SC21-7735-3

File No. S38-40

IBM System/38

IBM System/38 Operator's Guide

Program Number 5714-SS1

Fourth Edition (September 1982)

This is a major revision of, and obsoletes, SC21-7735-2 and SN21-8233. Changes or additions to the text and illustrations are indicated by a vertical line to the left of the change or addition. The following major technical changes have been made:

New displays and options available from the new displays have been added.

In addition, miscellaneous technical changes have been made throughout the manual.

This edition applies to release 4, modification 1 of the IBM System/38 Control Program Facility (Program 5714-SS1) 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.

This publication might 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.

Any reference to an IBM program product in this publication is not intended to state or imply that only IBM's program product may be used. Any functionally equivalent program may be used instead.

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 whatever. You may, of course, continue to use the information you supply.

©Copyright International Business Machines Corporation 1980, 1981, 1982

Contents

About This Manual

PURPOSE OF THIS MANUAL

This manual provides the information you need to operate the IBM System/38. The manual describes the following:

- How to operate the equipment on the IBM 5381 System Unit:
  - The operator/service panel
  - The system console
  - The diskette magazine drive
- Where to obtain information on devices attached to or communicating with System/38:
  - The IBM 5211 Printer
  - The IBM 3203 Printer
  - The IBM 3262 Printer
  - The IBM 5424 Multi-Function Card Unit (MFCU)
  - The IBM 3410/3411 Magnetic Tape Units
  - The components of the IBM 5250 Information Display System
- How to use the operator/service panel and the system console to prepare System/38 for operation.
- How to use Control Program Facility (CPF) commands and displays on the system console or another work station to:
  - Display information about the system
  - Start and stop subsystems
  - Control system equipment
  - Submit batch jobs for processing
  - Control individual jobs or groups of jobs that are using spooling
  - Send and display messages
  - Save and restore information
  - Obtain information for servicing
  - Terminate the system

The procedures described in this manual are based on the use of the CPF programming that is available from IBM. If your system does not use CPF, refer to the operating procedures for the programming being used.

Generally, you should be able to use this manual without prior knowledge of System/38; however, you should be familiar with the concepts in the prerequisite publication (see *What You Should Know*). A basic knowledge of data processing would be helpful in understanding some of the procedures in this manual. It is recommended that this manual be used in conjunction with introductory instruction in System/38 operations.

Note: This publication follows the convention that *he* means *he* or *she*.

WHAT YOU SHOULD KNOW

IBM System/38 Introduction, GC21-7728, provides an overview of the system equipment and capabilities.

IF YOU NEED MORE INFORMATION

IBM System/38 CL Reference Manual, SC21-7731, describes commands and parameters that are used for the various CPF functions.

IBM System/38 CPF Programmer's Guide, SC21-7730, explains how to use control language commands.

IBM System/38 Messages Guide: CPF, RPG III, IDU, SC21-7736, contains all messages, displayed and printed, pertaining to CPF and provides error recovery procedures related to those messages.

IBM System/38 Guide to Program Product Installation and Device Configuration, GC21-7775, describes programming installation procedures.

IBM System/38 Guide to Publications, GC21-7726, contains the titles and reading sequence of related publications. It also 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 3203 Printer Model 5 Component Description and Operator's Guide, GA33-1529, describes how to operate the 3203 Printer.

IBM 5211 Printer Models 1 and 2 Component Description and Operator's Guide, GA24-3658, describes how to operate the 5211 Printer.

IBM 3262 Printer Models A1 and B1 Component Description and Operator's Guide, GA33-1530, describes how to operate the 3262 Printer.

IBM 5424 and 5425 Multi-Function Card Units Operator's Guide and Programmer's Reference Manual, GA21-9167, describes how to operate the 5424 MFCU.

IBM 3410/3411 Operator's Guide, G232-0004, describes how to operate the 3410/3411 magnetic tape units.

IBM 5251 Models 1 and 11 Display Station, IBM 5252 Dual Display Station Operator's Guide, GA21-9248, describes how to operate directly attached display stations.

IBM 5251 Models 2 and 12 Display Station Operator's Guide, GA21-9323, describes how to operate remote display station/control units.

IBM System/38 Programmer's/User's Work Station Guide, SC21-7744, describes how to use the 5251 or 5252 display station as a System/38 work station. It describes the function of the programmer's menu, command prompting, and system request menu.

IBM 5256 Printer Operator's Guide, GA21-9260, describes how to operate the work station printer.

The IBM Diskette General Information Manual, GA21-9182, describes IBM diskettes and their handling.

Notes:

1. A set of insert tabs is included with this manual to help you locate each chapter quickly. You should insert the tabs between the chapters.
2. Requests for replacement insert tabs (SX21-9319) should be made to your IBM representative or the IBM branch office serving your locality.

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 or spooling functions that do not work as expected, or devices that do not work as expected.

IBM 5224 Printer Models 1 and 2 Operator's Guide, GA34-0092, describes how to operate the work station printer.

IBM 5225 Printer Models 1, 2, 3, and 4 Operator's Guide, GA34-0054, describes how to operate the 5225 Printer.

Your Role as the System Operator

As the System/38 operator, you are responsible for ensuring that work is processed in an orderly and efficient manner. Your duties usually include preparing the system for operation, controlling overall system operations, and supplying system equipment with necessary materials (paper, cards, diskettes).

To prepare the system for operation, you perform these functions:

- Set the switches on the operator/service panel to start a power-on sequence.

- Respond to displays on a special work station called the system console to complete the start-up of the system control programming.

Once the system is ready for operation, you control it by entering *commands* using the keyboard and display screen of the system console or another work station. All of the following actions that you might take in operating the system are done by entering commands:

- Display information about the status of the system.

- Start and stop subsystems that provide specialized operating environments for various types of work.

- Start, temporarily hold, and then resume or cancel specific input/output operations.

- Submit batch jobs.

- Hold and then release or cancel individual jobs as well as groups of jobs intended for a particular type of processing or output.

- Temporarily remove a device, control unit, or communication line from control of the system to correct a problem and then put it back under system control.

- Make a copy of information on one or more diskettes for storage away from the system and then return that information to the system from the diskettes.

- Send messages to and display messages from the system users and programs running on the system.

- Terminate the system.

Each command you are likely to use in operating the system is described individually in this manual. Whether you enter the commands separately or in groups depends on how your system is set up. Commonly used combinations of commands might be grouped together so that you can execute them as a unit. Some commands or groups of commands might be activated automatically when you start the system or a subsystem.

Much of the information you receive on system conditions is in the form of messages that appear on the screen of your work station. As a result of a message or as defined in your operating instructions, you may need to perform the following functions:

- Load or change forms, change a print belt, or replace a ribbon in a printer

- Load or unload cards, clear a card jam, replace a ribbon, or empty the chip box in a card unit

- Mount individual diskettes or magazines of diskettes in the diskette magazine drive

- Mount reels of magnetic tape in tape units

Before attempting System/38 operations, you should be familiar with the procedures in this operator's guide.

This guide is divided into two parts:

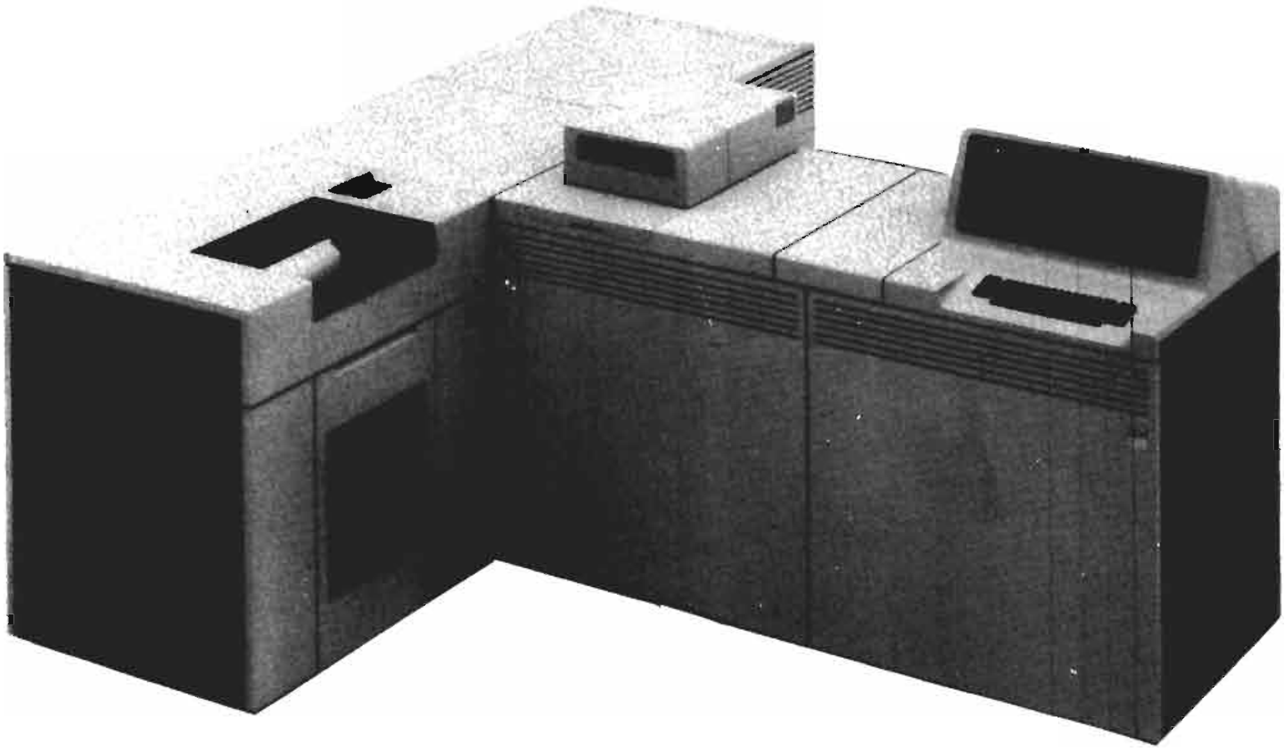
- *Part I. System Equipment* describes the basic system equipment and refers to the appropriate documents for information on devices that can be attached to the system.

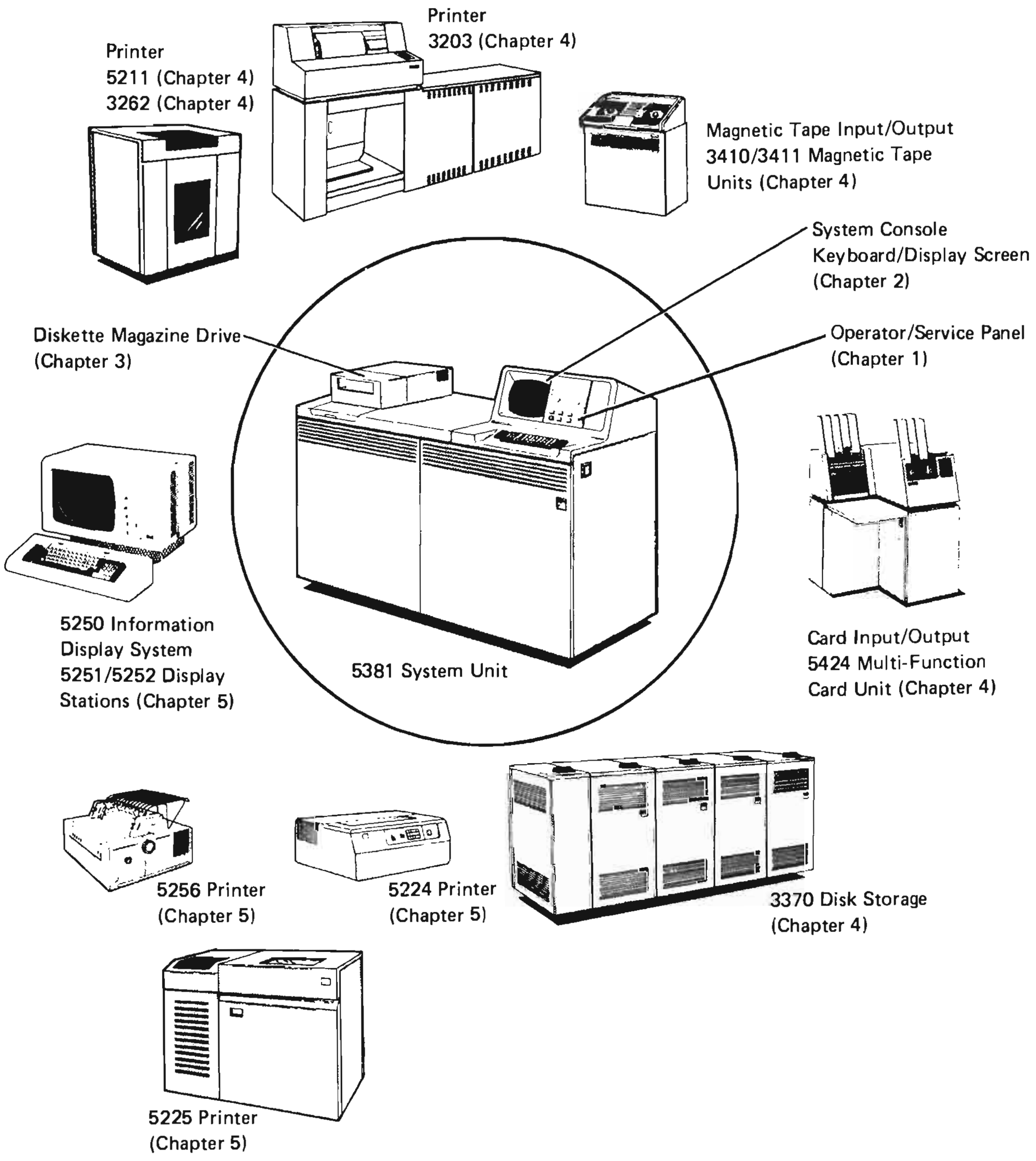
- *Part II. System Operation* describes how to operate and control the system.

You should be familiar with Part I and the documents referenced therein before proceeding to Part II. You should be familiar with Chapter 6 in Part II before proceeding to the remainder of Part II. Chapter 7 in Part II summarizes all the commands discussed throughout this manual.

The *System/38 Introduction* provides a good overview of system equipment and capabilities. Certain sections of this operator's guide assume you have read the Introduction.

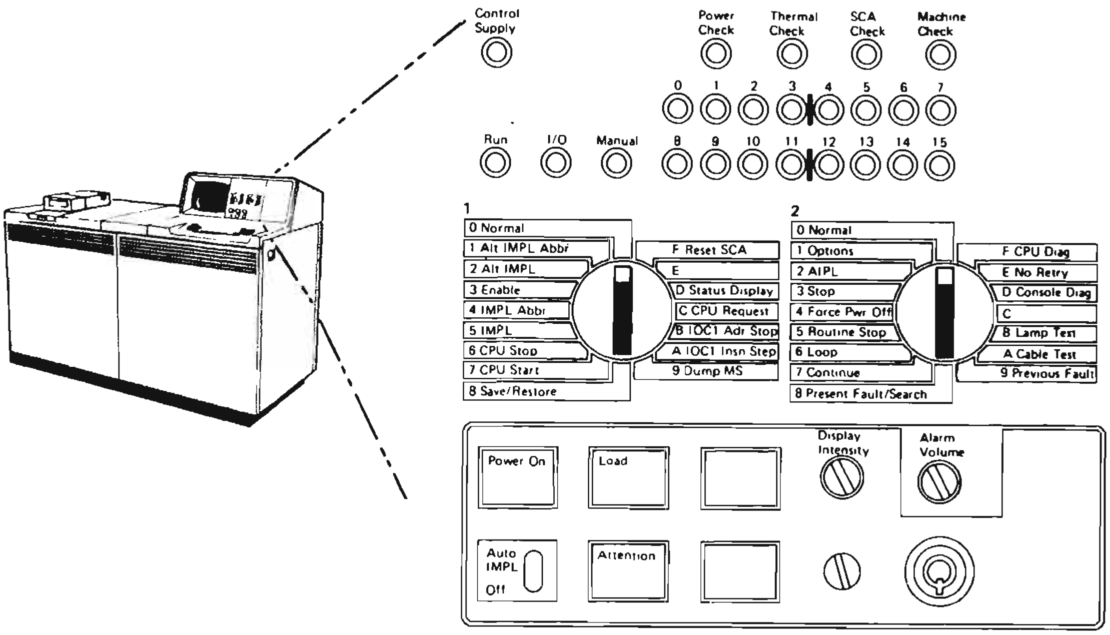
PART I
SYSTEM EQUIPMENT




Printer
5211 (Chapter 4)
3262 (Chapter 4)
Printer
3203 (Chapter 4)
Magnetic Tape Input/Output
3410/3411 Magnetic Tape
Units (Chapter 4)
System Console
Keyboard/Display Screen
(Chapter 2)
Diskette Magazine Drive
(Chapter 3)
Operator/Service Panel
(Chapter 1)
5250 Information
Display System
5251/5252 Display
Stations (Chapter 5)
5381 System Unit
Card Input/Output
5424 Multi-Function
Card Unit (Chapter 4)
5256 Printer
(Chapter 5)
5224 Printer
(Chapter 5)
3370 Disk Storage
(Chapter 4)
5225 Printer
(Chapter 5)

Chapter 1. IBM 5381 System Unit Operator/Service Panel

The operator/service panel (below) is located on the system console (see Chapter 2). It contains lights and switches that are used for communication between you and the system. Lights indicate conditions in the processing unit or the system. Switches are used to control operations of the system. You use this panel to start the system.



SYSTEM MODES

The system can be in one of two possible modes that affect how it responds to certain switch settings on the operator/service panel. The two modes are:

- User (normal) mode
- Maintenance mode

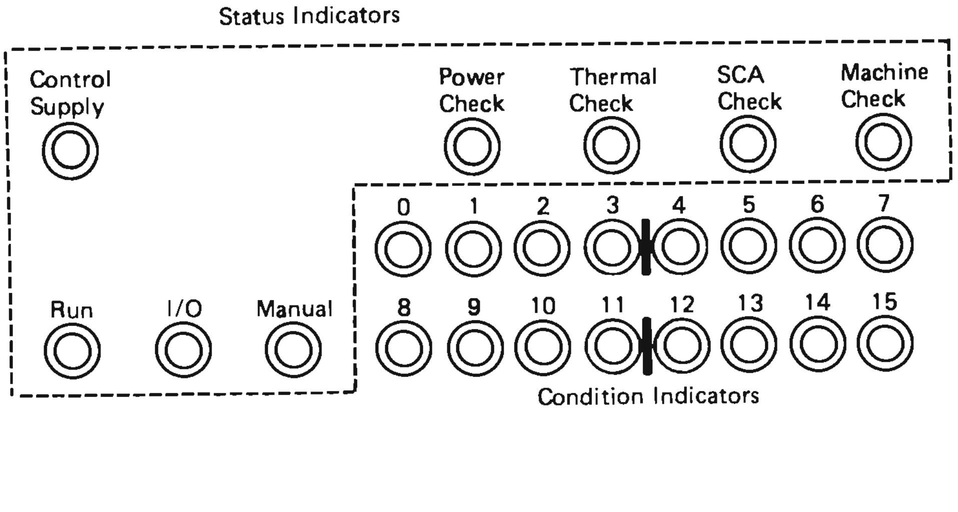
Normal operations are carried out in the user mode. The system is automatically in the user mode when started and remains in that mode unless a switch under the front cover of the processing unit is set to activate the maintenance mode. Your service representative uses the maintenance mode to diagnose problems with the system.

Note: Do not attempt to set the system in the maintenance mode unless you are specifically instructed to do so by your service representative. Attempting to operate the system in the maintenance mode could result in the loss of vital system information. You can ensure that the system remains in the user mode for normal operation by keeping closed all covers and access panels on the 5381 System Unit, except when loading diskettes in the diskette magazine drive (see Chapter 3).

LIGHTS

System Indicators

At the top of the operator/service panel are 24 indicator lights (below). The five on the top row and the three on the left of the lower row indicate machine status. The 16 lights numbered 0 through 15 indicate specific system conditions or faults, sometimes in combination with one or more of the eight status indicators.



Condition Indicators: During the IMPL (initial microprogram load) or one of the alternative sequences used to prepare the system for operation (see *Multifunction Rotary Switches*), these 16 lights flicker on and off or come on briefly. However, if any of these lights come on and remain on for more than a minute, along with the Manual indicator and/or one or more of the Power Check, Thermal Check, SCA Check, and Machine Check indicators, a fault has occurred in the system that has stopped the sequence. For some unusual conditions, these lights could also come on while the processor is still functioning. This is to inform you of an abnormal phase of machine processing that must be completed. The on/off pattern of the lights indicates a hexadecimal code that provides information about the problem or condition (see the *Problem Determination Guide* for the meaning of the hexadecimal codes).

Control Supply: This light is on when the main line circuit breaker is closed and the control supply power for the power controller is on. This light is usually on even when system power is off.

Power Check: This light turns on when a fault is detected in the power controller.

Thermal Check: This light turns on when a thermal switch trips, indicating that the safe operating temperature has been exceeded.

SCA (System Control Adapter) Check: This light turns on when a check is detected in the machine subsystem containing the system control adapter.

Machine Check: This light turns on when a machine failure is detected that causes a machine termination.

Run: This light is on when the processor is executing.

I/O: This light is on when any input/output device other than the SCA is operating. Examples of such devices are:

- The system console or any other work station
- A printer
- The diskette magazine drive
- The Multi-Function Card Unit
- Auxiliary storage (the permanent disk storage inside the 5381 System Unit and inside any 3370s on the system)

Manual: This light is on when processing has been stopped.

For problem determination procedures related to the operator/service panel indicators, see the *Problem Determination Guide*.

Attention and Switch Lights

There are two or three additional lights (depending on the features of your system) located in a group of controls at the bottom of the operator/service panel (see *Single-Function Controls*).

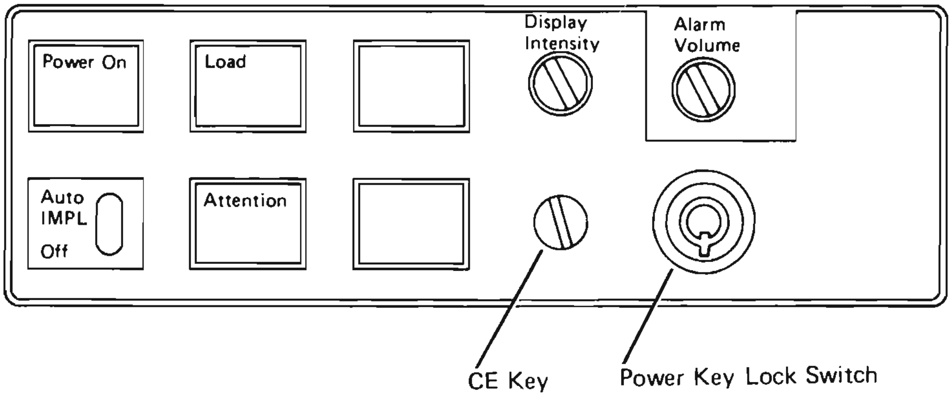
Attention: This light, which is part of the optional alarm feature, comes on when the Attention indicator on the system console screen comes on. The light turns off when the Attention indicator turns off.

Power-On Backlight: This light behind the Power On switch comes on when power is fully on to the system. It remains on as long as system power is on.

Load Backlight: This light behind the Load switch comes on when the switch is pressed, except for settings of the multifunction rotary switches that are used to determine power faults. It remains on until the switch is released and the system begins performing the function indicated by the setting on the multifunction rotary switches.

SINGLE-FUNCTION CONTROLS

At the bottom of the operator/service panel is a group of controls (below), each of which performs a separate function.



Power On Switch

When pressed, this switch initiates the power-on sequence. The actual sequence that takes place depends on how you set the two multifunction rotary switches discussed in the following section.

