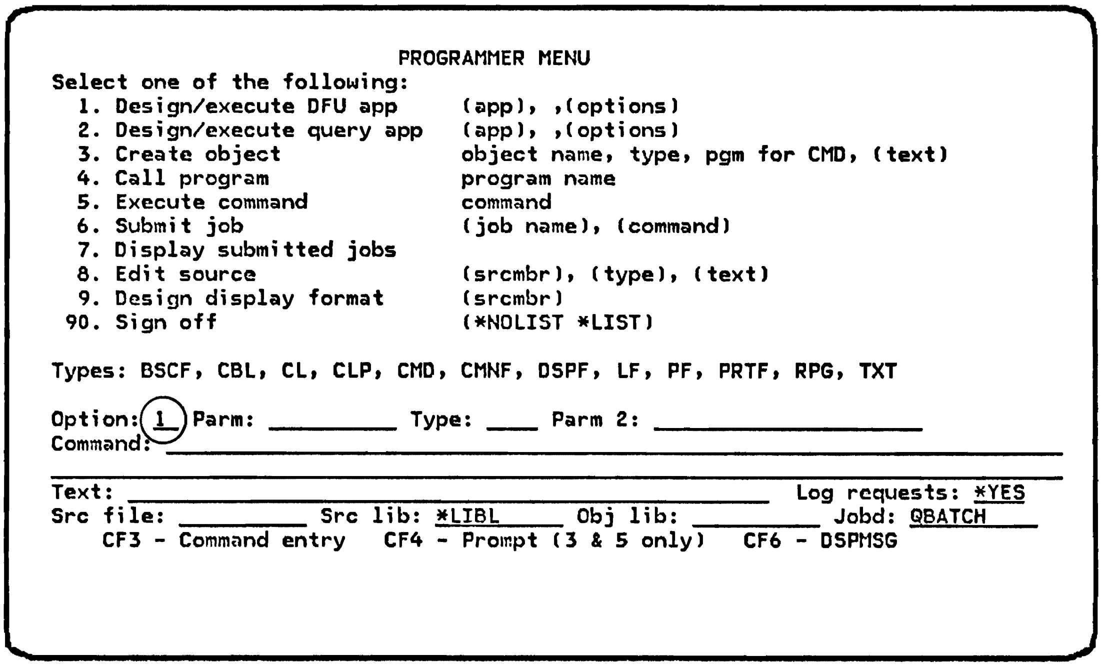
Option 5 can be used to execute a control language command on the programmer menu without switching to the command entry display.

All 10 options on the programmer menu are described in greater detail later in this chapter.

How To Request Functions from the Programmer Menu

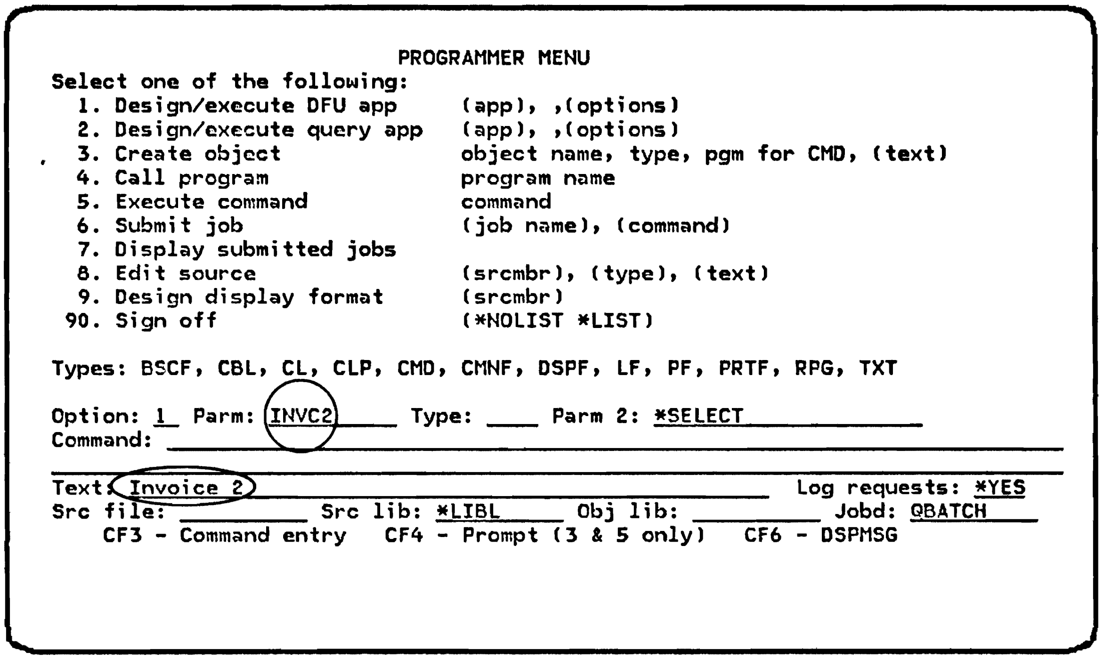
You can request one of the 10 numbered functions from the programmer menu by doing the following steps:

1. Key the number of the function in the Option field.



```
                              PROGRAMMER MENU
Select one of the following:
  1. Design/execute DFU app     (app), ,(options)
  2. Design/execute query app   (app), ,(options)
  3. Create object              object name, type, pgm for CMD, (text)
  4. Call program               program name
  5. Execute command            command
  6. Submit job                 (job name), (command)
  7. Display submitted jobs
  8. Edit source                (srcmbr), (type), (text)
  9. Design display format      (srcmbr)
 90. Sign off                   (*NOLIST *LIST)

Types: BSCF, CBL, CL, CLP, CMD, CMNF, DSPF, LF, PF, PRTF, RPG, TXT

Option: 1  Parm: __________ Type: ____ Parm 2: ____________________
Command: ____________________________________________________________
_____________________________________________________________________
Text: ___________________________________________ Log requests: *YES
Src file: __________ Src lib: *LIBL_____ Obj lib: __________ Jobd: QBATCH____
    CF3 - Command entry   CF4 - Prompt (3 & 5 only)   CF6 - DSPMSG
```

2. Key the parameters for the function in the input area.



```
                              PROGRAMMER MENU
Select one of the following:
  1. Design/execute DFU app     (app), ,(options)
  2. Design/execute query app   (app), ,(options)
  3. Create object              object name, type, pgm for CMD, (text)
  4. Call program               program name
  5. Execute command            command
  6. Submit job                 (job name), (command)
  7. Display submitted jobs
  8. Edit source                (srcmbr), (type), (text)
  9. Design display format      (srcmbr)
 90. Sign off                   (*NOLIST *LIST)

Types: BSCF, CBL, CL, CLP, CMD, CMNF, DSPF, LF, PF, PRTF, RPG, TXT

Option: 1  Parm: INVC2_____ Type: ____ Parm 2: *SELECT_____________
Command: ____________________________________________________________
_____________________________________________________________________
Text: Invoice 2__________________________________ Log requests: *YES
Src file: __________ Src lib: *LIBL_____ Obj lib: __________ Jobd: QBATCH____
    CF3 - Command entry   CF4 - Prompt (3 & 5 only)   CF6 - DSPMSG
```

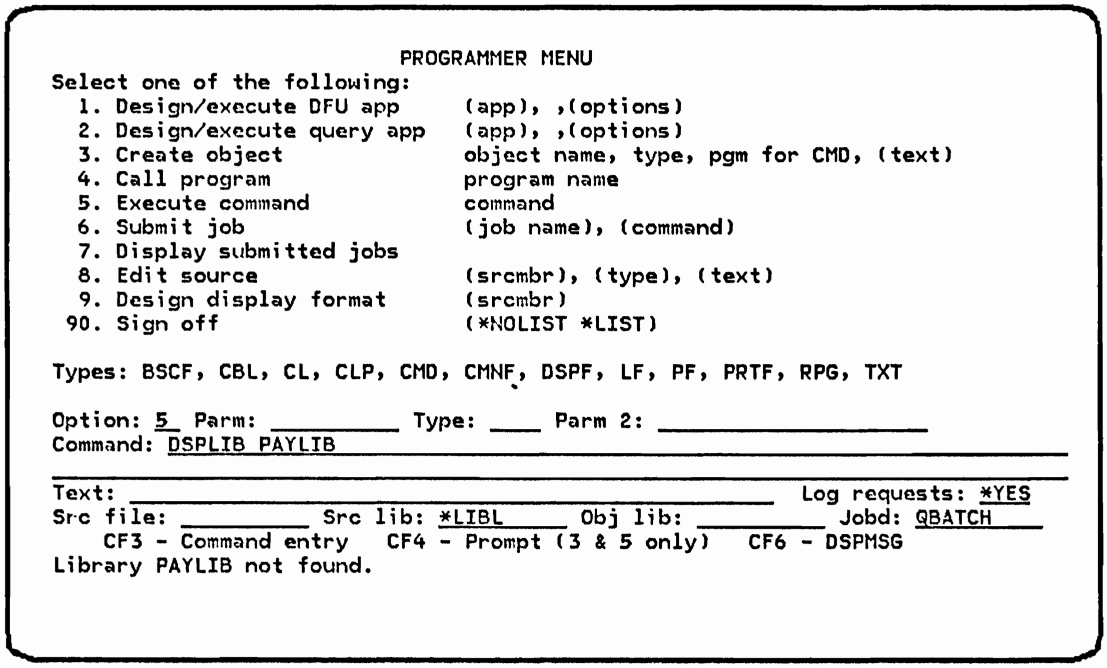
You can advance the cursor from field to field with the ⇥ (Field Advance) key or move the cursor back to a previous field with the ⇤ (Field Backspace) key. You can move the cursor from one line to another with the ↵ (New Line) key.

After you key data in an input field, you can press the Field Exit key to clear the remainder of the field and advance to the next input field.

All the fields in the input area are not used for each of the 10 functions. For example, the command input field is not used if you select option 3. You can leave data in these unused fields for subsequent operations, and the system will ignore them.

3. Press the Enter/Rec Adv key. When the function is done, the Input Inhibited light goes off and the programmer menu is shown again. The option field is blank, but the other entries from the previous programmer menu remain.

If an error occurred, an error message is shown on the bottom of the programmer menu. For example:



```
                           PROGRAMMER MENU
Select one of the following:
  1. Design/execute DFU app     (app), ,(options)
  2. Design/execute query app   (app), ,(options)
  3. Create object              object name, type, pgm for CMD, (text)
  4. Call program               program name
  5. Execute command            command
  6. Submit job                 (job name), (command)
  7. Display submitted jobs
  8. Edit source                (srcmbr), (type), (text)
  9. Design display format      (srcmbr)
 90. Sign off                   (*NOLIST *LIST)

Types: BSCF, CBL, CL, CLP, CMD, CMNF, DSPF, LF, PF, PRTF, RPG, TXT

Option: 5_ Parm: __________ Type: ____ Parm 2: _____________________
Command: DSPLIB PAYLIB_____________________________________________________
_______________________________________________________________________________
Text: __________________________________________________ Log requests: *YES
Src file: __________ Src lib: *LIBL______ Obj lib: __________ Jobd: QBATCH____
    CF3 - Command entry  CF4 - Prompt (3 & 5 only)  CF6 - DSPMSG
Library PAYLIB not found.
```

To correct an error, key the correct information in the input area on the programmer menu and then press the Enter/Rec Adv key.

Sometimes when a command fails, the message you receive may not sufficiently explain the problem. You may wish to see the second-level text for more information. To do this, position the cursor anywhere under the message and press the Help key. You may also wish to see the lower level messages for further help. You can do this by pressing the CF3 key to go to the command entry display and then pressing the CF7 key. This will display the last request from the menu and any lower level messages associated with that request.

Using Command Keys

You can use command keys 1, 3, 4, 6, 9, and 11 to request the following functions from the programmer menu.

CF1 The CF1 key can be used to leave the programmer menu and return to the place from which the programmer menu was requested. (If your sign on program calls QPGMMENU, the CF1 key causes a fresh copy of the programmer menu to be displayed.)

Some of the options on the programmer menu request other displays. You can return to the programmer menu from these displays by pressing the CF1 key.

CF3 The CF3 key can be used to request the command entry display from the programmer menu.

CF4 The CF4 key can be used to request command prompting assistance. This key is valid only for options 3 and 5 on the programmer menu.

CF6 The CF6 key can be used to display all messages that are currently on your work station message queue.

CF9 The CF9 key can be used after option 3 to interactively execute a create object request that cannot be submitted to the batch subsystem because it is longer than 256 characters.

CF11 The CF11 key can be used with option 3 to request that an existing object be replaced. The description of option 3 later in this chapter will provide more information about using the CF11 key.

If a create-object request is submitted to the job queue from the programmer menu, a message that indicates whether or not the job successfully executed is sent to your work station message queue and the Message Waiting light will come on and the keyboard buzzer will sound.

Option 1. Design/Execute DFU Application

Selecting option 1 results in a series of menus and displays to help you create, change, delete, execute, or manage a DFU application.

There are no required input fields for option 1.

Optional Input Fields for Option 1

The Parm, Obj lib, and Parm 2 input fields are optional if you select option 1.

Parm — Specify the application name that you want to process. If you leave the parm field blank, a default value of *PRV is assumed. *PRV specifies that DFU will use the name of the previous application that you accessed during your last work session.

Obj lib — Specify the library name that contains the application to be processed. If you leave the object library field blank, a default value of *PRV is assumed. *PRV specifies that the name of the previous library that was used for this function should be used again. *LIBL is also allowed and can be entered in this field.

Parm 2 — Specify up to two, single-digit option numbers for the first two menus that appear from DSNDFUAPP. If two option numbers are used, they must be separated by at least one blank. If you leave this field blank, a default value of *SELECT is assumed. DFU then displays the DFU menu so you can select the menu options you want.

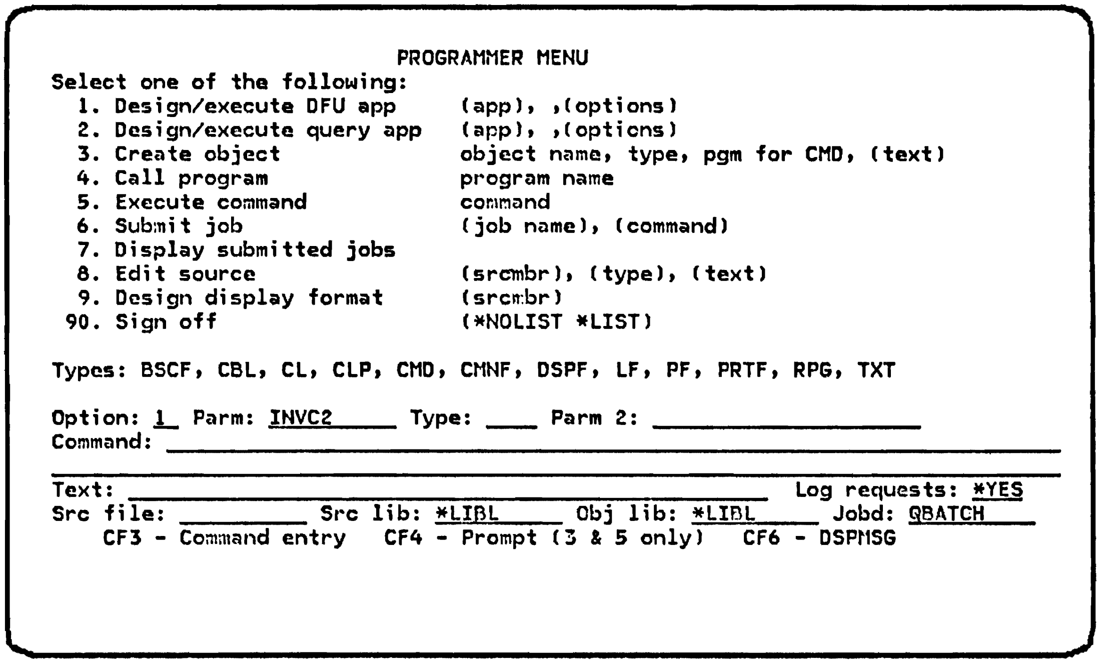
Ignored Input Fields for Option 1

The Type, Command, Text, Src file, Src lib, and Jobd input fields are ignored for option 1.

Example of How to Use Option 1

The following display shows an example of how option 1 on the programmer menu is used.

To create a DFU application named *INVC2* in *LIBL, enter:



```
                          PROGRAMMER MENU
Select one of the following:
  1. Design/execute DFU app     (app), ,(options)
  2. Design/execute query app   (app), ,(options)
  3. Create object              object name, type, pgm for CMD, (text)
  4. Call program               program name
  5. Execute command            command
  6. Submit job                 (job name), (command)
  7. Display submitted jobs
  8. Edit source                (srcmbr), (type), (text)
  9. Design display format      (srcmbr)
 90. Sign off                   (*NOLIST *LIST)

Types: BSCF, CBL, CL, CLP, CMD, CMNF, DSPF, LF, PF, PRTF, RPG, TXT

Option: 1_ Parm: INVC2_____ Type: ____ Parm 2: ____________________
Command: ________________________________________________________________
_________________________________________________________________________
Text: __________________________________________________ Log requests: *YES
Src file: _________ Src lib: *LIBL_____ Obj lib: *LIBL_____ Jobd: QBATCH____
    CF3 - Command entry   CF4 - Prompt (3 & 5 only)   CF6 - DSPMSG
```

Equivalent DSNDFUAPP Command

DSNDFUAPP INVC2

Option 2. Design/Execute Query Application

Selecting option 2 results in a series of menus and displays to help you create, change, delete, execute, or manage a query application.

There are no required input fields for option 2.

Optional Input Field for Option 2

The Parm, Obj lib, and Parm 2 input fields are optional if you select option 2.

Parm — Specify the application name that you want to process. If you leave the parm field blank, a default value of *PRV is assumed. *PRV specifies that query will use the name of the previous application that you accessed during your last work session.

Obj lib — Specify the library name that contains the application to be processed. If you leave the library field blank, a default value of *PRV is assumed. *PRV specifies that the name of the previous library that was used for this function should be used again. *LIBL is also allowed and can be entered in this field.

Parm 2 — Specify up to two, single-digit option numbers for the first two menus that appear from DSNQRYAPP. If two option numbers are used, they must be separated by at least one blank. If you leave this field blank, a default value of *SELECT is assumed. Query then displays the query menu so you can select the menu options you want.

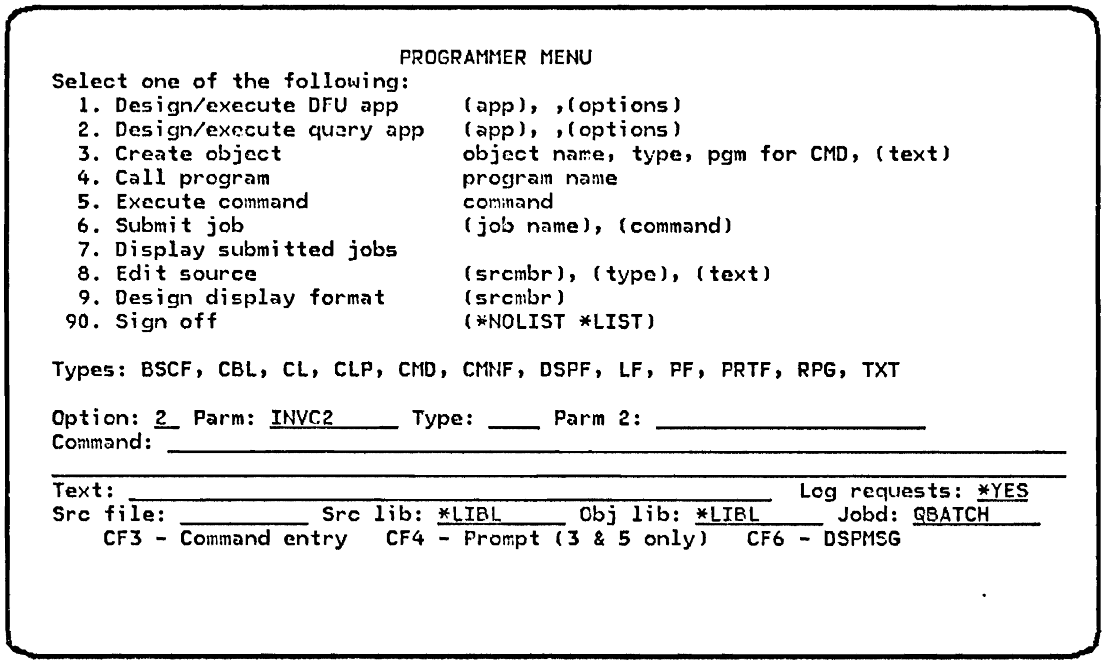
Ignored Input Fields for Option 2

The Type, Command, Text, Src file, Src lib, and Jobd input fields are ignored for option 2.

Example of How to Use Option 2

The following display shows an example of how option 2 on the programmer menu is used.

To design/execute a query application named INVC2 in *LIBL, enter:



```
                              PROGRAMMER MENU
Select one of the following:
  1. Design/execute DFU app     (app), ,(options)
  2. Design/execute query app   (app), ,(options)
  3. Create object              object name, type, pgm for CMD, (text)
  4. Call program               program name
  5. Execute command            command
  6. Submit job                 (job name), (command)
  7. Display submitted jobs
  8. Edit source                (srcmbr), (type), (text)
  9. Design display format      (srcmbr)
 90. Sign off                   (*NOLIST *LIST)

Types: BSCF, CBL, CL, CLP, CMD, CMNF, DSPF, LF, PF, PRTF, RPG, TXT

Option: 2_ Parm: INVC2_____ Type: ____ Parm 2: _____________________
Command: ____________________________________________________________
_____________________________________________________________________
Text: ___________________________________________ Log requests: *YES
Src file: __________ Src lib: *LIBL_____ Obj lib: *LIBL_____ Jobd: QBATCH____
    CF3 - Command entry   CF4 - Prompt (3 & 5 only)   CF6 - DSPMSG
```

Equivalent DSNQRYAPP Command

DSNQRYAPP   INVC2

Option 3. Create Object

Select option 3 on the programmer menu to create one of the following objects from a source member:

- Binary synchronous communication file
- COBOL program
- Communications file
- Control language program
- User-defined command
- DFU application
- Display file
- Logical file
- Physical file
- Printer device file
- Query application
- RPG program

The object creation is done by submitting a control language command as a batch job and by putting it on the job queue. The name the batch job is submitted under is derived from a combination of the object name (specified in the Parm field) and the user name of the interactive job. The other job attributes come from the job description specified.

If you are creating a new object, press the Enter/Rec Adv key after you fill out the menu. If you are replacing an existing object, press the CF11 key instead of the Enter/Rec Adv key.

If you press the Enter/Rec Adv key after selecting option 3, before submitting the job, the system determines whether or not the object exists. If it does, a message appears indicating that the object exists. After pressing the Error Reset key, you can press the CF11 key to delete the existing object, or enter a different request; press the Enter/Rec Adv key to execute the new request.

Parameters that you key on the programmer menu for option 3 indicate the name of the created object, the type of object to create, the library and source file that contain the source member from which the object is created, the library in which to put the object, any text you want associated with the object, and the job description that controls the batch job on the job queue. The source member from which the object is created is assumed to have the same name as its object.

The entries that you make on the programmer menu for option 3 are used as parameter input to a create object command. Defaults are assumed for many parameters. You can press the CF4 key instead of the Enter/Rec Adv key to display the defaults and the values that you have specified. You can change the parameter values before you submit the command. If, as a result of entering multiple values, the resulting command submitted on the SBMJOB command is longer than 256 characters, you receive a message (command too long to submit). After pressing the Error Reset key, you can press the CF9 key to execute the command interactively, or enter a different request and press the Enter/Rec Adv key to execute the new request.

If the command is less than 256 characters, a batch job is submitted. When the job completes, a message is sent to your work station message queue. The message indicates whether the job completed successfully. The keyboard buzzer sounds briefly to notify you that a message has arrived in your message queue. To display the message(s) in your work station message queue, press the CF6 key.

Required Input Fields for Option 3

The Parm and Type input fields are always required for option 3. The Parm 2 input field is required if the type field contains CMD.

Parm Specify the name of the object that you want to create. The system assumes that the source member for the object has the same name. The object name along with the user name is used as the name of the submitted job.

Type The type that you specify indicates the type of object to create. The type results in a default source file being used if the source file input field is blank. The source member is assumed to be in the default source file if you do not specify a source file.