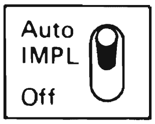
If the power is on, further pressing of the Power On switch is ignored and to execute the function indicated by the rotary switches, you must press the Load switch. (While the power is on, a backlight illuminates the switch.)

Load Switch

When this switch is pressed, the operations specified by the settings of the multifunction rotary switches are performed. If the positions of the rotary switches are not valid, pressing the Load switch results in no operation. When the Load switch is pressed, a backlight comes on to illuminate it and remains on briefly until the system begins the indicated operation.

Auto-IMPL Switch

If your system has the Auto-IMPL (automatic initial microprogram load) feature, the system will automatically initiate the power-on sequence after a power interruption and perform an IMPL (see *Normal Settings* later in this chapter). The switch must be set to Auto-IMPL.



The Auto-IMPL function is disabled by the machine under any of the following conditions:

- Auto-IMPL switch off

- System power down initiated by the Power Down System command with the RESTART parameter value *NO

- Forced power down by rotary switch

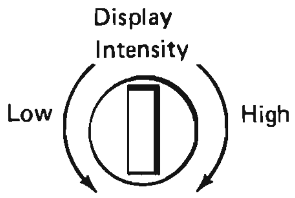
- Hardware failure detected

- System power down resulting from a power failure with the Power Warning feature using the system value QUPSMSGQ(*NONE)

Display Intensity Control

This control is used to vary the brightness of the system console screen.

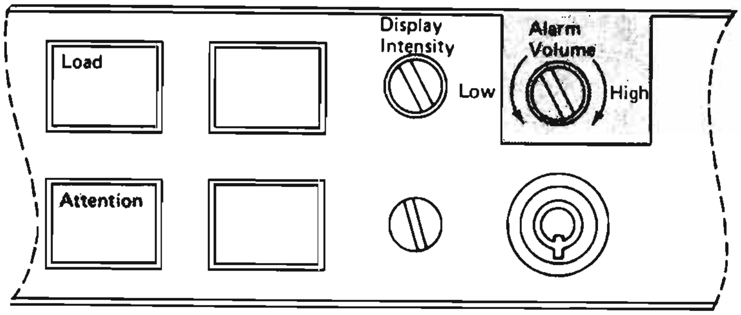
Turning the control to the right (clockwise) makes the screen brighter. Turning the control to the left (counterclockwise) makes the screen dimmer.



Alarm Volume Control/Attention Light

An alarm feature, consisting of an audible alarm and an Attention light, is available for the operator/service panel. The alarm and/or the Attention light come on under specific conditions to signal you that the system has a message for you. The conditions under which the alarm and Attention light come on are determined by system programming (see *Message Delivery Modes* in Chapter 11).

The Attention light comes on and turns off in conjunction with the Attention indicator on the system console screen. If the alarm and/or the Attention light are turned on to signal a message, they remain on until you display the message.



You can adjust the loudness of the alarm by using the Alarm Volume control. Turning the control to the right (clockwise) makes the alarm louder. Tuming the control to the left (counterclockwise) makes the alarm softer. You can make the alarm inaudible by turning the control fully counterclockwise.

If your system does not have the feature, there will be no Attention light or control to adjust the alarm volume.

CE (Service) Key

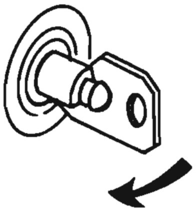
This lock is used only by the service representative while maintenance work is being done on an input/output device.

Power Key Lock Switch

Your system may have the optional Power Key Lock feature that prevents an unauthorized person from starting the system. When the Power Key Lock switch is in the locked position, the Power On switch is disabled.

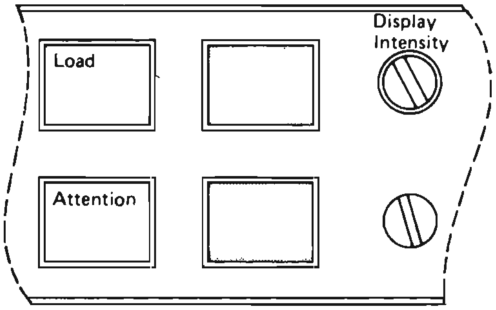
To unlock the Power On switch, insert the key in the Power Key Lock switch and turn it clockwise until it stops.

Locking the Power Key Lock switch has no effect after system power is on. Therefore, once the system has been powered on, you can lock the Power Key Lock switch again and remove the key.



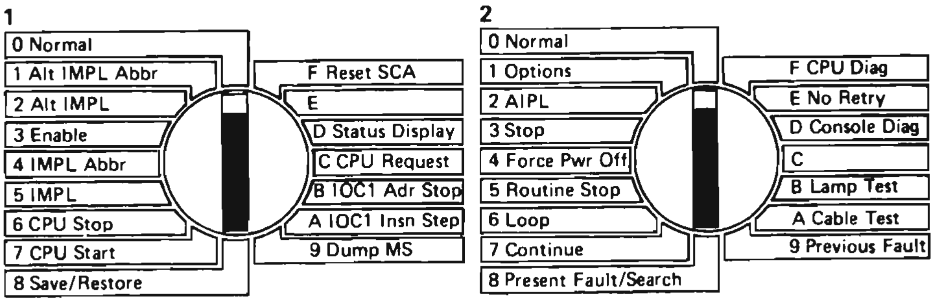
Reserved Switch Positions

There are two switch positions on the operator/service panel that are not used.



MULTIFUNCTION ROTARY SWITCHES

Between the indicator lights and the single-function controls are two rotary switches (below), each of which can be set to 16 possible positions. Each position represents a separate function, in some cases a sequence of operations the system is to perform.



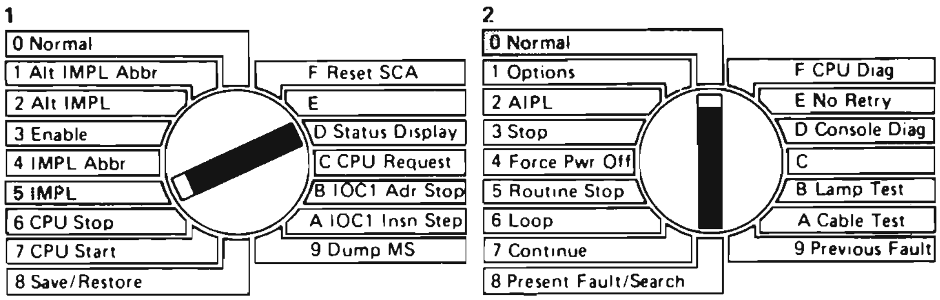
Setting the rotary switches does not, by itself, cause any action by the system. After selecting the desired function on the rotary switches, you must press the Load switch to activate the function. The IMPL sequence, or one of its variations (IMPL Abbr, Alt IMPL, Alt IMPL Abbr), that brings the system up to a working condition is also activated when you press the Power On switch to start the system. If both switches are normal when you power on, an IMPL is initiated.

The following summary describes the functions available to you by settings on the rotary switches. The settings not discussed are intended for use by service personnel and can be used only when the system is in the maintenance mode.

CAUTION:
After using any rotary switch setting, reset both switches to *0 Normal.* This will ensure that system operations are not disrupted in the event the Load switch is accidentally pressed. The Load switch is ignored when both rotary switches are set to *0 Normal.*

Normal Settings

IMPL



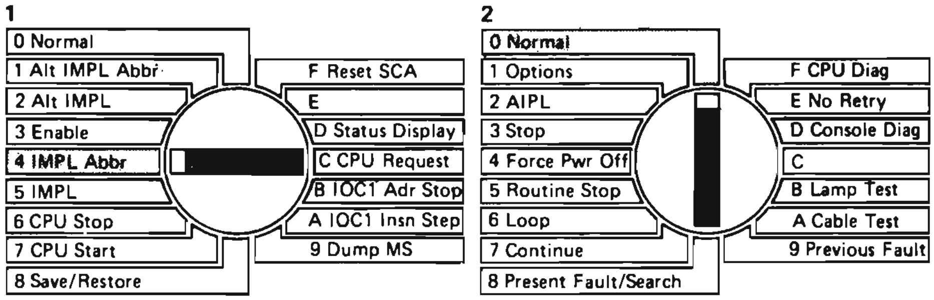
IMPL is used to perform an Initial Microprogram Load when power is already on to the system (Power On switch is lighted).

When the two rotary switches are set as shown and you press the Load switch (if power is already on), the system performs the IMPL (initial microprogram load) sequence. During the IMPL sequence, the microcode that controls fundamental machine operations is loaded from auxiliary storage. In addition, the hardware undergoes extensive diagnostic testing to ensure that the system is operating properly and all machine components are initialized (set to a known starting condition). After the microcode is loaded, the process begins that loads the Control Program Facility (CPF) or equivalent programming from auxiliary storage. The part of the sequence that loads CPF is referred to as the IPL (initial program load).

When the IMPL sequence is completed, the system is ready to begin normal operations, and a message display or the sign-on prompt appears on the system console screen (see *Preparing System/38 for Operation* in Chapter 8).

Note: If your system has the Auto-IMPL feature and the switch is set to Auto-IMPL, the system will automatically restart after a power interruption and perform the same IMPL sequence described here.

IMPL Abbr



If you need to do a second IMPL during a workday after the full IMPL sequence has already been done, you can use the abbreviated sequence provided by this setting.

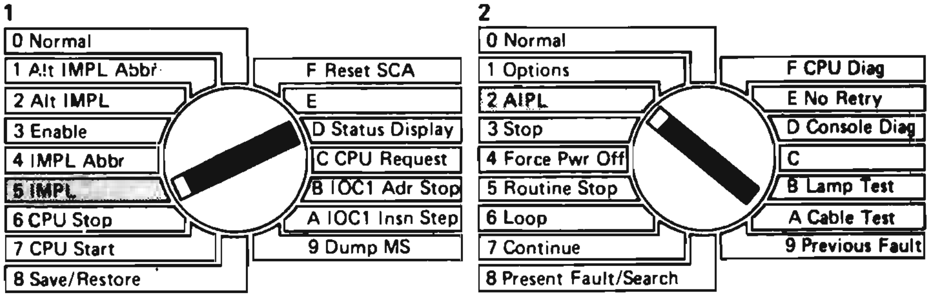
When the two rotary switches are set as shown and you press the Power On switch (if power is off) or the Load switch (if power is on), the system performs the IMPL Abbr (IMPL abbreviated) sequence. This sequence is the same as the IMPL sequence except that much of the hardware diagnostic testing is eliminated. When you use this setting, therefore, the system is ready for operation sooner than if you use the IMPL setting.

You should only use this setting when you are certain there are no hardware problems.

Settings to Install CPF

You use these settings to load a new or updated version of CPF into the system or to totally recover CPF after a system failure that resulted in the loss of essential CPF information.

IMPL with AIPL



This setting combines the normal IMPL with an AIPL (alternative initial program load) that copies CPF from diskette or tape into internal storage. Use this setting when you want to include the extensive hardware diagnostics, such as when starting the system at the beginning of a workday or when you want to ensure that the hardware is functioning properly.

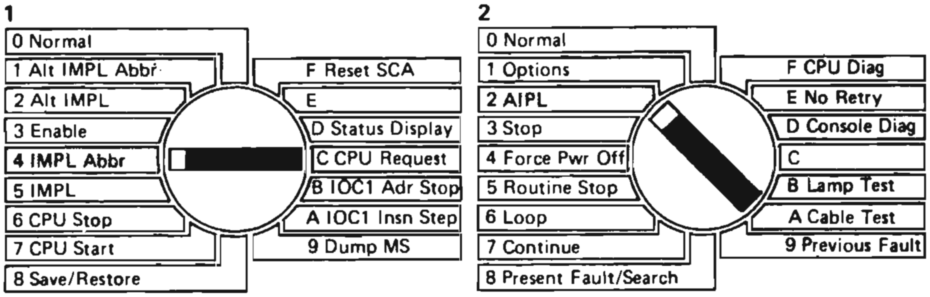
The procedure for using this setting is:

1. If you are installing CPF from diskette, place the diskette magazines containing the CPF information in the diskette magazine drive (see Chapter 3). If you are installing CPF from tape, place the CPF installation diskette into Magazine 1 Position 1, and mount the first CPF tape on the first tape unit.

2. Set the two rotary switches as shown.

3. Press the Power On switch (if power is off) or the Load switch (if power is on).

IMPL Abbr with AIPL



You use this setting to install CPF without performing the extensive hardware diagnostics. It combines the shorter IMPL Abbr sequence with the AIPL sequence that loads CPF into the system from diskette or tape.

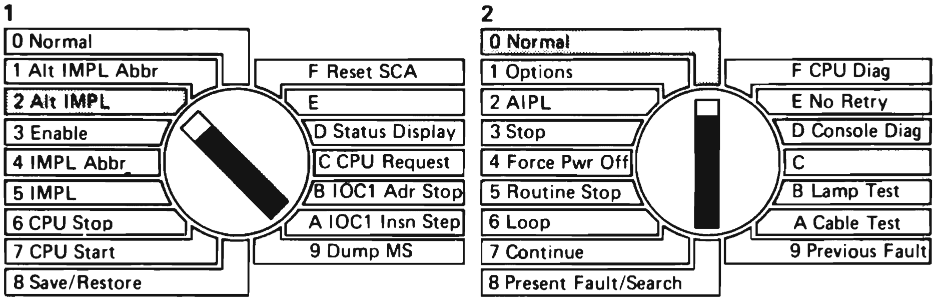
The procedure for using the setting is the same as for the IMPL-with-AIPL setting.

Do not use this setting if there is any question about the condition of the hardware.

Settings to Activate the Service Monitor Mode

These settings are used to load microcode into the system and invoke a service monitor mode that allows you to perform specific service type functions. Generally, you use the settings only under the instruction of your service representative.

Alt IMPL



The Alt IMPL (alternative initial microprogram load) sequence copies microcode from diskette to internal storage as well as conducting extensive hardware diagnostic and initialization procedures similar to the normal IMPL.

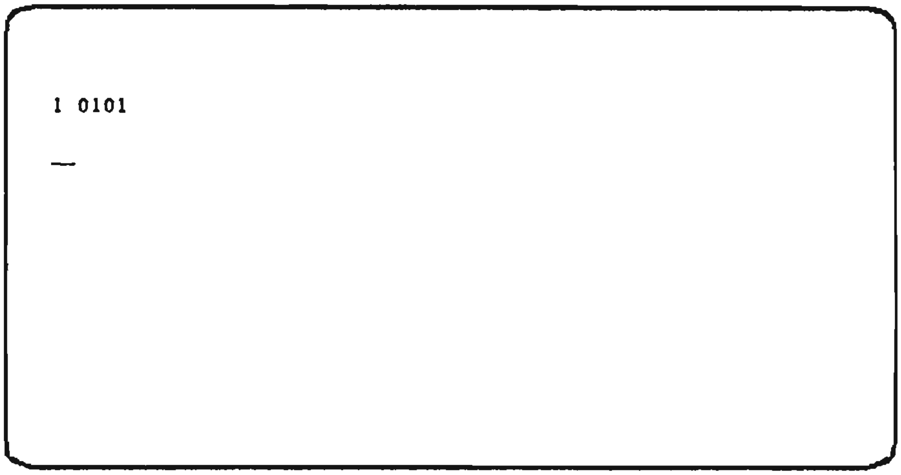
The procedure for using this setting is:

1. Place the diskette magazine or magazines containing the microcode in the diskette magazine drive (see Chapter 3) as appropriate for the service function being performed (for example, *Installing Microcode* described in Chapter 14).

2. Set the two rotary switches as shown.

3. Press the Power On switch (if power is off) or the Load switch (if power is on).

When the sequence is completed, the following display appears on the system console screen:



```
1 0101

_
```

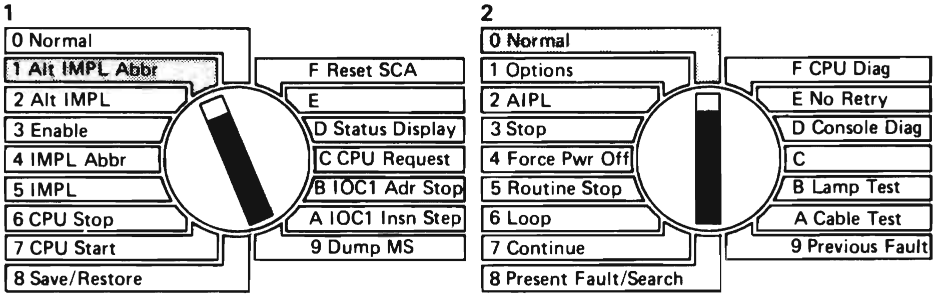
The numeric code indicates that the dedicated mode of the service monitor has been successfully activated. By entering a numeric code on the line provided below the code shown on the screen, you can start one of five special functions:

Code	Function
01	End dedicated service monitor mode
02	Start loading microcode
03	Start virtual storage stand-alone dump
04	Print information from stand-alone dump
05	Copy data from 62PC to 3370
06	Eject diskette

The use of functions 01, 02, and 05 is described in Chapter 14. The use of functions 03 and 04 is described in the *Problem Determination Guide*.

Normal system operations cannot be performed while the dedicated service monitor mode is active. After you have completed the desired function using the service monitor, you must do the normal IMPL or IMPL Abbr sequence (see *Normal Settings*) before system operations can begin.

Alt IMPL Abbr



The Alt IMPL Abbr (alternative initial microprogram load abbreviated) sequence copies microcode from diskette to internal storage like the Alt IMPL sequence, but it is completed sooner because it bypasses some of the hardware diagnostics. This setting activates the same service monitor mode as the Alt IMPL setting.

As with the Alt IMPL sequence, you must still do an IMPL or IMPL Abbr to prepare the system for normal operations after the service monitor functions are completed.

Default Settings for Power On

When the Power On switch is pressed to start the system, any rotary switch settings other than those described previously will be handled as invalid settings and one of two default power-on sequences will be used.

The IMPL sequence described in *Normal Settings* is performed *if*:

- Rotary switch 1 (the left switch) is set to any position *except*

  1 Alt IMPL Abbr
  2 Alt IMPL
  4 IMPL Abbr

- Rotary switch 2 (the right switch) is set to any position *except*

  2 AIPL

The IMPL-with-AIPL sequence described in *Settings to Install CPF* is performed *if*:

- Rotary switch 1 is set to any position *except*

  1 Alt IMPL Abbr
  2 Alt IMPL
  4 IMPL Abbr

- Rotary switch 2 is *set to*

  2 AIPL

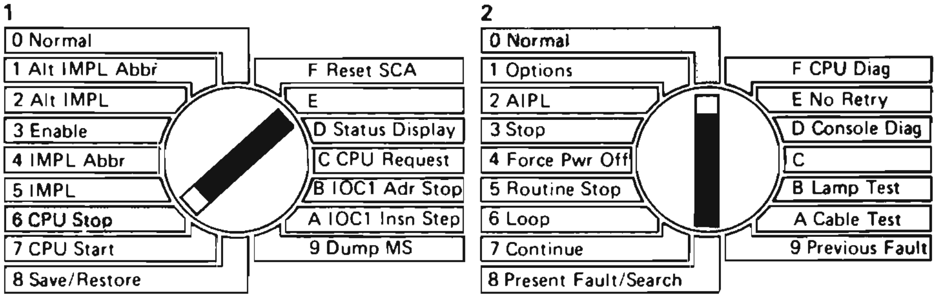
Remember that these default sequences will occur only if power is off and you press the Power On switch. If power is on and you press the Load switch, the function indicated on the rotary switches will be performed. A description of settings activated by the Load switch that you can use follows.

Settings Activated by Load Switch

The following settings can be activated only by the Load switch. If you set the rotary switches to any of these settings and press the Power On switch while power is off, the system will perform the normal IMPL sequence or the IMPL-with-AIPL sequence, depending on the setting. Generally, you would use these settings only when instructed to do so by your service representative.

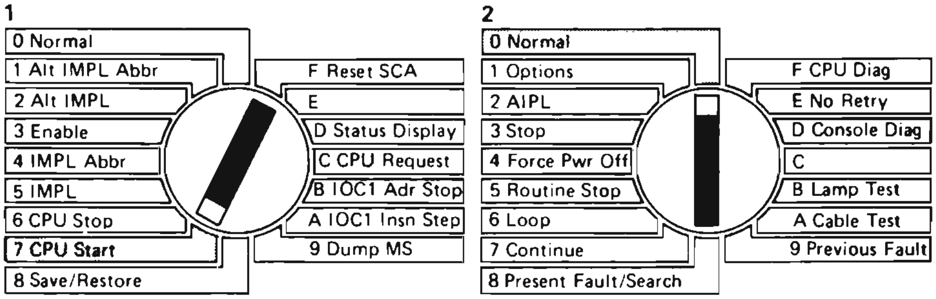
Except where noted, system power must be on (Power On switch lighted) for any of these settings to be used.

Stop Processing (CPU Stop)



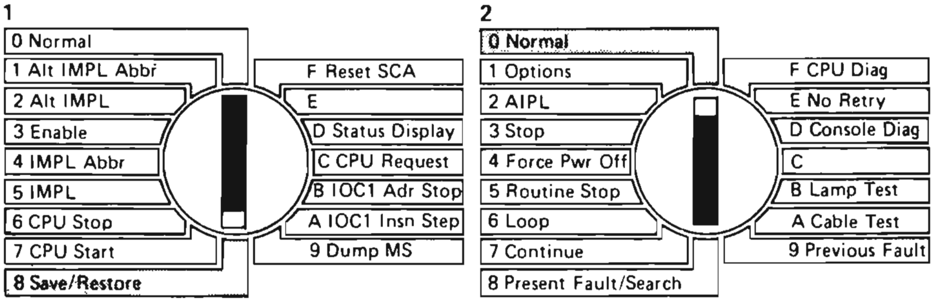
When you set the two rotary switches as shown and press the Load switch, processing proceeds to an interruptible point and stops.

Start Processing (CPU Start)



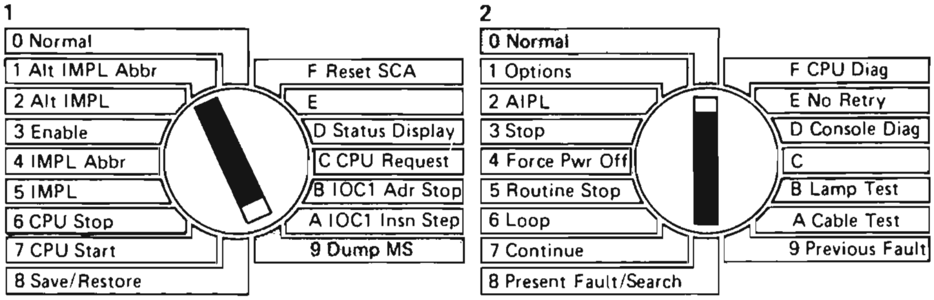
When you set the two rotary switches as shown and press the Load switch, processing that was previously stopped is resumed.

Eject Diskette (Save/Restore)



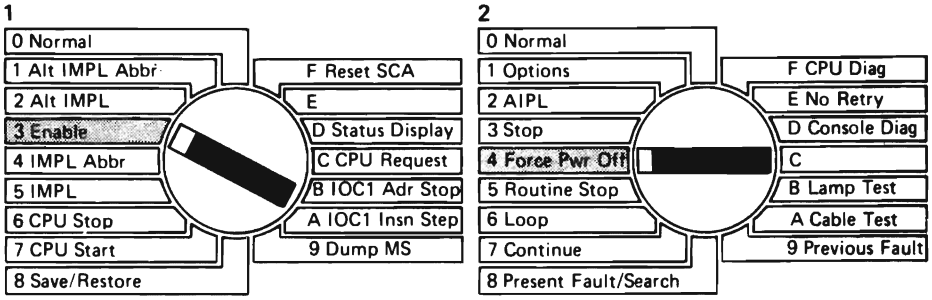
If a system failure occurs that incapacitates the diskette magazine drive (see Chapter 3), you can use this setting to remove diskettes from the drive. When you set the two rotary switches as shown and press the Load switch, the system is reset and a command is sent to the diskette magazine drive that returns the diskette being used to its normal position in a slot or magazine where you can remove it. You must perform the IMPL sequence again before system operations can be resumed.