Type	**Object Created**	**Command Used**	**Default Source File**
BSCF	Binary synchronous communications file	CRTBSCF	QDDSSRC
CBL	COBOL program	CRTCBLPGM	QCBLSRC
CLP	Control language program	CRTCLPGM	QCLSRC
CMD	Command definition	CRTCMD	QCMDSRC
CMNF	Communications file	CRTCMNF	QDDSSRC
DFU[1]	Data file utility application	CRTDFUAPP	QUDSSRC
DSPF	Display file	CRTDSPF	QDDSSRC
LF	Logical file	CRTLF	QDDSSRC
PF	Physical file	CRTPF	QDDSSRC
PRTF	Printer device file	CRTPRTF	QDDSSRC
QRY[1]	Query application	CRTQRYAPP	QUDSSRC
RPG	RPG program	CRTRPGPGM	QRPGSRC

[1] Options 1 and 2 support the creating and changing of these object types through prompting without using Utility Definition Specifications (UDS) source.

Parm 2 This input field is required only if the type input field is CMD. Specify the name of the command processing program in Parm 2.

Optional Input Fields for Option 3

The Text, Src file, Src lib, Obj lib, and Jobd input fields are optional for option 3.

Text Specify a text description of the object. This description is stored with the object and can be used to identify it.

Src file This input field is blank when the programmer menu initially appears. You can leave it blank for option 3 if you want to use a default source file. The explanation of the type input field indicates the default source files. You must specify a source file if the source member is not in the default source file.

Src lib Specify the name of the library that contains the source member from which the object is created, or use the default value, *LIBL, which causes the libraries in the job library list to be searched.

Obj lib This input field is blank when the programmer menu initially appears. You can leave it blank for option 3 if you want the created object placed in the general purpose library, QGPL. You must specify an object library if you do not want the object placed in QGPL.

Jobd Specify the name of the job description that controls the job on the job queue. The following parameters in the job description are important for option 3:

INLLIBL Determines the initial library list.
JOBQ Determines the job queue on which to put the batch job.
OUTQ Determines the default output queue used for the job printer output and job log.

The job description that you specify must be found by the use of the current library list. The default job description is QBATCH. It is suggested that these values can best be tailored to your needs if you use your own job description rather than QBATCH. For example, externally described files used by an RPG program are searched for when the program is compiled; only the current library list indicated in the job description can be used to find these files. Therefore, if the library list in the QBATCH job description is not correct, you should provide your own job description and specify the correct library list to be used for the RPG compilation.

Refer to the *CPF Programmer's Guide* for information about a job description.

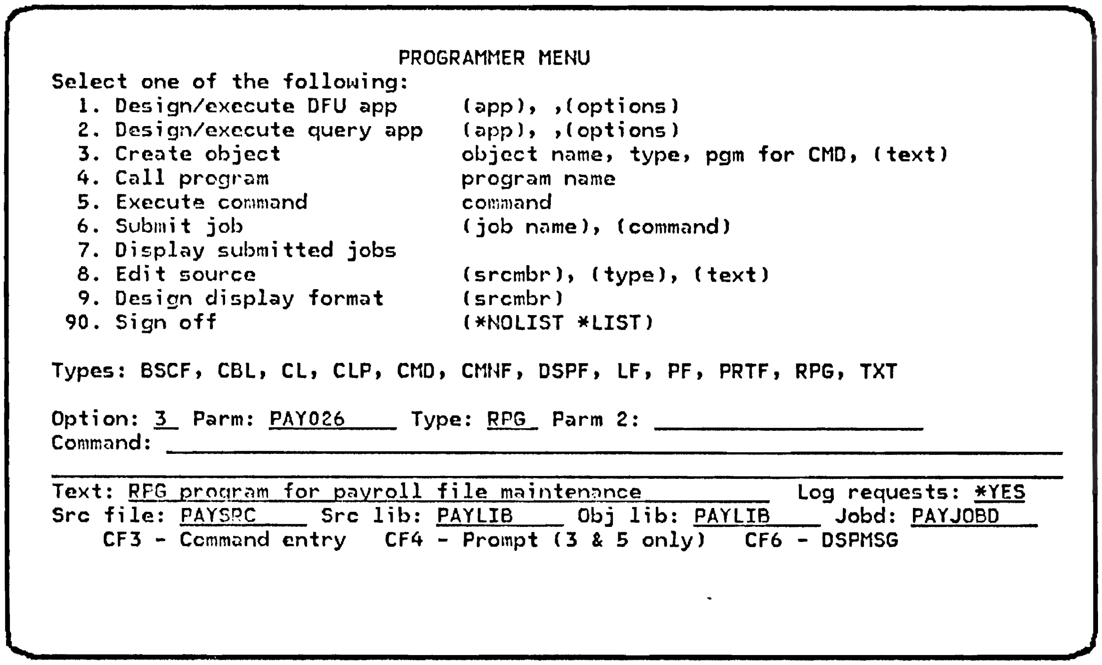
Ignored Input Field for Option 3

The Command input field is ignored for option 3.

Example of How to Use Option 3

The following display shows an example of how option 3 on the programmer menu is used.

To compile an RPG program *PAY026* enter:



```
                           PROGRAMMER MENU
Select one of the following:
  1. Design/execute DFU app     (app), ,(options)
  2. Design/execute query app   (app), ,(options)
  3. Create object              object name, type, pgm for CMD, (text)
  4. Call program               program name
  5. Execute command            command
  6. Submit job                 (job name), (command)
  7. Display submitted jobs
  8. Edit source                (srcmbr), (type), (text)
  9. Design display format      (srcmbr)
 90. Sign off                   (*NOLIST *LIST)

Types: BSCF, CBL, CL, CLP, CMD, CMNF, DSPF, LF, PF, PRTF, RPG, TXT

Option: 3_ Parm: PAY026____ Type: RPG_ Parm 2: ____________________
Command: ______________________________________________________________
_______________________________________________________________________
Text: RPG program for payroll file maintenance_________  Log requests: *YES
Src file: PAYSRC____ Src lib: PAYLIB____ Obj lib: PAYLIB____ Jobd: PAYJOBD__
    CF3 - Command entry   CF4 - Prompt (3 & 5 only)   CF6 - DSPMSG
```

Equivalent SBMJOB Command

```
SBMJOB  JOB(PAY026) JOBD(PAYJOBD) RQSDTA('CRTRPGPGM
           PGM(PAY026.PAYLIB)
           SRCFILE(PAYSRC.PAYLIB) SRCMBR(PAY026)
           TEXT('RPG program for payroll file maintenance')')
```

Option 4. Call Program

Select option 4 to call a program from the programmer menu. If you use option 4 to call a program, you cannot pass parameters to the program, the program must be in the library specified by the Obj lib input field, and you must wait until the program ends before you can enter another option from the programmer menu.

If you want to pass parameters to the program, you can select option 5 from the programmer menu, enter a CALL command, and specify the parameters to be passed.

If you want to call a program that is not in the library currently named in the Obj lib input field, you can enter that library name in the Obj lib field or enter a CALL command with option 5.

If you want to use your display station while the program executes, you can select option 6 to put the program on the job queue to be run as a separate batch job. To call the program, enter the CALL command in the Command input field.

Required Input Field for Option 4

The Parm input field is required for option 4 and must specify the name of the program to call.

Optional Input Field for Option 4

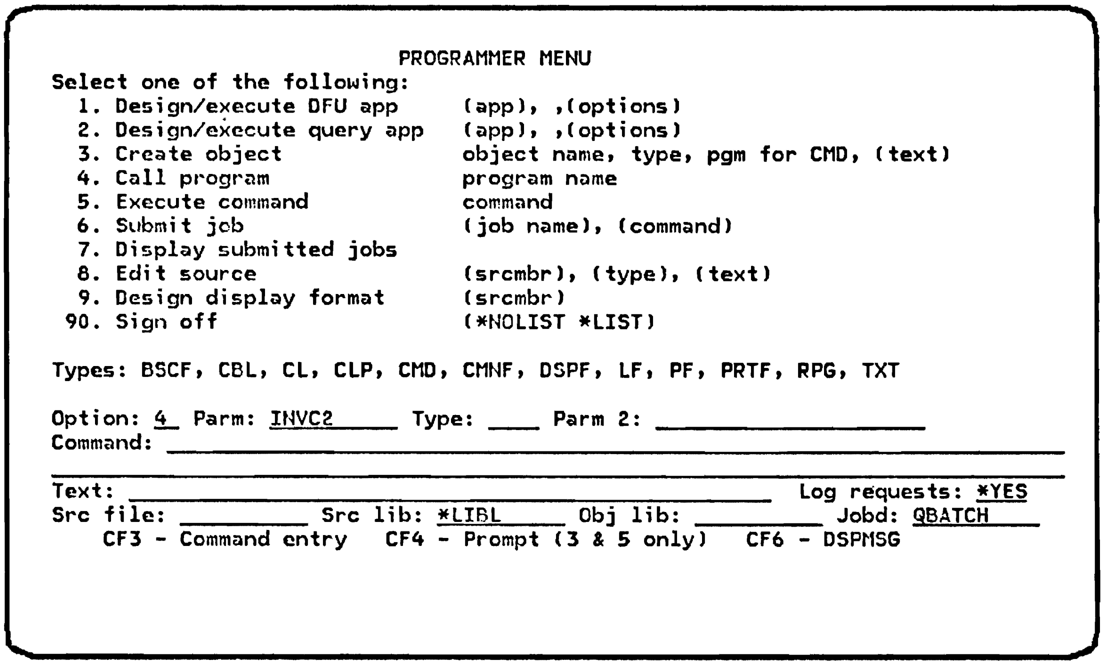
The Obj lib input field is optional for option 4 and can specify the name of the library that contains the program to call, or the field can be blank or *LIBL if the system can use the library list to find the program.

Ignored Input Fields for Option 4

The Type, Parm 2, Command, Text, Src file, Src lib, and Jobd input fields are ignored for option 4.

Example of How to Use Option 4

The following example shows how to call the program *INVC2*:



```
                           PROGRAMMER MENU
Select one of the following:
  1. Design/execute DFU app     (app), ,(options)
  2. Design/execute query app   (app), ,(options)
  3. Create object              object name, type, pgm for CMD, (text)
  4. Call program               program name
  5. Execute command            command
  6. Submit job                 (job name), (command)
  7. Display submitted jobs
  8. Edit source                (srcmbr), (type), (text)
  9. Design display format      (srcmbr)
 90. Sign off                   (*NOLIST *LIST)

Types: BSCF, CBL, CL, CLP, CMD, CMNF, DSPF, LF, PF, PRTF, RPG, TXT

Option: 4_ Parm: INVC2_____ Type: ____ Parm 2: _____________________
Command: ______________________________________________________________
_______________________________________________________________________
Text: __________________________________________________ Log requests: *YES
Src file: __________ Src lib: *LIBL_____ Obj lib: __________ Jobd: QBATCH____
    CF3 - Command entry   CF4 - Prompt (3 & 5 only)   CF6 - DSPMSG
```

Equivalent CALL Command

CALL PGM(INVC2.*LIBL)

Option 5. Execute Command

Select option 5 to execute a control language command. This option is useful because it allows you to enter a command from the programmer menu, which would otherwise require switching to the command entry display.

Required Input Field for Option 5

The command input field is required for option 5. Any interactive control language command that can be entered on the command entry display can be entered in the command input field.

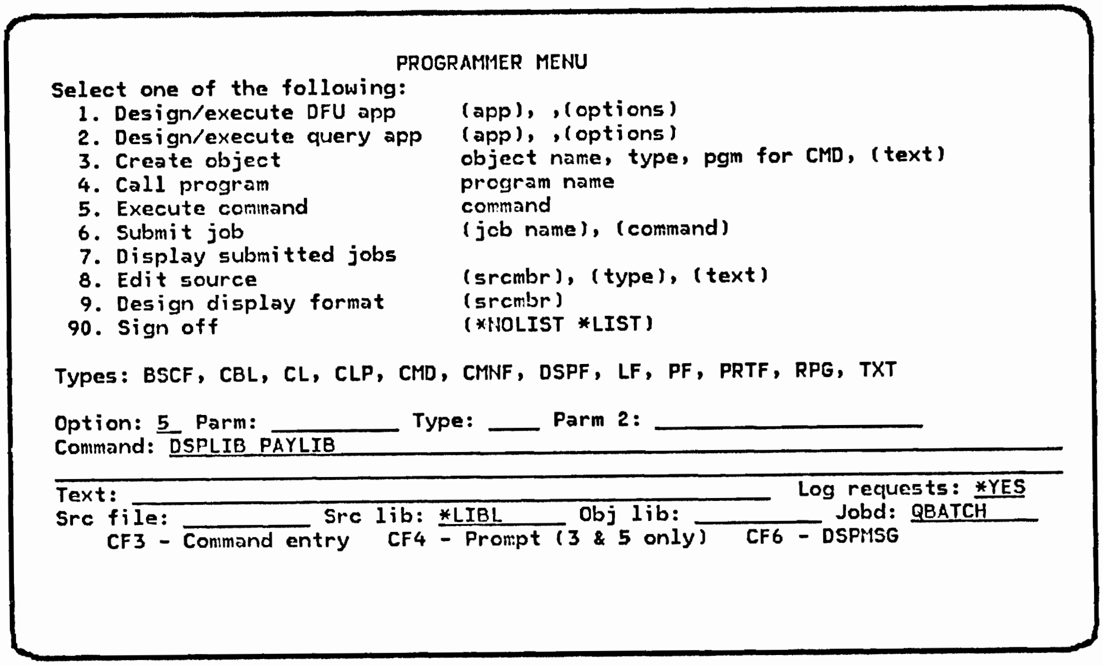
You can obtain prompting help for the command by pressing the CF4 key. Refer to the description of *Special Prompting Functions* earlier in this chapter for further information.

Ignored Input Fields for Option 5

The Parm, Type, Parm 2, Text, Src file, Src lib, Obj lib, and Jobd input fields are ignored for option 5.

Examples of How to Use Option 5

The following example shows how you would enter a Display Library (DSPLIB) command from the programmer menu.



```
                              PROGRAMMER MENU
Select one of the following:
  1. Design/execute DFU app      (app), ,(options)
  2. Design/execute query app    (app), ,(options)
  3. Create object               object name, type, pgm for CMD, (text)
  4. Call program                program name
  5. Execute command             command
  6. Submit job                  (job name), (command)
  7. Display submitted jobs
  8. Edit source                 (srcmbr), (type), (text)
  9. Design display format       (srcmbr)
 90. Sign off                    (*NOLIST *LIST)

Types: BSCF, CBL, CL, CLP, CMD, CMNF, DSPF, LF, PF, PRTF, RPG, TXT

Option: 5_ Parm: __________ Type: ____ Parm 2: _____________________
Command: DSPLIB PAYLIB___________________________________________________
_________________________________________________________________________
Text: ______________________________________________ Log requests: *YES
Src file: __________ Src lib: *LIBL_____ Obj lib: __________ Jobd: QBATCH____
    CF3 - Command entry   CF4 - Prompt (3 & 5 only)   CF6 - DSPMSG
```

Option 6. Submit Job

Select option 6 on the programmer menu to put a job on the job queue. Putting your batch jobs on the job queue helps improve response times for interactive functions and frees your work station from the submitted job so that you can use it while the job executes. You can display information about the jobs by using option 7 on the programmer menu.

Option 6 results in a Submit Job (SBMJOB) command being generated. Although there are no required entries for option 6, you usually fill in the Command field also. The command that you specify for option 6 is used for the RQSDTA parameter in the SBMJOB command. The job name is determined by what is specified in the Parm input field. The user name of your current interactive job is used for the user of the submitted job. Other job attributes are taken from the job description.

Notice that you do not have direct use of the prompting facilities. However, if prompting is needed, you can select option 5, plus the command you want, and press the CF4 key. On the display that you receive, fill in the prompts and press the CF1 key. You will receive a message that the command was canceled and the generated command is shown in the command input field. Enter option 6 to submit the command as a batch job.

Optional Input Fields for Option 6

The Parm and Command input fields are optional for option 6.

Parm Specify a name for the job. If you leave the parm field blank, the job description name is used for the job name. The user qualification of the job name is the user name of the current interactive job.

Command Specify any control language command that can be executed in a batch job. If you do not specify a command in the Command field, and one has already been specified in the job description, the job description request data is used.

The Jobd input field is used to specify a job description. A job description contains parameters to help the system control how the batch job is done. The following parameters in a job description arė important when you use option 6:

INLLIBL	Determines the initial library list.
JOBQ	Determines the job queue on which to put the batch job.
OUTQ	Determines the output queue used for the job listing and job log.
RQSDTA	Determines the function to be executed if no command is specified.

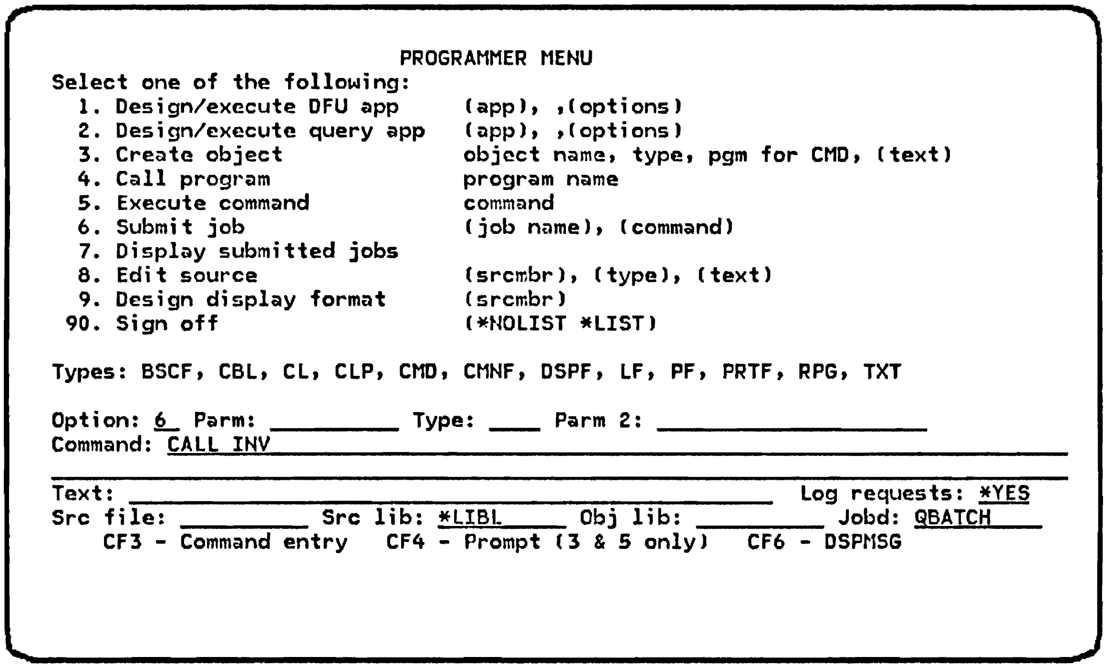
The job description that you specify must be found by use of the current library list. The default job description is QBATCH. If the jobs that you submit use objects in a library other than QGPL, you should consider using your own job description rather than QBATCH. Refer to the *CPF Programmer's Guide* for information about a job description.

Ignored Input Fields for Option 6

The Type, Parm 2, Text, Src file, Src lib, and Obj lib input fields are ignored for option 6.

Example of How to Use Option 6

The following display shows how the entries on the menu are used to submit a batch job that executes the program *INV*.



```
                            PROGRAMMER MENU
Select one of the following:
  1. Design/execute DFU app     (app), ,(options)
  2. Design/execute query app   (app), ,(options)
  3. Create object              object name, type, pgm for CMD, (text)
  4. Call program               program name
  5. Execute command            command
  6. Submit job                 (job name), (command)
  7. Display submitted jobs
  8. Edit source                (srcmbr), (type), (text)
  9. Design display format      (srcmbr)
 90. Sign off                   (*NOLIST *LIST)

Types: BSCF, CBL, CL, CLP, CMD, CMNF, DSPF, LF, PF, PRTF, RPG, TXT

Option: 6_ Parm: __________ Type: ____ Parm 2: _____________________
Command: CALL INV__________________________________________________________
___________________________________________________________________________
Text: __________________________________________________ Log requests: *YES
Src file: __________ Src lib: *LIBL_____ Obj lib: __________ Jobd: QBATCH____
    CF3 - Command entry   CF4 - Prompt (3 & 5 only)   CF6 - DSPMSG
```

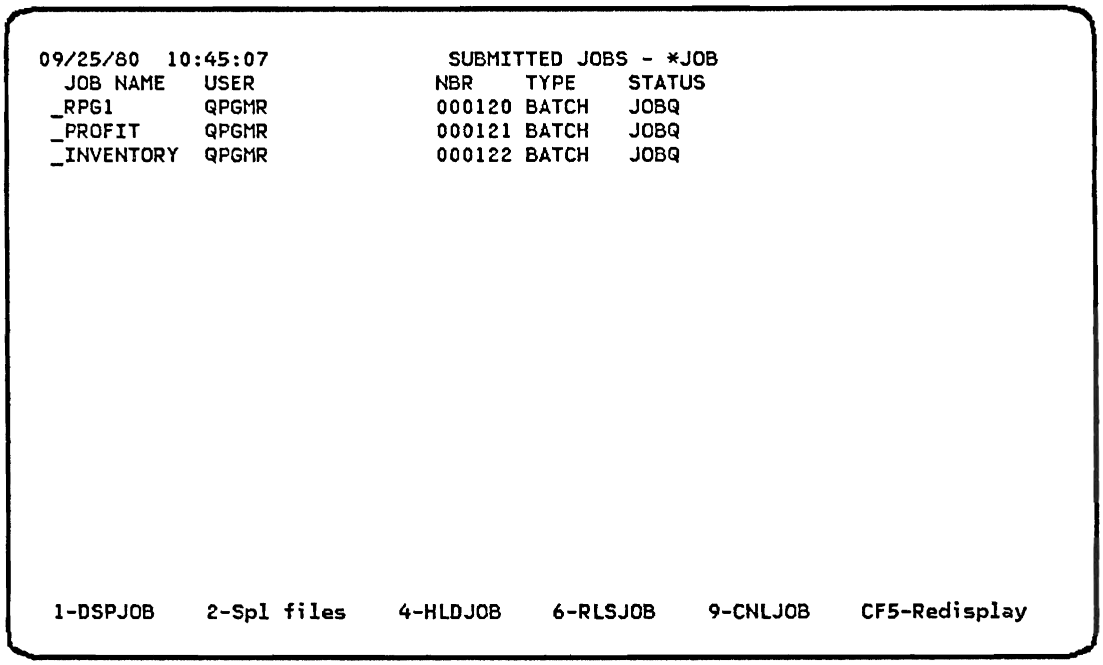
Equivalent SBMJOB Command

```
SBMJOB   JOB(QBATCH) JOBD(QBATCH.*LIBL) RQSDTA('CALL INV')
```

Option 7. Display Submitted Jobs

Select option 7 to display information about jobs that have been placed on the job queue from the programmer menu during your work session. Jobs can be submitted from the programmer menu by option 3 (Create Object) or option 6 (Submit Job). If you sign off, your association with these jobs through the session job number is lost, and you cannot use option 7 to display them in a subsequent session.