Dump Main Storage (Dump MS)



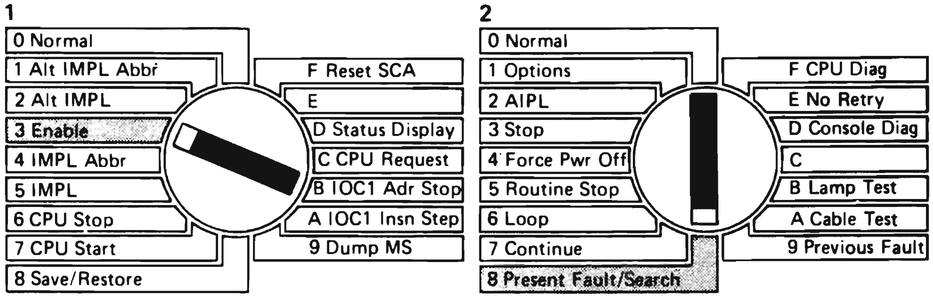
When you set the two rotary switches as shown and press the Load switch, the contents of main storage are copied onto the special diskettes that you have placed in the diskette magazine drive. Details on the use of this setting are given in the *Problem Determination Guide.*

Force Power Off (Force Pwr Off)



When you set the two rotary switches as shown and press the Load switch, the system is powered down immediately without regard to any jobs being processed. You should use this function only when it is not possible to power down the system by normal programming methods (see *Powering Down the System* in Chapter 8).

Determine Power Faults (Present Fault/Search)

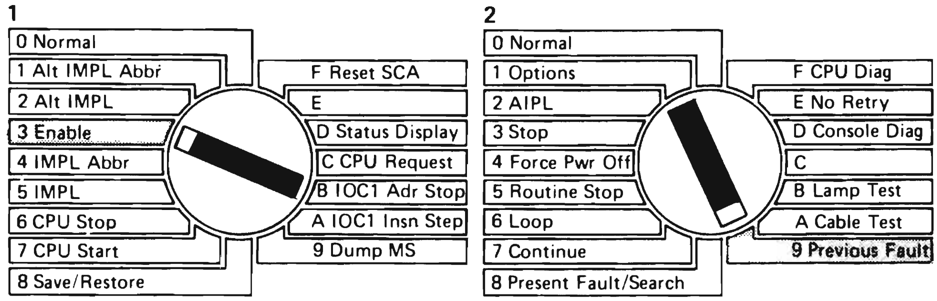


If one or more failures occur in system power equipment, you can use this setting to obtain information on each of those faults for service personnel. When you set the two rotary switches as shown and press the Load switch, the condition indicators numbered 10 to 15 come on in a pattern that indicates the highest priority fault recorded in the power controller. Each time you press the Load switch again, the condition indicators show a pattern that indicates the next highest priority fault. Write down which indicators come on each time you press the Load switch and save the information for your service representative.

When all faults have been indicated, all of the condition indicators numbered 10 to 15 come on. To turn off the indicators, move the rotary switches to a different position, such as 0 Normal.

This setting can be used when system power is off, but the control supply power must be on (Control Supply indicator on).

Determine Previous Power Fault (Previous Fault)

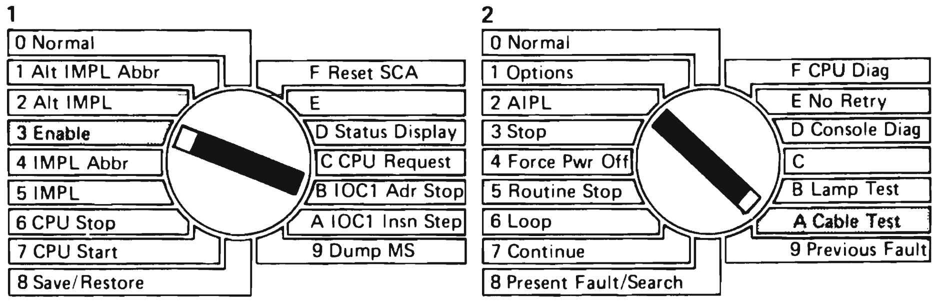


When you set the two rotary switches as shown and press the Load switch, the condition indicators numbered 10 to 15 come on in a pattern that indicates the last power fault recorded in the power controller. Write down which indicators are on, and save the information for your service representative.

To turn off the condition indicators, move the rotary switches to a different position, such as 0 Normal.

This setting can be used when system power is off, but the control supply power must be on (Control Supply indicator on).

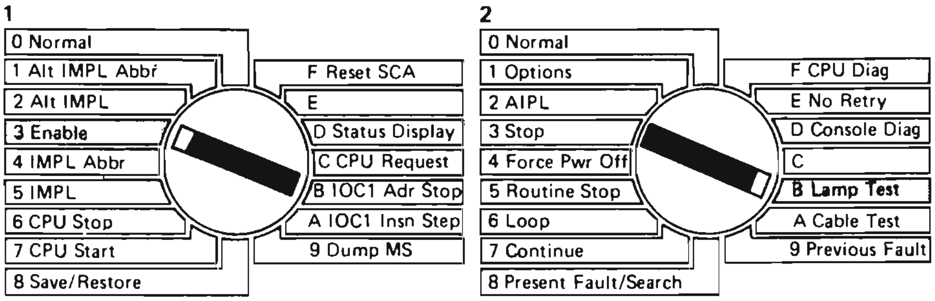
Check Cables (Cable Test)



You use this setting under instruction of your service representative to determine if cables are seated properly. When you set the two rotary switches as shown and press the Load switch, each condition indicator assigned to a cable group turns on if the group is seated properly.

To turn off the condition indicators, move the rotary switches to a different position, such as 0 Normal.

Check Lights (Lamp Test)

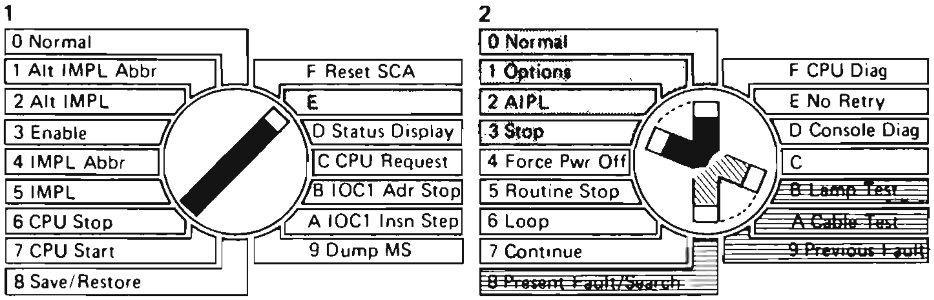


You use this setting under instruction of your service representative to determine if the lights on the operator/service panel are working. When you set the two rotary switches as shown and press the Load switch, all of the lights on the operator/service panel, each printer, and the MFCU come on.

To turn off the lights, move the rotary switches to a different position, such as 0 Normal.

This setting can be used when system power is off, but the control supply power must be on (Control Supply indicator on).

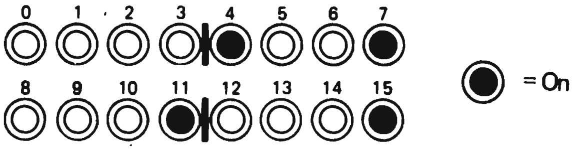
Communication Line Status



By setting the left rotary switch to the position labeled E and the right switch to position 0 through 3 for lines 1 through 4 or position 8 through B for lines 5 through 8, you can check the condition of system communication lines. The procedure is described in the *Problem Determination Guide*.

Settings for Address Regeneration

Eventually, internal storage addresses will be exhausted and will need to be regenerated. The regenerating process is a lengthy process which may take from one to seven hours depending on the amount of auxiliary storage on your system. To prevent the regeneration process from interrupting system operations, you are warned at IMPL time that approximately 95% of the available storage addresses are exhausted. The condition indicators display the hexadecimal code 0911; that is, they come on in the following pattern:



You can do either of the following:

- Regenerate addresses immediately.

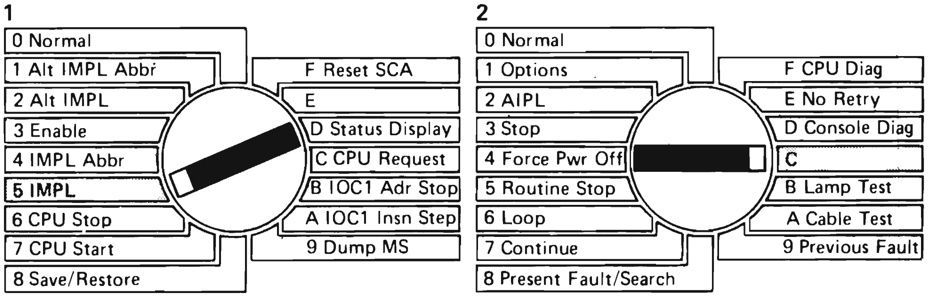
- Continue normal processing by doing another IMPL and regenerate addresses at the next convenient time, such as after system operations have been completed for the day.

Address regeneration can be done anytime, even if the warning has not been issued.

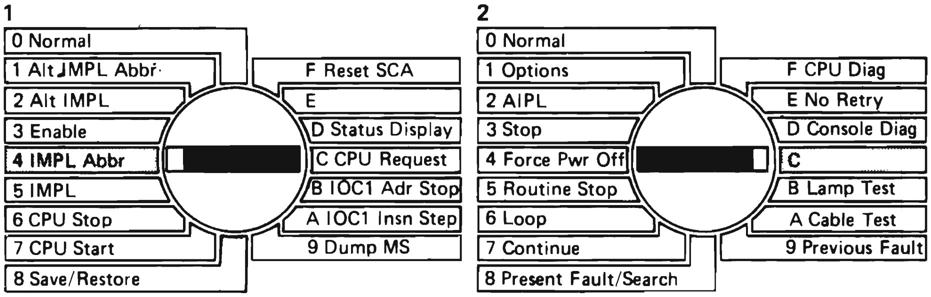
The following procedure is used regardless of when address regeneration is performed.

1. Set the rotary switches to one of the following settings.

Address Regeneration with IMPL



Address Regeneration with IMPL Abbr

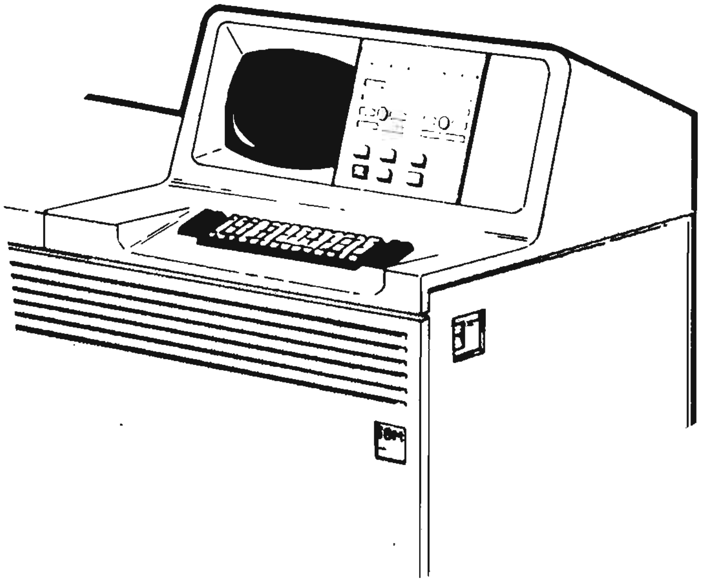


2. Press the Load switch.

While address regeneration is running, the condition indicators display the hexadecimal code 3F30.

Chapter 2. IBM 5381 System Unit System Console

Following the power-on sequence that you start at the operator/service panel, you use the system console (below) to prepare the system for normal operations. Once those operations begin, you can use the system console as a work station to monitor and control the system.

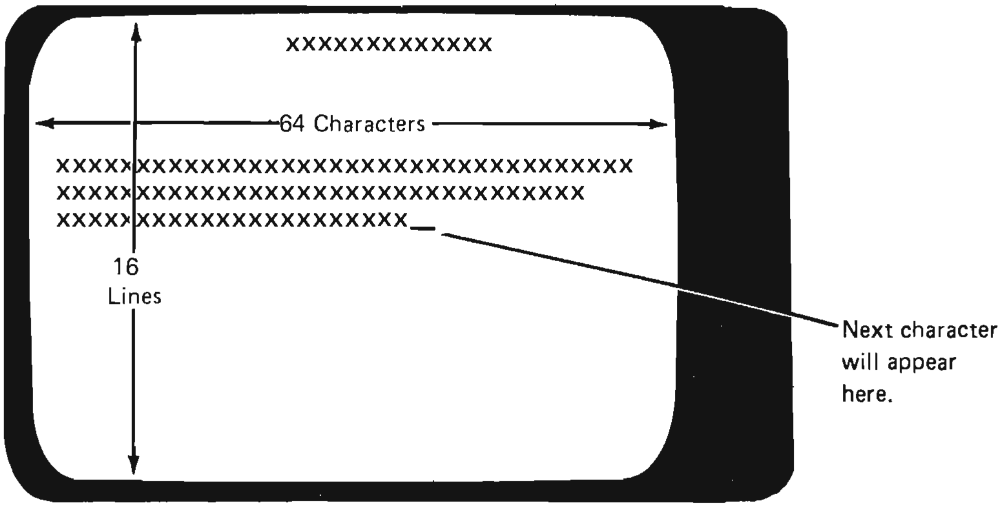


The system console is identified to the system by a device description. To specify the system console in a command, therefore, you use the name of its device description. IBM supplies a standard device description for the system console; its name is QCONSOLE.

DISPLAY SCREEN

The system console has a 1024-character screen (16 lines containing 64 characters each).

Each display appearing on the screen contains a cursor, resembling an underscore, that shows where the next character or symbol entered from the keyboard will appear (below).

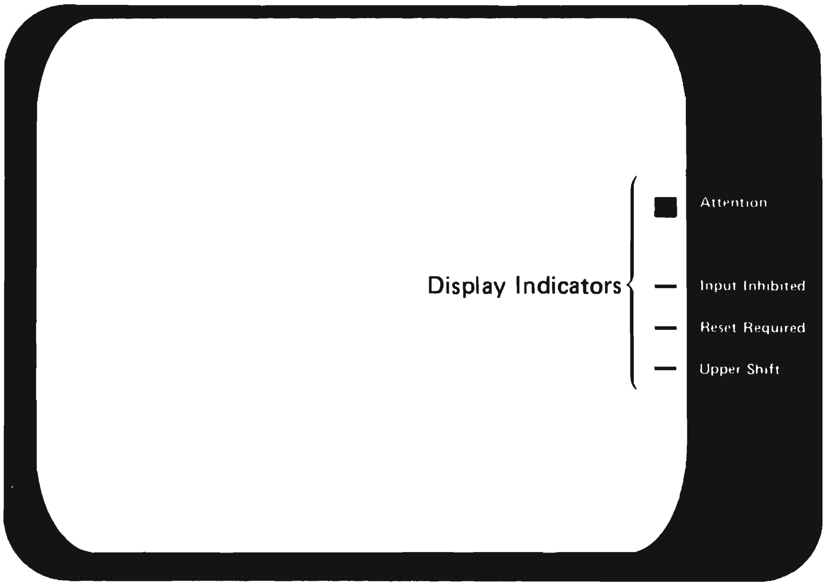

XXXXXXXXXXXXX
64 Characters
XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX
XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX
XXXXXXXXXXXXXXXXXXXXXXX_
16 Lines
Next character will appear here.

Note: Because the system console screen has a shorter line width (fewer characters per line) than the screens of the 5251 and 5252 display stations (see Chapter 5), certain displays will appear different on the system console than they do on the 5251 or 5252. For example, a statement 70 characters long will appear as a single line on a 5251 or 5252 screen, but will be carried over to a second line or truncated (cut off) on the system console.

CONTROLS AND INDICATORS

Display Indicators

Along the right edge of the screen are four visual indicators that inform you about the status of the system console or the system (below).



■ The indicator is ON.
— The indicator is OFF.

Attention: This indicator comes on (■) when there is a message for you. The indicator turns off (-) when you display the message (see *Message Queues* in Chapter 11).

For systems with the audible alarm feature, the Attention light on the operator/service panel turns on and the alarm sounds while this indicator is on. The light turns off when this indicator turns off.

Input Inhibited: This indicator is on (■) when the system cannot accept data through the keyboard and the keyboard is locked. For example, the indicator is on while the system is acting on data you have entered and is not yet ready to accept new data. When the system is ready to accept new data, the indicator goes off (-) and the keyboard unlocks. While this indicator is on, only the shift keys or the Sys Req key can be used; all other keys are ignored.

Reset Required: This indicator comes on (■) when a data entry error has occurred. For example, if you attempt to enter data into a protected field, the keyboard locks and the Reset Required indicator comes on. Pressing the Reset key on the keyboard turns this indicator off (-) and unlocks the keyboard so that you can correct your mistake and continue entering data.

Upper Shift: This indicator is on (■) when the keyboard is in upper shift; it is off (-) when the keyboard is in lower shift.

Controls

You can adjust the brightness of the display using the Display Intensity control on the operator/service panel (see *Single-Function Controls* in Chapter 1).

KEYBOARD

The system console keyboard contains 75 keys. The light keys in Figure 2-1 are used to enter data. The dark keys are used to perform functions, either on the screen or within the system.

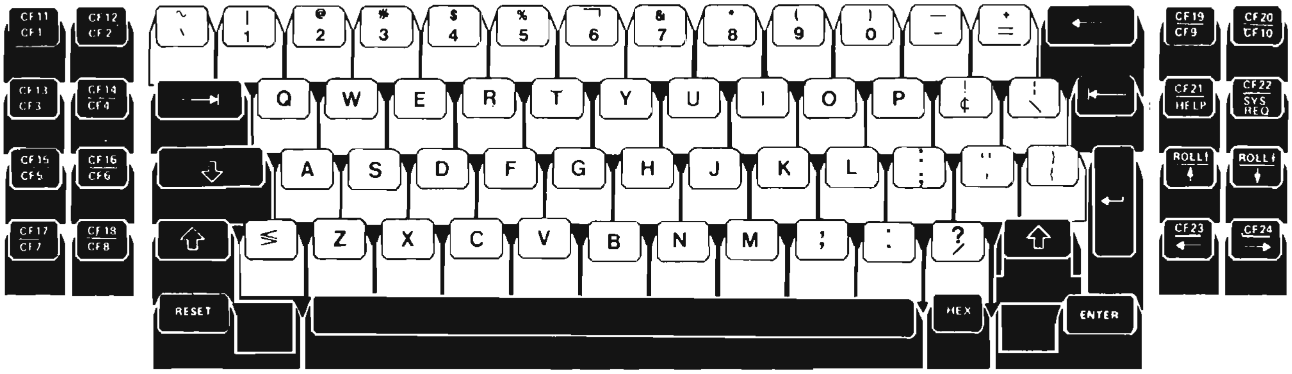


Figure 2-1. System Console Keyboard

Most of the keys are typamatic. This means that the function or character of the key is repeated as long as the key is held down. The unshaded keys in Figure 2-2 are typamatic.

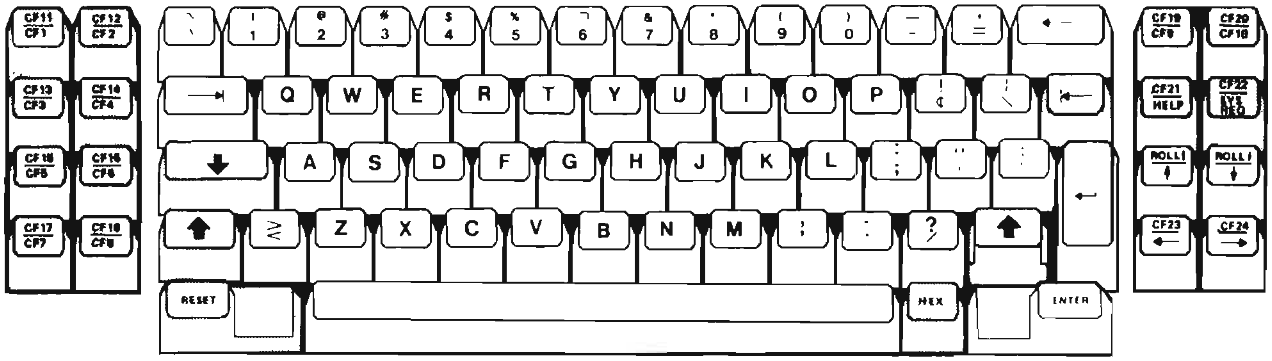


Figure 2-2. Typamatic Keys (unshaded)

As on a typewriter, certain keys are dual character; that is, the upper symbol or function indicated on those keys is obtained by an upper shift using the Shift or Shift Lock keys.

Data Keys

The data keys (light keys in Figure 2-1) are arranged in a similar order to the keys on a typewriter, and they are used the same way. When you press a data key, the indicated character or symbol appears on the screen.

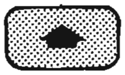
All data keys are typamatic.

Shift/Space Keys

These keys work like the Shift/Lock keys and the Spacebar on a typewriter.

Shift Keys

or



When you hold down a Shift key:

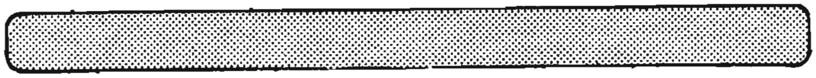
- The special character identified on the upper half of numeric and symbol keys can be entered.

- The function indicated on the upper half of function control and command function (CF) keys can be performed.

Shift Lock Key

When you press this key, the keyboard locks in upper shift. You can return the keyboard to lower shift by pressing and then releasing either of the Shift keys.

Spacebar



When you press the Spacebar, the cursor moves one position to the right and a blank is entered in the position above the cursor. When the cursor reaches the end of an input field, it moves to the beginning of the next input field. The blank entered is treated as a character, which means it destroys any data in the position where it is placed. Therefore, the Spacebar should not be used for repositioning the cursor.

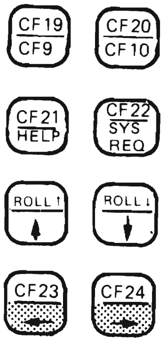
Because the Spacebar is typamatic, the cursor continues to move, and a blank is placed in each position where it has moved, as long as you hold down the key.

Cursor Movement Keys

Entering new data or changing existing data could require moving the cursor a single position, several positions, or an entire field or line at a time. There are a variety of keys to perform those moves. Some cursor movement functions are on the lower half of dual-function keys that perform different functions in upper shift. An arrow on each key shows the direction in which the key will move the cursor. Use of these keys has no effect on the information on the screen, nor does it cause any information to be entered. All cursor movement keys except the Field Backspace key are typamatic.

A description of each cursor movement key follows.

Horizontal Movement



These keys, when the keyboard is in lower shift, move the cursor to the next right or left position.

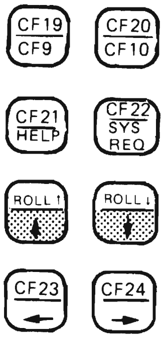
To move the cursor to the right, press CF24 → .

When the cursor moves off the right side of the screen, it reappears one line lower on the left side of the screen. When it moves off the last position on the bottom line of the screen, it reappears in the first position on the top of the screen.

To move the cursor to the left, press CF23 ← .

When the cursor reaches the left margin, it moves to the right margin one line higher. When it moves past the first position on the top line, it reappears at the last position on the bottom line.

Vertical Movement



The lower shift function of these keys moves the cursor up or down one line.

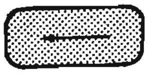
To move the cursor up, press ROLL ↑ ↑ .

When the cursor moves off the top of the screen, it reappears on the bottom line in the same vertical column it moved from. It does not shift to the right or left when it moves off the screen.

To move the cursor down, press ROLL ↓ ↓ .

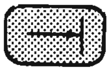
When the cursor moves off the bottom of the screen, it reappears in the same vertical column at the top.

Character Backspace



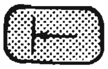
When you press this key, the cursor moves back to the previous enterable character position (not necessarily the last position). For example, if the cursor is in the first position of an input field, pressing the Character Backspace key will move the cursor back to the last enterable character position of the previous input field. (An input field is an area on a display where you can enter information.)

Field Advance



When you press this key, the cursor moves to the first position of the next input field. If the cursor is in the last field on the screen, it moves to the first position of the first field on the screen.

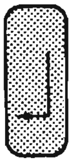
Field Backspace



When you press this key, the cursor moves back (left) to the first character position of the field it is in. If the cursor is already in the first position of a field, it moves back to the first position of the preceding input field.

The Field Backspace key is the only cursor movement key that is not typamatic. The cursor does not continue to move if this key is held down.

New Line



When you press this key, the cursor moves from its current position to the first input position of the first input field on the next line of the screen. If the cursor is on the last input line of the screen, it moves to the first input position at the top of the screen.

Function Control Keys

Function control keys enable you to:

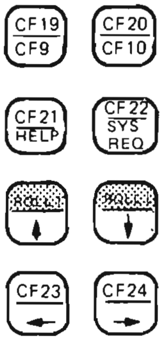
- Enter input into the system.
- Adjust the information displayed on the screen.
- Create a second interactive job at the system console.
- Obtain additional information on the nature of an error condition.
- Recover from an entry error that locks the keyboard.
- Select special characters not shown on the keyboard.

Enter

ENTER

Press the Enter key when you have completed keying in your input (if any) and the information on the screen is ready to be entered into the system for processing. The next appropriate display then appears on the screen.

Roll



Use the upper shift function of these keys to move the information on the screen up or down. This movement repositions the information on the screen and allows you to view data either above or below the data currently on the screen. The roll function is normally valid only when there is repetitive information with a similar format that cannot all be shown on the screen at the same time.

To move the information up so that you can view data that is below the data currently displayed (roll up function):

1. Press and hold ⇧.

2. Press Roll ↑ .

To move the information down so that you can view data that is above the data currently displayed (roll down function):

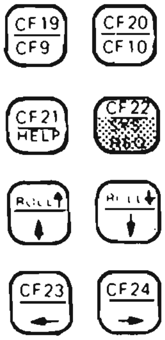
1. Press and hold ⇧

2. Press Roll ↓.

How far the information on the screen rolls up or down each time you press one of these keys depends on your system program, which controls the roll function for each individual job. On CPF displays, you can use the Roll keys to obtain additional information if a + appears in the last character position (right side) on the last line of the current set of information being displayed. The cursor must be positioned on the line, or in the area, that is to be rolled. Each time you press a Roll key, the information shown on the display is usually replaced completely by a new set of information. When you reach the end of the display (the + disappears), pressing the Roll Up key has no effect; when you reach the beginning of the display, pressing the Roll Down key has no effect.

Since these keys are typamatic, the program may repeat the roll function if you continue to hold down the key.

Sys Req (System Request)



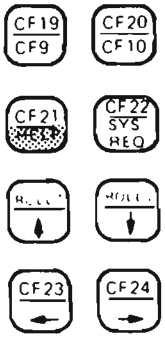
Use the lower shift function of this key to interrupt your interactive job and obtain a special menu display that allows you to:

- Transfer to a second interactive job at the same work station
- Cancel your previous request
- Display the status of your current job or another job
- Display the messages in a message queue
- Send a message
- Sign off

To enter a system request, press .

For additional details of the system request function, see *Using the System Request Menu* in Chapter 6.

Help



The lower shift function of this key allows you to obtain a description of a current error condition or further information from the system.

If you press the Help key during an error condition, a message is displayed that describes the error. If you press the Help key when there is no error condition, a list of command function key assignments is displayed.

To use the help function, press .

Reset

When you make an entry error that locks the keyboard (the Reset Required indicator on the screen shows ■), press the Reset key to unlock the keyboard (and turn off the indicator). This key, however, will not unlock a keyboard that is locked by system action (the Input Inhibited indicator shows ■).

Hex

HEX

When you press the Hex key, you can enter special characters by entering a two-digit hexadecimal code associated with each of the special characters. These special characters can be entered into the system as data in the same way as alphabetic or numeric characters.

To enter a special character:

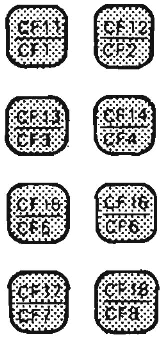
1. Press and hold down the Hex key.

2. Press the key for the first character of the hexadecimal code associated with the special character (only 4 through 9 or A through F is valid).

3. Press the key for the second character of the hexadecimal code associated with the special character (0 through 9 or A through F is valid).

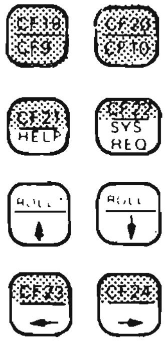
The special character will then appear on the system console screen. If your system console has the Multinational Character Set feature, the special character displayed will be as shown in Figure 2-3 for the code you entered (see *Multinational Character Set* in this chapter). If you do not have the feature, some of the special characters associated with particular codes will be different than shown in Figure 2-3. The actual character displayed will depend on the characteristics of your system console keyboard.

Command Function Keys



Programs within the system allow you to perform special functions by using the command function keys, identified by the numbers CF1 through CF24, on the left and right of the keyboard.

To obtain functions indicated by numbers on the lower half of the keys (CF1 through CF10), press the key having the number of the desired function. To obtain functions indicated by numbers on the upper half of the keys (CF11 through CF24), hold down the Shift key while pressing the key with the desired CF number.



Note: Command functions CF23 and CF24 are upper shift functions of typamatic keys. Therefore, the program may repeat the CF23 and CF24 function if you continue to hold down either key while in upper shift.

The function performed by each CF key may vary, depending on the program in control. The CF keys you are likely to use and the functions they perform are summarized in Appendix A. The CF key assignments for all IBM-supplied functions are given in an appendix to the *System/38 Work Station User's Guide*.

Note that the CF24 key on the system console performs the same print function as the Print key on the 5251 and 5252 display stations except that the output with the CF24 key is directed to the QPRINT output queue.

Multinational Character Set

The Multinational Character Set is an optional feature that allows you to enter additional EBCDIC or diacritic characters from the system console. Some of the characters in this character set are different from those in the standard EBCDIC character sets provided for the system console. For example, the following two characters are different from those in the standard English EBCDIC character set:

- For the hexadecimal code 4F, the character | of the English EBCDIC character set is replaced by ! in this character set.

- For the hexadecimal code 5F, the character ¬ of the English EBCDIC character set is replaced by ^ in this character set.

Using Diacritic Keys

Using a diacritic key, you can place a diacritic (modifying) mark above a character to indicate a different phonetic value for that character.

Various diacritic keys are available. They are:

` (Grave Accent)
´ (Acute Accent)
~ (Tilde)
^ (Circumflex)
¨ (Diaeresis)
¸ (Cedilla)

However, your keyboard may not have all of these.

To enter a diacritic above a character, press the diacritic key and then the character key. The system console then checks to see that the diacritic key and the character key pressed are valid combinations. The valid combinations for each diacritic are:

Diacritic	Hexadecimal Code	Allowable Characters
´ (Grave Accent)	79	A E I O U
` (Acute Accent)	BE	A E I O U
~ (Tilde)	A1	A N O
^ (Circumflex)	5F	A E I O U
¨ (Diaeresis)	BD	A E I O U Y (Y is allowable only as a lowercase character)
¸ (Cedilla)	9D	C

The indicated characters are also allowed in lowercase. After the diacritic and character combination is checked and found to be correct, the cursor moves to the next position. If the character combination is not valid, the Reset Required indicator comes on (■). Press the Reset key and continue to enter the correct combination. The diacritic, if correctly entered, does not have to be entered again.

Using the Hexadecimal Code to Enter Diacritics or Characters

If you have the multinational character set option for your system console, you can enter any of the special characters (diacritics, EBCDIC characters, or accented characters) shown on the chart in Figure 2-3 by entering a two-digit hexadecimal code associated with the character.

The hexadecimal code for the selected character can be found by:

1. Following the column that the selected character is in to the top of the chart. The character at the top of the chart is the first character of the hexadecimal code.

2. Following the row that the selected character is in to the left of the chart. The character at the left of the chart is the second character of the hexadecimal code.

First character

	4	5	6	7	8	9	A	B	C	D	E	F
0		&	-	ø	Ø	˚	µ	¢	{	}	\	0
1		é	/	É	a	j	~	£	A	J		1
2	â	ê	Â	Ê	b	k	s	¥	B	K	S	2
3	ä	ë	Ä	Ë	c	l	t	₧	C	L	T	3
4	à	è	À	È	d	m	u	ƒ	D	M	U	4
5	á	í	Á	Í	e	n	v	§	E	N	V	5
6	ã	î	Ã	Î	f	o	w	¶	F	O	W	6
7	å	ï	Å	Ï	g	p	x	¼	G	P	X	7
8	ç	ì	Ç	Ì	h	q	y	½	H	Q	Y	8
9	ñ	ß	Ñ	`	ı	r	z	¾	I	R	Z	9
A	[	]	¦	·	«	ª	¡	¬	SHY Note	>	²	³
B	.	$	,	#	»	º	¿	|	ô	û	Ô	Û
C	<	*	%	@	ð	æ	Đ	≠	ö	ü	Ö	Ü
D	(	)	_	'	≤	¸	↑	¨	ò	ù	Ò	Ù
E	+	;	>	=	þ	Æ	Þ	´	ó	ú	Ó	Ú
F	!	¯	?	"	±	¤	®	‗	õ	ÿ	Õ	

Second character

Note: This character is identical to a hyphen or minus sign and is used when hyphenating a word at the end of a sentence.

Figure 2-3. Multinational Character Set

To enter the selected character on the system console, do the following:

1. Press and hold down the Hex key.

2. Press the key for the first character of the hexadecimal code (only 4 through 9 or A through F is valid).

3. Press the key for the second character of the hexadecimal code (0 through 9 or A through F is valid).

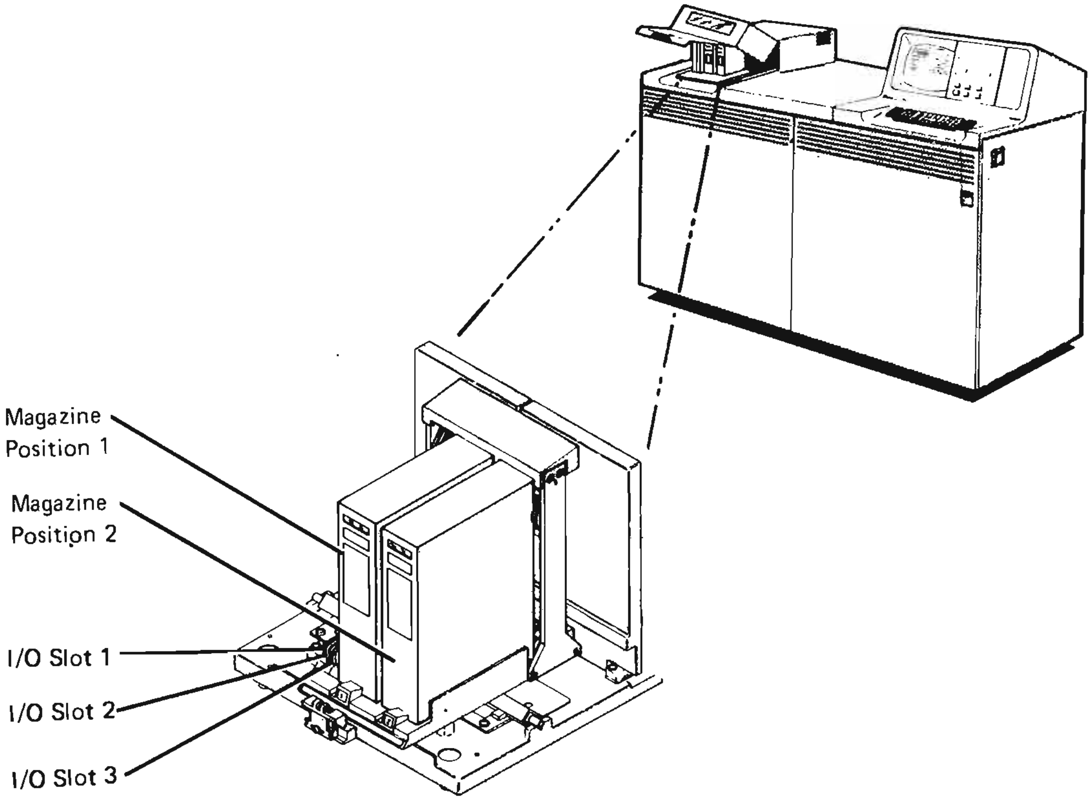
If you want to place a diacritic above the character, repeat the procedure for the diacritic without moving the cursor. If you are entering *only* a character or diacritic, press the Spacebar or the appropriate cursor movement key to advance the cursor before repeating the procedure. The Hex key can be held down until all characters have been entered.

After you have entered the hexadecimal code for the character, the character will be shown on the screen.

Chapter 3. IBM 5381 System Unit Diskette Magazine Drive

The System/38 diskette magazine drive is used to transfer information between system internal storage and removable diskettes. These diskettes provide a source of information for the system and a means of storing information external to the system. You use the diskette magazine drive for data exchange with other systems or offline diskette devices, as an input/output device for spooling operations (see Chapter 10), and to perform save/restore operations (see Chapter 13).

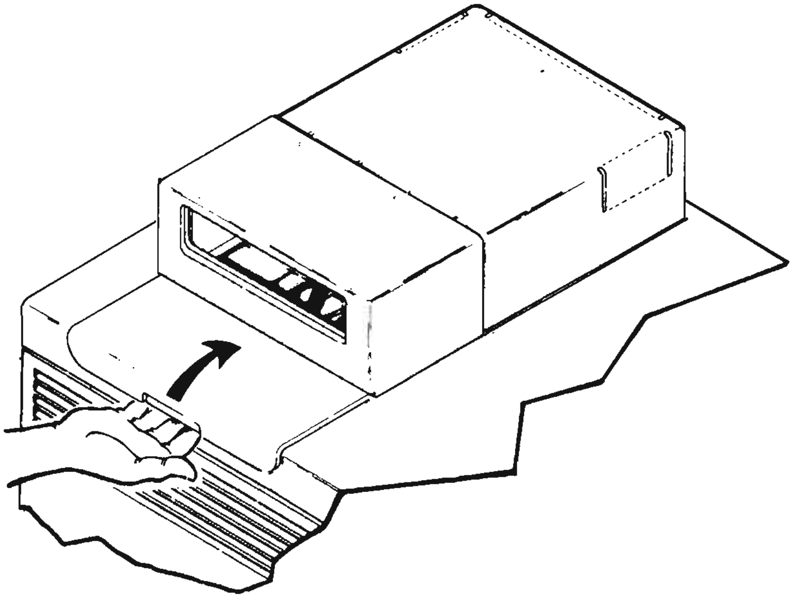
The diskette magazine drive can hold two magazines, each containing 10 diskettes, plus individual diskettes in three separate slots (below). Once you have loaded the magazines or individual diskettes properly in the drive, all other operations needed to read data from diskettes or write data onto diskettes are performed automatically under programming instructions. Any problems that occur with the device during system operations are indicated by messages (see *Operator/Device Interaction* in Chapter 9).



To specify the diskette magazine drive in a command, you use the name of the device description that identifies it to the system. IBM supplies a standard device description for the diskette magazine drive; its name is QDKT.

OPENING AND CLOSING THE COVER

To open the cover of the diskette magazine drive, place your fingers under the front of the cover and push upward (below). The cover will remain open without being held.



To close the cover, push down on the front edge slowly until it has returned to its closed position.

Note: Do not leave the cover open. The carriage bed that positions the diskette or magazine for use will not move while the cover is open.

CONTROLS AND INDICATORS

Interlock/Indicator

When a particular diskette is to be used, the carriage bed moves horizontally to align the diskette at the guide-in window (Figure 3-1). The diskette is then moved into the guide-in window to the read/write position in the drive station. Similarly, when the read/write operation is completed, the diskette is pushed out of the window to its original position in the magazine or I/O slot. A fixed indicator in front of the carriage bed (Figure 3-1) is aligned with the guide-in window and thus shows which diskette is currently being used.

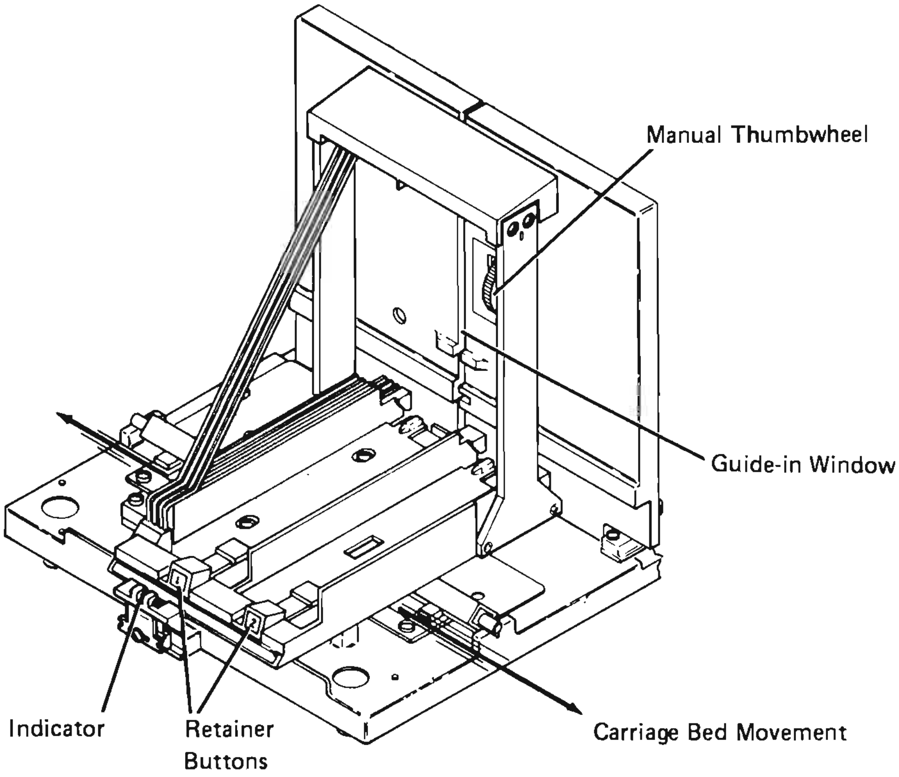


Figure 3-1. Diskette Magazine Drive Controls and Indicator

The indicator also serves as an interlock to prevent a magazine from being removed while in use. To remove an interlocked magazine, you can usually move the carriage bed so that the magazine is clear of the interlock/indicator. If the carriage bed cannot be moved, you can release the magazine by pushing down on the indicator and the magazine retainer button at the same time.

Magazine Retainer Buttons

Each magazine is held in place by a spring-loaded retainer button at the front of the carriage bed (Figure 3-1).

To release a magazine, press down on its retainer button. The retainer button is locked by the interlock/indicator when the carriage bed is positioned to read or write a diskette from the magazine.

Manual Thumbwheel

If a diskette remains in the drive station after the system has powered down, you can remove it manually using the thumbwheel beside the guide-in window (Figure 3-1).

CAUTION
Never attempt to use the thumbwheel while the system power is on. You could damage the diskette drive mechanism.

Push in on the thumbwheel to engage it; then turn it slowly downward until the diskette starts to move (see *Removing Diskettes from the Drive Station* later in this chapter).

USING DISKETTES

Basic Concepts

An IBM diskette (Figure 3-2) is a thin, flexible disk permanently enclosed in a semirigid, protective, plastic jacket. When the diskette is properly inserted in the diskette drive, the disk turns freely within the jacket. The diskette jacket contains a liner material that cleans the disk as it turns. The disk surface is coated with a magnetic recording material.

The system writes data on the recording surface by generating magnetically charged spots (magnetic fields) at specific addresses (locations). These addresses provide direct access to specific information. Data written at an address remains there until it has been replaced by new data or blanks. To read data, the system finds the desired address and converts the magnetic fields back into machine-readable code. Reading the diskette does not erase or change the data stored on it.

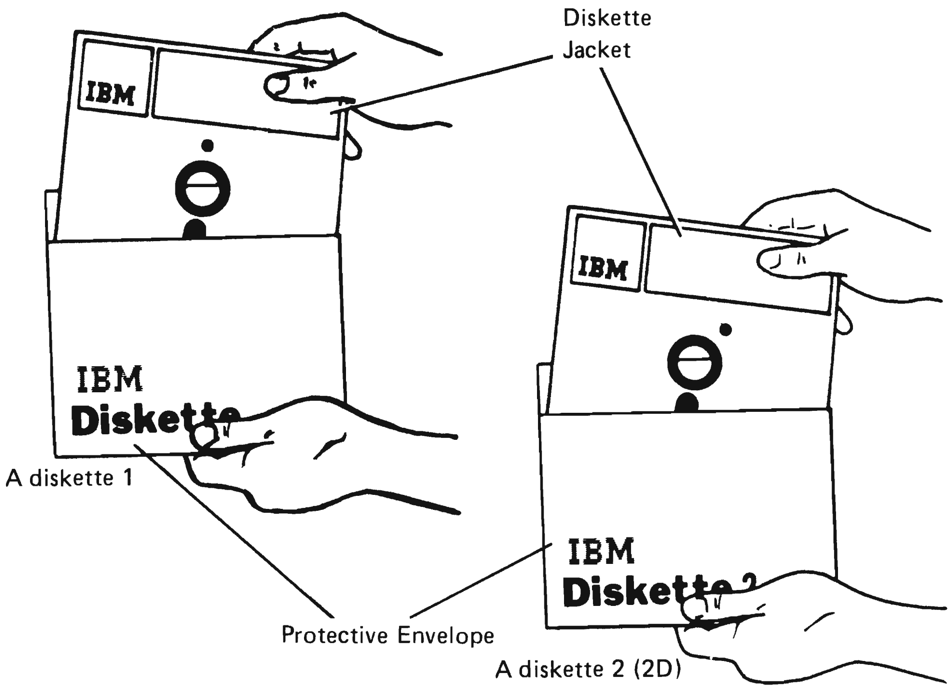


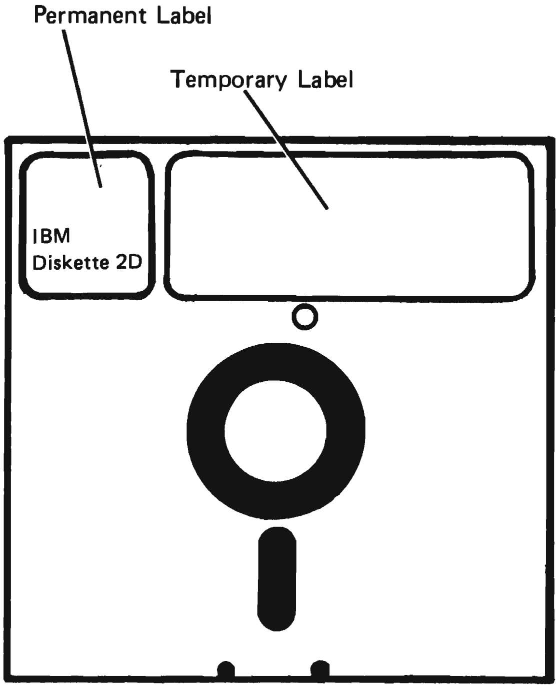
Figure 3-2. IBM Diskettes

The recording surface on one or both sides of a diskette is divided into tracks. A *track* is the recording area that passes the read/write head when the diskette makes a complete revolution. For diskettes that have a recording surface on both sides (double-sided diskettes), two tracks on opposite sides will pass the two read/write heads while the diskette is in a particular position. This pair of associated tracks is called a *cylinder*. The tracks, or cylinders, on a diskette are divided into portions of equal size that are called *sectors*.