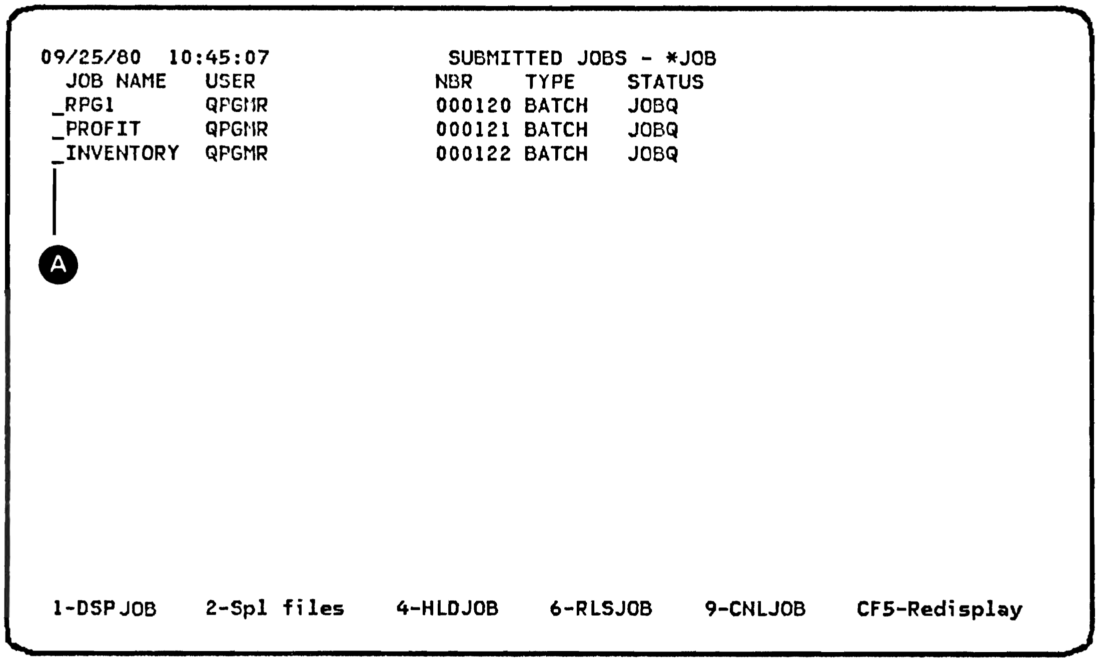
The following display is an example of the information shown by option 7:



```
09/25/80  10:45:07              SUBMITTED JOBS - *JOB
  JOB NAME   USER                 NBR    TYPE    STATUS
 _RPG1       QPGMR                000120 BATCH   JOBQ
 _PROFIT     QPGMR                000121 BATCH   JOBQ
 _INVENTORY  QPGMR                000122 BATCH   JOBQ




  1-DSPJOB    2-Spl files    4-HLDJOB    6-RLSJOB    9-CNLJOB    CF5-Redisplay
```

If you want to display the submitted jobs in a subsequent session, use the DSPSBMJOB command. See *Command Descriptions* in the *CL Reference Manual.*

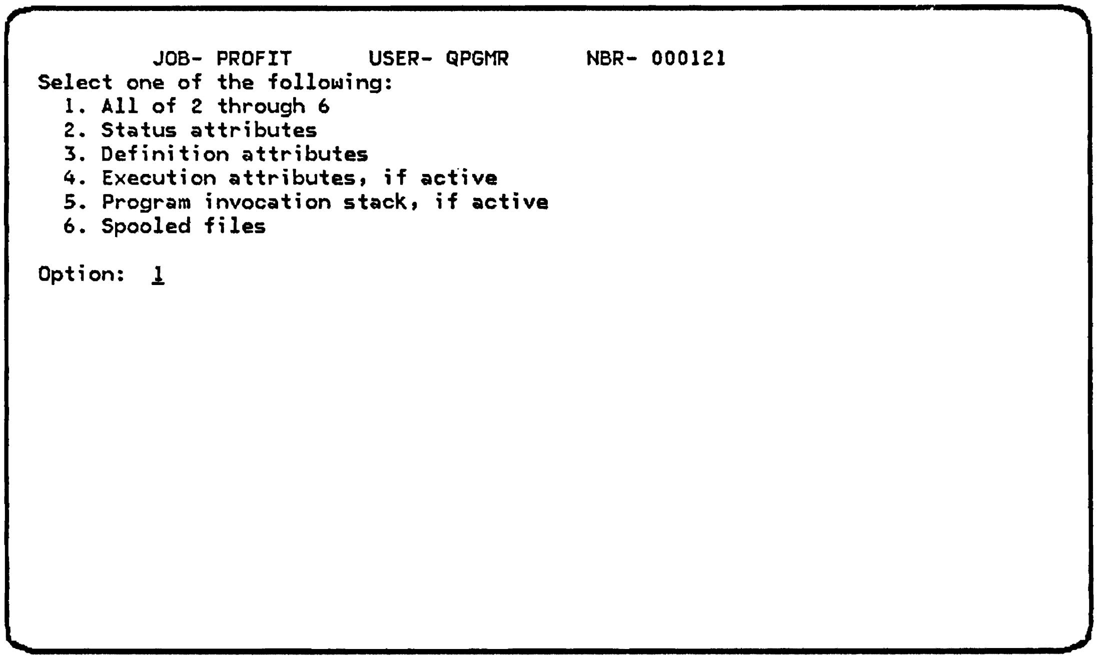
Each job name on the display is preceded by a one-position input field Ⓐ.



```
09/25/80  10:45:07               SUBMITTED JOBS - *JOB
  JOB NAME   USER                NBR    TYPE    STATUS
 _RPG1       QPGMR               000120 BATCH   JOBQ
 _PROFIT     QPGMR               000121 BATCH   JOBQ
 _INVENTORY  QPGMR               000122 BATCH   JOBQ








 1-DSPJOB    2-Spl files    4-HLDJOB    6-RLSJOB    9-CNLJOB    CF5-Redisplay
```

You can enter a 1, 2, 4, 6, or 9 in the input field to request the following functions:

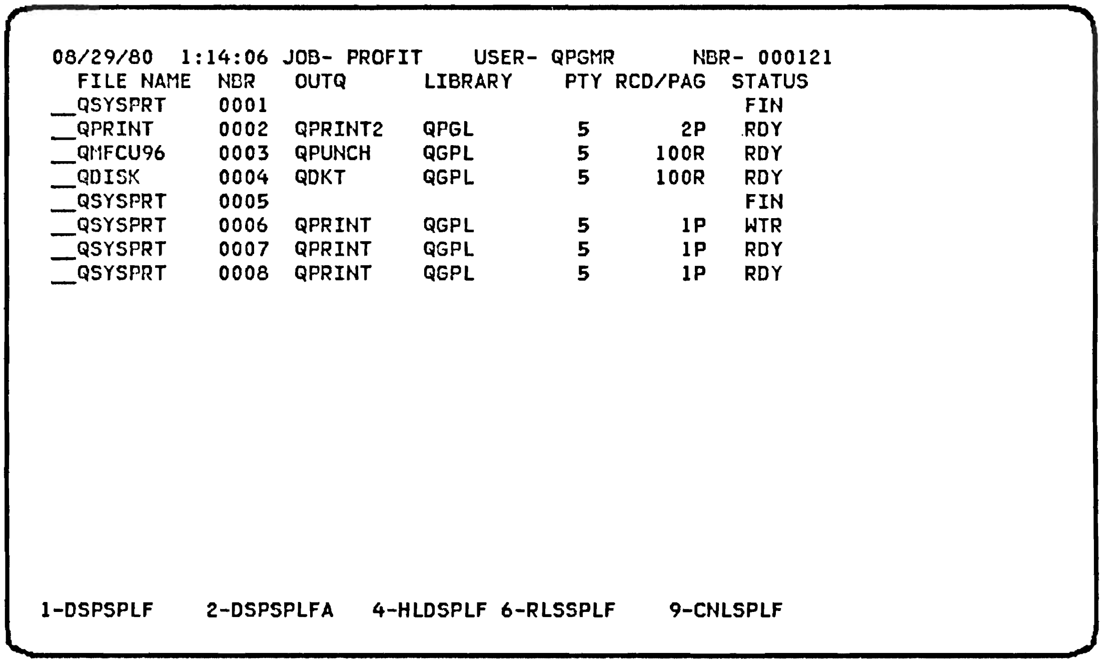
1 Specify 1 to obtain the job display menu for the job.



```
        JOB- PROFIT      USER- QPGMR      NBR- 000121
Select one of the following:
  1. All of 2 through 6
  2. Status attributes
  3. Definition attributes
  4. Execution attributes, if active
  5. Program invocation stack, if active
  6. Spooled files

Option:  1
```

Refer to the DSPJOB command in the *CL Reference Manual* for more information about this display.

2 Specify 2 to obtain spool file information about a job. The following display appears. (Refer to the DSPJOB command in the *CL Reference Manual* for information about this display.)



```
08/29/80  1:14:06 JOB- PROFIT    USER- QPGMR      NBR- 000121
  FILE NAME  NBR   OUTQ       LIBRARY    PTY RCD/PAG  STATUS
__QSYSPRT    0001                                      FIN
__QPRINT     0002  QPRINT2    QPGL        5       2P  .RDY
__QMFCU96    0003  QPUNCH     QGPL        5     100R   RDY
__QDISK      0004  QDKT       QGPL        5     100R   RDY
__QSYSPRT    0005                                      FIN
__QSYSPRT    0006  QPRINT     QGPL        5       1P   WTR
__QSYSPRT    0007  QPRINT     QGPL        5       1P   RDY
__QSYSPRT    0008  QPRINT     QGPL        5       1P   RDY









1-DSPSPLF    2-DSPSPLFA   4-HLDSPLF 6-RLSSPLF    9-CNLSPLF
```

4 Specify 4 to hold the job. The spool files produced by the job will not be held.

6 Specify 6 to release the job. The job must have been previously held.

9 Specify 9 to cancel the job. The job log and any spool files produced by that job are not canceled.

Ignored Input Fields for Option 7

All input fields on the programmer menu are ignored for option 7.

Option 8. Edit Source

Select option 8 on the programmer menu to use the source entry utility (SEU) to edit (create or change) a source member in an existing source file. Option 8 results in an Edit Source (EDTSRC) command being generated.

There are no required input fields for option 8.

Optional Input Fields for Option 8

The Parm, Type, Text, Src file, and Src lib input fields are optional for option 8.

Parm Specify the name of the source member to be edited. If this input field is left blank, SEU will present a member selection list. From this list a member to be changed can be selected or the name of a member to be created can be entered.

Type Specify the type of source that you want to edit. The type also results in a default source file being used if the Src file input field is blank. The following list shows the default source files:

Type Specified	**Requested SEU Function**	**Default Source File**
BSCF	Edit DDS for binary synchronous communications file	QDDSSRC
CBL	Edit COBOL source statements	QCBLSRC
CL	Edit control language statements	QCLSRC
CLP	Edit control language statements for a control language program	QCLSRC
CMD	Edit a command definition	QCMDSRC
CMNF	Edit DDS for a communications file	QDDSSRC
DFU	Edit UDS for a DFU definition	QUDSSRC
DSPF	Edit DDS for a display file	QDDSSRC
LF	Edit DDS for a logical file	QDDSSRC
PF	Edit DDS for a physical file	QDDSSRC
PRTF	Edit DDS for a printer file	QDDSSRC
QRY	Edit UDS for a query definition	QUDSSRC
RPG	Edit RPG source statements	QRPGSRC
TXT	Edit text statements	QTXTSRC

If no type is specifed, SEU uses the same type used when the member was last edited. For a new member, the type defaults to the type related to the source file being used; if both the Type field and Src file field are blank, source file QTXTSRC and type TXT are used.

Text Enter up to 50 characters of text to describe the member that is being created or changed. If the text field is blank when a member is being changed, the member text will not be changed.

Src file Specify the name of the source file that contains or will contain the member to be edited. This input field can be left blank if you wish to use a default source file, which will be determined by the type specification entered in the Type input field. If both the Src file field and the Type input field are blank, the source file QTXTSRC is used.

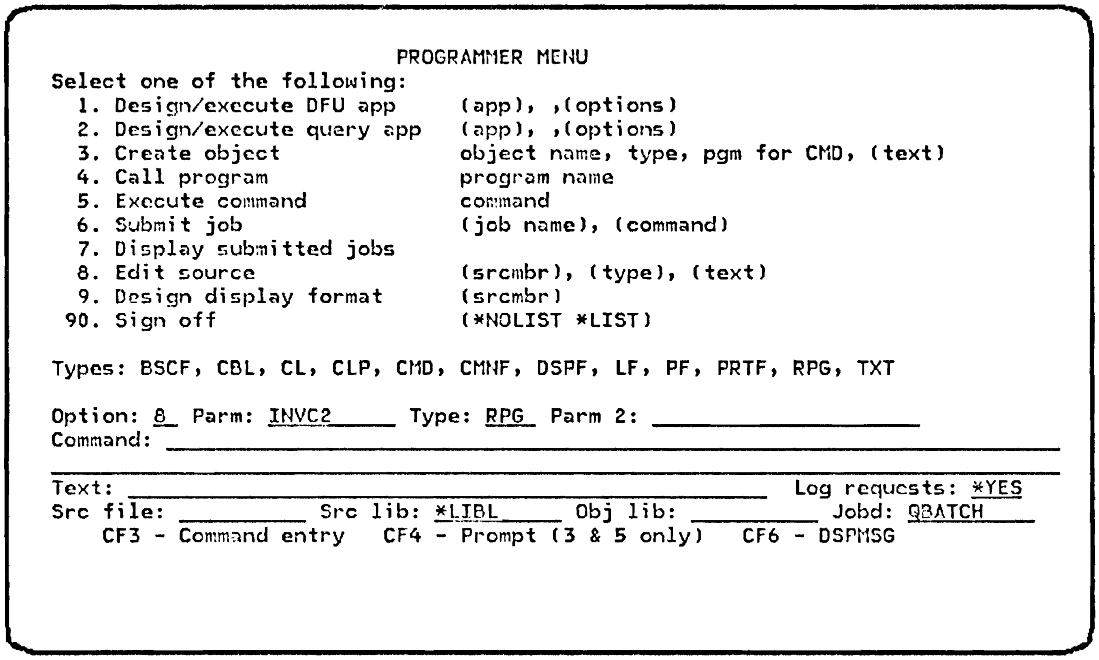
Src lib Specify the name of the library in which the source file exists. If this input field is blank or contains *LIBL (the initial value when the programmer menu is invoked), your library list will be searched to find the default or specified source file.

Ignored Input Fields for Option 8

The Parm 2, Command, Obj lib, and Jobd input fields are ignored for option 8.

Example of How to Use Option 8

The following display shows how the entries on the programmer menu are used to edit the RPG source for the *INVC2* program in the *QRPGSRC* file.



```
                            PROGRAMMER MENU
Select one of the following:
  1. Design/execute DFU app      (app), ,(options)
  2. Design/execute query app    (app), ,(options)
  3. Create object               object name, type, pgm for CMD, (text)
  4. Call program                program name
  5. Execute command             command
  6. Submit job                  (job name), (command)
  7. Display submitted jobs
  8. Edit source                 (srcmbr), (type), (text)
  9. Design display format       (srcmbr)
 90. Sign off                    (*NOLIST *LIST)

Types: BSCF, CBL, CL, CLP, CMD, CMNF, DSPF, LF, PF, PRTF, RPG, TXT

Option: 8_ Parm: INVC2_____ Type: RPG_ Parm 2: ____________________
Command: _____________________________________________________________
______________________________________________________________________
Text: __________________________________________________ Log requests: *YES
Src file: __________ Src lib: *LIBL_____ Obj lib: __________ Jobd: QBATCH____
    CF3 - Command entry   CF4 - Prompt (3 & 5 only)   CF6 - DSPMSG
```

Equivalent EDTSRC Command

EDTSRC   SRCFILE(QRPGSRC.*LIBL) SRCMBR(INVC2) TYPE(*RPG)

Option 9. Design Display Format

Select option 9 on the programmer menu to use the screen design aid (SDA) to create, change, or test a display record format.

There are no required input fields for option 9.

Optional Input Fields for Option 9

The Parm, Src file, Src lib, Obj lib, and Jobd input fields are optional for option 9.

Parm Specify the name of a new or existing source member that contains, or will contain, the DDS for the display formats to be created, updated or tested by SDA. If this input field is left blank, the source member name is not displayed on the design record format menu. You may enter a source member name when the design record format menu is displayed; or, if you do not enter a source member name, SDA displays a list of all members in the source file.

Src file Specify the name of an existing DDS source file which contains, or will contain, the source member to be used by SDA. If this input field is left blank when creating a display device file, SDA will assume a value of QDDSSRC for the DDS source file prompt on the design record format menu. You may change this value on the design record format menu. When creating a menu, whether you key in a source file name or leave this parameter blank, SDA assumes a value of QCLSRC for the CL source file name on the initial menu definition display. You may change this value on the initial menu definition display.

Src lib Specify the name of the library in which the source file exists. If this input field is blank or contains *LIBL (the initial value when the programmer menu is invoked), SDA will assume a value of *LIBL for the library prompt on the design record format menu. You may change this value when the design record format menu is displayed.

Obj lib Specify the name of the library in which programs and display files created by SDA will be stored. If this input field is left blank, SDA will assume a value of QGPL for the library prompt on the DDS save/create device file menu. You may change this value when the DDS save/create device file menu is displayed.

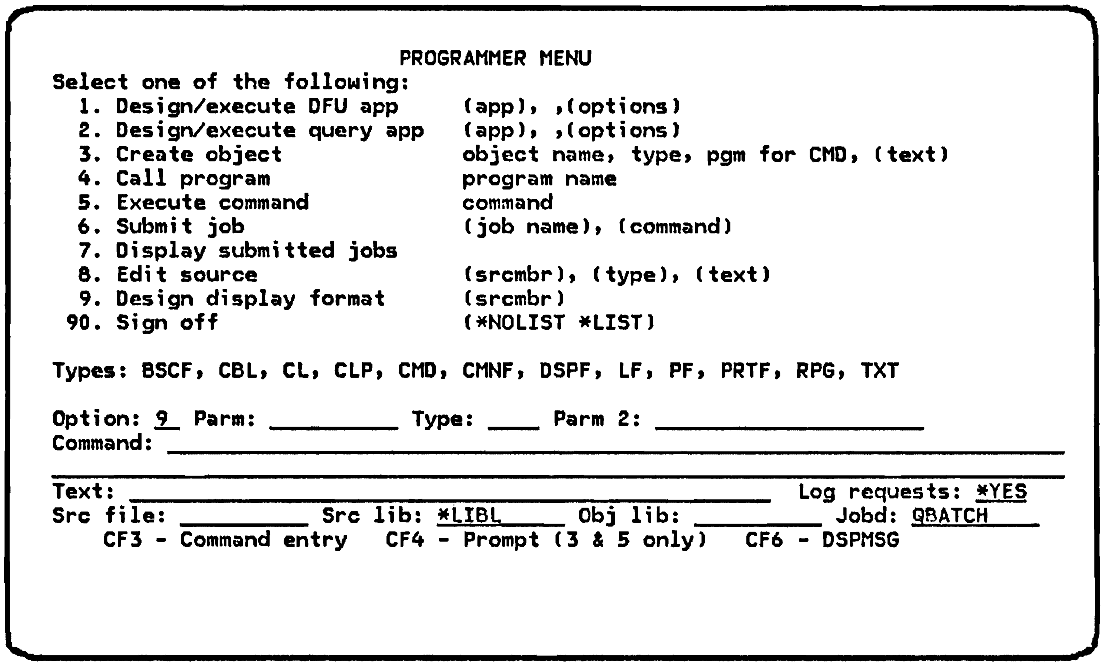
Jobd Specify the name of the job description to be used with jobs being submitted by SDA for execution in batch. If this input field is left blank, SDA will assume a value of QBATCH for the job description prompt on the DDS save/create device file menu. You may change this value when the DDS save/create device file menu is displayed.

Ignored Input Fields for Option 9

The Type, Parm 2, Command, and Text input fields are ignored by option 9.

Example of How to Use Option 9

The following display shows how the entries on the programmer menu are used to display the SDA option menu.



```
                          PROGRAMMER MENU
Select one of the following:
  1. Design/execute DFU app     (app), ,(options)
  2. Design/execute query app   (app), ,(options)
  3. Create object              object name, type, pgm for CMD, (text)
  4. Call program               program name
  5. Execute command            command
  6. Submit job                 (job name), (command)
  7. Display submitted jobs
  8. Edit source                (srcmbr), (type), (text)
  9. Design display format      (srcmbr)
 90. Sign off                   (*NOLIST *LIST)

Types: BSCF, CBL, CL, CLP, CMD, CMNF, DSPF, LF, PF, PRTF, RPG, TXT

Option: 9_ Parm: __________ Type: ____ Parm 2: _____________________
Command: ____________________________________________________________________
__________________________________________________________________________________
Text: __________________________________________________  Log requests: *YES
Src file: __________ Src lib: *LIBL_____ Obj lib: __________ Jobd: QBATCH____
    CF3 - Command entry   CF4 - Prompt (3 & 5 only)   CF6 - DSPMSG
```

Option 90. Sign Off

Select option 90 on the programmer menu to end your job at a work station.

There are no required input fields for option 90 although the Parm field must be blank if one of the values listed below is not selected.

Optional Input Fields for Option 90

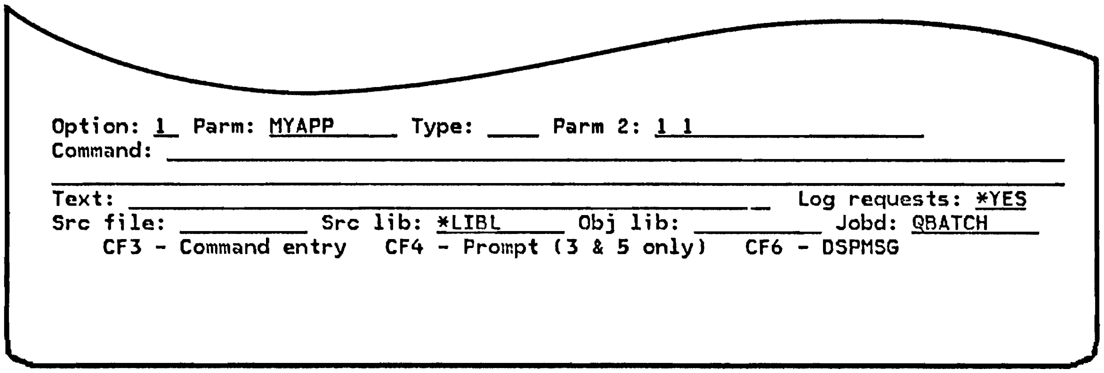
Parm   If you specify *NOLIST, the job log will not be printed. This is the default if the field is left blank.

If you specify *LIST, the job log will be printed.

How to Use Log Requests Field on the Programmer's Menu

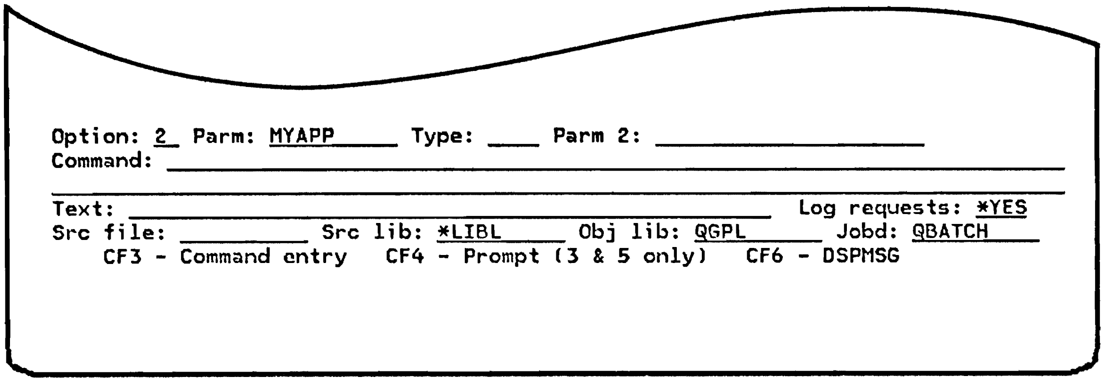
The log requests field will optionally place a request message (which consists of parameters you keyed on the menu) on the job log. From the programmer's menu you can view the messages that go into the job log by pressing the CF3 key. (Roll keys are valid if the messages do not fit on one screen.) Valid entries for the log requests field are *YES and *NO.

The following examples show various options and request messages that could occur if *YES is specified. If *NO is specified, these request messages are not sent to the job log.