Labels

A permanent label (below) is attached to the jacket of each diskette before it is shipped. This label indicates the type of diskette and provides space for you to write in information describing the diskette and its condition, such as:

- Sector size and recording code (EBCDIC or ASCII)
- Volume identifier
- Date first used



Temporary labels are supplied with the diskettes for recording changing items such as:

- Data stored on the diskette (job numbers, names, and dates)
- Who entered the data
- Date of data verification

These temporary labels can be attached next to the permanent label on the diskette jacket or to the envelope in which you store the diskette when not in use.

The labels also serve as an aid in positioning the diskette properly when you load it into a magazine.

Diskette Handling Precautions

Observe the following precautions when handling diskettes.

- Do not use a diskette contaminated by sticky or abrasive substances, even if the contaminant is only on the diskette jacket. The read/write heads can be damaged and they can contaminate other diskettes.

- Return the diskette to its protective envelope as soon as it is removed from a drive or magazine. The envelope protects it from static, spills, scratches, and fingerprints.

- Do not touch or attempt to clean the exposed diskette surface. Be especially careful to avoid grasping the diskette in the area of the head slot.

- Do not fold or bend the diskette.

- Do not use clips or rubber bands on a diskette.

- Do not place heavy objects on diskettes. The weight can cause serious damage.

- Do not eat, drink, or smoke while handling the diskette.

- Do not expose diskettes to heat greater than 51.5°C (125°F) or direct sunlight.

- Do not place diskettes near materials that might be magnetized. Data can be lost from a diskette exposed to a magnetic field.

- Attach labels only in the label area, being careful not to cover the index hole. Remove any old labels first. Do not use tape on the diskette.

- Do not write on diskettes outside the label area. Use only fiber-tip pens when writing on labels attached to the diskette and then only with the diskette in its protective envelope. If possible, write the label before attaching it to the diskette.

- Do not make erasures on or near the diskette.

- Store diskettes needed for immediate use flat in their protective envelopes, in stacks of 10 or less.

- When storing diskettes vertically, support the diskettes so they do not lean or sag.

- For longer storage, use the original shipping carton, with each diskette in its protective envelope. Shipping cartons can be stored vertically or horizontally.

Handling Defective Cylinders

With use, areas can develop on the disk surface on which readable records cannot be written. The diskette magazine drive is capable of reinitializing diskettes and assigning up to two alternative cylinders. Do not use a diskette with a defective area before reinitializing it to bypass the cylinder containing the defective area.

If diskette errors occur, you must make a decision regarding replacement of individual diskettes. The following procedures can help with this decision:

- After reinitializing or renaming a new diskette to assign a volume identifier, record that volume identifier on the diskette temporary label.

- Keep a log of diskette volume identifiers and the initial date used so you can estimate wear by the age of the diskette.

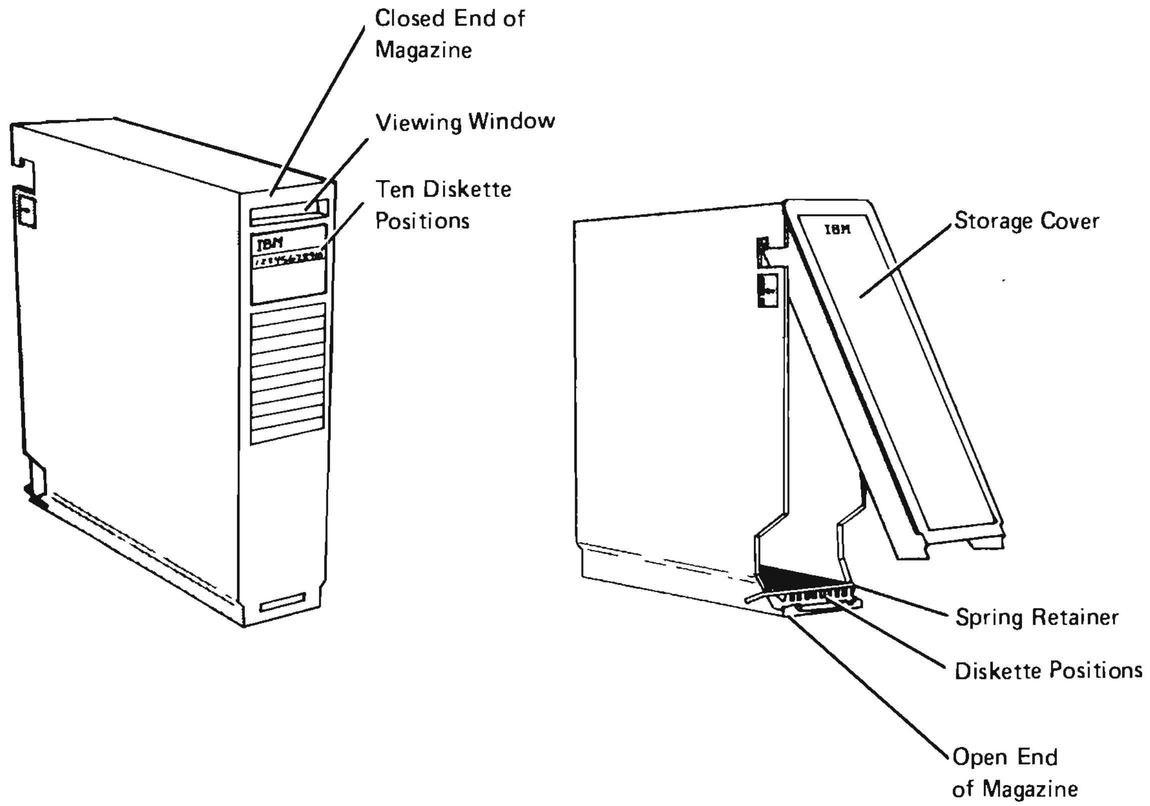
- Whenever a diskette error occurs repeatedly, reinitialize the diskette as soon as possible. Before initializing, copy any data from the diskette. As part of the initialization process, the diskette magazine drive assigns cylinder and sector numbers to the diskette, bypasses the defective cylinder, and assigns the cylinder number of the defective cylinder to the next good cylinder. Two defective cylinders per diskette can be replaced in this manner.

- Periodically examine the log of diskette volume identifiers and the permanent labels on the diskettes. If this log indicates a diskette is too old for continued use, or if there are more than two defective cylinders, replace the diskette.

For further details on diskettes and their handling, refer to *The IBM Diskette General Information Manual.*

LOADING AND UNLOADING MAGAZINE DISKETTES

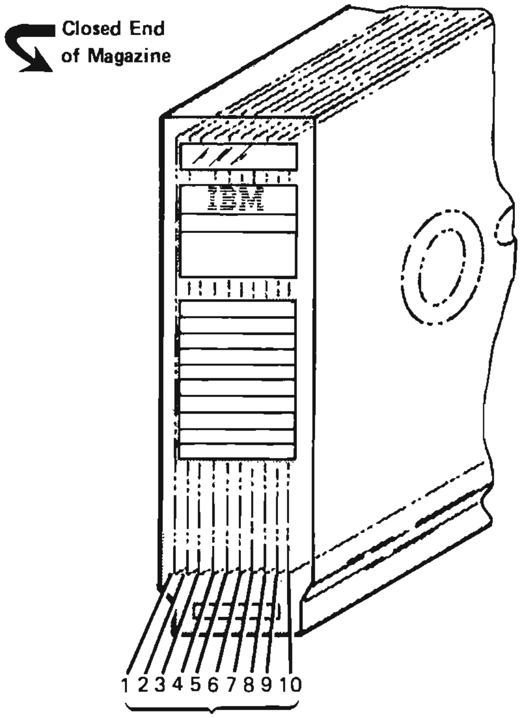
The magazines that hold diskettes used in the diskette magazine drive are identical and interchangeable (below). A diskette from one magazine can be used in another magazine. For the diskette drive to work properly, you must be sure that the diskettes are inserted correctly in the magazines.

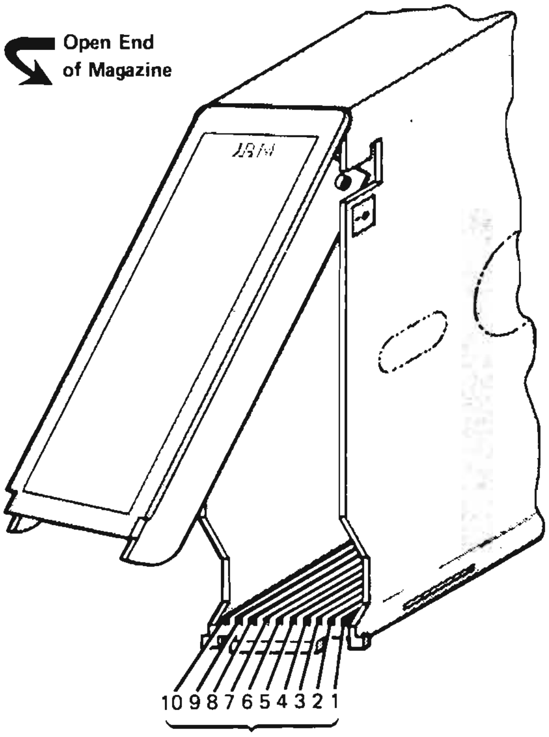


The magazine, including the cover and labels, is IBM part number 2462521.

Positioning the Diskette

Diskettes are held in the magazine by slots at the top and bottom of the magazine. These slots are numbered 1 through 10 from right to left when facing the open end of the magazine, or from the left to right when facing the closed end (below). The diskette is properly aligned when its top and bottom edges are inserted in the slots at the top and bottom of the magazine that have the same slot number.





Each diskette used for save/restore operations has a volume identifier that specifies which slot it is in. Therefore, if you remove such a diskette from a magazine, take care to reinsert it in the same slot position you took it from. If you need to use the diskette in a different position, you must change its volume identifier (see *Preparing Diskettes* in Chapter 13).

Inserting a Diskette

All magazines have a label on both sides near the open end. The label on the left side, as viewed facing the open end, shows a replica of the labels on the diskette. When inserting a diskette in a magazine, you must insert the edge containing the labels first, and the label side of the diskette must be toward the side of the magazine showing the replica of the labels (below).

To load a diskette into a magazine:

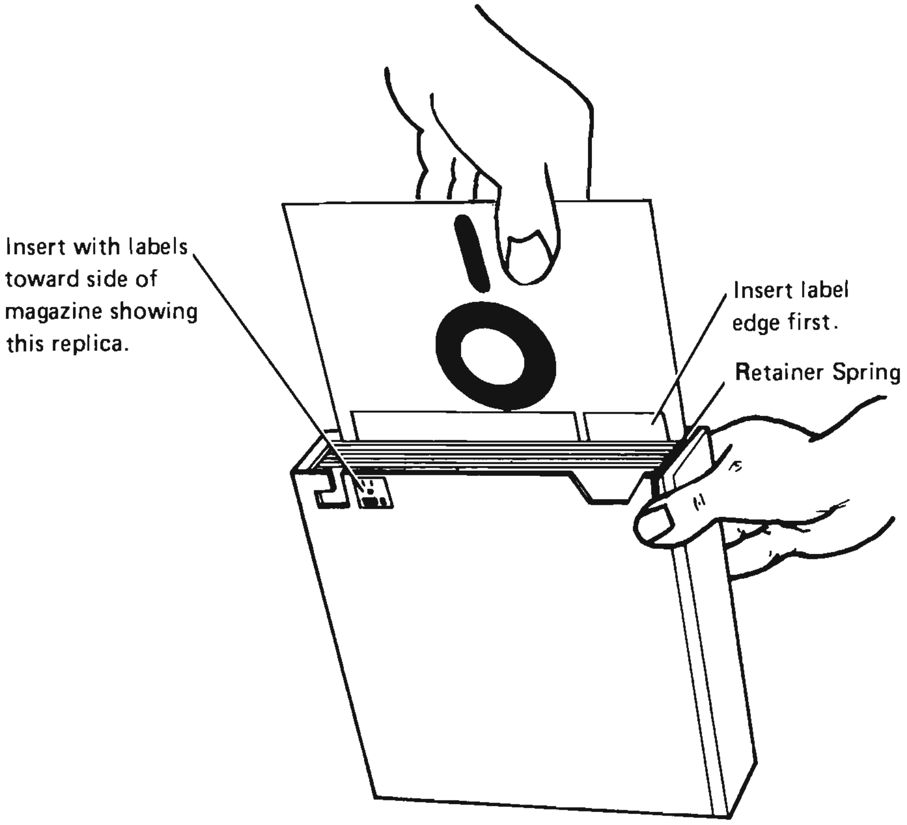
1. Remove the diskette from its protective envelope (see Figure 3-2).

2. Press and hold down the diskette retainer spring and insert the diskette, label edge first, into the selected slot position (below).

3. Verify that the diskette is aligned properly.

CAUTION
The diskette should slide in easily. If it does not, remove it and correct the obstruction before attempting to insert it again.

4. Gently push the diskette in until the rear edge of the diskette clears the retainer spring. The spring will restore itself and thereby lock the diskette in place.


Insert with labels toward side of magazine showing this replica.
Insert label edge first.
Retainer Spring

Removing a Diskette

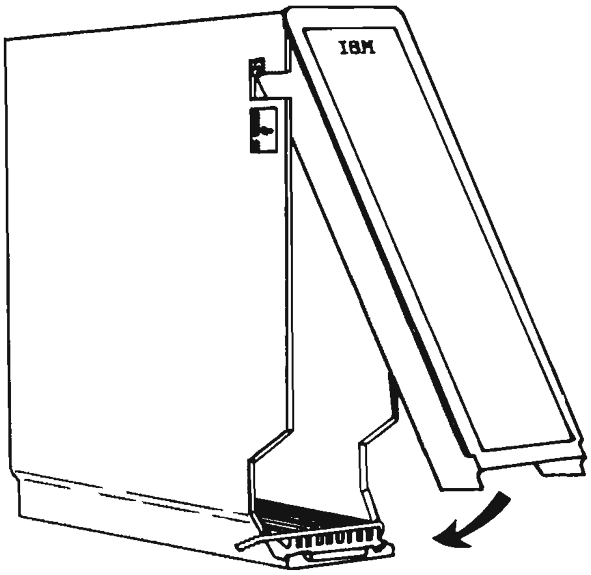
To unload a diskette from a magazine:

1. Press and hold down the diskette retainer spring.

2. Gently pull the diskette out.

LOADING AND UNLOADING MAGAZINES

A protective cover is supplied with each magazine. This cover should be placed on the open end of the magazine (below) to protect the diskettes whenever the magazine is stored. However, the cover *must be removed* before the magazine is loaded into the diskette magazine drive.



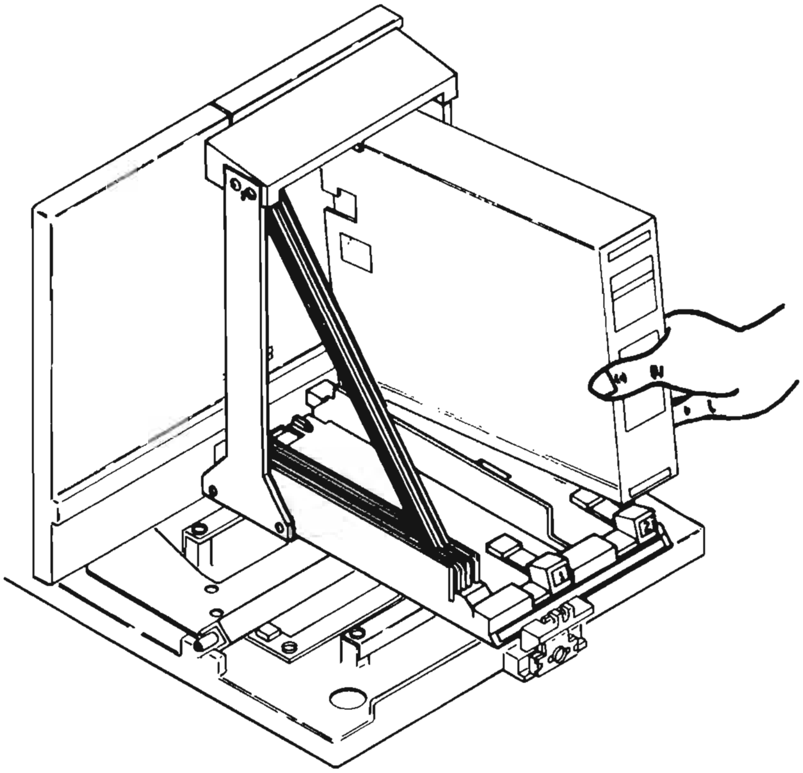
Inserting a Magazine

To load a magazine into one of the two magazine positions:

1. Place the magazine between the guide rails of the position selected, with the magazine's open end facing the guide-in window; that is, with the open end away from you (below).

CAUTION
Do not attempt to load the magazine when the selected magazine position is aligned with the guide-in window; the retainer button for that position will be locked.

2. Push the magazine in toward the guide-in window until the magazine retainer button snaps up, locking the magazine in place.

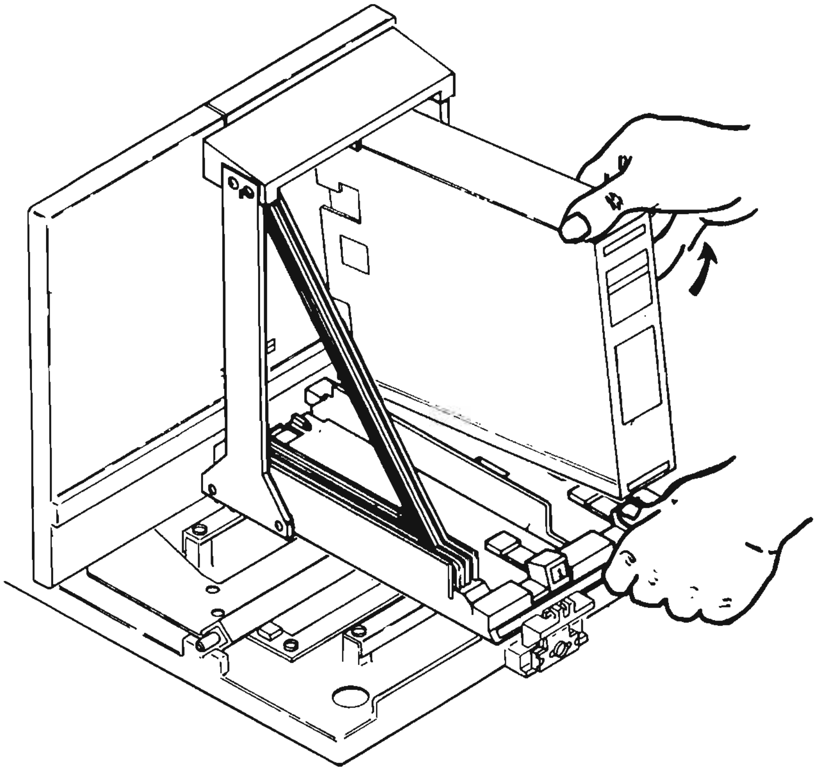


Removing a Magazine

To unload a magazine from the device (below):

1. Press the magazine retainer button and allow the magazine to be ejected to its stop.

2. Lift the magazine's closed end (the end nearest you) slightly to clear the stop, and remove the magazine.



LOADING AND UNLOADING INDIVIDUAL DISKETTES

Individual diskettes not stored in magazines should be stored in their protective envelopes. However, they must be removed from the envelopes before they are used in the diskette magazine drive.

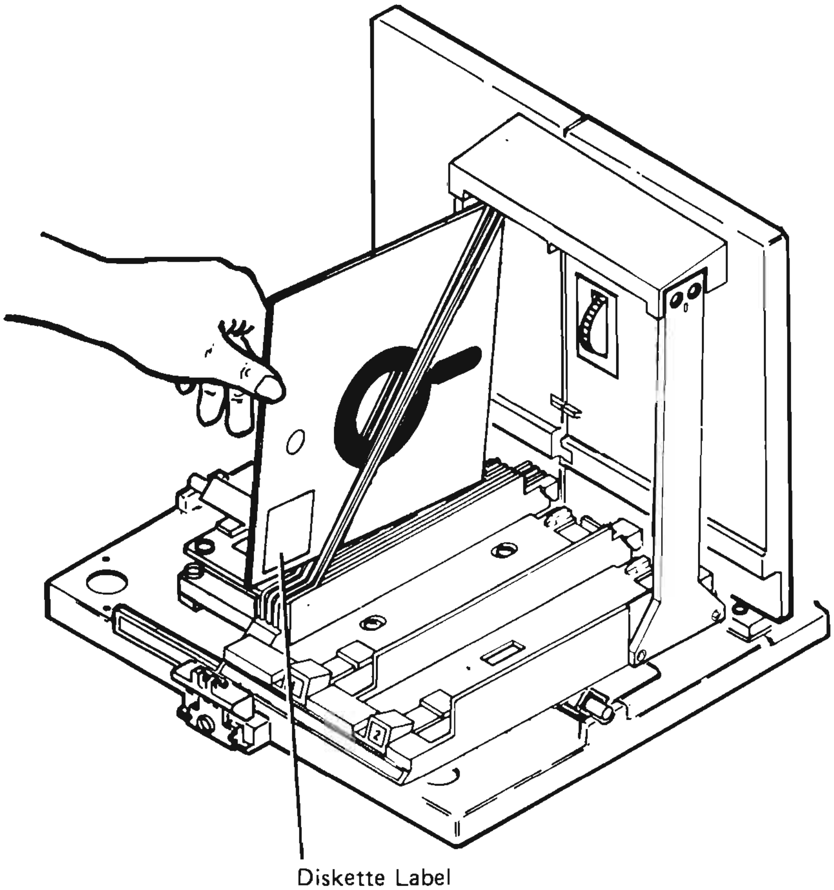
Inserting a Diskette in an I/O Slot

To load a diskette into an I/O slot:

1. Remove the diskette from its envelope (see Figure 3-2).
2. Hold the diskette with the label edge toward the front of the system unit and insert the opposite edge between the guide wires of the selected slot (below).

CAUTION
The diskette should slide in easily. If it does not, remove it and correct the obstruction before attempting to insert the diskette again.

3. Push the diskette in toward the guide-in window until the label edge of the diskette nearest you clears the stop.



Removing a Diskette from an I/O Slot

To unload a diskette from an I/O slot:

1. Gently pull the diskette toward you until the edge of the diskette touches the stop.

2. Lift the edge of the diskette slightly to clear the stop and slide the diskette out.

REMOVING DISKETTES FROM THE DRIVE STATION

Diskettes are normally removed from the drive station by programming control without your intervention, or after you have responded to a message on your work station screen by indicating that the diskette should be removed. If a system failure occurs, however, a diskette may be left in the drive station. There are two procedures you can use to remove the diskette.

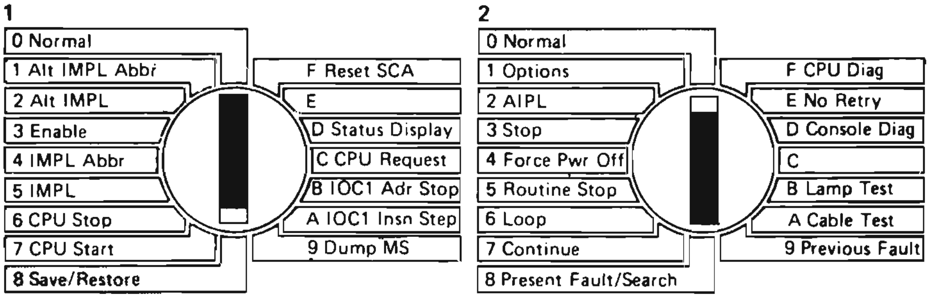
System Failure (Power On)

If a diskette remains in the drive station after a system failure that does not result in the loss of system power (the Power On switch on the operator/service panel is still lighted), you can remove the diskette by doing another IMPL or using the Save/Restore setting of the rotary switches.