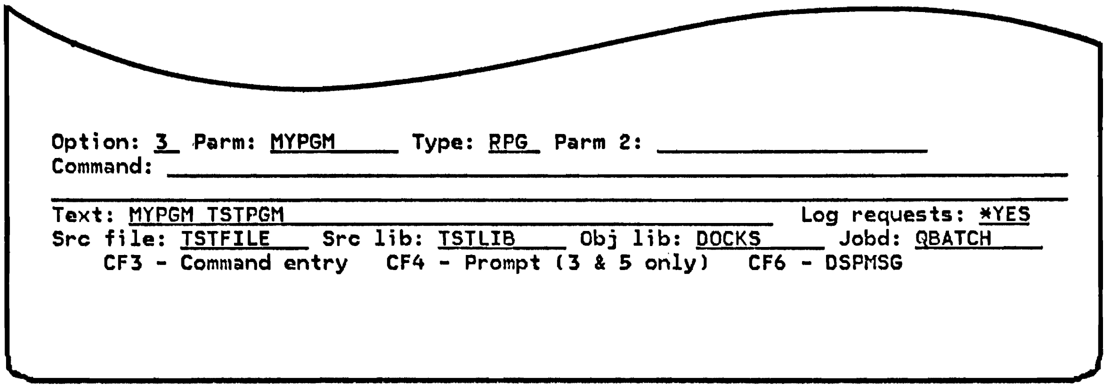
```
Option: 1_ Parm: MYAPP_____ Type: ____ Parm 2: 1 1________________
Command: ____________________________________________________________
_____________________________________________________________________
Text: _________________________________________ Log requests: *YES
Src file: __________ Src lib: *LIBL_____ Obj lib: __________ Jobd: QBATCH____
    CF3 - Command entry   CF4 - Prompt (3 & 5 only)   CF6 - DSPMSG
```

Option 1. DSNDFUAPP   Parm-MYAPP Parm2-1 1



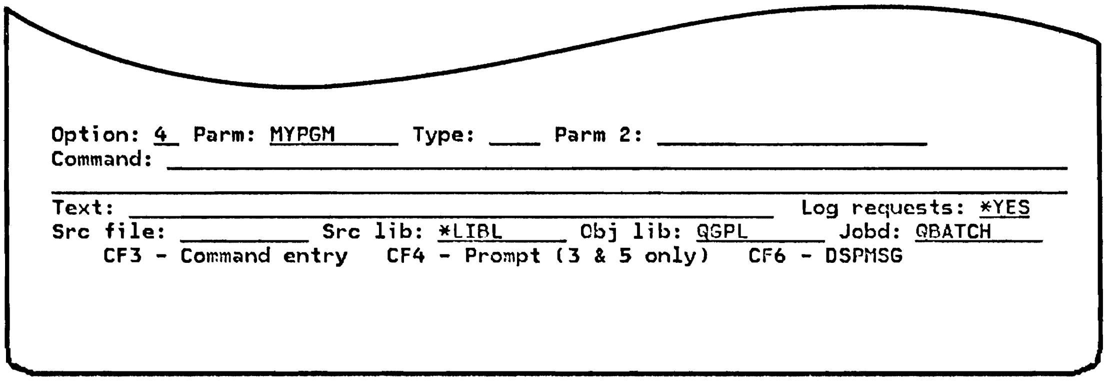
```
Option: 2_ Parm: MYAPP_____ Type: ____ Parm 2: ____________________
Command: ____________________________________________________________
_____________________________________________________________________
Text: _________________________________________ Log requests: *YES
Src file: __________ Src lib: *LIBL_____ Obj lib: QGPL______ Jobd: QBATCH____
    CF3 - Command entry   CF4 - Prompt (3 & 5 only)   CF6 - DSPMSG
```

Option 2. DSNQRYAPP   Parm-MYAPP Obj lib-QGPL



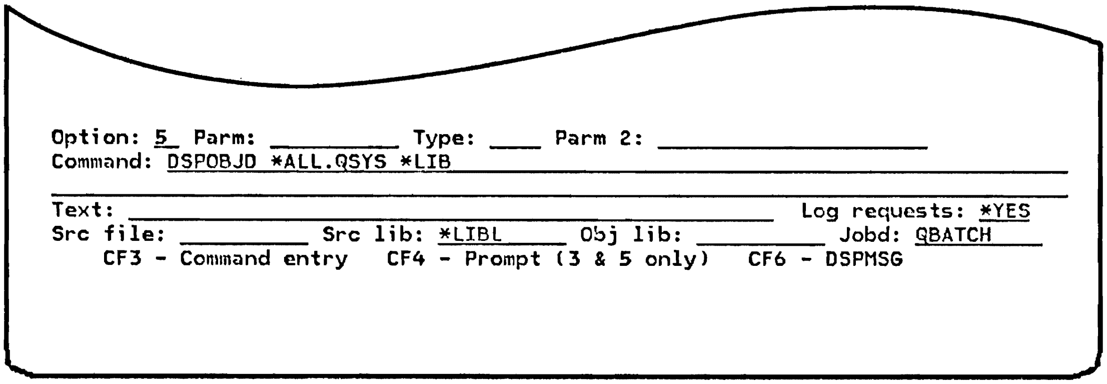
```
Option: 3_ Parm: MYPGM_____  Type: RPG_ Parm 2: ____________________
Command: __________________________________________________________
___________________________________________________________________
Text: MYPGM TSTPGM_____________________________  Log requests: *YES
Src file: TSTFILE___ Src lib: TSTLIB____ Obj lib: DOCKS____ Jobd: QBATCH___
    CF3 - Command entry   CF4 - Prompt (3 & 5 only)   CF6 - DSPMSG
```

Option 3. Crt obj   Parm-MYPGM Type-RPG Text-MYPGM TSTPGM
Src file-TSTFILE Src lib-TSTLIB Obj lib-DOCKS Jobd-QBATCH



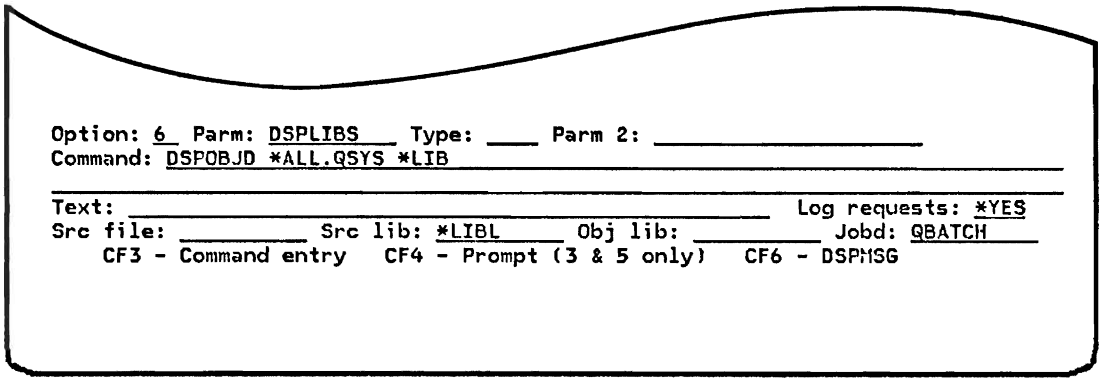
```
Option: 4_ Parm: MYPGM_____  Type: ____ Parm 2: ____________________
Command: __________________________________________________________
___________________________________________________________________
Text: _________________________________________  Log requests: *YES
Src file: __________ Src lib: *LIBL_____ Obj lib: QGPL______ Jobd: QBATCH____
    CF3 - Command entry   CF4 - Prompt (3 & 5 only)   CF6 - DSPMSG
```

Option 4. CALL   Parm-MYPGM Obj lib-QGPL



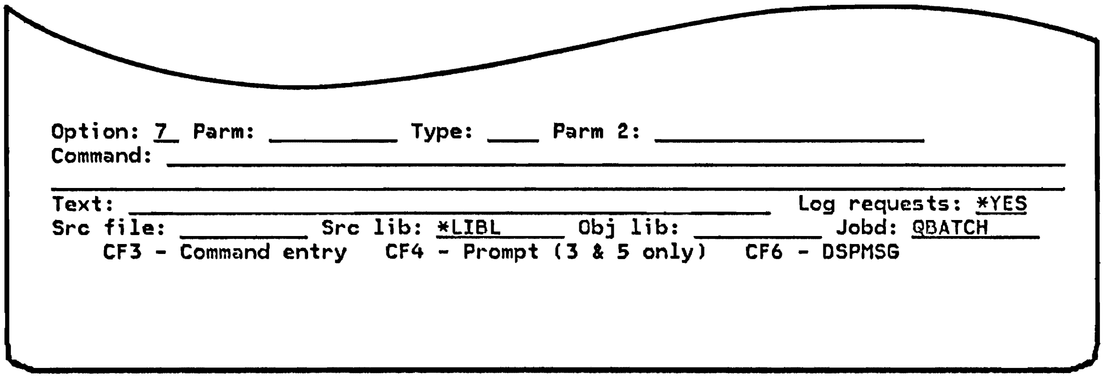
```
Option: 5_ Parm: __________  Type: ____ Parm 2: ____________________
Command: DSPOBJD *ALL.QSYS *LIB____________________________________
___________________________________________________________________
Text: _________________________________________  Log requests: *YES
Src file: __________ Src lib: *LIBL_____ Obj lib: __________ Jobd: QBATCH____
    CF3 - Command entry   CF4 - Prompt (3 & 5 only)   CF6 - DSPMSG
```

Option 5. Execute
DSPOBJD *ALL.QSYS *LIB



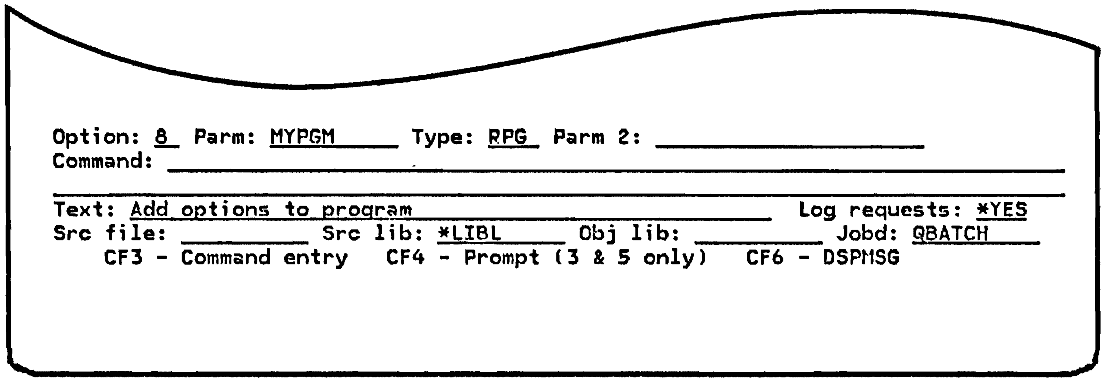
```
Option: 6  Parm: DSPLIBS    Type: ____  Parm 2: ____________________
Command: DSPOBJD *ALL.QSYS *LIB
______________________________________________________________________
Text: __________________________________________________  Log requests: *YES
Src file: __________  Src lib: *LIBL      Obj lib:            Jobd: QBATCH
    CF3 - Command entry   CF4 - Prompt (3 & 5 only)   CF6 - DSPMSG
```

Option 6. SBMJOB Parm-DSPLIBS Command-DSPOBJD *ALL.QSYS *LIB Jobd-QBATCH



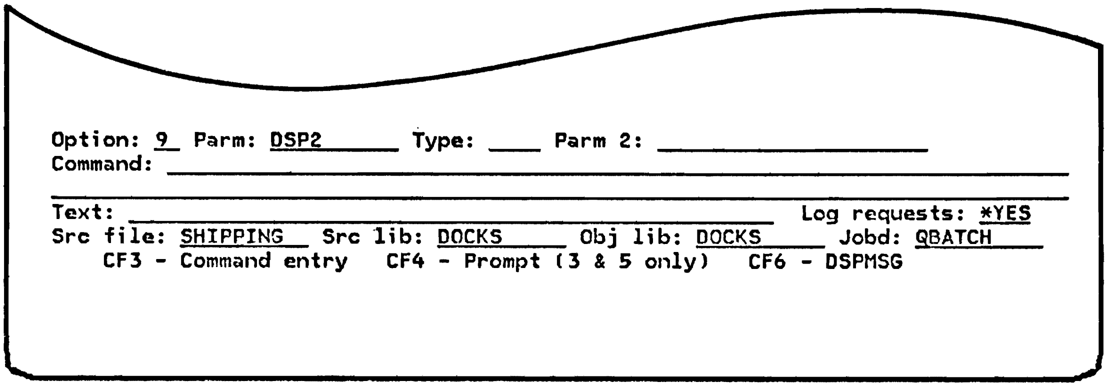
```
Option: 7  Parm: __________  Type: ____  Parm 2: ____________________
Command: ____________________________________________________________
______________________________________________________________________
Text: __________________________________________________  Log requests: *YES
Src file: __________  Src lib: *LIBL      Obj lib: __________  Jobd: QBATCH
    CF3 - Command entry   CF4 - Prompt (3 & 5 only)   CF6 - DSPMSG
```

Option 7. DSPSBMJOB



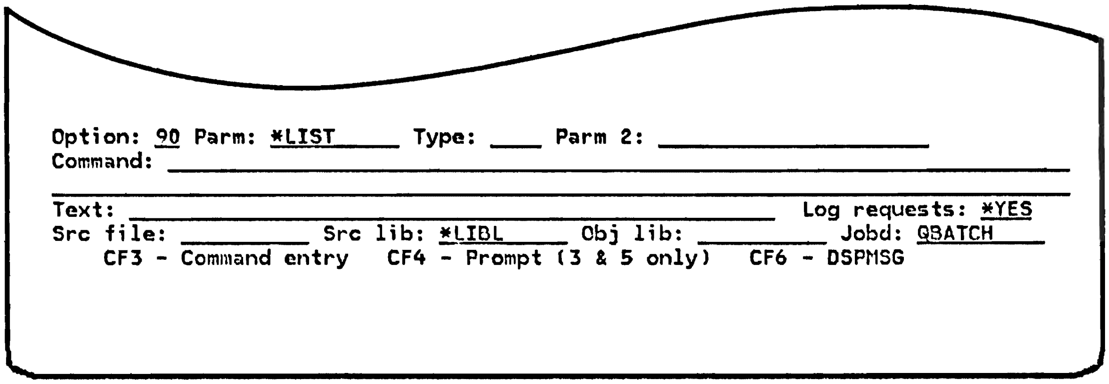
```
Option: 8  Parm: MYPGM      Type: RPG  Parm 2: ____________________
Command: ____________________________________________________________
______________________________________________________________________
Text: Add options to program                              Log requests: *YES
Src file: __________  Src lib: *LIBL      Obj lib: __________  Jobd: QBATCH
    CF3 - Command entry   CF4 - Prompt (3 & 5 only)   CF6 - DSPMSG
```

Option 8. EDTSRC Parm-MYPGM Type-RPG Text-Add options to program Src lib-*LIBL



```
Option: 9_ Parm: DSP2______ Type: ____ Parm 2: _____________________
Command: ____________________________________________________________
_____________________________________________________________________
Text: ________________________________________ Log requests: *YES
Src file: SHIPPING__ Src lib: DOCKS_____ Obj lib: DOCKS_____ Jobd: QBATCH____
    CF3 - Command entry   CF4 - Prompt (3 & 5 only)   CF6 - DSPMSG
```

Option 9. DSNFMT   Parm-DSP2 Src file-SHIPPING Src lib-DOCKS
Obj lib-DOCKS Jobd-QBATCH



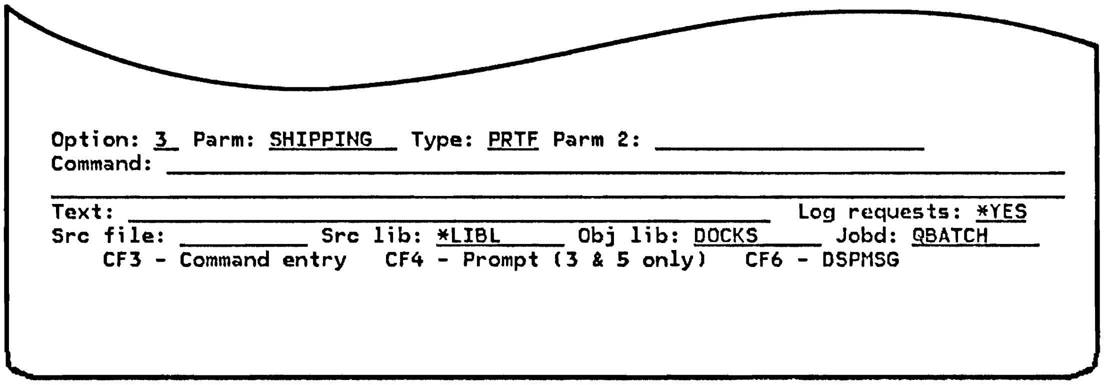
```
Option: 90 Parm: *LIST_____ Type: ____ Parm 2: _____________________
Command: ____________________________________________________________
_____________________________________________________________________
Text: ________________________________________ Log requests: *YES
Src file: __________ Src lib: *LIBL_____ Obj lib: __________ Jobd: QBATCH____
    CF3 - Command entry   CF4 - Prompt (3 & 5 only)   CF6 - DSPMSG
```

Option 90. SIGNOFF   Parm-*LIST

Prompting

If you press the CF4 key while using option 3, two request messages are placed on the job log if *YES is specified in the Log requests field. The first message has the option number and description; it also indicates whether the CF4, CF9, or CF11 keys were used (or jobd if the CF9 key was not used). The second message has the command string as the result of prompting.

In the following example you want to create a print file named SHIPPING in a library DOCKS (that already exists). After you key the parameters on the programmer's menu, press the CF4 key and change any other parameters (in this case, TEXT and COPIES). Press Enter/Rec Adv to execute the command. After the menu reappears with a message stating that an object already exists, press the CF11 key. This will interactively delete the object before proceeding with the create.

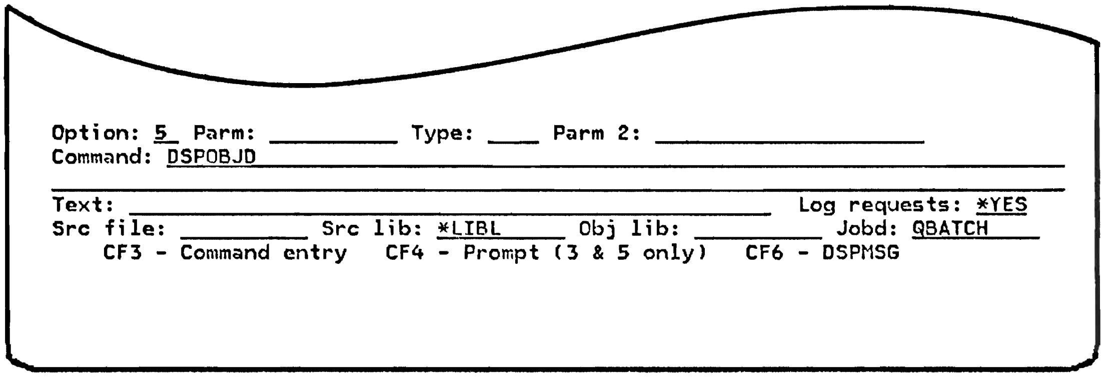


```
Option: 3_ Parm: SHIPPING__ Type: PRTF Parm 2: ____________________
Command: ____________________________________________________________
_____________________________________________________________________
Text: __________________________________________ Log requests: *YES
Src file: __________ Src lib: *LIBL_____ Obj lib: DOCKS____ Jobd: QBATCH____
    CF3 - Command entry   CF4 - Prompt (3 & 5 only)   CF6 - DSPMSG
```

```
Option 3.  Crt obj JOBD-QBATCH  CF4  CF11
CRTPRTF FILE(SHIPPING.DOCKS)  SRCFILE(QDDSSRC.*LIBL) COPIES(2)
TEXT('SHIPPING FILE FOR DOCKS')
```

Pressing the CF4 key while using option 5 also results in two messages being logged if *YES is specified. The first message has the option number and description. The second message is the command string as a result of prompting. This command string is also returned to the command line of the menu. (If the command string is longer than the command field, the last three characters on the command field are replaced by three periods.)

In the following example, you want to execute a command (DSPOBJD) but are not sure of the parameters to enter. Key in the command on the programmer's menu and press the CF4 key to enter the parameters.



```
Option: 5_ Parm: __________ Type: ____ Parm 2: ______________________
Command: DSPOBJD_____________________________________________________
_____________________________________________________________________
Text: __________________________________________ Log requests: *YES
Src file: __________ Src lib: *LIBL_____ Obj lib: __________ Jobd: QBATCH____
    CF3 - Command entry   CF4 - Prompt (3 & 5 only)   CF6 - DSPMSG
```

```
Option 5.  Execute
DSPOBJD OBJ(*ALL.QSYS) OBJTYPE(*ALL) DETAIL(*FULL)
```

This page is intentionally left blank.

HOW TO USE SYSTEM REQUEST

There may be occasions when you need to temporarily exit from your interactive job to perform other functions at the work station, such as display your messages, and then return to the original job. You can do this by using the system request function, which is available to anyone who is signed on the system at a work station. However, if you received the routing display after sign-on, the system request function cannot be requested until after valid routing data has been entered and another display appears (for example a menu or the command entry display).

The system request menu remains available to you until you sign off or your interactive job is terminated by some other means, such as when the system is powered down.

To get the system request menu, you use the following procedures:

1. Press and hold down the Shift key.

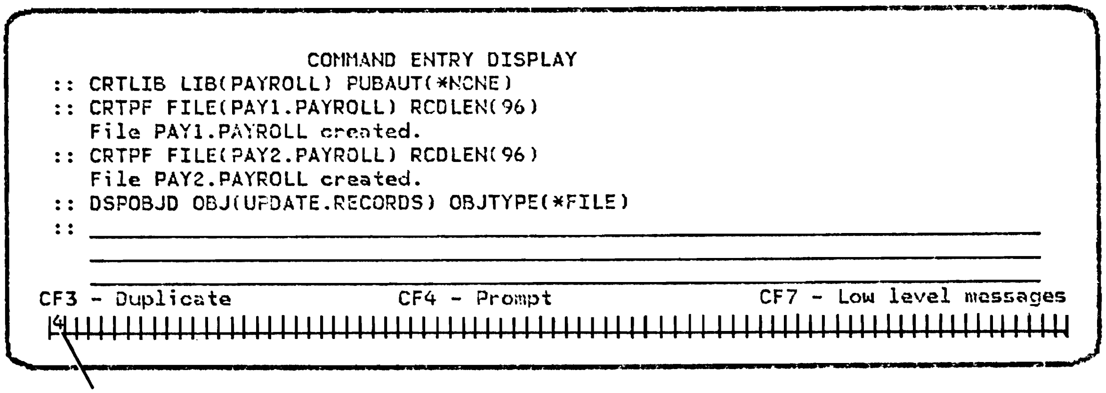
2. Press the Sys Req key. When you press the Sys Req key, the cursor is positioned on the bottom line of the screen, where vertical bars appear, and any messages on that line are cleared.

3. Press the Enter/Rec Adv key. (You can release the other two keys before pressing this key.)

Your current interactive job is suspended at the point where you left it, and you receive the system request menu (Figure 5-7).

If you want to bypass the system request menu, press and hold down the Shift key and press the Sys Req key. The vertical bars that appear at the bottom of the display can be used as an input field. Simply enter the desired option number from the system request menu, optionally followed by at least one blank, enter any parameters, and press the Enter/Rec Adv key. Your requested function will be performed and you will return to the display that you made your request from, or you will receive the system request menu with messages at the bottom of the display. You cannot bypass the system request menu from the console.

For example, if you want to display the messages from the work station message queue you would enter a 4 and press the Enter/Rec Adv key. The next display would be the message queue display.

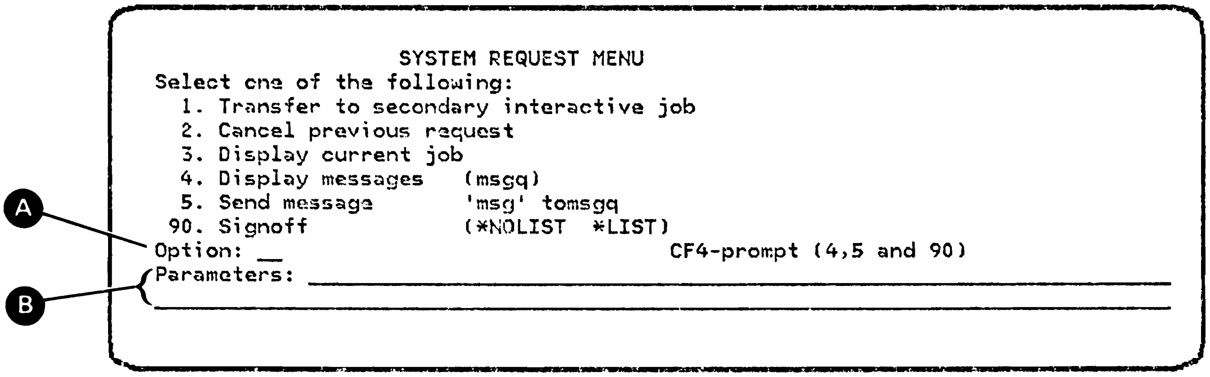


```
                        COMMAND ENTRY DISPLAY
 :: CRTLIB LIB(PAYROLL) PUBAUT(*NCNE)
 :: CRTPF FILE(PAY1.PAYROLL) RCDLEN(96)
    File PAY1.PAYROLL created.
 :: CRTPF FILE(PAY2.PAYROLL) RCDLEN(96)
    File PAY2.PAYROLL created.
 :: DSPOBJD OBJ(UPDATE.RECORDS) OBJTYPE(*FILE)
 :: ____________________________________________________________________
    ____________________________________________________________________
    ____________________________________________________________________
CF3 - Duplicate              CF4 - Prompt                CF7 - Low level messages
|4|||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||
```

Enter a 4 here.

The system request menu provides six options. A brief description of the function to be performed by each option is shown following the option number on the menu. For all options except option 5, you can select the function by keying the option number into the Option field Ⓐ and then pressing the Enter/Rec Adv key. The requested function is performed. If an optional parameter value is not entered for a given function (option), the default for the corresponding command's parameter is used. The default for the major parameter values is shown in parentheses next to the option description. For option 5, the two values that you *must* enter are indicated next to the option description.

The first option allows you to create a secondary interactive job, or if it is already created, to transfer to it. Option 2 allows you to cancel the request that was active when you entered the system request menu. Option 3 allows you to display the status of your current job. Each of the remaining options allows you to request the full function of the associated specific CPF commands. You can enter parameter values of the command in the Parameters field Ⓑ in the same manner you would if you were entering the entire command (see *Command Syntax* earlier in this chapter). Default parameter values are provided for parameters that you choose not to enter. If you need assistance in entering the parameter values, you can request parameter prompt displays for the command associated with the option by pressing the CF4 key. All prompting functions described under *Special Prompting Functions* earlier in this chapter are available for use when requesting options 4 through 6 from the system request menu. If you specify parameter values in the Parameters field before requesting the prompt, those values are shown on the prompt. You can enter the parameter values you want on the prompt screen, and then press the Enter/Rec Adv key to execute the command. If you enter an incorrect value, you receive an error message at the bottom of your screen. The function you requested is not executed until you key in the correct information on top of your previous input and press the Enter/Rec Adv key.



```
                        SYSTEM REQUEST MENU
Select one of the following:
  1. Transfer to secondary interactive job
  2. Cancel previous request
  3. Display current job
  4. Display messages    (msgq)
  5. Send message        'msg' tomsgq
 90. Signoff             (*NOLIST  *LIST)
Option: __                              CF4-prompt (4,5 and 90)
Parameters: ______________________________________________________
__________________________________________________________________
```

Figure 5-7. System Request Menu

If you want to return to the system request menu without executing the command, press the CF1 key. If instead you want to execute the command, press the Enter/Rec Adv key on the last display or press the CF16 key. After the command is executed, you again receive the system request menu. An error message appears on the bottom of the menu if the command did not complete normally.

To return to your interactive job from the system request menu, press the CF1 key or press the Enter/Rec Adv key with the option field left blank. The job is resumed at the point where you left it, and you receive the display that was on your screen when you entered the system request.

The following summarizes how you use each of the options on the system request menu:

Option 1: Using this option, you can create a secondary interactive job at the same work station and then transfer back and forth between the primary and secondary job. The first time you request the system request menu and enter option 1, you receive the sign-on prompt. When you sign on, a secondary job is created and you receive the basic working display of the new job (such as the command entry display). Your primary job remains suspended as long as you are in the secondary job. The next time you request the system request menu and enter option 1, your current job is suspended and you return to the other job at the point where you left it. When you sign off either of the jobs, you are automatically returned to the remaining job.

Option 2: This option cancels the previous request that was active when you entered the system request. You can use this option, for example, to cancel a command that you do not want to complete. After the cancel function is complete, you receive the display that was on your screen when you entered the system request. For additional information, refer to the *PRV parameter value under the CNLRQS command in the *CL Reference Manual*.

Note that this option is not allowed when your job is in such a state that processing cannot be interrupted. If you select option 2, you receive an error message if this condition exists.

Option 3: This option allows you to display the status of your current job. When you enter option number 3 in the Option field, you receive the job display menu for the job from which you requested the system request menu. When you press the Enter/Rec Adv key after viewing the selected options, you return to the system request menu. For additional information, refer to the * parameter value for the JOB keyword under the DSPJOB command in the *CL Reference Manual*.

Option 4: This option allows you to display the messages in a message queue. You receive a separate message display (see *Displaying the Messages* in Chapter 3). If you enter only the option number 4 in the Option field, you receive a display of messages in the message queue associated with the work station you are using. If you also specify the name of a message queue in the Parm field, you receive a display of the messages in the specified message queue. For example, to display the messages in the message queue DEPT255 you would enter *DEPT255* in the Parm field. When you press the Enter/Rec Adv key after examining the messages, you return to the system request menu. The CF4 key can be used to request prompting for all the parameters on the associated Display Message (DSPMSG) command. For additional information about the DSPMSG command, see the *CL Reference Manual.*

Option 5: This option allows you to send a message to a specified message queue (see *How to Send Messages to the System Operator* in Chapter 4). After entering a 5 in the Option field, you enter the message, enclosed in apostrophes, in the Parm field, followed by the name of the message queue where the message is to be sent. For example, to send the message:

```
Start ORDER subsystem
```

to the system operator message queue (QSYSOPR), you enter in the Parm field

```
'Start ORDER subsystem' QSYSOPR
```

The CF4 key can be used to request prompting for all the parameters on the associated Send Message (SNDMSG) command. For additional information on the SNDMSG command, see the *CL Reference Manual.*

Option 90: This option allows you to end the interactive job from which you entered system request. Each interactive job has a job log that can be printed after the job has ended. If you enter only the option number 90 in the Option field, the requesting job ends without printing the job log for the job. To print the job log, you must also specify *LIST in the Parm field. Signing off from your job results in your receiving the sign-on display or your secondary job being resumed (see Option 1). The CF4 key can be used to request prompting for all the parameters on the associated sign off (SIGNOFF) command. For additional information on the SIGNOFF command, see the *CL Reference Manual.*

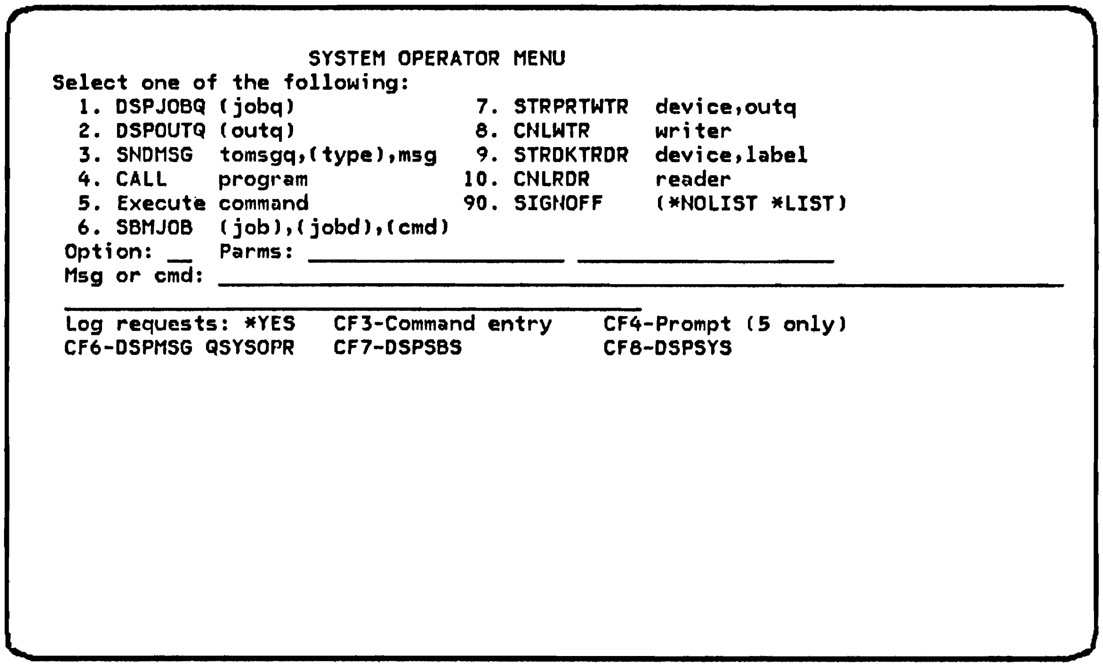
The system request is a useful function that can aid you in your work on the system. Using it *improperly*, however, could tie up system resources that are needed by other system users. If you use a system request to exit your interactive job at a work station while the system is processing a function you have requested through that job, system resources being used to process the function are held until you return to the job and the function completes. The resources being held are not available to other system users. Therefore, unless you have a problem with your interactive job that requires an immediate exit from the job, you should use the system request only when the system has completed your previous request and is waiting for you to enter a new request.

THE SYSTEM OPERATOR MENU

The system operator menu serves as the basic working display for users who sign on the system as a system operator; that is, for those using the QSYSOPR user profile.

The menu provides a direct method of executing the commands that are most likely to be used in operating the system.

The following is the system operator menu.



```
                          SYSTEM OPERATOR MENU
Select one of the following:
  1. DSPJOBQ (jobq)                 7. STRPRTWTR  device,outq
  2. DSPOUTQ (outq)                 8. CNLWTR     writer
  3. SNDMSG  tomsgq,(type),msg      9. STRDKTRDR  device,label
  4. CALL    program               10. CNLRDR     reader
  5. Execute command               90. SIGNOFF    (*NOLIST *LIST)
  6. SBMJOB  (job),(jobd),(cmd)
 Option: __  Parms: ____________________ ____________________
 Msg or cmd: ______________________________________________________________
 ______________________________________________
 Log requests: *YES   CF3-Command entry    CF4-Prompt (5 only)
 CF6-DSPMSG QSYSOPR   CF7-DSPSBS           CF8-DSPSYS
```

For information on using the system operator menu, refer to the *System/38 Operator's Guide*.

Chapter 6. Message Handling Using Commands

TYPES OF MESSAGES

The system can handle two basic kinds of messages.

Predefined Messages: A predefined message has a fixed form that is defined by a message description stored in a message file. Predefined messages are the means of communication between the various types of programs running on the system as well as between those programs and users of the system. CPF error messages are predefined messages.

Impromptu Messages: An impromptu message can have any form up to a maximum length of 132 characters as long as it is enclosed in apostrophes when entered into the system. For example, a properly entered impromptu message is:

```
'Dept meeting at 3:00 in room 2A'
```

The apostrophes tell the system to transmit the message exactly as entered. When you send a message to another system user, you are using an impromptu message.

Details of the various message types and how they are used by the system can be found in the *CPF Programmer's Guide.* The discussion here is restricted to how you handle messages at your work station.

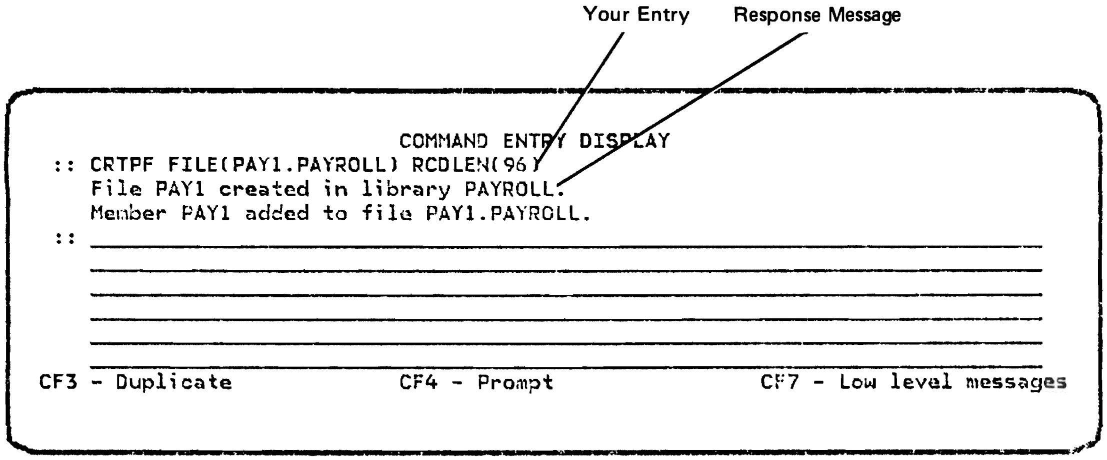
HOW MESSAGES ARE D!SPLAYED

In terms of the way you receive them at a work station, messages can be categorized as follows:

- Messages in response to your requests
- Messages sent to your message queue

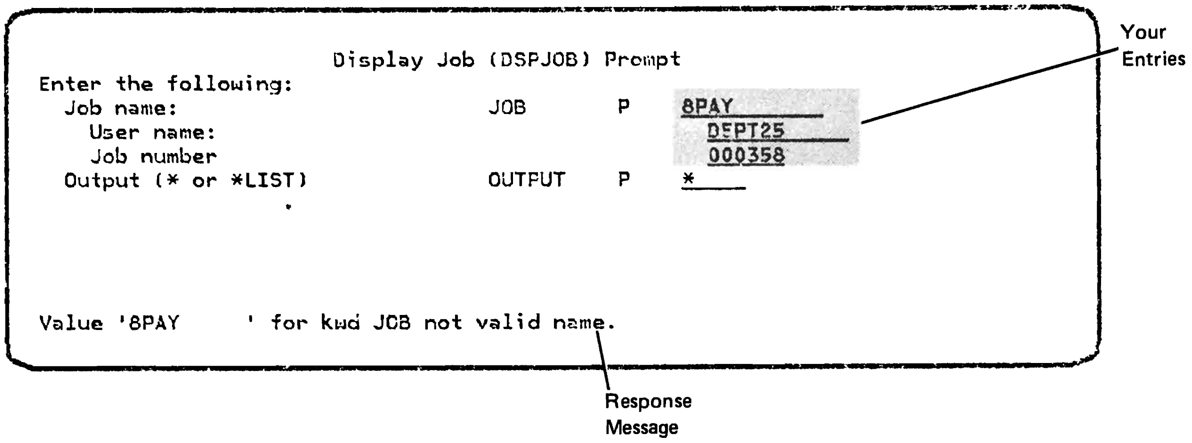
Response Messages

When you enter a command, you are sending a request to the system. Messages in response to that request are sent to the display on which you entered the request and displayed automatically. Those response messages, which are usually predefined messages, appear directly below your request on the command entry display



```
                          COMMAND ENTRY DISPLAY
 :: CRTPF FILE(PAY1.PAYROLL) RCDLEN(96)
    File PAY1 created in library PAYROLL.
    Member PAY1 added to file PAY1.PAYROLL.
 :: ____________________________________________________________________
    ____________________________________________________________________
    ____________________________________________________________________
    ____________________________________________________________________
    ____________________________________________________________________
    ____________________________________________________________________
CF3 - Duplicate             CF4 - Prompt             CF7 - Low level messages
```

or on the bottom line of the screen for other types of CPF displays.



```
                      Display Job (DSPJOB) Prompt
Enter the following:
  Job name:                        JOB        P     8PAY______
    User name:                                        DEPT25____
    Job number                                        000358
  Output (* or *LIST)              OUTPUT     P     *____



Value '8PAY      ' for kwd JOB not valid name.
```

Response messages on the command entry display regarding a successfully executed command are intended for your information only and do not require an action by you. Response messages regarding an error on the command entry display or another display do require an action by you if you want the request to be processed.

The action could be to use the procedures described earlier for obtaining information about the error and correcting it (see *How to Handle Response Messages* in Chapter 3). Or you could bypass the error and go on to another function. On the command entry display, this would be done merely by ignoring the error and entering another command in the new input field provided below the error message. On a display requested from the command entry display, you could exit the requested function (by pressing the CF1 key or the Enter/Rec Adv key). This returns you to a display that allows function requests where another option can be selected.

If you make a keying error on any display, you initially receive only an error code rather than an error message. These error codes are described in the operator's guides for the 5251 and 5252 display stations and the *IBM 5250 Information Display System Functions Reference Manual*, SA21-9247. You can, however, press the Help key to replace the error code by a descriptive message. Regardless of which display you are using, you must press the Error Reset key before you can correct a keying error. On some types of displays, in particular, menus and prompts, you must also press the Error Reset key to correct a request error indicated by an error message.

During any type of error condition at your work station, you can use the system request function to temporarily escape from your interactive job (see *How to Use System Request* in Chapter 5).

Messages Sent to a Message Queue

Except for messages in response to your requests, all messages sent to you at a work station are transmitted to a message queue associated with the work station. The message queue then delivers the messages to your work station. Using a command, you can define the way in which the messages are delivered to you (see *How to Set the Message Delivery Mode* later in this chapter).

Messages can be delivered to you at a work station from two types of message queues (Figure 6-1):

- Work station message queues
- User message queues

Each work station has a system message queue permanently assigned to it. A message sent to a work station message queue is available to anyone currently signed on at the work station. A work station message queue and the job supporting the work station have the same name as the work station. You can find out what this name is by requesting the job display for your interactive job which gives the job name. This can be done by simply entering DSPJOB (Display Job command) with no parameter values on the command entry display or programmer's menu on that work station.

A message queue can also be created for individual users of the system by using the Create Message Queue (CRTMSGQ) command. The user message queue exists independently of the work station message queue and can be used at any work station. A message sent to a user message queue is available only to the user or users authorized to use the queue. If you are authorized to use a particular user message queue, you can display messages from it, but you will not be notified of messages in the queue unless you change its delivery mode using the change message queue (CHGMSGQ) command. If you change your user message queue to break mode or notify mode, using the DLVRY parameter, the message queue is allocated exclusively to your interactive job and will deliver messages only to your work station until you change the delivery mode to hold or default mode. Other users cannot receive or display messages from your user message queue when it is in break or notify mode. When you sign off, your work station message queue is reset to hold mode and is reallocated. For additional information, see *Changing a Message Queue* in the *System/38 Operator's Guide.*

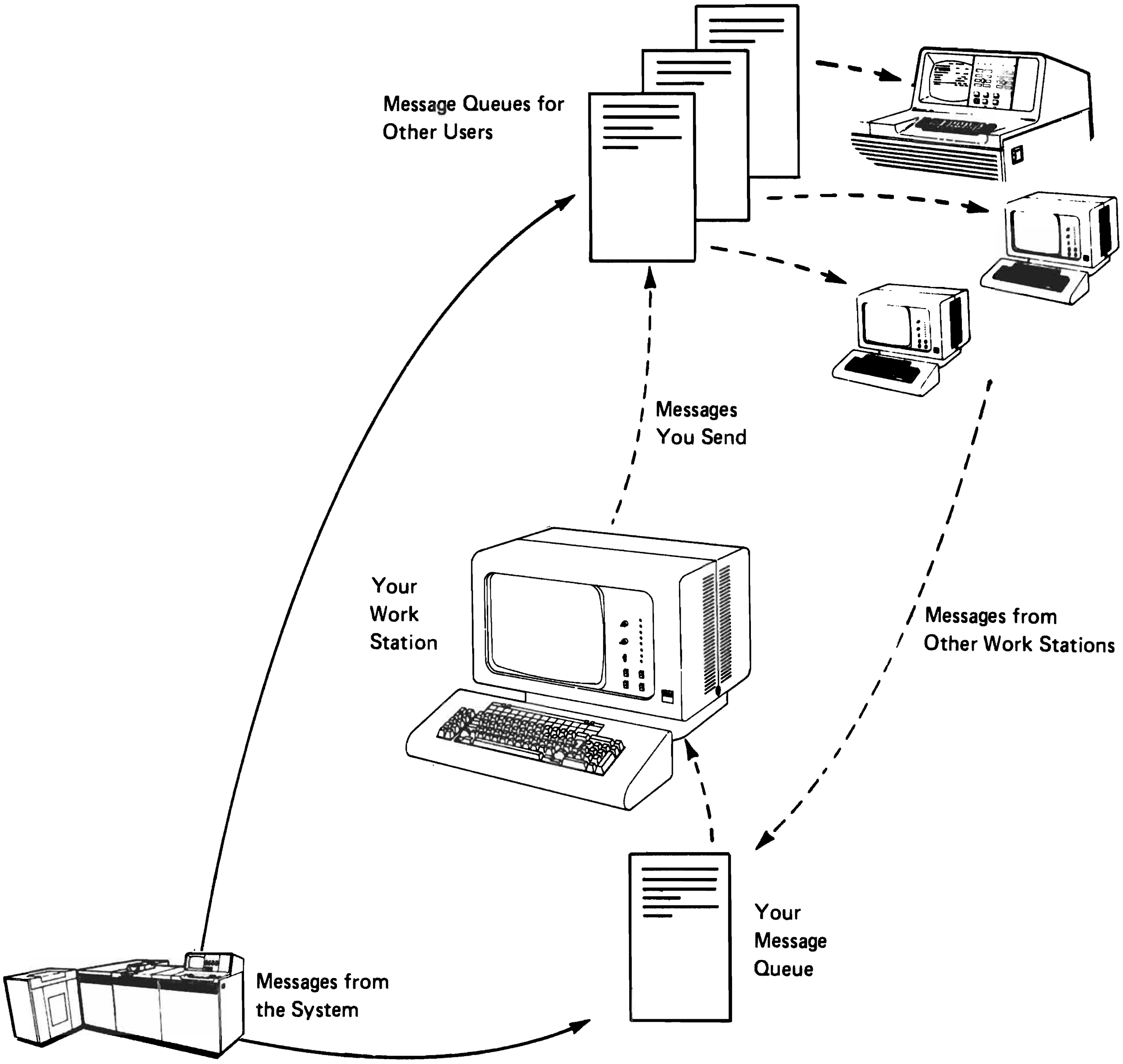


Figure 6-1. Work Station and User Message Queues

Whether a message is sent to your work station message queue or user message queue, you obtain it the same way. For details on displaying messages sent to a message queue, see the discussion later in this chapter on *How to Display Messages from a Message Queue.*

HOW TO SET THE MESSAGE DELIVERY MODE

What Are Message Queue Delivery Modes?

A work station or user message queue can deliver messages to your work station in one of four modes. The current delivery mode and severity of a message queue are displayed on the first line of the DSPMSG display. The four delivery modes are:

- *Break Mode*: If you set your message queue to the break mode and it contains any messages you have not displayed, your work station job is interrupted and the messages are displayed. From that point on until the message queue is changed to a different delivery mode, each time a new message arrives in the queue, your work is interrupted to deliver the message. However, a message severity level can be set for the queue such that only messages having a severity level of greater than or equal to the queue will be delivered in break mode (see CHGMSGQ command in the *CL Reference Manual*).

  The way in which the message is delivered depends on what program was specified when the message queue was changed to the break mode (see the following discussion of *Using the Change Message Queue Command*). If *DSPMSG was specified on the CHGMSGQ command, the following occurs each time the message queue contains a message you have not displayed:
  - If you are using the work station, the keyboard buzzer sounds briefly.
  - If you are using the system console and you have the alarm feature, the alarm sounds until a response is made.
  - Your current display is saved and replaced on the screen by a message display containing the message.

  If another specialized program was specified, the way in which you are notified and the message delivered is determined by the program.

- *Notify Mode:* Your work is not interrupted, but the Message Waiting indicator on your screen comes on (■) and the keyboard buzzer sounds briefly to notify you when there are messages to be displayed.

  You display the messages whenever you choose by entering the Display Messages command.

- *Hold Mode*: The message queue holds the messages until you ask for them, using the Display Messages command. You are not notified when a message arrives.

- *Default Mode*: All messages are ignored and any messages requiring a reply are sent the default reply specified for the message. None of the messages are retained in the queue.

Message Queue Delivery Modes at Sign-On and Sign-Off

Sign-On

When you sign on a work station, the work station message queue is automatically reset to the notify mode. Therefore, if there are any messages in the work station message queue, the Message Waiting indicator and the keyboard buzzer come on.

If you also have a user message queue, it remains in the hold mode until it is specifically allocated to your work station after you are signed on.

Sign-Off

When you sign off a work station, the work station message queue remains in or reverts to the hold delivery mode. Your user message queue is also reset to the hold mode unless it was set to the default mode; in that case it remains in the default mode.

Using the Change Message Queue Command

To change the delivery mode of a message queue, you use the Change Message Queue (CHGMSGQ) command and specify the delivery mode in the DLVRY parameter. If you specify the break mode in the DLVRY parameter, you can use the PGM parameter to specify which program is to be invoked to deliver the messages. The program specified previously is not changed if you do not enter the PGM parameter. However, if no program was specified previously and you do not enter the PGM parameter, the *DSPMSG program is assumed. This means that once you have entered the CHGMSGQ command specifying the break mode, any messages in the message queue are immediately delivered to your work station.

Note: You should avoid setting more than one message queue to the notify mode at the same time. This could result in confusion as to which message queue is signaling you.

As with all CL commands, you can enter the CHGMSGQ command directly on the command entry display or you can request system prompting (see *System-Assisted Entry* in Chapter 5). (For details on the various parameters of the CHGMSGQ command, see the *CL Reference Manual*.)

Examples:

The following examples assume you are entering the CHGMSGQ command on the command entry display.

1. To change the work station WS3 message queue so that messages are retained but you are not notified when they arrive, enter

   ```
   CHGMSGQ  MSGQ(WS3)  DLVRY(*HOLD)
   ```

2. To change the user message queue designated as DEPT22 so that any new messages are immediately delivered, enter

   ```
   CHGMSGQ  MSGQ(DEPT22)  DLVRY(*BREAK)
   ```

HOW TO DISPLAY MESSAGES FROM A MESSAGE QUEUE

Break Messages and Break Delivery

There may be occasions when you receive a break message from the system operator to report some system condition that affects your work. When a break message is sent to your message queue, the keyboard buzzer sounds briefly. Regardless of how your message queue is set to deliver messages, a break message is always delivered immediately.

As described earlier in this chapter, a message queue set to the break delivery mode delivers messages according to a specified program. If that program is *DSPMSG, messages are delivered to your work station immediately in the same way as a break message. However, if a severity code was specified, only messages having severity equal to or greater than the specified severity will cause break delivery.