CAUTION
When you use the Save/Restore setting, main storage is cleared and you must do an IMPL to resume system operations.

The procedure for removing the diskette using the Save/Restore switch setting is:

1. Set the two rotary switches to:



2. Press the Load switch.

3. Press down at the same time on the interlock/indicator and the magazine retainer button (see *Interlock/Indicator* in this chapter) and gently remove the magazine. If, in removing the magazine, you discover that the diskette still remains partly or completely in the drive station, do not attempt to remove it. Contact your service representative and report that you have a jammed diskette.

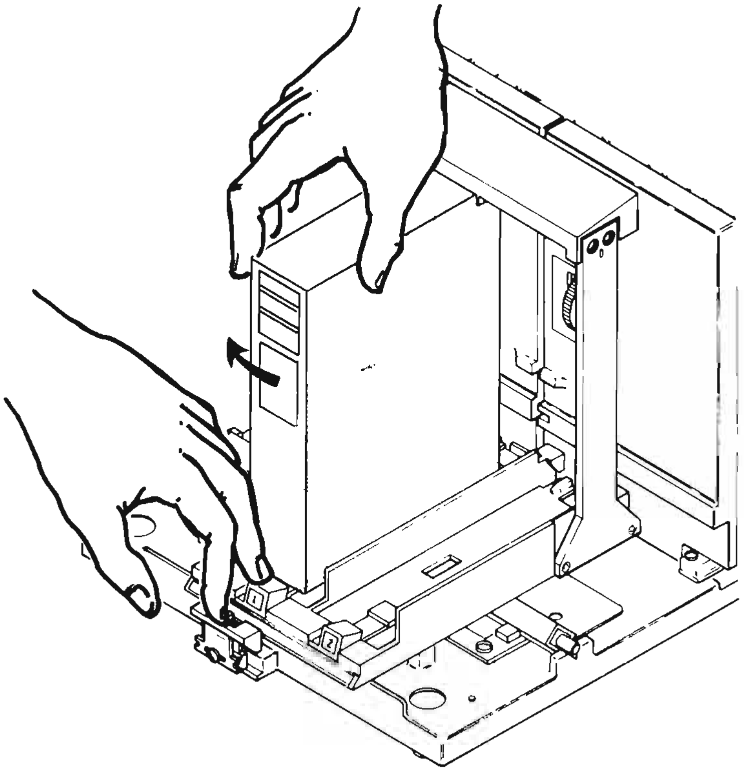
System Failure (Power Off)

If a diskette remains in the drive station after a system failure that results in the loss of system power, you must remove the diskette manually.

Magazine Diskette

To remove a magazine diskette:

1. Press down on the interlock/indicator and the magazine retainer button at the same time and gently remove the magazine (below).



2. Push in on the manual thumbwheel and turn it downward until the diskette starts to move (Figure 3-3).

3. Release the thumbwheel and gently pull the diskette out.

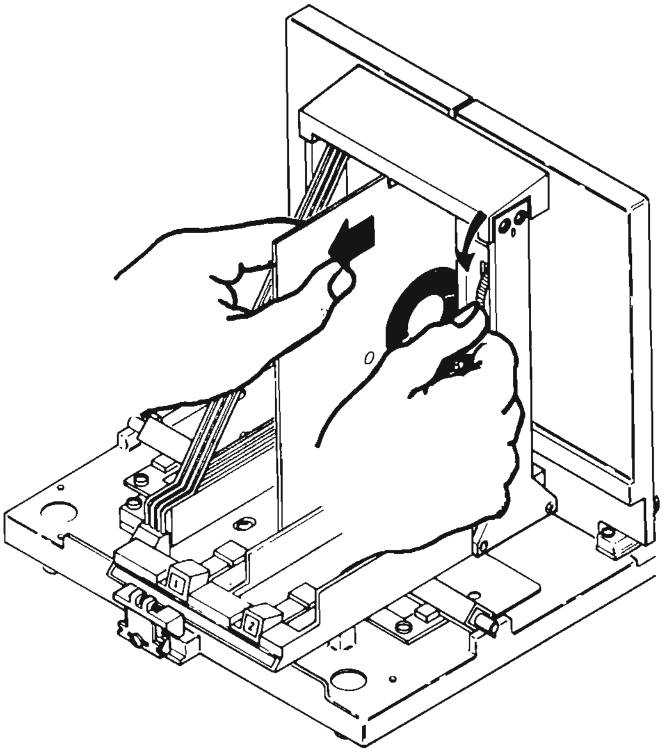


Figure 3-3. Removing Diskette from Drive Station

I/O Slot Diskette

To remove a diskette from I/O slot 1, 2, or 3:

1. Push in on the manual thumbwheel and turn it downward until the diskette starts to move (as in Figure 3-3).

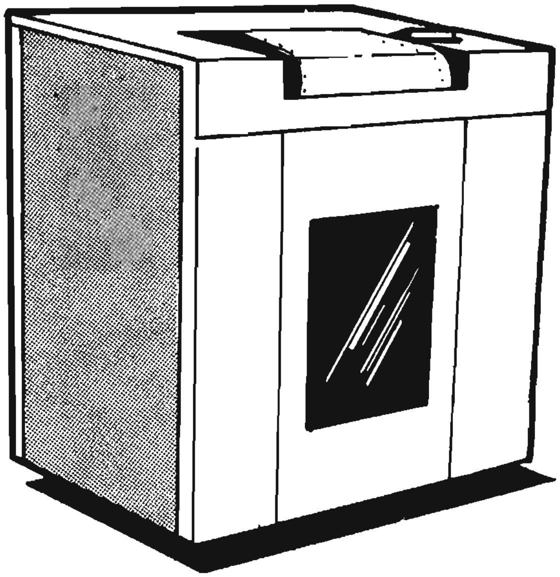
2. Release the thumbwheel and gently pull the diskette out.

If the diskette does not move when you turn the thumbwheel, contact your service representative and report that you have a jammed diskette.

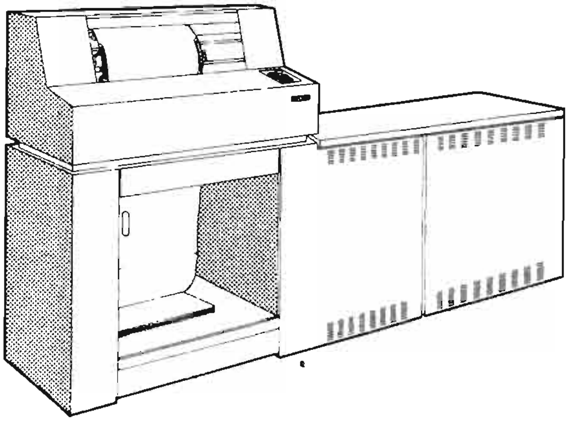
Chapter 4. IBM System/38 Peripheral Devices

5211 PRINTER; 3262 PRINTER; 3203 PRINTER

Your system may include one or two system printers. The system printers for the System/38 are the IBM 5211, IBM 3203, or the IBM 3262. The 5211 and the 3262 have a similiar exterior appearance.



5211 Printer; 3262 Printer (5211 shown)



3203 Printer

The 3203, 5211, and 3262 are line printers; that is, instead of printing a single character at a time, they print a complete line of up to 132 characters at a time. Several different type styles are available on interchangeable print belts or interchangeable cartridges. The printing speed will vary depending on the size of the character set, the number of characters, the actual characters being printed, and the form skips associated with the output to be printed. The printers can print an original copy with up to five carbon copies, on paper from 88.9 mm to 387.35 mm (3.5 in to 15.25 in) wide [406.4 mm (16 in) wide on the 3262] [508 mm (20 in) wide on the 3203] and from 76.2 mm to 355.6 mm (3 in to 14 in) long on the 5211 and 3262, [76.2 mm to 610 mm (3 in to 24 in) long on the 3203.

Each printer is identified to the system by a separate device description. To identify a particular printer in a command, you use the name of its device description. The device description is created by the CE at installation time. The name of the first system printer should be QSYSPRT.

Your interaction with the 5211 Printer or the 3262 Printer or the 3203 Printer in terms of system operations is described in the section on *Operator/Device Interaction* in Chapter 9. Specific procedures for preparing and maintaining the printers for operation are given in the separate operator's quides for the printers as follows:

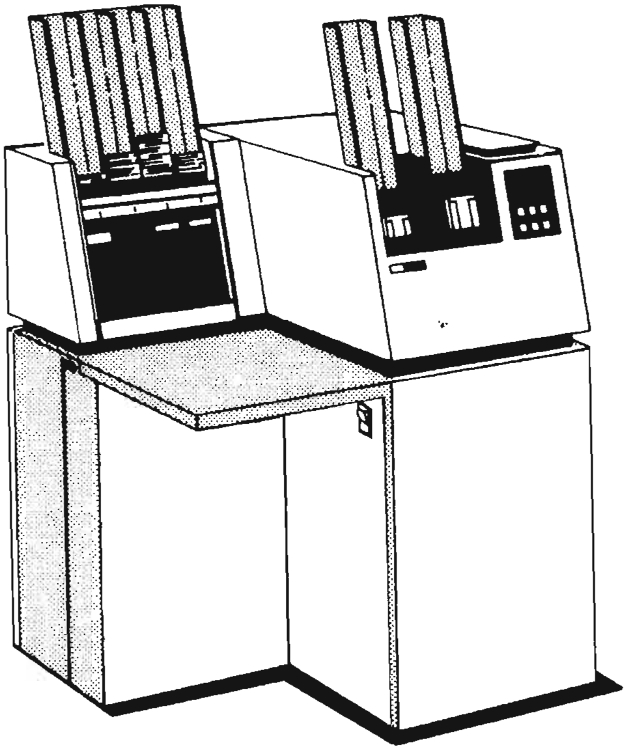
- *IBM 5211 Printer Models 1 and 2 Component Description and Operator's Guide,* GA24-3658

- *IBM 3262 Printer Models A1 and B1 Component Description and Operator's Guide,* GA33-1530

- *IBM 3203 Printer Model 5 Component Description and Operator's Guide,* GA33-1529

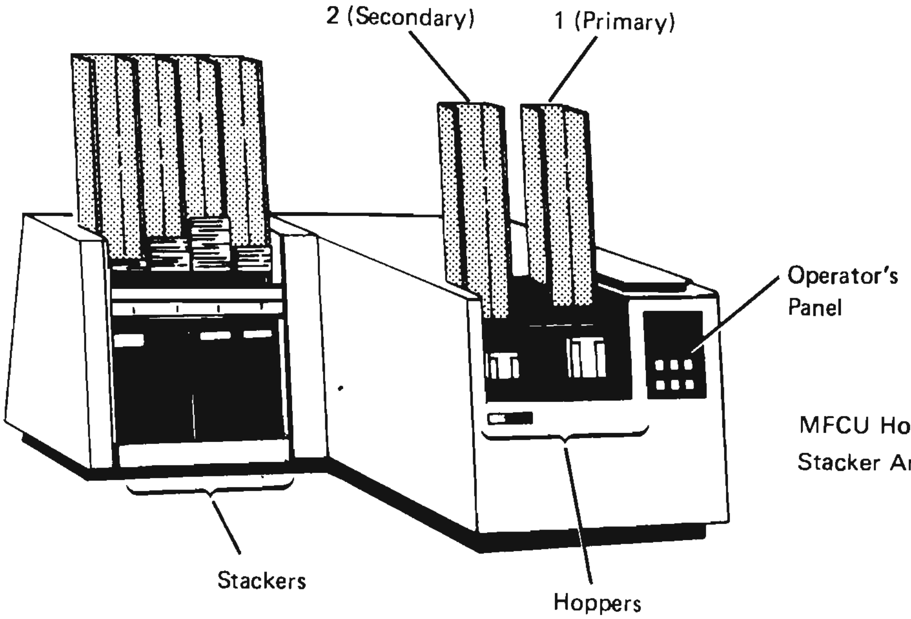
5424 MULTI-FUNCTION CARD UNIT

The IBM 5424 Multi-Function Card Unit (MFCU) provides input to and/or produces output from System/38 on 96-column cards. The configuration of the 5424 as used with System/38 is shown below.



Multi-Function Card Unit

As an input device the MFCU can read cards from either of the two hoppers and stack cards in any of the four stackers (below). As an output device the MFCU can punch, print, and stack cards in any of the four stackers.



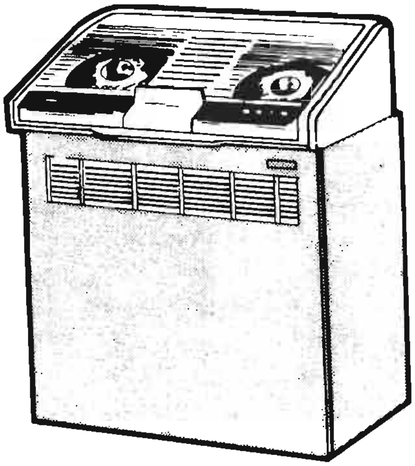
MFCU Hopper and Stacker Arrangement

To specify the MFCU in a command, you use the name of the device description that identifies it to the system. IBM supplies a standard device description for the MFCU; its name is QCARD96.

Your interaction with the MFCU in terms of system operations is described in the section on *Operator/Device Interaction* in Chapter 9. Specific procedures for preparing the MFCU for operation and maintaining it are given in the *IBM 5424 and 5425 Multi-Function Card Units Operator's Guide and Programmer's Reference Manual*, GA21-9167.

3410/3411 MAGNETIC TAPE UNITS

The 3410/3411 magnetic tape units (below) provide input to and/or output from System/38. Most save/restore operations that can be done on diskettes can also be done on magnetic tape using the 3410/3411; however, system objects and libraries can be saved or restored only on diskettes, not on magnetic tape. The 3410 and 3411 are the same except that the 3411 contains a control unit.



3410/3411 Magnetic Tape Unit

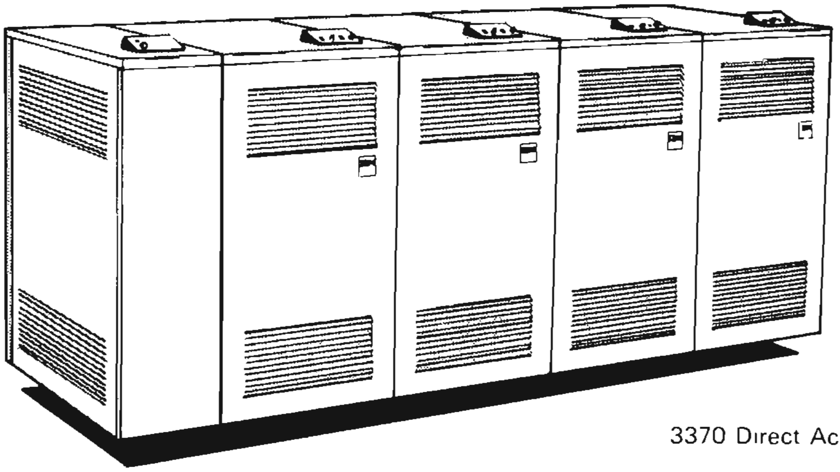
Each 3410 is identified to the system by a separate device description. To specify a 3410 in a command, you use the name of its device description. The 3411 is identified to the system by a control unit description. To specify the 3411 in a command, you use the name of its control unit description. The device description is created by the CE at installation time. The names of these device and control unit descriptions should be:

Unit	Device Description	Control Unit Description
3410	QTAPE2 QTAPE3 QTAPE4	
3411	QTAPE1	QTAPE

The 3410/3411 tape units are operated under the control of system programming. Your interaction with the tape units in terms of system operations is described in the section on *Operator/Device Interaction* in Chapter 9. Specific procedures for loading and unloading tape reels and maintaining the tape units are summarized on the reference card, *IBM 3410/3411 Operator's Guide*, G232-0004.

IBM 3370 DIRECT ACCESS STORAGE DEVICE

The IBM 3370 (below) provides additional direct access storage to augment the auxiliary storage inside the system unit. Once installed, the 3370 is handled by the system as a part of auxiliary storage.



3370 Direct Access Storage Device

Because the direct access storage is controlled entirely by the system, no operator interaction with the 3370 is required. Each 3370 unit does, however, have three switches what must be in a specific position for normal operations. The proper switch positions are as follows:

Switch	Correct Position
Enable On–Delayed Power Off	Enable On
Read/Write–Read Only	Read/Write
Ready (push button)	(Do not press)

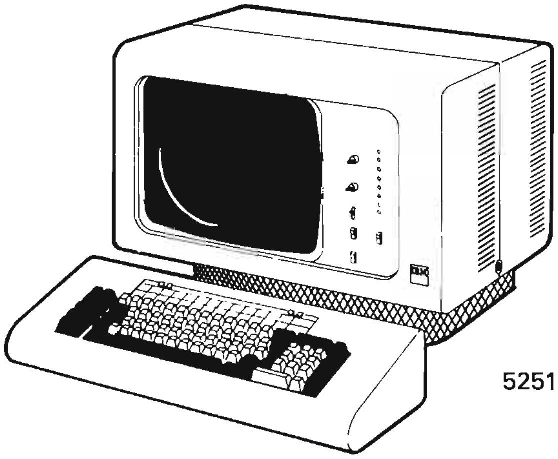
Once these switches are set correctly, they should not be touched unless you are instructed to do so by your service representative.

Chapter 5. IBM 5250 Information Display System

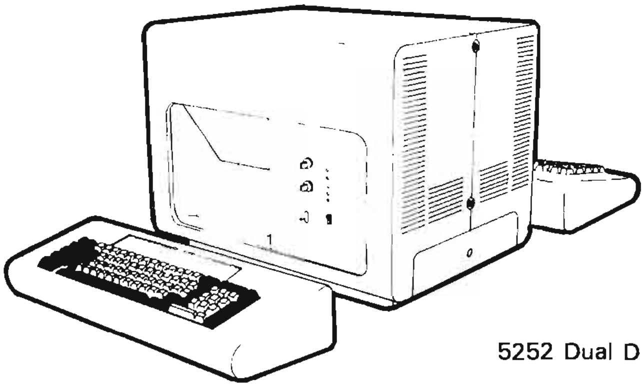
The 5250 Information Display System provides System/38 users with interactive work stations, supported by a tabletop printer, for entering information in the system and obtaining information from the system. The 5250 components are the 5251 Display Station, the 5252 Dual Display Station, and the 5256 Printer. Once you have prepared the system for operation at the system console, you can use a 5251 or 5252 instead of the system console to monitor and control system operations.

5251/5252 DISPLAY STATIONS

The 5251 Display Station (below) has a single display screen and keyboard. The 5252 Dual Display Station (below) has two back-to-back screens, each with a separate keyboard, that can be used independently. The operation of either 5252 display unit is the same as the operation of the 5251.



5251 Display Station



5252 Dual Display Station

Some versions of the 5251 are directly connected to the system through a cable (Models 1 and 11); other versions are used as remote terminals that are connected to the system through communication media such as modems and telephone lines (Models 2 and 12). The 5252 can only be cable-connected. At remote locations, a combination of remote and cable-connected display stations may be used. For example, a 5251 Model 1 and a 5252 may be connected by a cable to a 5251 Model 12, which serves as their controller and provides the communication link with the system.

All display stations to be used as work stations on the system must be defined to the system (the procedure is described in the *System/38 Guide to Program Product Installation and Device Configuration*). Once this procedure has been completed, a display station becomes available for use when:

- The subsystem in System/38 that is controlling the display station is active (see *Subsystems* in Chapter 8).

- The Power switch on the display station control panel is set to Power On.

- The communications link to the system is established (necessary only for remote display stations).

To identify a display station in a command, you specify the name of the device description that was created when the display station was defined to the system. Similarly, to identify a control unit or a line through which a display station communicates with the system, you specify the name of the appropriate control unit description or line description.

If your intervention is required in making a connection between the system and a remote terminal, refer to the procedures described in the section on *Handling Switched Lines* in Chapter 9.

If a failure occurs in the communications link between a display station and the system (such as a dropped line), you are notified by a message from the system (see *Handling Error Messages* in Chapter 11). You may not be aware of a problem with a display station until you are contacted by the person who is using it. The actions you should take in resolving work station problems are included in the problem determination procedures in Chapter 14.

For complete operating procedures on the 5251 and 5252, refer to:

- *IBM 5251 Models 1 and 11 Display Station, IBM 5252 Dual Display Station Operator's Guide*, GA21-9248 (directly attached display stations)

- *IBM 5251 Models 2 and 12 Display Station Operator's Guide*, GA21-9323 (remote communicating display stations)

Other details on the use of the 5251 or 5252 as a work station for System/38 are given in the *System/38 Work Station User's Guide*.

Differences Between Display Stations and the System Console

Several differences between the operation of the 5251/5252 and the operation of the the system console should be noted. (The system console is described in Chapter 2.)

Display Screen

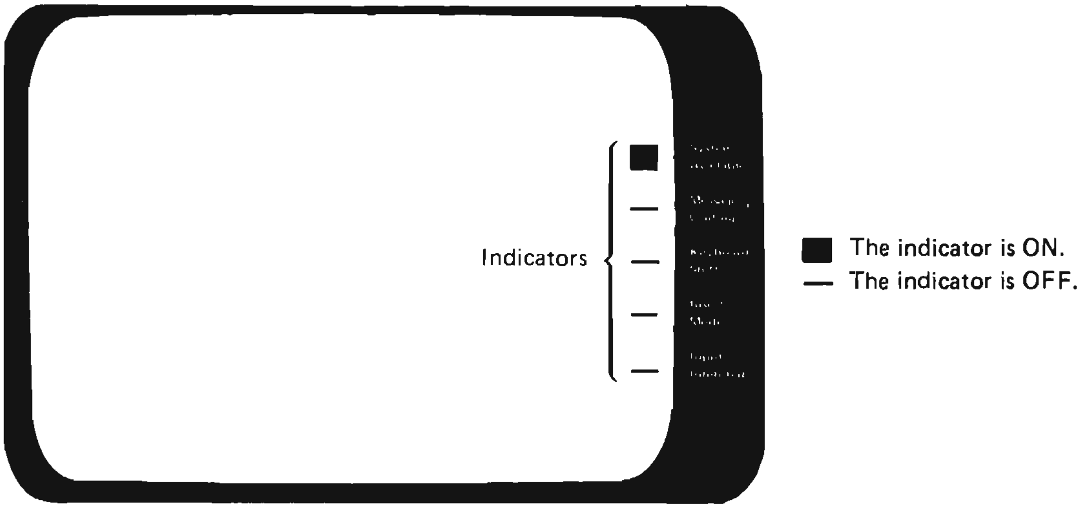
The size of the display screen on the 5251 depends on the model you are using. Models 1 and 2 have a screen that can display a maximum of 960 characters on 12 lines of 80 characters each. Models 11 and 12 have a larger screen that can display up to 1920 characters on 24 lines of 80 characters each.

Each 5252 screen has the same capability as the 960-character screen on the 5251 Models 1 and 2.

Regardless of the screen size, displays used with System/38 CPF programs will generally appear the same on the 5251 and 5252 as on the system console. Because the system console screen can display fewer characters per line, however, some information displayed as a single line on the 5251 or 5252 screens will be folded (carried over to a second line) or truncated (cut off) on the system console screen.

The 5251/5252 display screens have the additional capability for blinking images or reverse image (dark characters on a light background) that is not available on the system console. For example, if the system cannot process a request you have entered because you have provided incorrect information, the input area in error may be shown in reverse image on the screen of the 5251 or 5252, but not on the system console.

Each 5251/5252 display screen has five status indicators on the side of the screen (below), compared with four on the system console.