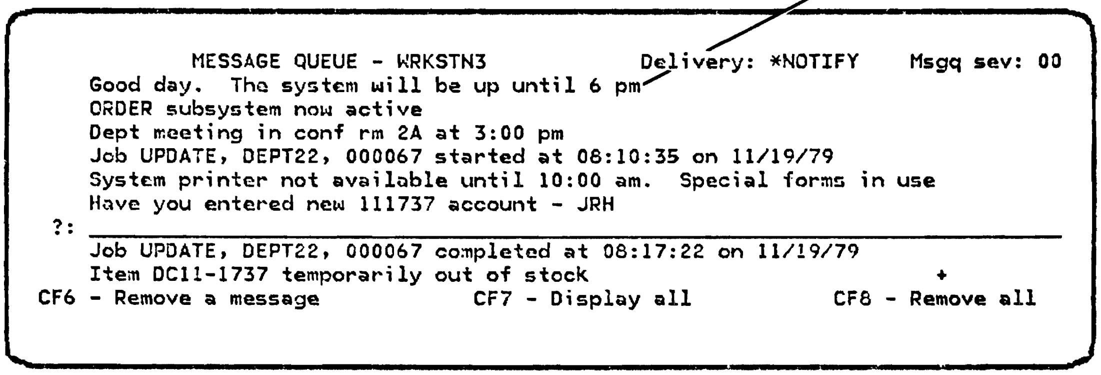
Notify and Hold Delivery

If a message queue is set to either the notify or hold mode, you obtain the messages by using the Display Messages (DSPMSG) command or the display messages option on the system request menu. In the notify mode, the arrival of a message causes the Message Waiting indicator to come on (unless it is already on) and the keyboard buzzer to sound. In the hold mode, this does not happen.

The DSPMSG command allows you considerable versatility in specifying which messages you want to see. You can restrict the messages to a certain type by using the MSGTYPE parameter, or to only those messages above a certain severity level by using the SEV parameter. Furthermore, by specifying OUTPUT(*LIST), you can have the specified messages printed instead of displayed on your work station screen.

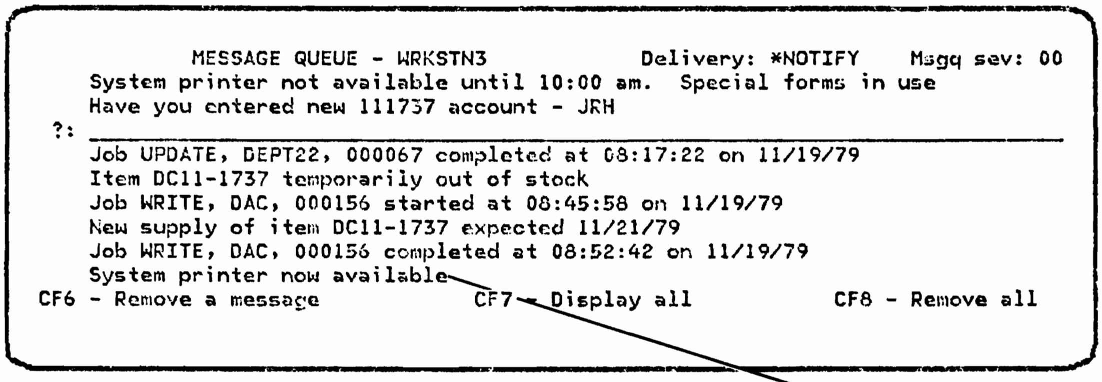
If you are displaying the messages at your work station, you can use the START parameter to specify where the display begins. By specifying START (*FIRST), you specify that the display is to begin at the top of the screen with the first message placed on the message queue, such as

First Message in Queue



```
           MESSAGE QUEUE - WRKSTN3           Delivery: *NOTIFY    Msgq sev: 00
    Good day.  The system will be up until 6 pm
    ORDER subsystem now active
    Dept meeting in conf rm 2A at 3:00 pm
    Job UPDATE, DEPT22, 000067 started at 08:10:35 on 11/19/79
    System printer not available until 10:00 am.  Special forms in use
    Have you entered new 111737 account - JRH
 ?: ___________________________________________________________________________
    Job UPDATE, DEPT22, 000067 completed at 08:17:22 on 11/19/79
    Item DC11-1737 temporarily out of stock                              +
CF6 - Remove a message           CF7 - Display all           CF8 - Remove all
```

You obtain any later messages that do not fit on the screen by pressing the Roll Up key. If you do not specify the parameter or if you do specify START(*LAST), the display begins with the last message at the bottom of the screen, such as



```
              MESSAGE QUEUE - WRKSTN3          Delivery: *NOTIFY    Msgq sev: 00
      System printer not available until 10:00 am.  Special forms in use
      Have you entered new 111737 account - JRH
   ?: ____________________________________________________________________________
      Job UPDATE, DEPT22, 000067 completed at 08:17:22 on 11/19/79
      Item DC11-1737 temporarily out of stock
      Job WRITE, DAC, 000156 started at 08:45:58 on 11/19/79
      New supply of item DC11-1737 expected 11/21/79
      Job WRITE, DAC, 000156 completed at 08:52:42 on 11/19/79
      System printer now available
  CF6 - Remove a message              CF7 - Display all            CF8 - Remove all
```

Last Message
in Queue

You obtain any earlier messages that will not fit on the screen by pressing the Roll Down key. (For details on the various parameters of the DSPMSG command, see the *CL Reference Manual.*)

Examples:

The following examples assume you are entering the DSPMSG command on the command entry display.

1. To display on your work station screen all types of new messages that are on the work station message queue, enter

   DSPMSG

2. To print all types of messages on the DEPT22 user message queue, enter

   DSPMSG MSGQ(DEPT22) OUTPUT(*LIST)

3. To display at your work station only inquiry messages (those requiring a reply) on the JONES message queue that have a severity code equal to or greater than 20 and start with the last message on the bottom of your screen, enter

   DSPMSG MSGQ(JONES) MSGTYPE(*INQ) SEV(20)

Using Message Displays

Whether you receive the messages automatically or request them your current display is replaced by a messages queue display of the type discussed in Chapter 3. For information on interpreting and using message displays, refer to *Message Displays* in Chapter 3.

When you are finished with the messages queue display, you can return to the command entry display (or your previous display) by pressing the Enter/Rec Adv key. (See Figure 3-9 in Chapter 3.)

After you receive a message on your display and determine that the message is no longer needed, you should delete the message. (See *Deleting Messages* in Chapter 3.) Deleting the message makes it easier to view your current messages and frees up system storage.

HOW TO SEND MESSAGES

You can send an impromptu message from your work station to one or more work station or user message queues by entering the Send Message (SNDMSG) command. You specify the message in the MSG parameter and where it is to be sent in the TOMSGQ parameter. The text of the message must be enclosed in apostrophes, such as:

'Jones account delayed'

You can require a reply to your message (by specifying *INQ in the MSGTYPE parameter), and have that reply sent to either your work station or to a specific user message queue (RPYMSGQ parameter). To send a message to the system operator, you specify QSYSOPR in the TOMSGQ parameter. (For additional details on the SNDMSG command, see the *CL Reference Manual.*)

Examples:

The following examples assume you are entering the SNDMSG command on the command entry display.

1. To send the message

   Please start PAYROLL subsystem

   to the system operator, enter

   ```
   SNDMSG  MSG('Please start PAYROLL subsystem')
    TOMSGQ(QSYSOPR)
   ```

2. To send the message

   Monday dept meeting canceled

   to work station message queues WRKSTN4 and WRKSTN5, enter

   ```
   SNDMSG  MSG('Monday dept meeting canceled')
    TOMSGQ(WRKSTN4  WRKSTN5)
   ```

   Note: To determine the name of the work station message queue, see *Messages Sent to a Message Queue* earlier in this chapter.

3. To send the message

   Will you be done at 4:00?

   to the user message queue SMITH and require a reply to your user message queue (MYQUEUE), enter

   ```
   SNDMSG  MSG('Will you be done at 4:00?')
    TOMSGQ(SMITH)  MSGTYPE(*INQ)  RPYMSGQ(MYQUEUE)
   ```

4. To send the break message

   Do you have account number 1234?

   to work station message queue WRKSTN1 and require a reply to your user message queue (WRKSTN2), enter

   ```
   SNDBRKMSG MSG('Do you have account number 1234?')
   TOMSGQ(WRKSTN1) MSGTYPE(*INQ) RPYMSGQ(WRKSTN2)
   ```

Chapter 7. Signing Off through a Command

WHAT SIGN-OFF MEANS

When you sign off, you are telling the system that your interactive job at the work station is ended. To use the work station again, you must sign on as you did before; this creates a new interactive job.

HOW TO SIGN OFF

You sign off at a work station by using the Sign Off (SIGNOFF) command. You can either enter the complete command on the command entry display or request a prompt for the command by entering a partial command and pressing the prompt key (CF4). The actual form of the command you use depends on how your work station is connected to the System/38 and whether you want to print the job log for your job.

Job Log

Each job on the system, including your interactive job, has an associated job log for retaining job information. For example, your requests entered on the command entry display and the system responses to the requests could be retained in the job log for your job. The information that is actually retained in the job log, if any, is determined by how your interactive job was defined. (See the description of the LOG parameter for the Create Job Description command in the *CL Reference Manual*.) When you sign off, you can request that your job log be printed with other spooled output from your job. To have the job log printed, include the LOG parameter in the form

```
LOG(*LIST)
```

when you enter the SIGNOFF command.

Example:

You are signing off from a work station connected to the system through a dial-up line. You want to drop the line and print your job log; therefore, you enter the SIGNOFF command in the form

```
SIGNOFF LOG(*LIST) DROP(*YES)
```

Note: If your remote work station has control of an application and the application ends with your receiving the sign-on prompt, you can specify whether the line is to be dropped, by signing on and then using the SIGNOFF command as described above.

Work Stations Not Connected by Switched Line

If you are using a work station directly connected to the system (not through a switched communications line) and you do not want to print your job log, you sign off by using the simplest form of the Sign Off command. You merely enter

```
SIGNOFF
```

If you do want to print your job log, you should enter

```
SIGNOFF LOG(*LIST)
```

If you are not sure how your work station is connected to the system, use the DROP parameter to default to *DEVD.

Work Stations Connected by Switched Line

If you are using a remote work station connected to the system through a switched communications line (such as a dial-up line), you can do one of the following:

1. Enter only the command name

   ```
   SIGNOFF
   ```

   In this case, the disposition of the communications line is determined based on the work station device description stored in the system.

2. Include the DROP parameter in the form

   ```
   DROP(*YES)
   ```

   In this case, the communications line is released if no other work station on the line is signed on.

3. Include the DROP parameter in the form

   ```
   DROP(*NO)
   ```

   In this case, the communications line is held. You would use this form of the command, for example, when another user is waiting to sign on the work station as soon as you sign off.

   In each of the above examples, you have the option of specifying LOG(*LIST) if you want to print your job log.

Note that if your remote work station has control of an application and the application ends with your receiving the sign-on prompt, you can specify whether the line is to be dropped, by signing on and then using the SIGNOFF command as described above.

Chapter 8. Requesting Special Applications

HOW TO REQUEST IDU

The Interactive Data Base Utilities (IDU) is a licensed program consisting of four separate utilities that assist work station users and programmers in performing certain types of tasks. If these utilities are installed on your system, you can request an IDU function by entering the appropriate IDU command on the command entry display, on the programmer menu, or through prompting, as described in Chapter 5.

Source Entry Utility

The source entry utility (SEU) aids you in entering and maintaining source statements for System/38, such as RPG specifications and COBOL source. Details on the use of SEU can be found in the *SEU Reference Manual and User's Guide.*

Data File Utility

The data file utility (DFU) aids you in entering, updating, and displaying data file records. Details on the use of DFU can be found in the *DFU Reference Manual and User's Guide.*

Query Utility

The query utility allows you to select data from a file in the data base and generate a report from that data. The report can be displayed, or printed, or both. Details on the use of query can be found in the *Query Utility Reference Manual and User's Guide.*

Screen Design Aid Utility

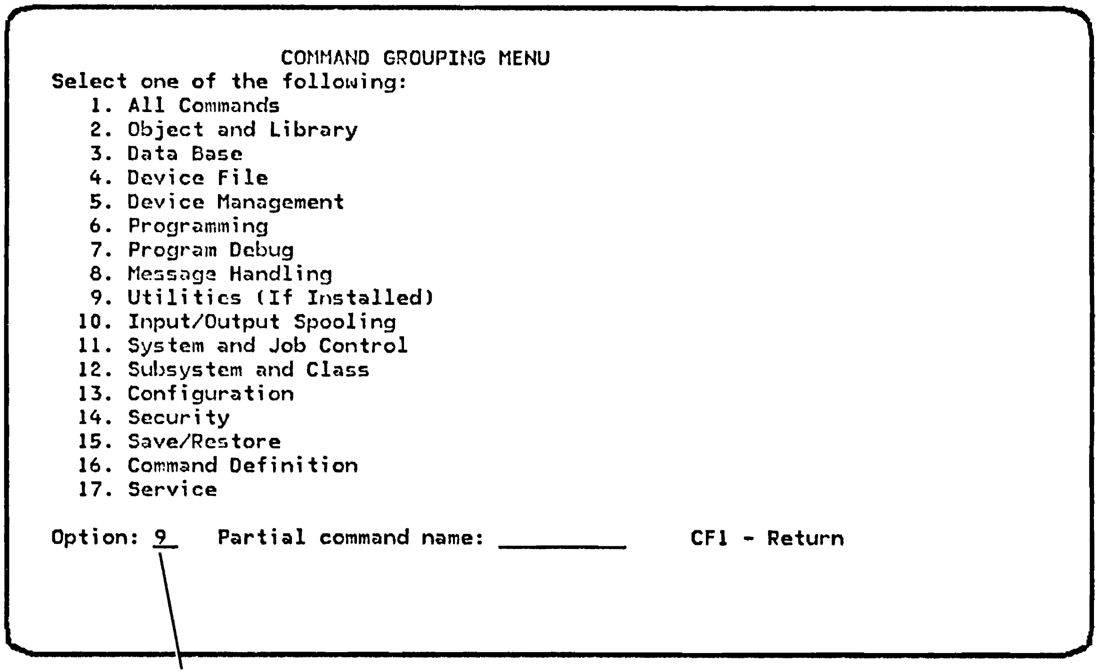
The screen design aid (SDA) allows you to interactively design, create, maintain, and test display formats and menus which are to be incorporated into your applications. SDA generates data description specifications (DDS) for the displays and menus you design and also generates the CL program source code necessary to support your menus. In additon, SDA initiates the compilation of the menu programs and creates the display device files. For more information refer to the *SDA Reference Manual and User's Guide.*

Requesting Utilities Through Prompting

You can select and execute an IDU command through the same system prompting facilities that are used for CPF commands (see *Command Prompting* in Chapter 5). The procedure is:

1. Press the prompt (CF4) key to obtain the command grouping menu.

2. Select option 9 on the menu.

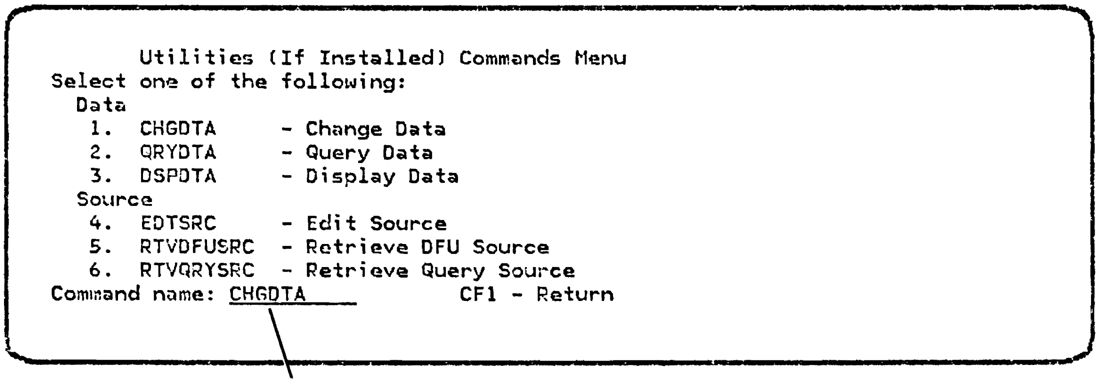


```
                    COMMAND GROUPING MENU
Select one of the following:
   1. All Commands
   2. Object and Library
   3. Data Base
   4. Device File
   5. Device Management
   6. Programming
   7. Program Debug
   8. Message Handling
   9. Utilities (If Installed)
  10. Input/Output Spooling
  11. System and Job Control
  12. Subsystem and Class
  13. Configuration
  14. Security
  15. Save/Restore
  16. Command Definition
  17. Service

Option: 9    Partial command name: __________     CF1 - Return
```

Key in 9.

You then receive a menu of all utility commands. The Roll key can be used to see the remaining commands in a group.

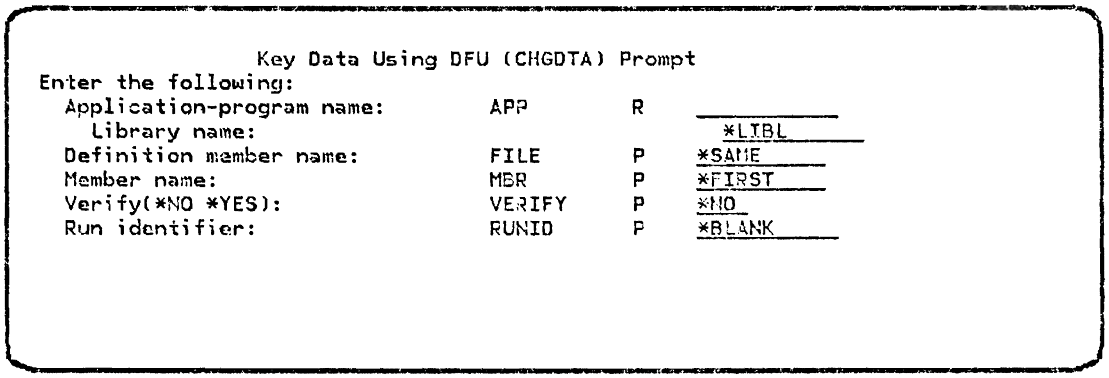
3. Enter a command name on the menu and press the Enter/Rec Adv key.



```
       Utilities (If Installed) Commands Menu
Select one of the following:
  Data
   1.  CHGDTA     - Change Data
   2.  QRYDTA     - Query Data
   3.  DSPDTA     - Display Data
  Source
   4.  EDTSRC     - Edit Source
   5.  RTVDFUSRC  - Retrieve DFU Source
   6.  RTVQRYSRC  - Retrieve Query Source
Command name: CHGDTA           CF1 - Return
```

Key in the command name here.

You receive the prompt for that command.

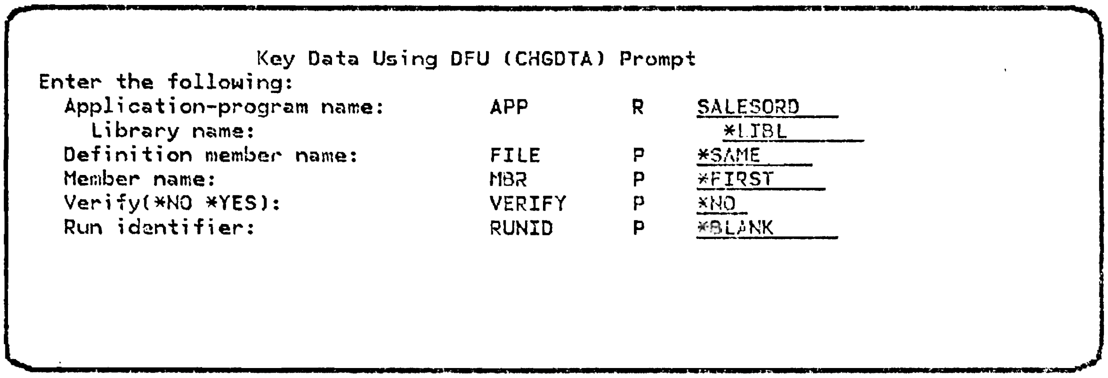


```
                   Key Data Using DFU (CHGDTA) Prompt
Enter the following:
  Application-program name:        APP       R     __________
    Library name:                                    *LIBL
  Definition member name:          FILE      P     *SAME
  Member name:                     MBR       P     *FIRST
  Verify(*NO *YES):                VERIFY    P     *NO
  Run identifier:                  RUNID     P     *BLANK
```

4. Complete the prompt and press the Enter/Rec Adv key.

Example—Using DFU:

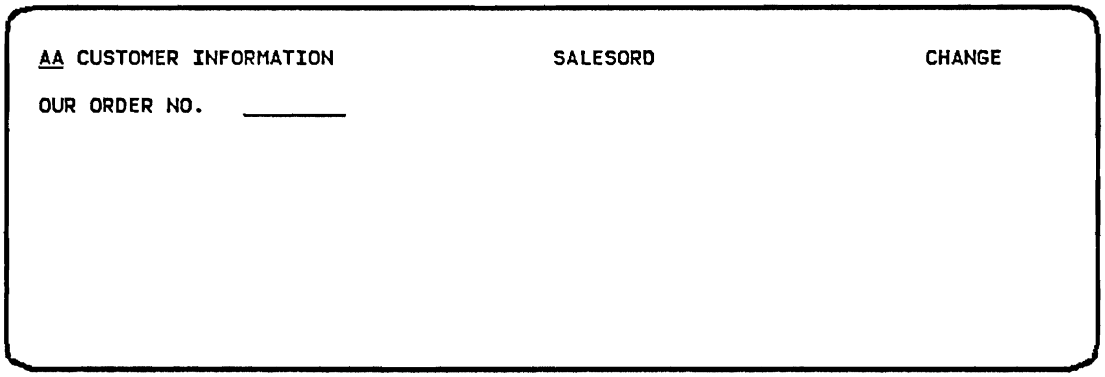
A DFU application has been created and you have obtained the prompt for the Change Data command of the data file utility. You want to obtain information using the SALESORD DFU application. Therefore, you key SALESORD into the APP input field.



```
                   Key Data Using DFU (CHGDTA) Prompt
Enter the following:
  Application-program name:        APP       R     SALESORD
    Library name:                                    *LIBL
  Definition member name:          FILE      P     *SAME
  Member name:                     MBR       P     *FIRST
  Verify(*NO *YES):                VERIFY    P     *NO
  Run identifier:                  RUNID     P     *BLANK
```

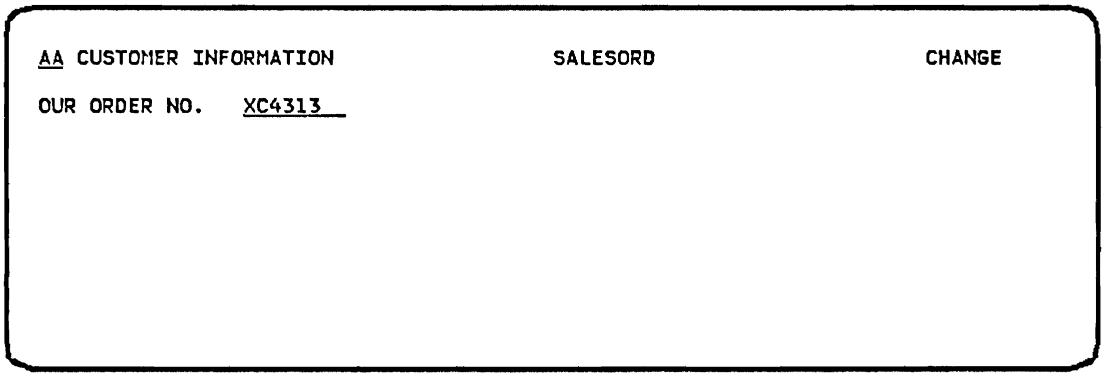
Then press the Enter/Rec Adv key.

DFU responds with a display requesting the order number:



```
AA CUSTOMER INFORMATION                SALESORD                 CHANGE

OUR ORDER NO.    ________
```

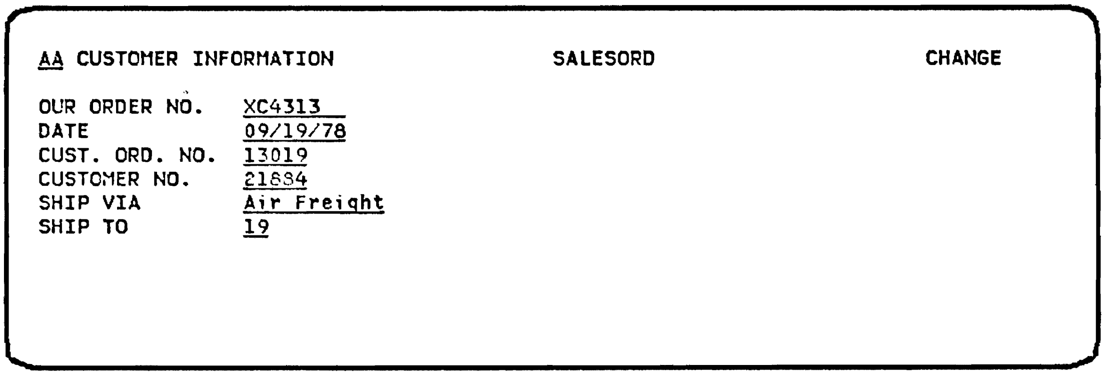
You key in the order number XC4313.



```
AA CUSTOMER INFORMATION                SALESORD                 CHANGE

OUR ORDER NO.    XC4313
```

Then press the Enter/Rec Adv key.

DFU responds with the information for the order number you specified:



```
AA CUSTOMER INFORMATION                SALESORD                 CHANGE

OUR ORDER NO.    XC4313
DATE             09/19/78
CUST. ORD. NO.   13019
CUSTOMER NO.     21884
SHIP VIA         Air Freight
SHIP TO          19
```

HOW TO CALL A PROGRAM

Some System/38 functions, such as IDU, are requested through specific commands. Other functions may not have specific commands associated with them. If you want to use an application program that is not associated with a specific command, you request the program by using the CALL command. The CALL command performs the same function as option 1 of the program call menu.

In its simplest form, the command consists of the word CALL followed by the name of the program to be called, such as

```
CALL  INVENTORY
```

When you enter the command in this form, you are using positional coding. The keyword-with-value form of the same example would be

```
CALL  PGM(INVENTORY)
```

You can also pass parameters to the called program. For example, if you want to call the program STOCK and pass the parameters NEW, CURRENT, and SHIP to it, you would enter (in keyword-with-value form)

```
CALL  PGM(STOCK)  PARM(NEW CURRENT SHIP)
```

or (in positional form)

```
CALL  STOCK  (NEW CURRENT SHIP)
```

Note that if parameters are required they must be entered exactly as the program expects to receive them.

For additional details on the CALL command, see the *CL Reference Manual*.

Appendix A. Summary of Command Function Keys–CPF and IDU

The lists on the following pages summarize the general command function key assignments for CPF and the Interactive Data Base Utilities - DFU, Query, SEU, and SDA - that are available from IBM. Each list shows:

- The name of the command function key.
- The command function (CF) number of the key.
- An illustration of the appropriate key on the data-entry or typewriter-like keyboard that you press to obtain the function (remember, you must press the Cmd key first).
- A description of the function.

To help you find the correct key on your keyboard, the pages following the lists show:

- The complete 5251/5252 keyboards.
- The relation of the top row keys on the keyboards to the CF numbers shown on the lists.

The *IBM System/38 Keyboard Template*, GX21-7756, is available for use above the keyboard. It will assist you in remembering which command function keys are used by a particular licensed program.

CPF Command Function Key Summary

Name of Function	Key Name	Valid For	Data-Entry Keyboard	Typewriter-Like Keyboard	Function of Key
Command Complete (Prompter)	CF16	Ⓑ	▨ $ *	▨ $ 4	Indicates to the system that command is complete as entered and to execute the command.
Command Prompt	CF4	Ⓐ	$ *	$ 4	Requests command name or command parameter prompting.
Display Command String (Prompter)	CF14	Prompting	▨ # @	▨ @ 2	Displays resulting command in string form as it would be entered on the command entry display.
Display Error Messages (Prompter)	CF15	Prompting	▨ ' %	▨ # 3	Obtains a display of all error messages currently outstanding on a command.
Display Key Assignments (Prompter)	CF13	Prompting	▨ # @	▨ ! 1	Displays all CF key functions available to assist in entering parameters of a command.
Enter/Rec Adv	Enter /Rec Adv	Most CPF Displays	ENTER/ REC ADV	ENTER/ REC ADV	Advances to next display if +++ is shown in the upper right corner of the display. If no +++ is shown, returns to the display from which the request was made.
Help	Help	Most CPF Displays	HELP	HELP	Displays second-level text if you position the cursor under a message.
Previous Display	CF2	Some CPF Displays	# @	@ 2	Backs up to the previous display in a series if Enter/Rec Adv pressed because +++ appeared on previous display.
Print (Console only)	CF24	All CPF	N/A	N/A	Prints the current system console display on the system printer or an alternate printer (performs the same function as the Print key on the 5251/5252 work stations).
Redisplay	CF5	Most CPF Displays	. <	% 5	Causes the current display to be reshown with the most current data available.
Return	CF1	Most CPF Displays	# @	! 1	Provides a permanent exit from a function and returns to basic working display, such as the command entry display, from which the function was requested.
Revert to Command Defaults (Prompter)	CF18	Prompting	▨ DUP	▨ ¬ 6	Returns a prompt display to its original form with all values set to default. Any values previously entered are lost.
Roll	Roll	Most CPF Displays	ROLL↑ ROLL↓	ROLL↑ ROLL↓	Displays additional data on the current display.

Legend:

▨ Shaded keys are uppershift keys

N/A Not applicable

Ⓐ Command Entry Display, Programmer, System Operator, System Request, and Configuration Menus, and within prompting for embedded commands.

Ⓑ Prompting and Ⓐ.

DFU Definition Command Function Key Summary

Name of Key	CF No.	Typewriter-Like Keyboard	Function of Key
Add After	CF9	(9	Adds an entry format after the entry format prompt currently shown (modify mode).
Add Before	CF8	* 8	Adds an entry format before the entry format prompt currently shown.
Advance	CF10	) 0	Allows you to advance to the next prompt (at the same level) or to the modify menu.
Delete	CF11	— -	Allows you to delete: – An entry format – An audit control
Display Messages	CF6	¬ 6	Displays the work station messages; pending operations are suspended.
Exit Application	CF1	| 1	Displays DFU menu from any DFU display; pending operations are ignored. Exits DFU when DFU menu is displayed. When using interactive definition prompts, Exit key displays exit application definition menu.
Fold/Truncate	CF3		Reverses the format of the Review Application Fields and the Review DDS displays. The format of the displays is changed from truncated to folded or from folded to truncated. This function is meaningful only on the 16-line display (console).
Previous Display	CF2	@ 2	Returns you to the previous display; pending operations are ignored.
Review Application Fields	CF14	▨ @ 2	Displays the data fields that the current application refers to.
Review DDS	CF15	▨ # 3	Displays all fields contained in record formats that your definition refers to.
Review Format	CF16	▨ $ 4	Displays the execution-time format as it is currently being defined or modified.
Status	CF13	▨ | 1	Displays the current status of the definition process.

Legend:

▨ Shaded keys are uppershift keys

DFU Execution Command Function Key Summary

Name of Key	CF No.	Typewriter-Like Keyboard	Function of Key
Add	CF9	(9	Displays a skeleton data entry format.
Auto Duplicate On/Off	CF6	¬ 6	Reverses the status of the auto dup indicator (A or off).
Auto Increment On/Off	CF7	& 7	Reverses the status of the auto increment indicator (I or off).
Auto Record Advance On/Off	CF18	▨ ¬ 6	Reverses the status of the auto record advance indicator.
Change	CF10	) 0	Displays a key skeleton format to allow you to request the record to be changed.
Data Error Detection On/Off	CF19	▨ & 7	Reverses the status of the data error detection indicator.
Delete	CF11	— .	When viewing records in display mode, pressing the Delete key deletes the first (or only) record displayed. Otherwise, pressing the Delete key displays a skeleton format to enter a key field of a record to be deleted.
Exit Application	CF1	| 1	Ends the current utility processing and displays a count of the utility actions performed.
Nullify	CF22	▨) 0	Cancels the most recently entered delete request(s).
Print Accumulators	CF17	▨ % 5	Displays the accumulator subtotals for the current record(s) and writes them in the QDTAPRT print file. If record auditing is occuring, these subtotals are also written to the QDTALOG print file.
Print Records	CF15	▨ # 3	Prints the currently displayed record(s).
Display	CF8	* 8	Displays a skeleton format to allow entry of the key fields for the records to be displayed.
Review Application Fields	CF14	▨ @ 2	Displays labels and descriptions of the fields in the current application format.
Search Next On Format	CF20	▨ * 8	Locates the next record that can be displayed in the same format as the format being displayed or a new primary format ID is modified.
Select Format	CF5	% 5	Allows you to select another primary format.
Status	CF13	▨ | 1	Displays information about the current application.

Legend:

▨ Shaded keys are uppershift keys

Query Definition Command Function Key Summary

Name of Key	CF No.	Typewriter-Like Keyboard	Function of Key
Add After	CF9	(9	Creates a new prompt from changes keyed on an existing prompt, and adds the new prompt to the definition of a query application. The new prompt is placed in the definition after the original prompt, and the original prompt is restored in the definition.
Add Before	CF8	* 8	Creates a new prompt from changes keyed on an existing prompt, and adds the new prompt to the definition of a query application. The new prompt is placed in the definition before the original prompt, and the original prompt is restored in the definition.
Advance	CF10	) 0	In a sequence of modifications to a definition, displays the next primary prompt, if any, that is related to the last option you chose on the query modify menu.
Delete	CF11	_ -	Deletes an entire prompt from the definition of a query application. If the deleted prompt required responses to subsequent prompts, those prompts are also deleted.
Display Activity Log	CF7	▨ & 7	Displays the query activity log; pending operations are suspended.
Display Messages	CF6	¬ 6	Displays the work station messages; pending operations are suspended.
Exit Application	CF1	| 1	Displays query menu from any query display; pending operations are ignored. Exits query when the query menu is displayed. When using the interactive definition prompts, displays an exit application definition menu.
Previous Display	CF2	@ 2	Returns to previous display; pending operations are ignored.
Review Application Fields	CF14	▨ @ 2	Displays a description of the fields selected or specified in the definition of a query application.
Review DDS	CF15	▨ # 3	Displays a description of all record formats defined for the file referred to by a query application, and permits a review of field descriptions for each record format.
Status	CF13	▨ | 1	Displays the status of the definition of a query application.

Legend:

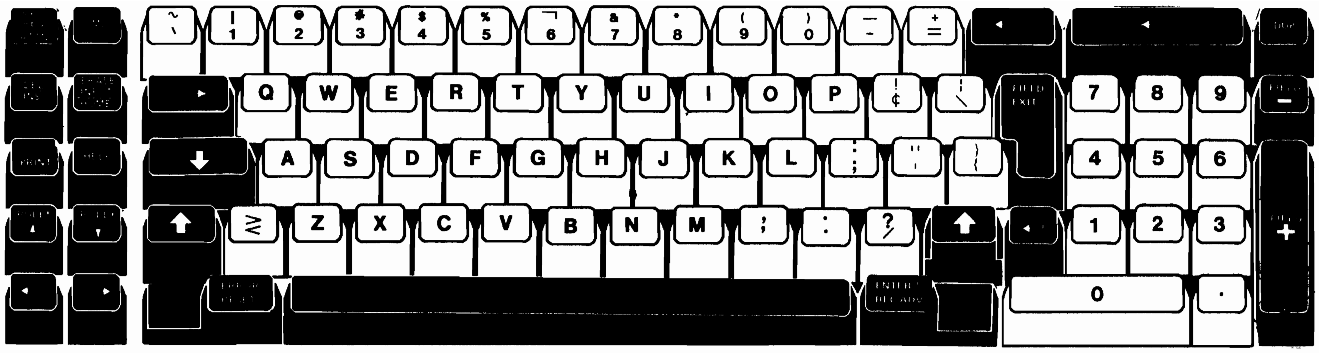
▨ Shaded keys are uppershift keys

SEU Command Function Key Summary

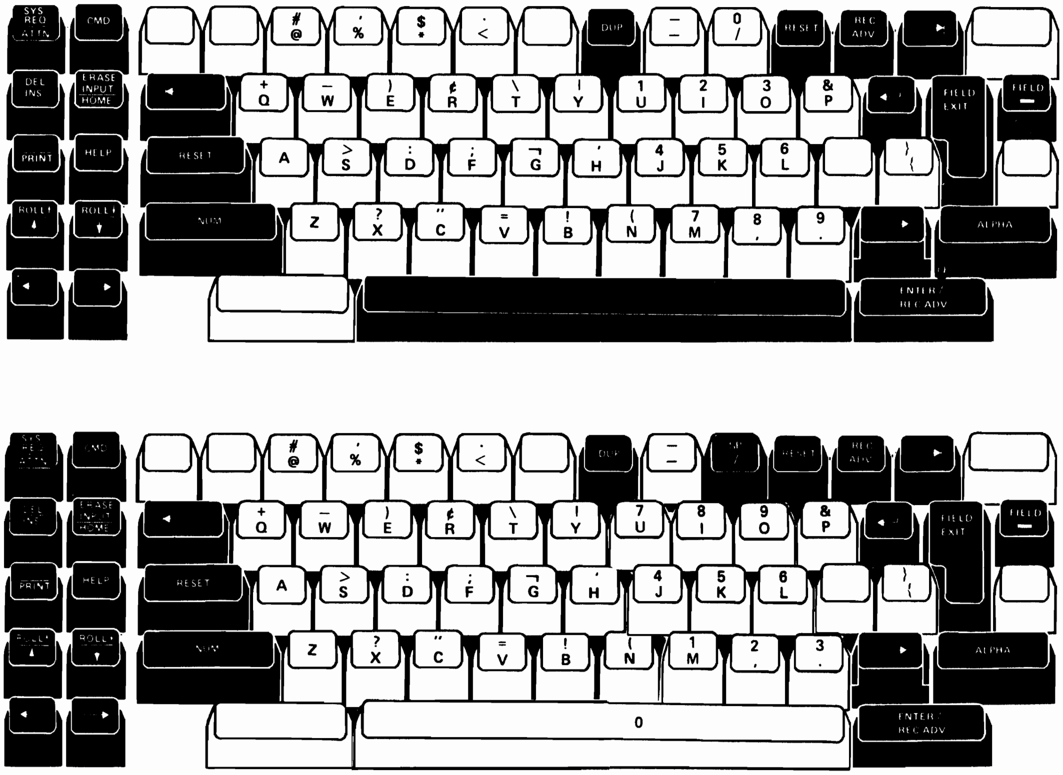
Name of Key	CF No.	Typewriter-Like Keyboard	Function of Key
Cancel Pending Operation	CF6	¬ 6	Cancels pending operations.
Exit SEU	CF1	| 1	Displays the SEU exit display. (If you press CF1 while viewing the help text, the Exit SEU key returns you to the display you were viewing when you pressed the Help key.)
Prompt	CF4	$ 4	For non-CL source types, creates a prompting section that contains the record indicated by the cursor position. For CL source types, invokes CPF prompting support to help you enter or change the control language command indicated by the cursor position.
Scan/Substitute Backward	CF8	* 8	Searches for the previous occurrence of a scan string and, if a substitute string is specified, substitutes the substitute string on next enter.
Scan/Substitute Forward	CF7	& 7	Searches for the next occurrence of a scan string and, if a substitute string is specified, substitutes the substitute string on next enter.
Services Display	CF5	% 5	Displays the SEU services display to allow you to perform the browse, scan/substitute, and syntax check operations.
Uppercase/ Lowercase	CF12	+ =	Reverses the status of the uppercase/lowercase indicator.

SDA Command Function Key Summary

Name of Key	CF No.	Typewriter-Like Keyboard	Function of Key
Condition Work Display	CF5	% 5	Displays the work display conditioning display. Used only on the work display.
Data Base Field Select	CF11	— .	Displays the data base file display. Used only on the work display.
Display Field List	CF10	) 0	Displays the field list display. Used only on the work display.
Exit/Save	CF1	| 1	Ignores information keyed in on a current display and exits to an option that enables you to save DDS or CL. Exits the help text displays.
Menu Definition Command Prompt	CF4	$ 4	Requests command name or command parameter prompting. Used only on the menu definition display.
Previous Display	CF2	@ 2	Saves changes, performs diagnostics, displays error messages, and backs up to the previous display in a series.
Print Work Display	CF6	¬ 6	Prints the current work display. Used only on the work display.
Subfile Display Size Prompt	CF9	(9	Displays a prompt on the bottom line of the work display that enables you to change the SFLLIN and SFLPAG sizes that were in effect for the current record format being defined. Used only on the work display.
Switch Field Name Lists	CF7	& 7	Displays on the bottom line of the work display a list of selected data base fields. Redisplays fields that were placed on the work display on the bottom line of the work display. Used only on the work display.
Work Display Ruler	CF8	* 8	Places a position, column ruler on the work display where you position the cursor. The ruler does not interfere with your data. Pressing CF8 again removes the ruler. Used only on the work display.



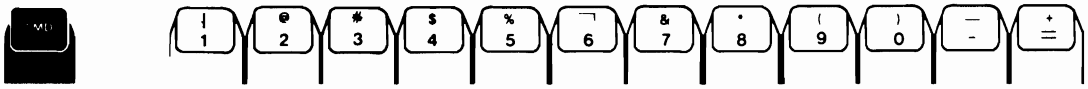
5251/5252 Typewriter-Like Keyboard



5251/5252 Data-Entry Keyboards

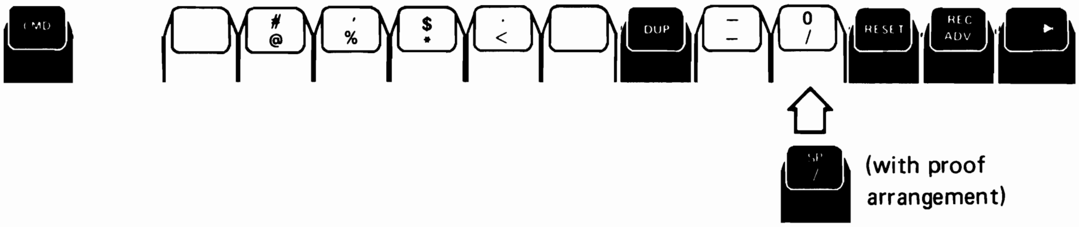
Typewriter-like Keyboard

Upper Shift	CF13	CF14	CF15	CF16	CF17	CF18	CF19	CF20	CF21	CF22	CF23	CF24
Lower Shift	CF1	CF2	CF3	CF4	CF5	CF6	CF7	CF8	CF9	CF10	CF11	CF12



Data-Entry Keyboard

Upper Shift	CF13	CF14	CF15	CF16	CF17	CF18	CF19	CF20	CF21	CF22	CF23	CF24
Lower Shift	CF1	CF2	CF3	CF4	CF5	CF6	CF7	CF8	CF9	CF10	CF11	CF12



(with proof arrangement)

CF Key Numbers for 5251/5252 keyboards

Appendix B. Problem Determination

SYSTEM FAILURE

The Indications

If a system failure occurs that results in the termination of your interactive job at a work station, you will usually be notified by a message, or messages, on your work station screen. You handle these termination messages in the same way that you handle any other message (see Chapter 3).

If you have tried to start an interactive job at a work station without success, the system could be down or merely unavailable for your use at present. The indications are:

- Your work station screen is blank except for the cursor and screen indicators.
- The System Available indicator on the work station screen is off; that is, it shows (-) instead of (■).
- You receive no response to your attempted entries.

What You Can Do

Contact the system operator or data processing manager to determine whether the system is down or unavailable for your use. The failure could be the system as a whole or the programming through which you are communicating with the system.

If you received a termination message, press the Help key to obtain the second-level message display for the message. The second-level text on the display will describe what to do.

If there was a system failure, refer to the recovery guidelines later in this chapter after the system is restarted.

WORK STATION FAILURE

The Indications

The problem is most likely in your work station if:

- The work station screen is completely blank (no cursor or screen indicators).
- The System Available screen indicator is on (■), but the characters you attempt to key in do not appear on the screen. (Remember, however, that information keyed into nondisplay fields such as the sign-on prompt will not appear on the screen.) If you are using a remote work station, the System Available indicator could be on (indicating that the local control unit is functioning) even though the system is not available.
- The Line Sync light is not on.

What You Can Do

If your work station screen is completely blank, ensure that the power cord is connected to a grounded power outlet that has power and that the Display Intensity Control is not turned all the way down. Also, if your display station has the Keylock feature, ensure that it is unlocked by turning the key clockwise approximately one-half turn (until it stops).

If you are unable to key information onto the screen, ensure that the cable connecting the keyboard to the base of the work station is in place and has not been damaged.

If there is another work station in your area, ensure that it is working properly. If the other work station is operating and in normal contact with the system, you know that the problem is probably isolated to your work station. On the other hand, if all other work stations are down, the problem is likely a general power failure, which may also have affected the system itself.

If the problem appears to be isolated to your work station, contact the system operator to determine if the system is waiting for you to respond to a message or whether the interactive job at your work station was terminated. If neither one is true, have the system operator cancel the interactive job. Obtain a copy of the job log from the system operator and look through the job log for any error message(s) that might indicate the cause of the problem.

If you are still unable to determine the cause of the problem, refer to the specific problem determination procedures in the separate 5251/5252 operator's guides (see *About This Manual*) or the *Problem Determination Guide.*

When the problem has been corrected, refer to the following recovery guidelines.

RECOVERY

When your work station and/or the system is back in operation, your work station screen will show the sign-on prompt.

Your previous interactive job was terminated, and you will have to sign on to start a new interactive job. Some data may have been lost.

The recovery actions you need to take are usually indicated by the second-level text for messages related to the termination of your interactive job. Other recovery actions are generally the responsibility of a programmer. Therefore, unless you have programmer responsibilities, you should request the assistance of a programmer in performing general recovery procedures.

The system operator can determine which jobs were being processed and which files were open at the time of an abnormal termination. The same type of information is contained in a job log that is usually printed for each job that is terminated abnormally.

Based on this information, you can (with the help of a programmer):

- Display the contents of open files to determine whether any data was lost. You could, for example, display the data in spooled output files by using the Display Spooled File (DSPSPLF) command.

- Submit again any requests that were not processed as a result of the termination.

- As needed, resubmit any batch jobs that were lost as a result of the termination. Jobs that were on a job queue (input or output) remain on the queue unless the queues are cleared when the system is restarted. However, jobs that were taken off a queue and were being processed when the termination occurred may have been lost.

If you were using a particular application when a termination occurred, refer to the documentation for that application.

The System/38 glossary defines terms that are used in the customer documentation for System/38.

This glossary includes definitions developed by the American National Standards Institute (ANSI) and the International Organization for Standardization (ISO). This material is reproduced from the *American National Dictionary for Information Processing*, copyright 1977 by the Computer and Business Equipment Manufacturers Association. Copies of this dictionary may be purchased from the American National Standards Institute, 1430 Broadway, New York, New York 10018. Definitions from this dictionary are identified by (ANSI) at the beginning of the definition.

abnormal termination: System termination by a means other than the successful execution of the Power Down System (PWRDWNSYS) command. See also *system termination.*

access path: The means by which CPF provides a logical organization to the data in a data base file so that the data can be processed by a program. See also *arrival sequence access path* and *keyed sequence access path.*

add rights: The authority to add an entry to an object. Contrast with *delete rights*, *read rights*, and *update rights.*

addressability: The ability to locate an object in online storage.

allocate: To assign a resource for use in performing a specific task. Contrast with *deallocate.*

American National Standard Code for Information Interchange: The standard code used for information interchange between data processing systems, data communications systems, and associated equipment. The code uses a coded character set consisting of 7-bit coded characters (8 bits including parity check). The set consists of control characters and graphic characters. Abbreviated ASCII.

arithmetic instruction: An instruction that specifies an arithmetic operation.

arrival sequence access path: An access path that is based on the order in which records are stored in a physical file. Contrast with *keyed sequence access path.*

ASCII: See *American National Standard Code for Information Interchange.*

attribute: A characteristic; for example, attributes of a field include its length and data type, and attributes of a job include its user name and job date.

authority: The right to access objects, resources, or functions.

authorization: The process of giving a user either complete or restricted access to an object, resource, or function.

autostart job: A job that is automatically initiated when a subsystem is started.

autostart job entry: A work entry in a subsystem description that specifies a job to be automatically initiated each time the subsystem is started.

auxiliary storage: All addressable storage other than main storage. Auxiliary storage is located in the system's nonremovable disk enclosures.

basic working display: The display that serves as the base from which you make requests of the system at a work station. When the request is completed, you return to the display. It is usually the display you receive when you sign on.

batch device: Any device that can read serial input or write serial output, or both, but cannot communicate interactively with the system. Examples of batch devices are card devices, printers, and diskette units.

batch job: A group of processing actions submitted as a predefined series of actions to be performed with little or no interaction between the user and the system.

batch processing: A method of executing a program or a series of programs in which one or more records (a batch) is processed with little or no interaction with the user or operator. Contrast with *interactive processing*.

branching: The technique of bypassing specific instructions or operations to alter the sequential execution of instructions in a program.

branching instruction: An instruction that changes the sequence of program execution.

breakpoint: A place in a program (specified by a command or a condition) where the system halts execution and gives control to the work station user or to a specified program.

byte: A group of eight adjacent binary digits that represents one EBCDIC character.

call: (1) To instruct that a program is to begin execution. (2) An instruction to a program to begin execution. (3) In data communications, the action necessary to make a connection between two stations on a switched line.

called program: A program whose execution is requested by another program (a calling program) or by a command.

calling program: A program that requests the execution of another program (a called program).

character string: (1) A string consisting of any of the 256 EBCDIC characters that are used as a value. (2) (ANSI) In COBOL, a sequence of characters that form a COBOL word, a literal, a PICTURE character-string, or a comment-entry.