The five indicators have the following meaning:

System Available: When this indicator is on (■), the display station is communicating normally with its control unit; when it is off (-), the display station is not communicating with its control unit and cannot be used. The system might not be available for use even when the indicator is on. There is no similar indicator on the system console screen.

Message Waiting: This indicator performs the same function as the Attention indicator on the system console screen. It comes on (■) when there is a message for you; it is off (-) when there are no messages. A keyboard alarm also buzzes briefly when this indicator comes on.

Keyboard Shift: This indicator performs the same function as the Upper Shift indicator on the system console screen. It is on (■) when the keyboard is in upper shift; it is off when the keyboard is in lower shift.

Insert Mode: This indicator comes on (■) when you press the Insert Mode key and the display station is in insert mode; it is off (-) when the display station is not in insert mode. The insert mode allows you to insert a character where another character already exists without destroying the existing character. There is no similar insert mode function on the system console.

Input Inhibited: This indicator performs the same functions as the Input Inhibited and Reset Required indicators on the system console screen. The indicator comes on (■) when keyboard input is not accepted (keyboard locked); it is off (-) when input from the keyboard is accepted.

Keyboard

Each 5251 and 5252 keyboard can have one of three possible arrangements. The typewriter-like keyboard (Figure 5-1) has an arrangement similar to the system console keyboard, except for the function keys on the left and right of the keyboard and the calculator-like numeric pad on the right of the keyboard. The basic data entry keyboard has an arrangement similar to a keypunch (top of Figure 5-2). With the optional proof feature, the numeric keys on the data entry keyboard are arranged like an adding machine (bottom of Figure 5-2).

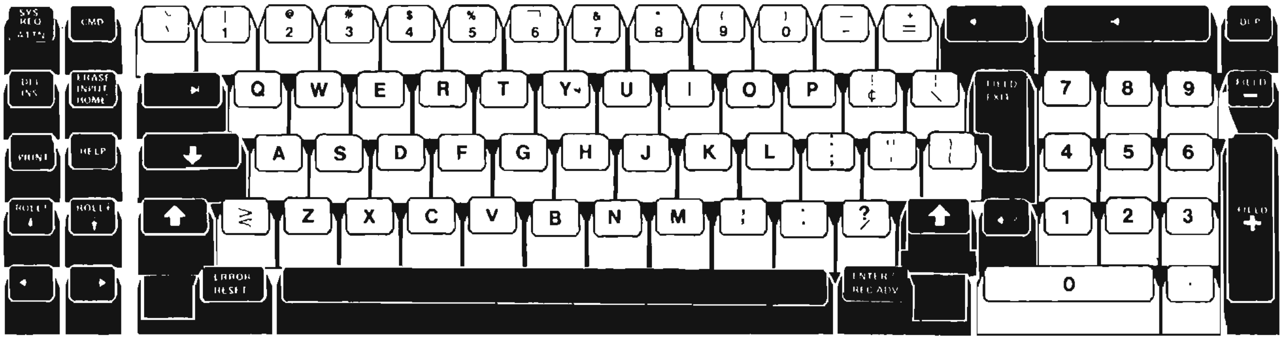


Figure 5-1. 5251/5252 Typewriter-Like Keyboard

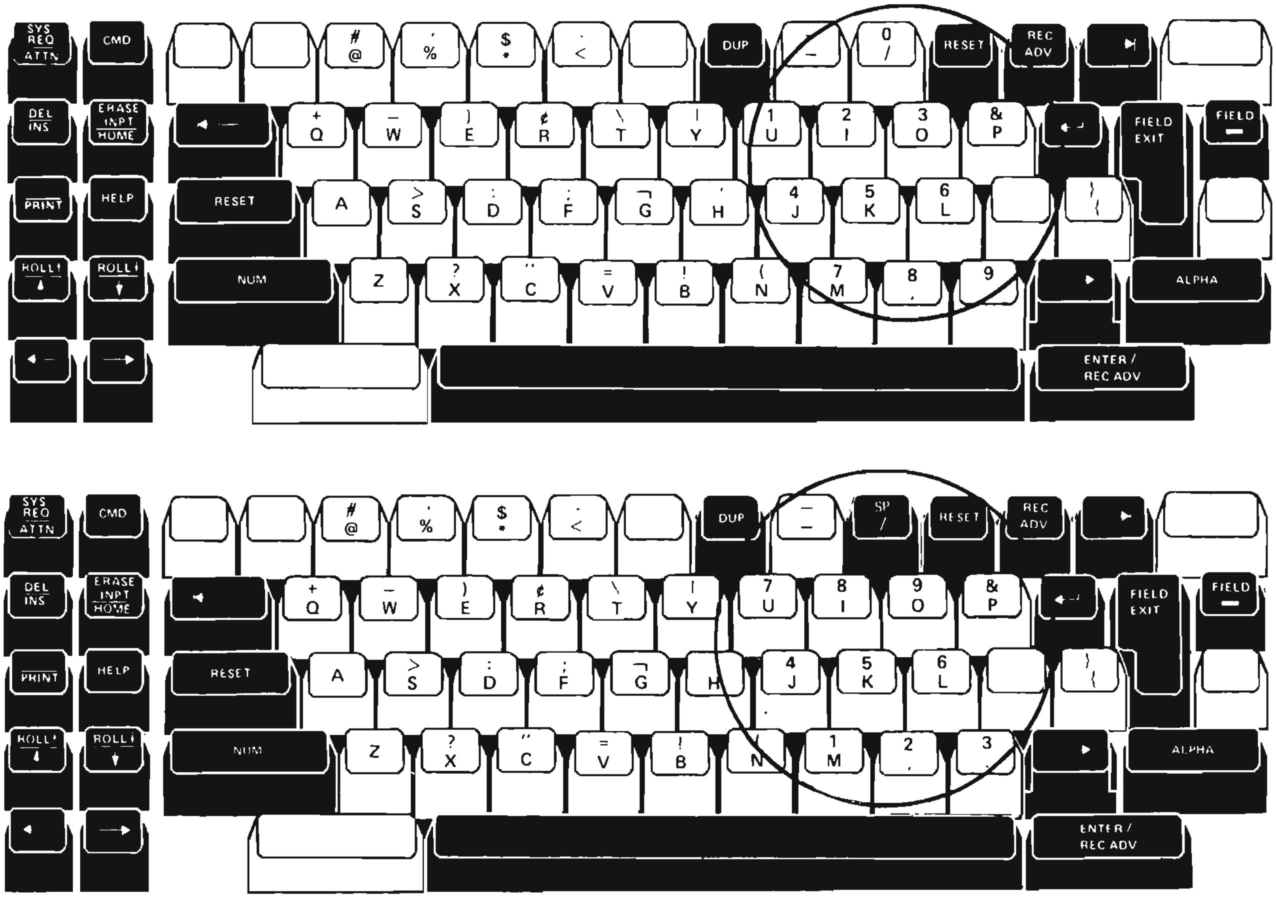
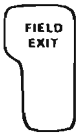


Figure 5-2. 5251/5252 Data Entry Keyboards

In particular, note the following differences between the 5251/5252 keyboards and the system console.

Keying Errors: A 5251 or 5252 display station checks for keying errors as you key in your input. The system console does not. If a keying error is detected by a 5251 or 5252, the keyboard is locked and an error code is displayed on the screen. A discussion of *Keying Errors* is included in Chapter 11.

Field Exit Key: In addition to a Field Advance key similar to that on the system console, each 5251/5252 keyboard has a Field Exit key.



As does the Field Advance key, the Field Exit key moves the cursor from the input field where it is currently positioned to the beginning of the next input field. (An input field is an area on a display where you are allowed to enter information.) Unlike the Field Advance key, however, the Field Exit key will usually also insert nulls from the original position of the cursor to the end of the field. An exception is a right-adjust field, where your input is shifted to the right and nulls are inserted in the vacated positions. In either situation, inserting the nulls has the effect of blanking out the remainder of the field. For example, if you positioned the cursor at the beginning of an input field and pressed the Field Exit key, nulls would be inserted in the entire field, erasing any information that had been in the field, and the cursor would move to the beginning of the next input field. The system console has no equivalent key. To blank out all or part of a field on the system console, you must insert blanks by pressing the Spacebar.

Error Reset: The 5251/5252 display stations have the capability to check for keying errors. When a keying error is detected, the 5251/5252 keyboard locks and an error code appears on the message line of the display currently on the screen (see *Keyboard Errors on 5251/5252* in Chapter 11). You must press the Reset key or the Error Reset key to unlock the keyboard and continue. Depending on the display, the 5251/5252 keyboard may also lock when a error message appears on the message line. Again, you must press the Reset key to continue. The system console does not check for the keying errors, nor does the system console keyboard lock when an error message is received. Pressing the Reset key is not required in either case.

Special Characters: A few of the characters in the special character set for the 5251/5252 keyboards are different from the equivalent special characters for the system console keyboard. Therefore, if any of those characters are entered at one device and transmitted to the other device, those characters will be different when displayed on the other device. If both devices have the Multinational Character Set feature, there will be no difference between the characters.

Command Functions: The 5251/5252 keyboards do not have the separate command function keys that are on the system console keyboard (keys labeled CF1 through CF24 as shown in Figure 2-1). To obtain a command function on the 5251 or 5252, you press the Cmd key followed by one of the top row keys, either with or without an upper shift.

The relationship of the top row keys on the 5251/5252 keyboards to the CF keys on the system console keyboard is shown below.

To obtain a command function from CF1 to CF12 on the 5251 or 5252:

1. Press the Cmd key.
2. Press the appropriate top row key as shown below.

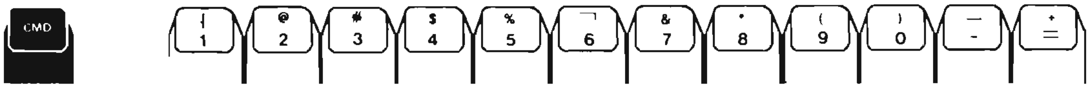
To obtain a command function from CF13 to CF24:

1. Press the Cmd key.
2. Press and hold the Shift key.
3. Press the appropriate top row key as shown below.

If you have pressed the Cmd key, but decide not to use a command function, you can cancel the selection procedure by pressing the Error Reset key. (On the data entry keyboards, use the Reset key located above the Numeric Shift key.)

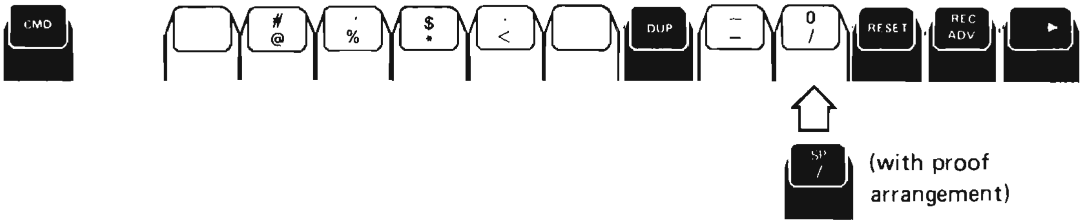
Typewriter-Like Keyboard

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)

Special Function Key Use for System/38

Function keys on the 5251/5252 keyboards (dark keys in Figures 5-1 and 5-2) generally perform the functions described in the 5251/5252 operator's guides that were referenced earlier in this chapter. 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: As on the system console, the Sys Req key on the 5251/5252 keyboards is used to obtain the system request menu (see *Using the System Request Menu* in Chapter 6). However, you must hold down the Shift key when you press the Sys Req key on the 5251/5252 keyboards, and you must press the Enter/Rec Adv key after you have pressed the Sys Req key.

Attention (Attn) Key: The Attn key has no function in IBM-supplied CPF programming. The interrupt or escape function usually associated with the Attn key is performed by the Sys Req key. There is no Attn key on the system console.

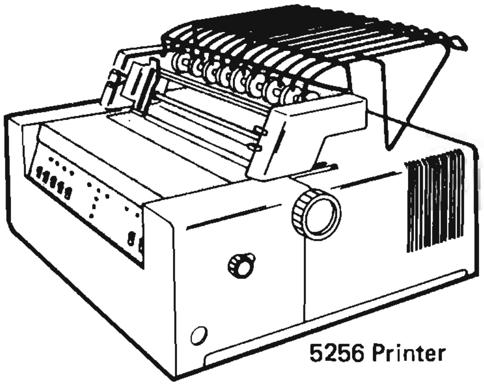
Erase Input Key: Pressing the Erase Input key will erase the information you have entered in all input fields on your current display. There is no Erase Input key on the system console. To erase information from an input field on the system console, you must space through the field manually with the Spacebar.

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. There is no Home key on the system console.

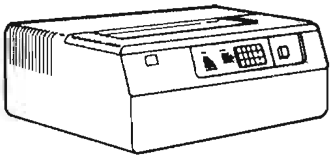
Print Key: When you press the Print key, the information in the display currently on your screen can be printed. The printing may be done on a 5256 Printer located near the display station you are using or spooled for another printer specified for the program currently in control of your display station. The program in control determines whether printing is allowed and how it is done. The CF24 key on the system console performs the same print request function as the Print key.

5224 PRINTER; 5225 PRINTER; 5256 PRINTER

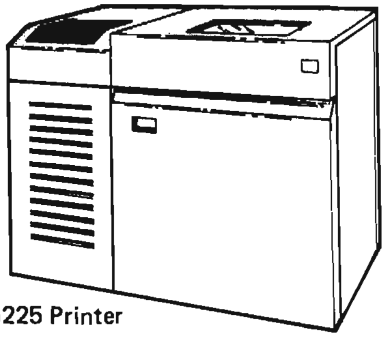
The 5224, 5225, and 5256 are work station printers that support the printing needs of users at one or more 5251 or 5252 display stations. They are cable-connected to a display station.



5256 Printer



5224 Printer



5225 Printer

If you have display stations in your area, you may be responsible for maintaining the printers supporting them. Your duties will then include putting each printer in a ready condition, loading and aligning forms, and replacing the print ribbon. For complete operating procedures on the printer see the *IBM 5256 Printer Operator's Guide*, GA21-9260, the *IBM 5224 Models 1 and 2 Operator's Guide*, GA34-0092, or the *IBM 5225 Models 1, 2, 3, and 4 Operator's Guide*, GA34-0054.

As for display stations, the printers must be defined to the system before they can be used. You identify it in a command by specifying the name of its device description.

PART II
SYSTEM OPERATION

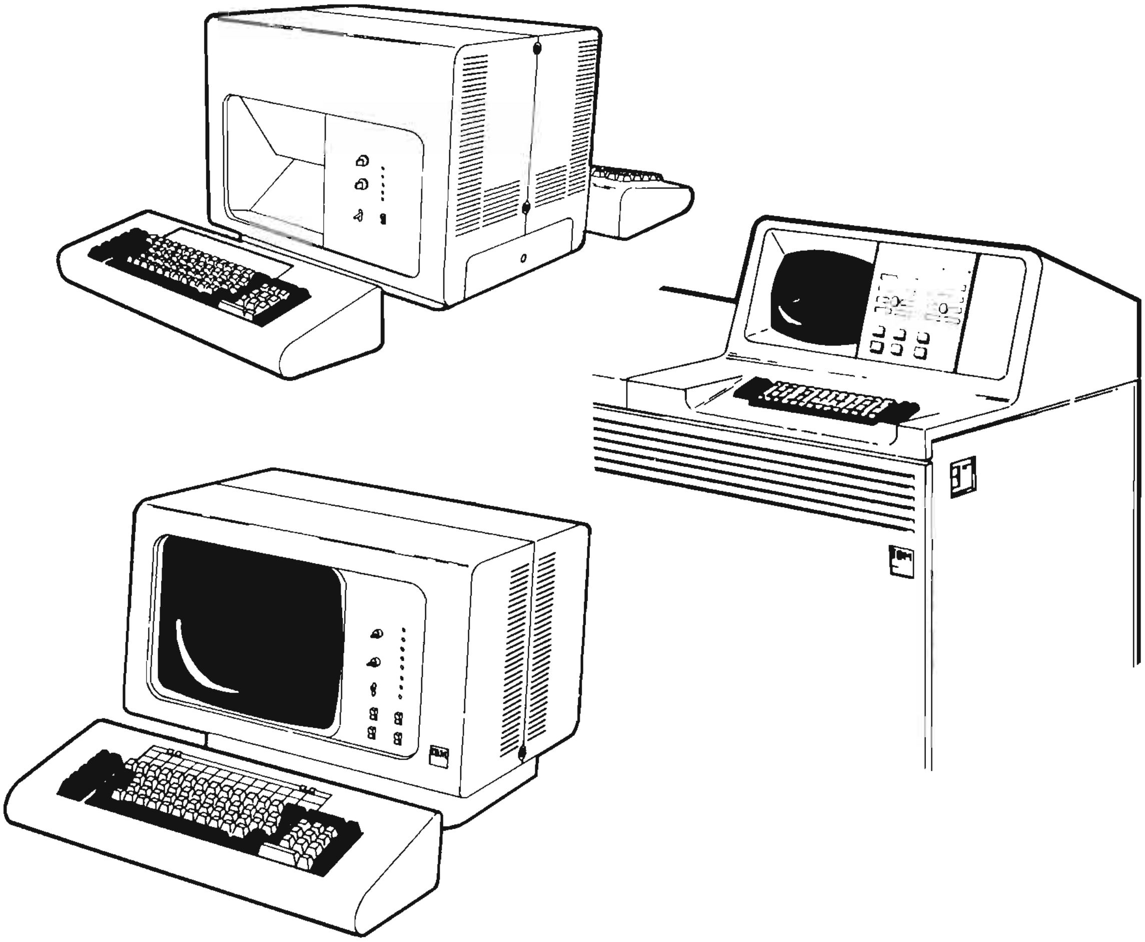
This page is intentionally left blank.

Chapter 6. Using a Work Station

A work station is an interactive terminal that you use to communicate with and control the system. Interactive means that the system acts on each request individually, as if you are conversing with the system.

Your work station may be the system console or a 5251 or 5252 Display Station. Other than its special capabilities in starting the system (see Chapter 8), the system console is used the same way as any other work station.

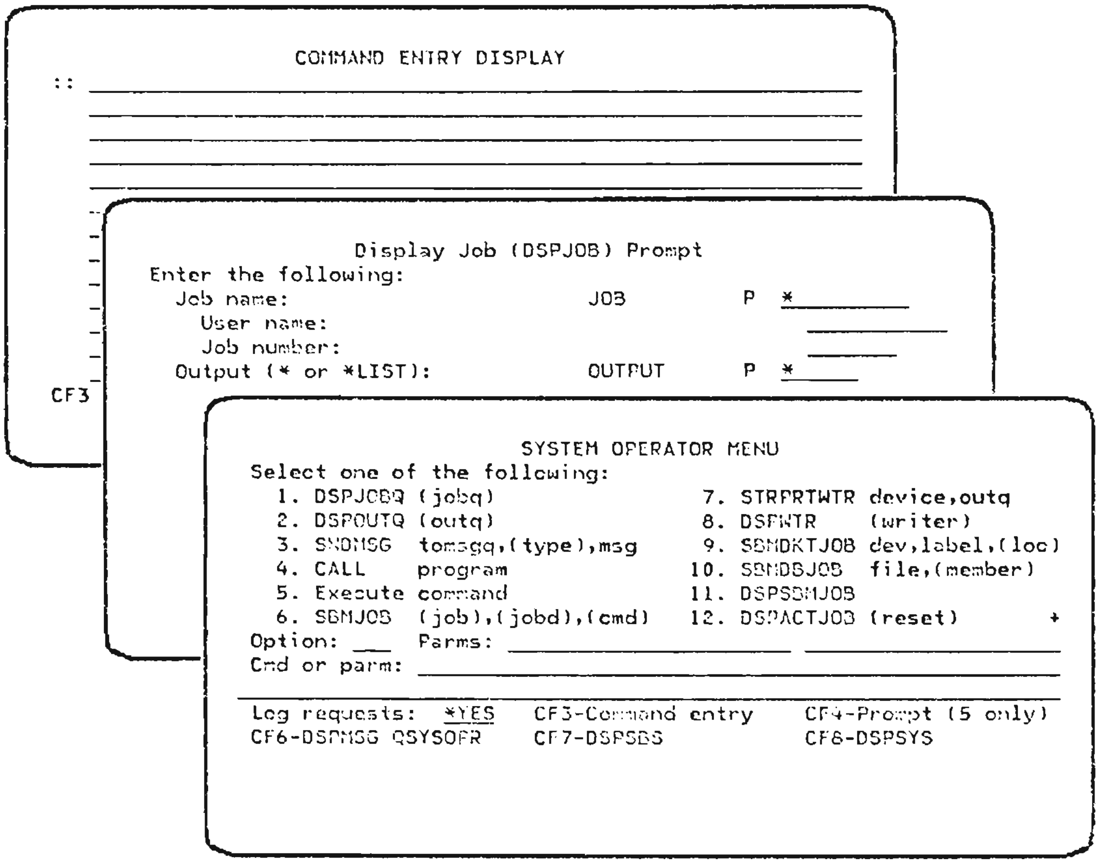
Before using the system console, you should review the procedures in Chapter 2. Before using a 5251 or 5252 Display Station, you should review the procedures in Chapter 5 and the appropriate operator's guide referenced in that chapter.



DISPLAYS

You control the system by interacting with displays on a work station. The displays and the means of interacting with them that are described in this manual are part of the Control Program Facility (CPF), which is available from IBM.

CPF displays are formatted such that you can enter information only in certain areas, called input fields. Except for the sign-on display discussed later in this chapter, each input field is indicated by a continuous underscore (Figure 6-1).



```
                         COMMAND ENTRY DISPLAY
 :: ____________________________________________________________
    ____________________________________________________________
    ____________________________________________________________
    ____________________________________________________________
    ____________________________________________________________
 CF3
```

```
                    Display Job (DSPJOB) Prompt
Enter the following:
  Job name:                               JOB        P  *_________
    User name:                                            ___________
    Job number:                                           _______
  Output (* or *LIST):                    OUTPUT     P  *____
```

```
                          SYSTEM OPERATOR MENU
Select one of the following:
  1. DSPJOBQ (jobq)                     7. STRPRTWTR device,outq
  2. DSPOUTQ (outq)                     8. DSPWTR    (writer)
  3. SNDMSG  tomsgq,(type),msg          9. SBMDKTJOB dev,label,(loc)
  4. CALL    program                   10. SBMDBJOB  file,(member)
  5. Execute command                   11. DSPSBMJOB
  6. SBMJOB  (job),(jobd),(cmd)        12. DSPACTJOB (reset)         +
Option: ___  Parms: ______________________  _____________________
Cmd or parm: _____________________________________________________
_________________________________________________________________
Log requests:  *YES    CF3-Command entry      CF4-Prompt (5 only)
CF6-DSPMSG QSYSOPR     CF7-DSPSBS             CF8-DSPSYS
```

Figure 6-1. Typical Operator Displays

When a display containing more than one input field appears on your screen, the cursor is located at the first position of the first input field. To insert information into an input field, you key it in using the work station keyboard as you would a typewriter. As you key in each character, it appears on the screen and the cursor moves to the next input position. When the cursor reaches the end of one input field, it moves to the beginning of the next input field. If the information you have keyed in does not fill the input field so that the cursor does not automatically move to the next input field, you can advance the cursor to the next field yourself by pressing the Field Advance key ⇥ on the system console (or the Field Exit key on a 5251 or 5252).