CL: See *control language*.

class: An object that contains the execution parameters for a routing step. The system-recognized identifier for the object type is *CLS.

close: A data manipulation function that ends the connection between a file and a program. Contrast with *open*.

command: (1) A statement used to request a function of the system. A command consists of the command name, which identifies the requested function, and parameters. (2) In SNA, any field set in the transmission header (TH), request header (RH), and sometimes portions of a request unit that initiates an action or that begins a protocol.

command definition: An object that contains the definition of a command (including the command name, parameter definitions, and validity checking information) and identifies the program that performs the function requested by the command. The system-recognized identifier for the object type is *CMD.

command function key: At a work station, a keyboard key that is used with the command (CMD) function control key to request preassigned functions. At the system console, a keyboard key, called a CF key, that is used to request preassigned functions.

command processing program: A program that processes a command. This program performs some validity checking and executes the command so that the requested function is performed. Abbreviated CPP.

communications line: The physical link (such as a wire or a telephone circuit) that connects one or more work stations to a communications control unit, or connects one control unit to another. Contrast with *data link*.

compilation: Translation of a source program (such as RPG or COBOL specifications) into an executable program.

compiler listing: A printout that is produced by compiling a program or creating a file and that optionally includes, for example, a line-by-line source listing, a cross-reference list, diagnostic information, and for programs, the description of externally described files.

completion message: A message that conveys completion status of work.

control language: The set of all commands with which a user requests functions. Abbreviated CL.

control language program: An executable object that is created from source consisting entirely of control language commands.

control language variable: A program variable that is declared in a control language program and is available only to the program.

Control Program Facility: The system support licensed program for System/38. It provides many functions that are fully integrated in the system such as work management, data base data management, job control, message handling, security, programming aids, and service. Abbreviated CPF.

control unit: Circuitry or a device that coordinates and controls the operation of one or more input/output devices (such as work stations) and synchronizes the operation of such devices with the operation of the system as a whole. Same as *controller*.

controller: See *control unit*.

CPF: See *Control Program Facility*.

CPU: Central processing unit. See *processor*.

create: (1) The function used to bring an object into existence in the system. (2) To bring an object into existence in the system.

cursor: A movable spot of light, resembling a bright underscore, that shows where the next character will appear on the work station screen when a key on the keyboard is pressed.

data area: An object that is used to communicate data such as CL variable values between the programs within a job and between jobs. The system-recognized identifier for the object type is *DTAARA.

data base: The collection of all data base files stored in the system.

data base file: An object that contains descriptions of how input data is to be presented to a program from internal storage and how output data is to be presented to internal storage from a program. See also *physical file* and *logical file*.

data description specifications: A description of the user's data base or device files that is entered into the system using a fixed-form syntax. The description is then used to create files. Abbreviated DDS.

data file: Any nonsource file. A data file is created by the specification of FILETYPE(*DATA) on a create file command.

data file utility: The utility of the Interactive Data Base Utilities licensed program that is used to create, maintain, and display records in a data base file. Abbreviated DFU.

data link: The communications lines, modems, control units, work stations, and other communications equipment used for the transmission of data between a receiving station and a transmitting station in a data network. Contrast with *communications line*.

data rights: The authority to read, add, update (modify), or delete data contained in an object.

data type: An attribute used for defining data as numeric or character.

DDS: See *data description specifications*.

deallocate: To release a resource that is assigned to a specific task. Contrast with *allocate*.

default delivery: The method of delivering messages to a message queue in which messages are placed on the queue without interrupting the job, and the default reply is sent for any messages requiring a reply.

default user name: A CPF-provided name for user identification for an installation that does not want to require separate user identifications.

default value: A value assumed when no value has been specified.

delete rights: The authority to delete an entry from an object or to delete the object itself. Contrast with *add rights*, *read rights*, and *update rights*.

DFU: See *data file utility*.

display: A visual presentation of information on a work station screen, usually in a specific format. Display is often used as a shortened version of information display.

display screen: An electronic display tube, similar to a TV picture tube, used to display information entered or received at the system console or a work station.

display station: An input/output device containing a display screen and an attached keyboard that lets a user send information to or receive information from the system.

dump: To copy data in a readable format from main or auxiliary storage onto an external medium such as tape, diskette, or printer.

edit code: A letter or number indicating what kind of editing should be done before a field is displayed or printed.

edit description: An object that contains a description of a user-defined edit code. The system-recognized identifier for the object type is *EDTD.

edit word: A user-defined word with a specific format that indicates how editing should be done.

embedded command: A command that is specified as a value in a parameter of another command.

enter: To press the Enter/Rec Adv key (on a work station keyboard) or the Enter key (on the system console) or a command function key to transfer keyed-in information to the system for processing. See also *key in.*

execute: To cause a program, command, utility, or other machine function to be performed.

external storage: Data storage other than main or auxiliary storage.

externally described data: Data contained in a file for which the fields in the records are described to CPF, by using data description specifications, when the file is created. The field descriptions can be used by the program when the file is processed. Contrast with *program-described data.*

field: An area that is reserved and used for a particular item of information.

file: A generic term for the object type that refers to a data base file, a device file, or a set of related records treated as a unit. The system-recognized identifier for the object type is *FILE.

file overrides: The file attributes specified at execution time that will override the attributes specified in the file description or in the program.

first-level message: The initial message that is presented to the user. The initial message contains general information or designates an error. Contrast with *second-level message.*

function check: A notification (by a message) that an unexpected condition has stopped the execution of a program.

function check, machine: A notification of the malfunction of a machine instruction.

general-purpose library: The library provided by CPF to contain user-oriented, IBM-provided objects and user-created objects that are not explicitly placed in a different library when they are created. Named QGPL.

get operation: An input operation that obtains a record from an input file and passes it to a program.

handset: A telephone receiver and transmitter mounted in a handle.

help text: Information that is associated with an information display, a menu, or a prompt that explains options or values displayed. Help text is requested by pressing the Help key.

hexadecimal: Pertaining to a numbering system with a base of 16. Valid numbers are the digits 0 through 9 and the characters A through F, where A represents 10 and F represents 15.

high-level language: A programming language that relieves the programmer from the rigors of machine level or assembler level programming; for example, RPG III and COBOL. Abbreviated HLL.

high-level message: A message that is sent to the program message queue of the program receiving the request. The message is displayed or provided for the user who entered the request. Contrast with *low-level message.*

history log: A log of information about system status and events. Named QHST.

HLL: See *high-level language.*

hold delivery: The method of delivering messages to a message queue that holds the messages until the user requests them.

host system: The controlling or highest level system in a data communications configuration. For example, a System/38 is the host system for the work stations connected to it.

IDU: See *Interactive Data Base Utilities*.

impromptu message: A message that is created when it is sent. Contrast with *predefined message*.

informational message: A message that conveys information about the normal condition of a function.

inline data file: A file described by a //DATA command that is included as part of a job when the job is read from an input device by a reader program.

input: Information (or data) to be processed.

input-capable field: Any field that can receive input from a user.

input field: A field in a display file into which da*ι*a can be entered. An input field is passed from the device to the program when the program reads the record containing that field.

input stream: A group of records submitted to the system as batch input that contains CL commands for one or more jobs and/or the data records for one or more inline data files.

inquiry message: A message that conveys information and that requests a reply.

integrity: The protection of data and programs from inadvertent destruction or alteration.

interactive: Pertaining to a program or system that alternately accepts input and then responds. An interactive system is conversational; that is, a continuous dialog exists between the user and the system.

Interactive Data Base Utilities: A System/38 licensed program that consists of DFU, SEU, query, and SDA. Abbreviated IDU.

interactive job: A job in which the processing actions are performed in response to input provided by a work station user. During a job, a dialog exists between the user and the system.

interactive processing: Pertaining to a program or procedure that alternately accepts input and then responds to the input. Contrast with *batch processing*.

internal storage: All main and auxiliary storage in the system.

invoke: To instruct a specific program to start executing. Same as *call*.

job: A single identifiable sequence of processing actions that represents a single use of the system. A job is the basic unit by which work is identified on the system.

job log: A record of requests submitted to the system by a job, the messages related to the requests, and the actions performed by the system on the job. The job log is maintained by CPF.

job name: The name of a job as identified to the system. For an interactive job, the job name is the name of the work station at which the job was initiated; for a batch job, the job name is specified in the command used to submit the job. Contrast with *qualified job name*.

job number: A number assigned to a job as it enters the system to distinguish the job from other jobs.

job queue: An object that contains a list of batch jobs submitted to the system for execution and from which the batch jobs are selected for execution by CPF. The system-recognized identifier for the object type is *JOBQ.

job queue entry: A work entry in a subsystem description that specifies the job queue from which the subsystem can accept batch jobs and transferred jobs.

key field: A field in a record whose contents are used to sequence the records of a particular type within a file member.

key in: The action of pressing keys on a keyboard to specify information that is to be processed. See also *enter*.

keyed sequence access path: An access path to a data base file that is ordered on the contents of key fields contained in the individual records. Contrast with *arrival sequence access path*.

library: An object that serves as a directory to other objects. A library is used to group related objects and to find objects by name when they are used. The system-recognized identifier for the object type is *LIB.

library list: An ordered list of library names used to find an object. The library list indicates which libraries are to be searched and the order in which they are to be searched. The system-recognized identifier is *LIBL. *LIBL specifies to the system that a job's current library list is to be used to find the object.

list element: One of several values specified in a list parameter.

list parameter: A parameter defined to accept a list of multiple like values or unlike values.

local work station: A work station that is connected directly to System/38 without need for data transmission facilities. Contrast with *remote work station*.

logical file: A description of how data is to be presented to or received from a program. This type of data base file contains no data, but it provides an ordering and format for one or more physical files. Contrast with *physical file*.

logical record: (ANSI) In COBOL, the most inclusive data item. The level number for a logical record is 01.

low-level message: A message that is sent to the program message queue of the lower-level program invocation. A low-level message is normally not displayed. Contrast with *high-level message*.

machine interface: The instruction set and interface to the machine. The instruction set is called the System/38 instruction set. Abbreviated MI.

main storage: All storage in a computer from which instructions can be executed directly.

member: A description of a named subset of records in a physical or logical file. Each member conforms to the characteristics of the file and has its own access path. All I/O requests are directed to a specific member of a data base file.

menu: A display in which a list of options is shown.

message: A communication sent from one person or program to another person or program.

message description: The information describing a particular message. A message description is stored in a message file.

message file: An object that contains message descriptions. The system-recognized identifier for the object type is *MSGF.

message queue: An object on which messages are placed when they are sent to a person or program. The system-recognized identifier for the object type is *MSGQ.

MI: See *machine interface*.

mixed list: A series of unlike values for a parameter that accepts a set of separately defined values. Contrast with *simple list*.

modem: A mechanism that modulates and demodulates signals transmitted over data communications facilities.

nested command: A command or group of commands whose execution is conditioned on the evaluation of a preceding or associated command. Nesting is a structured form of branching. In CL programs, the nested command is embedded in an associated command. If the nested command is a DO command, the entire do group is nested.

normal termination: System termination that results from the successful execution of the Power Down System (PWRDWNSYS) command. Contrast with *abnormal termination*.

notify delivery: The method of delivering messages to a message queue in which the work station user is notified that a message is on the queue. The notification is by means of an attention light or an audible alarm.

notify message: A message that describes a condition for which a program requires a reply from its caller, or a default reply is sent to the program.

object: A named unit that consists of a set of attributes (that describe the object) and, in some cases, data. An object is anything that exists in and occupies space in storage and on which operations can be performed. Some examples of objects are programs, files, and libraries.

object authority: The right to use or control an object. See *object rights* and *data rights*.

object name: The name of an object. Contrast with *qualified object name*.

object owner: A user who creates an object or to whom the ownership of an object has been transferred. The object owner has complete control over the object.

object rights: The authority that controls what a system user can do to an entire object. For example, object rights include deleting, moving, or renaming an object. There are three types of object rights: object existence, object management, and operational.

object user: A user who has been authorized by the object owner, the security officer, or a user with object existence rights to perform certain functions on an object.

open: The function that connects a file to a program for processing. Contrast with *close*.

operator/service panel: A panel located adjacent to the system console on the system unit. This panel contains lights and switches that are used primarily when the system is started or serviced.

output/input field: A field in a display file that is used for both output and input operations.

output queue: An object that contains a list of output files to be written to an output device by a writer. The system-recognized identifier for the object type is *OUTQ.

overflow indicator: An indicator that signals when the overflow line on a page has been printed or passed. The indicator can be used to specify which lines are to be printed on tho next page.

page: (1) A 512-byte block of information that can be transferred between auxiliary storage and main storage. (2) Each group of records in a subfile that are displayed concurrently. (3) One printer form.

page fault: A program notification that occurs when a page that is marked as not in main storage is referred to by an active page.

page-in: The process of transferring a page from auxiliary storage to main storage.

page-out: The process of transferring a page from main storage to auxiliary storage.

parameter: (1) Data passed to or received from another program. (2) In CPF, an argument that identifies an individual value or group of values to be used by a command to tailor a function requested through the command.

password: A unique string of characters that a system user enters to identify himself to the system.

physical file: A description of how data is to be presented to or received from a program and how data is actually stored in the data base. A physical file contains one record format and one or more members. Contrast with *logical file*.

predefined message: A message whose description is created independently of when it is sent and is stored in a message file. Contrast with *impromptu message*.

priority: The relative significance of one job to other jobs in competing for allocation of resources.

problem determination: The process of determining the source of a problem as a component problem, a machine failure, a common carrier link, a user-supplied element, or a user error.

problem determination procedure: A prescribed sequence of steps taken to identify the source of a problem.

process: (1) An MI object consisting of a group of interdependent programs (activations and associated invocations) and the environment required for their execution. A process can be totally independent of other processes, or it can be dependent on and communicate with other processes. (2) In COBOL, any operation or combination of operations on data.

processor: The functional unit that interprets and executes instructions. Same as *CPU*.

program-described data: Data contained in a file for which the fields in the records are described in the program that processes the file. Contrast with *externally described data.*

prompt: A displayed request for information or user action. The user must respond to allow the program to proceed.

public authority: The authority to an object granted to all users.

put operation: An output operation that writes a record to an output file.

QGPL: See *general-purpose library.*

qualified job name: A job name and its associated user name and a system-assigned job number. Contrast with *job name.*

qualified object name: An object name and the name of the library containing the object. Contrast with *object name.*

query: (1) A utility that is part of the Interactive Data Base Utilities licensed program. (2) A request to extract, from a file, one or more records based upon some combination of data.

queue: A line or list formed by items in the system waiting for service; for example, work to be performed or messages to be displayed.

read rights: The authority to read the entries in an object. Contrast with *add rights, delete rights,* and *update rights.*

reader: A program that reads jobs from an input device or a data base file and places them on a job queue.

record: (1) An ordered set of fields that make up a single occurrence of the basic unit of data transferred between a file and a program. (2) In COBOL, a set of one or more related data items that are grouped for processing. Records can be defined for an input/output device or for internal processing. See also *logical record.*

record format: The definition of how data is structured in the records contained in a file. The definition includes the record name, field names, and field descriptions (such as length and data type). The record formats used in a file are contained in the file's description.

remote work station: A work station whose connection to the processing system uses modems and common carrier or private data transmission facilities. Contrast with *local work station.*

reserved word: (1) A word in the RPG language (such as *PRINT or UDATE) that is associated with a specific language function. A reserved word cannot be used as a programmer-defined symbolic name. (2) A special word that has a specific meaning to the system as defined in a programming language. (3) A predefined word used in a COBOL source program for syntactical purposes. It must not appear in a program as a user-defined name or system-name.

restore: To transfer specific objects or libraries from magnetic media such as diskettes or tape to internal storage by reconstructing them in internal storage. Contrast with *save.*

routing data: A character string that CPF compares with character strings in the subsystem description routing entries to select the routing entry that is to be used to initiate a routing step. Routing data can be provided by a work station user, specified in a command, or provided through the job description for the job.

routing entry: An entry in a subsystem description that specifies the program to be invoked to control a routing step that executes in the subsystem.

routing step: The processing performed as a result of invoking a program specified in a routing entry.

save: To duplicate specific objects or libraries by transferring them from internal storage to magnetic media such as diskettes or tape. Contrast with *restore.*

screen design aid: The utility of the Interactive Data Base Utilities licensed program that is used to interactively design, create, and maintain display record formats and menus. Abbreviated SDA.

SDA: See *screen design aid.*

second-level message: A message that provides additional information to that already provided in a first-level message. See also *second-level message display.*

second-level message display: A display containing the second-level message text (if any) and additional message information. This display is obtained by pressing the Help key while a first-level message is displayed.

security: The control of access to, or use of, data or functions.

security officer: The individual at an installation who is designated to control the authorization of functions and data in System/38.

security officer user profile: The CPF-supplied user profile that has authority to control the authorization of functions and data used in the installation. Named QSECOFR.

sequential-by-key processing: A method of file processing that reads records from a keyed sequence file in the order in which the keys are arranged in the access path.

sequential file: A file in which records are processed in the order that they are stored in the file.

SEU: See *source entry utility.*

shared access path: An access path used by more than one file to provide access to data common to the files.

shared record format: A record format that is used in more than one externally described file.

simple list: A list of like values, for example, a list of user names. Contrast with *mixed list.*

single value: A value that can be specified in place of multiple values for a list parameter in a CL command.

source entry utility: The utility of the Interactive Data Base Utilities licensed program that is used to create and change source members. Abbreviated SEU.

source file: A file created by the specification of FILETYPE(*SRC). A source file can contain source statements for such items as high-level language programs and data description specifications.

source listing: A portion of a compiler listing that contains source statements and diagnostics. See also *compiler listing.*

source statement: A statement written in symbols of a programming language. For example, RPG, COBOL, or DDS specifications are source statements.

special authority: The right to perform certain system control operations, such as save system and job control operations.

spooled file: A generic term for three types of files: a device file that provides access to an inline data file or that creates a spooled output file, an inline data file, or a spooled output file.

spooling: The CPF-provided execution-time support that reads and writes input and output streams on an intermediate device in a format convenient for later processing or output.

string: A linear sequence of entities such as characters.

subfile: A group of records of the same record format that can be displayed concurrently at a work station. The system sends the entire group of records to the work station in a single operation and receives the group in another operation.

subsystem: An operating environment, defined by a subsystem description, through which CPF coordinates work flow and resource usage.

syntax checking: A function of the command analyzer, a compiler, or SEU that checks single statements for violations of the rules governing the structure of the statement.

system console: The keyboard and display screen on the system unit that serve as a work station for communicating with and controlling the system. See also *operator/service panel* and *work station.*

system library: The library provided by CPF to contain system-oriented objects provided as part of CPF. Named QSYS.

system operator: The person who operates the system and looks after the peripheral equipment necessary to initiate computer runs or finalize the computer output in the form of completed reports and documents.

system operator message queue: The message queue used by the system operator to receive and reply to messages from the system, work station users, and application programs. Named QSYSOPR.

system termination: The state in which all processing on the system is stopped. Depending on the cause of the termination, system power could be shut off (such as by a power interruption or by entering the Power Down System (PWRDWNSYS) command) or could remain on (such as caused by a machine error condition). See also *abnormal termination* and *normal termination*.

tag: One or more characters, attached to a set of data, that contains information about the set, including its identification.

temporary library: A library that is automatically created for each job to contain temporary objects that are created by that job. The objects in the temporary library are deleted when the job ends. Named QTEMP.

termination: The act of putting the system or an element of the system (such as CPF or a subsystem) in the state where it no longer performs its normal function. See also *system termination*.

trace: The process of recording the sequence in which the statements in a program are executed and, optionally, the values of the program variables used in the statements.

typamatic key: A key that repeats its function when pressed and held down.

update rights: The authority to change the entries in an object. Contrast with *add rights*, *delete rights*, and *read rights*.

user identification: System recognition of a system user so that only the facilities and data he is authorized to use are made available to him.

user name: The name by which a particular user is known to the system.

user password: A unique string of characters that a system user enters to identify himself to the system.

user profile: An object that contains a description of a particular user or group of users. A user profile contains a list of authorizations to objects and functions. The system-recognized identifier for the object type is *USRPRF.

validity checker: A user-written program that tests commands for errors in the parameter values. Validity checking is done in addition to the checking done by the command analyzer.

validity checking: Operations performed against a field value to ensure that the field contains appropriate data. Checking can be done on a single field (for example, the field must be plus) or on multiple fields (for example, if FLDA contains a 1, FLDB can contain only a 2 or 3).

variable: A named modifiable value. The value can be accessed or modified by referring to the name of the variable.

virtual storage: The combination of main storage and auxiliary storage, treated as a single addressable unit. Abbreviated VS.

VS: See *virtual storage*.

work entry: An entry in a subsystem description that specifies a source from which jobs can be accepted to be executed in the subsystem.

work station: A device that lets a person transmit information to or receive information from a computer as needed to perform his job.

work station entry: A work entry in a subsystem description that specifies the work stations from which users can sign on to the subsystem or from which interactive jobs can transfer to the subsystem.

work station message queue: A message queue that is associated with a particular work station and that is used for sending and receiving messages sent to the work station. The name of the message queue is the same as the name of the work station.

writer: A CPF program that writes spooled output files from an output queue to an external device, such as a printer.

Index

IBM System/38 Programmer's/User's Work Station Guide

READER'S COMMENT FORM

SC21-7744-2

Please use this form only to identify publication errors or to request changes in publications. Direct any requests for additional publications, technical questions about IBM systems, changes in IBM programming support, and so on, to your IBM representative or to your nearest IBM branch office.

□ If your comment does not need a reply (for example, pointing out a typing error) check this box and do not include your name and address below. If your comment is applicable, we will include it in the next revision of the manual.

□ If you would like a reply, check this box. Be sure to print your name and address below.

Page number(s): Comment(s):

Please contact your nearest IBM branch office to request additional publications.

Name ____________________________

Company or
Organization ____________________________

Address ____________________________

____

City State Zip Code

IBM may use and distribute any of the information you supply in any way it believes appropriate without incurring any obligation whatever. You may, of course, continue to use the information you supply.

No postage necessary if mailed in the U.S.A.

SC21-7744-2

Cut Along Line

Fold and tape

Please do not staple

Fold and tape

NO POSTAGE
NECESSARY IF
MAILED IN THE
UNITED STATES

BUSINESS REPLY MAIL

FIRST CLASS PERMIT NO. 40 ARMONK, N. Y.

POSTAGE WILL BE PAID BY . . .

IBM CORPORATION
General Systems Division
Development Laboratory
Publications, Dept. 245
Rochester, Minnesotá 55901

Fold and tape

Please do not staple

Fold and tape

IBM System/38 Programmer's/User's Work Station Guide (File No. S38-40) Printed in U.S.A. SC21-7744-2

IBM System/38 Programmer's/User's Work Station Guide

READER'S COMMENT FORM

SC21-7744-2

Please use this form only to identify publication errors or to request changes in publications. Direct any requests for additional publications, technical questions about IBM systems, changes in IBM programming support, and so on, to your IBM representative or to your nearest IBM branch office.

□ If your comment does not need a reply (for example, pointing out a typing error) check this box and do not include your name and address below. If your comment is applicable, we will include it in the next revision of the manual.

□ If you would like a reply, check this box. Be sure to print your name and address below.

Page number(s): Comment(s):

Please contact your nearest IBM branch office to request additional publications.

Name ______________________________

Company or
Organization ______________________________

Address ______________________________

____

City State Zip Code

IBM may use and distribute any of the information you supply in any way it believes appropriate without incurring any obligation whatever. You may, of course, continue to use the information you supply.

No postage necessary if mailed in the U.S.A.

Cut Along Line

Fold and tape

Please do not staple

Fold and tape

NO POSTAGE
NECESSARY IF
MAILED IN THE
UNITED STATES

BUSINESS REPLY MAIL

FIRST CLASS PERMIT NO. 40 ARMONK, N. Y.

POSTAGE WILL BE PAID BY . . .

IBM CORPORATION
General Systems Division
Development Laboratory
Publications, Dept. 245
Rochester, Minnesota 55901

Fold and tape

Please do not staple

Fold and tape

IBM

IBM®

IBM System/38 Programmer's/User's Work Station Guide (File No. S38-40) Printed in U.S.A. SC21-7744-2

SC21-7744-2