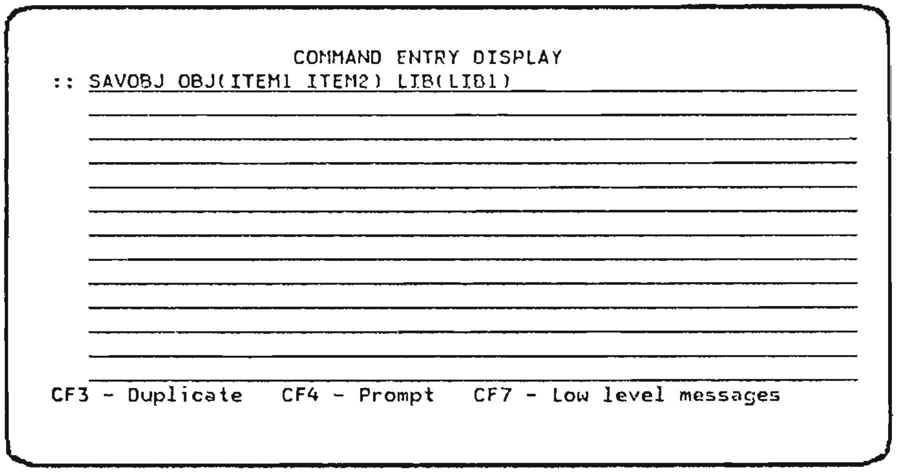
As indicated in Figure 6-1, a single input field can extend to more than one line of a display. When inserting information into such a field, you key in your input as if you were typing it on one continuous line. When you reach the end of one line, the cursor automatically moves to the beginning of the next line and any subsequent characters you key in appear on that 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. Note, however, that the remainder of the line will be reproduced as spaces in any subsequent display of your input.

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

When you have completed all additions or changes to the display currently on your screen, you send the information to the system for processing by pressing the Enter key ENTER or in some cases one of the command function keys. The information is not acted on by the system until you press the Enter key or a command function key.

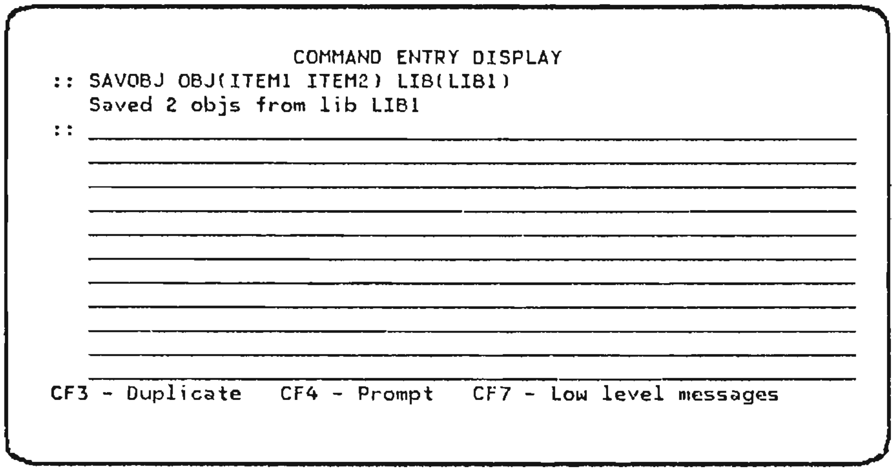
When the system processes the information you entered, its response is indicated on your screen in a variety of ways. The response could be a modification of your current display to incorporate information related to your request (Figure 6-2), an entirely new display providing information you requested (Figure 6-3), a message at the bottom of your current display (Figure 6-4), or merely repositioning the cursor for your next entry.

Display with Your Entry



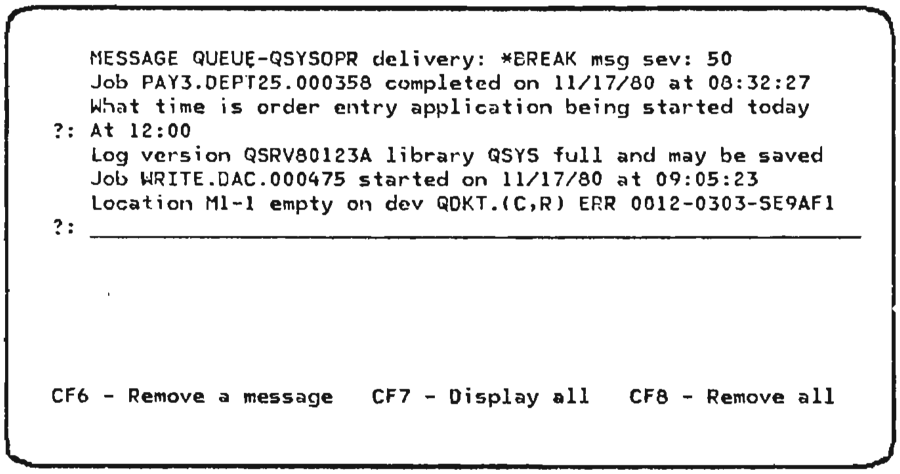
```
                    COMMAND ENTRY DISPLAY
 :: SAVOBJ OBJ(ITEM1 ITEM2) LIB(LIB1)











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

Modified Display after System Response



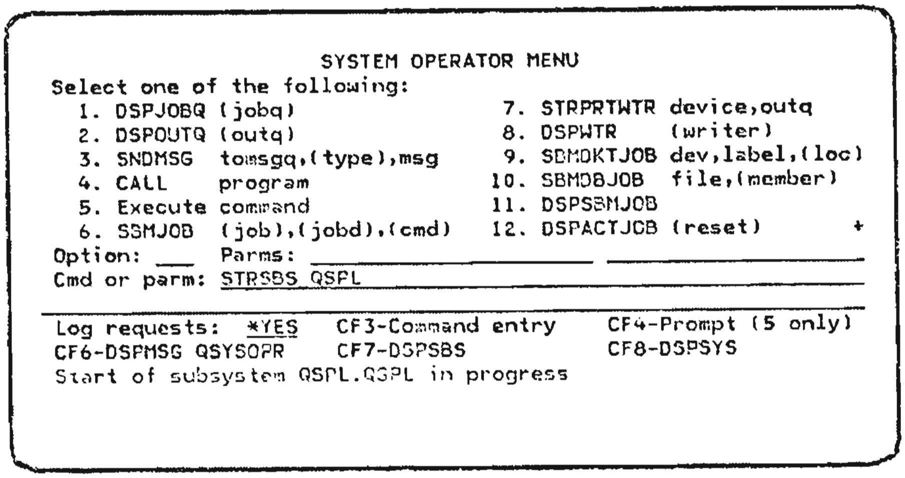
```
                    COMMAND ENTRY DISPLAY
 :: SAVOBJ OBJ(ITEM1 ITEM2) LIB(LIB1)
    Saved 2 objs from lib LIB1
 ::










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

Figure 6-2. Modified Display as Response



```
   MESSAGE QUEUE-QSYSOPR delivery: *BREAK msg sev: 50
   Job PAY3.DEPT25.000358 completed on 11/17/80 at 08:32:27
   What time is order entry application being started today
?: At 12:00
   Log version QSRV80123A library QSYS full and may be saved
   Job WRITE.DAC.000475 started on 11/17/80 at 09:05:23
   Location M1-1 empty on dev QDKT.(C,R) ERR 0012-0303-SE9AF1
?: ____________________________________________________________



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

Figure 6-3. New Display as Response



```
                     SYSTEM OPERATOR MENU
Select one of the following:
  1. DSPJOBQ (jobq)                   7. STRPRTWTR device,outq
  2. DSPOUTQ (outq)                   8. DSPWTR    (writer)
  3. SNDMSG  tomsgq,(type),msg        9. SBMDKTJOB dev,label,(loc)
  4. CALL    program                 10. SBMDBJOB  file,(member)
  5. Execute command                 11. DSPSBMJOB
  6. SBMJOB  (job),(jobd),(cmd)      12. DSPACTJOB (reset)      +
Option: ___  Parms: ______________________ ____________________
Cmd or parm: STRSBS QSPL_______________________________________
_______________________________________________________________
Log requests:  *YES   CF3-Command entry     CF4-Prompt (5 only)
CF6-DSPMSG QSYSOPR    CF7-DSPSBS            CF8-DSPSYS
Start of subsystem QSPL.QGPL in progress
```

Figure 6-4. Response Message on Current Display

WORK STATION SIGN-ON

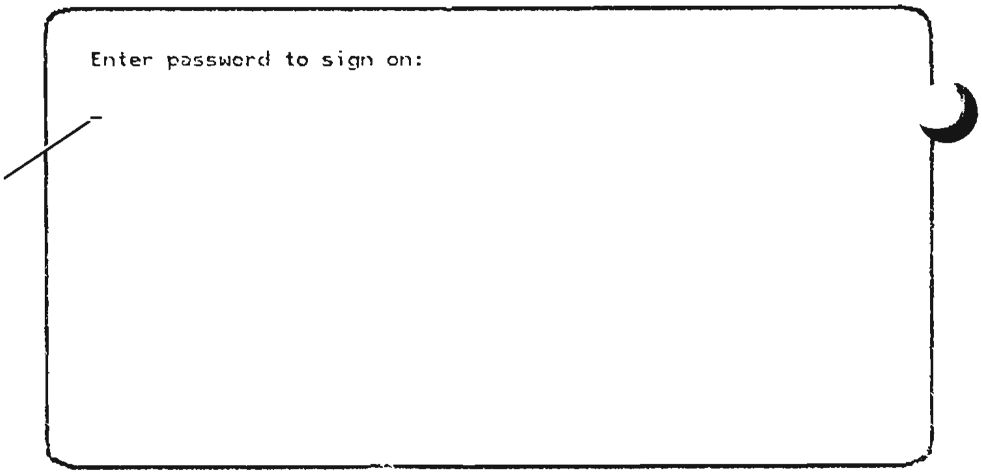
You sign on the system console as a part of the procedure for starting the system (see Chapter 8). You also may need to sign on a 5251 or 5252 work station before you can use it.

What Sign-On Means

When you sign on a work station, you create an interactive job, or session, with the system. That interactive job remains in effect until you sign off or cancel the job, power down the system, or terminate the subsystem that the work station was assigned to.

Sign-On Procedure

When the system is powered on and the power-on sequence is successfully completed, a sign-on prompt (below) is displayed on the system console screen. Once the system begins normal operations, any work station that is available for use but not currently signed on will display the same prompt, provided that the work station is assigned to an active subsystem.



```
Enter password to sign on:

_
```

The cursor appears here.

To sign on, you enter your password on the display. The cursor is automatically positioned at the beginning of the input field. Note that this input field is not underlined. It is a nondisplay field, which means that your entered password does not appear on the screen. This is to ensure that your password remains secret.

When you press the Enter key after keying in your password, the password is matched to a user profile stored in the system. This user profile tells the system what you are and are not authorized to do.

If the sign-on is successful, the sign-on prompt will be replaced by a new display, such as the start CPF prompt or the system operator menu (depending on whether you are starting the system or signing on after the system is already operating).

If Sign-On Fails

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

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

If your sign-on still fails, see the problem determination procedure in the *Problem Determination Guide.*

CONTROL LANGUAGE

When you request the system to take some action, your request is entered into the system in the form of a CPF control language (CL) command. CPF includes an extensive set of CL commands, each designed to perform a single function. You will probably not be using all of the CL commands. Those commands you are likely to use in operating the system are listed in Chapter 7.

The following discussion summarizes the fundamentals of CL commands and related CPF terminology that you may need in using the commands. For a complete description, see the *CPF Reference Manual—Control Language.*

CPF Terminology

Objects

The basic unit on which CPF operations can be performed is an *object.* Each object is a separate entity that has its own name by which users identify it to the system.

Typical CPF objects include:

- A *program,* containing executable code created in a source language such as CL or RPG, that performs a specific task

- A *file* that contains or provides access to a group of related data

- A *subsystem description* or *device description* that defines the characteristics and use of a particular subsystem or device

- A *command definition* stored in the system that describes how the system is to respond to a command you have entered

Most objects are grouped in *libraries,* which are objects themselves and are therefore identifiable to the system as separate entities. You can perform operations on individual objects in a library or on an entire library.

To perform an operation on an object, you must have *authority* to use it. This authority is given to you by either the object's owner, the security officer for your system, or someone else authorized to grant authority.

Object Names: All objects are referenced by name. There are three types of object names:

- Simple Object Name: The name of the object itself. Examples are:

  ITEM1

  PAY1

  UPDATE

- Qualified Object Name: The name of the object plus the name of the library in which it is located (connected by a period). Examples are:

  ```
  ITEM1.LIB1
     /    \
  Object Library
  Name   Name
  ```

  ```
  PAY1.PAYROLL
     /    \
  Object Library
  Name   Name
  ```

- Generic Name: A group of characters common to several object names (always followed by an asterisk). Examples are:

  ITEM*

  PAY*

  If a generic name is specified in a command, the system will search for all objects whose names begin with those characters.

An object that is supplied with the system has a name beginning with the letter Q. Examples are:

QSYS (system library)

QGPL (general purpose library)

QBATCH (batch subsystem description)

QSPL (spooling subsystem description)

All IBM-supplied objects are listed in the *CPF Programmer's Guide*.

Job Names: Job names can be qualified in a manner similar to a qualified object name. Instead of adding a library qualifier as in object names, however, you qualify a job name by adding the name of the user who submitted it or both the user name and the job number assigned to it when it entered the system. Examples are:

```
INVPRT
 ↑
 |
Job (no qualifiers)

PAY.DEPT25
 ↑   ↑
 |   |
Job User  (qualified job name)

CKPRT.SMITH.715475
 ↑      ↑      ↑
 |      |      |
Job    User  Job Number  (fully qualified job name)
```

Predefined Values: Any character, group of characters, or word preceded by an asterisk is a predefined value. Examples are:

*N (indicates null value for optional parameters)

*NO (indicates no change or no action for the keyword value)

*MOUNTED (indicates currently mounted diskette)

Standard Libraries

Several libraries are supplied with the system. They are:

- QSYS (system library): Contains IBM-supplied programs and data essential to system operation.
- QGPL (general purpose library): Contains user-created programs and data. This library contains IBM-supplied sample user objects and defaults when shipped.
- QSPL (spooling library): Contains IBM-supplied objects used for spooling.
- QTEMP (temporary library): Created by the system to contain temporary objects used by the jobs.
- QSRV (service library): Contains IBM-supplied programs and data for servicing the system.
- QRECOVERY (recovery library): Contains information for recovery after a system failure.
- QRCL (reclaim library): This library is created if needed during execution of a Reclaim Storage (RCLSTG) command.

Your system may have additional IBM-provided libraries if your installation is using IBM program products, such as RPG III (QRPG library) or IDU (QIDU library).

Command Syntax

In its most commonly used form, a CL command consists of a *command name* followed by one or more *parameters*, such as:

$$\underbrace{\text{SAVOBJ}}_{\textbf{Command Name}}\ \underbrace{\text{OBJ(ITEM1 ITEM2)}}_{\textbf{Parameter}}\ \underbrace{\text{LIB(LIB1)}}_{\textbf{Parameter}}$$

Blanks (spaces) serve as separators between the command name and the parameters.

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:

SAVOBJ (Save an Object)

SNDBRKMSG (Send a Break Message)

SBMJOB (Submit a Job)

STRCRDRDR (Start a Card Reader)

PWRDWNSYS (Power Down the System)

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 final abbreviation is shortened to a single character. Examples are:

CPYF (Copy a File)

CHGOUTQ (Change an Output Queue)

A few command names are composed of a complete word that describes the action to be taken. Examples are:

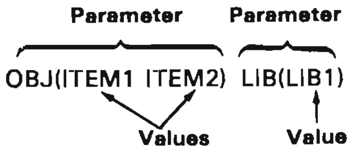
CALL (Call a program)

SIGNOFF (Sign Off a work station)

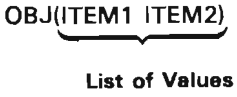
Parameter Set

A command may have one or more parameters, or no parameters at all. For some commands, certain parameters are required and must be specified 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. If more than one value is specified for a parameter, that set of values is called a *list*.

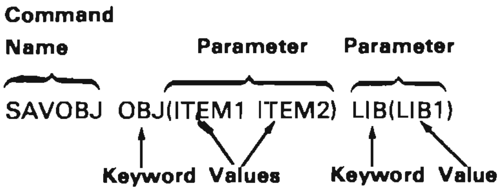


When you enter a command, you must determine the values to be used in each parameter of the command.

Although a command may have many parameters, you can usually execute the command by entering only a few of them. 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 one missing. By not specifying an optional parameter, you are, in effect, entering the default value of that parameter.

The parameters of a command, each with one or more values, can be coded in either of two ways:

1. *Keyword-with-Value Parameter*: The parameter consists of a keyword followed immediately by one or more values in parentheses.

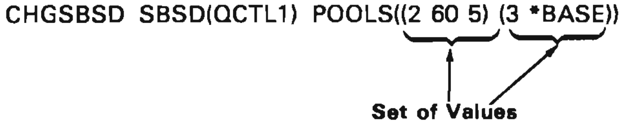


```
SAVOBJ OBJ(ITEM1 ITEM2) LIB(LIB1)
```

The parameters can be entered in any order:

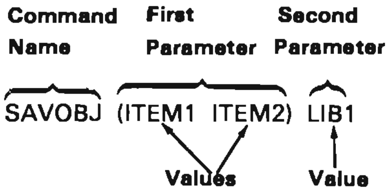
```
SAVOBJ OBJ(ITEM1 ITEM2) LIB(LIB1)

SAVOBJ LIB(LIB1) OBJ(ITEM1 ITEM2)
```

Where a parameter of a command can have sets of separately defined values (a *mixed list*), you must use a second set of parentheses to group each set of values within the parameter such as



```
CHGSBSD SBSD(QCTL1) POOLS((2 60 5) (3 *BASE))
```

2. *Positional Parameter*: 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 values must be entered in a specific sequence. For example, if the command just shown were entered in reverse order

   ```
   SAVOBJ LIB1 (ITEM1 ITEM2)
   ```

   the system would associate the values with the wrong parameter, or could not process the command at all. Note that parentheses are required only if there is more than one value in a parameter. As for the keyword-with-value form, a second set of parentheses must be used to group separate sets of related values within a parameter:

   ```
   CHGSBSD QCTL1 ((2 60 5) (3 *BASE))
   ```

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

   ```
   SAVOBJ (ITEM1 ITEM2) LIB1 *N *N *N SVVOL1
   ```

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

   ```
   SAVOBJ (ITEM1 ITEM2) LIB1
   ```

A command can be coded with some of its parameters in the positional form and others in the keyword form. If this mixed coding is used, however, the positional parameters must appear first:

```
SAVOBJ (ITEM1 ITEM2) LIB1 OBJTYPE(*ALL) LOC(*M1)
```

There is a limit on the number of positional parameters that can be safely used in CL programs. This limit is specified on the syntax diagram by the symbol ←⟨P⟩. On the prompt screens, the parameters that can be safely coded positionally are designated by the letter P.

Separators

When entering a command, you must place at least one space:

- Between the command name and the first parameter
- Between all parameters
- Between values within a parameter

```
SAVOBJ OBJ(ITEM1 ITEM2) LIB(LIB1)
```

At least one space is required.

```
SAVOBJ (ITEM1 ITEM2) LIB1
```

You are not limited to a single space, however. You can separate the command name, parameters, and values by any number of spaces. For example, you could put the command name and each parameter on a separate line:

```
SAVOBJ

OBJ(ITEM1  ITEM2)

LIB(LIB1)
```

If you are using the keyword-with-value coding, there must be no space between the keyword and the parenthesis that separates it from the value or values:

```
SAVOBJ  OBJ(ITEM1  ITEM2)  LIB(LIB1)
```

No space is allowed here.

Other Command Elements

Occasionally you may encounter a command that also has a label or a qualified command name, such as:

```
LOOP1: SAVOBJ.QSYS OBJ(ITEM1 ITEM2) LIB(LIB1)
  ↑           ↑
Label    Library Qualifier
```

The label is used to identify a command when it is called out in a branching command within a CL program.

A qualified command name is similar to the qualified object name described earlier in this chapter. The library qualifier, connected to the command name by a period, defines the library in which the system can find the command definition.

Job control commands used in batch jobs must be preceded by a double slash, such as

```
// JOB WKPAY CHECKS

// ENDJOB
```

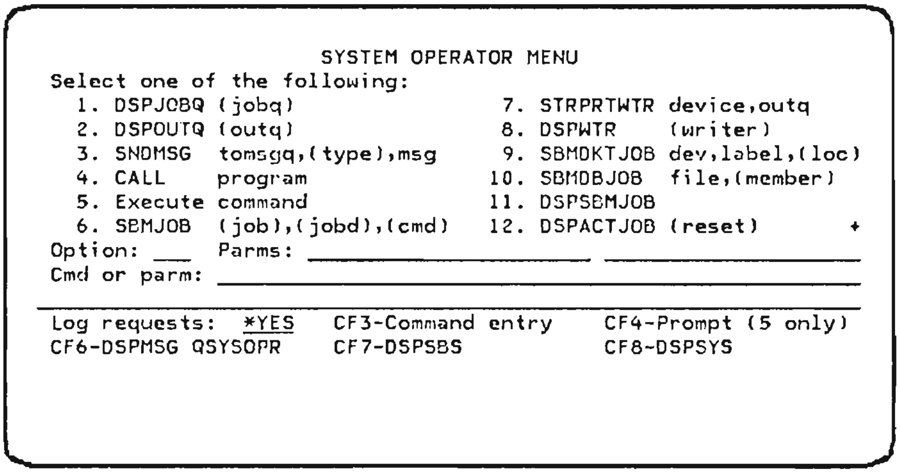
Conventions Used in Representing Commands

This manual follows the convention that parts of commands shown in uppercase must be entered exactly as shown, but the parts shown in lowercase are merely descriptive names for which you must substitute a specific value. The CPF displays you see on your work station screen generally follow the same convention.

In the examples in this manual, all commands are shown in uppercase to distinguish the actual values that you are to enter from the descriptive names of those values. However, you can also enter commands in lowercase.

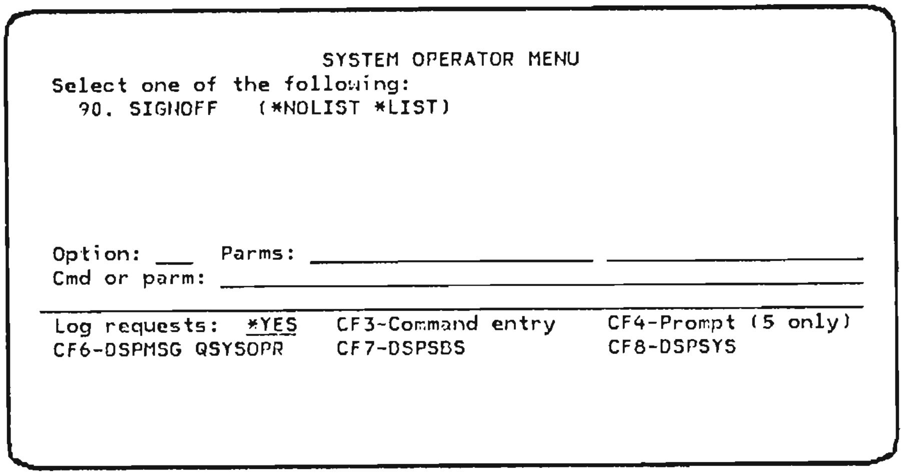
USING THE SYSTEM OPERATOR MENU

When you sign on the system with a password such as SYSOPR that is associated with the QSYSOPR user profile, you receive the system operator menu (below) as your basic working display. The menu is provided through the QOPRMENU program.



```
                         SYSTEM OPERATOR MENU
Select one of the following:
  1. DSPJOBQ (jobq)                  7. STRPRTWTR device,outq
  2. DSPOUTQ (outq)                  8. DSPWTR    (writer)
  3. SNDMSG  tomsgq,(type),msg       9. SBMDKTJOB dev,label,(loc)
  4. CALL    program                10. SBMDBJOB  file,(member)
  5. Execute command                11. DSPSBMJOB
  6. SBMJOB  (job),(jobd),(cmd)     12. DSPACTJOB (reset)        +
Option: ___ Parms: ______________________ _________________
Cmd or parm: ______________________________________________
___________________________________________________________
Log requests:  *YES   CF3-Command entry     CF4-Prompt (5 only)
CF6-DSPMSG QSYSOPR    CF7-DSPSBS            CF8-DSPSYS
```

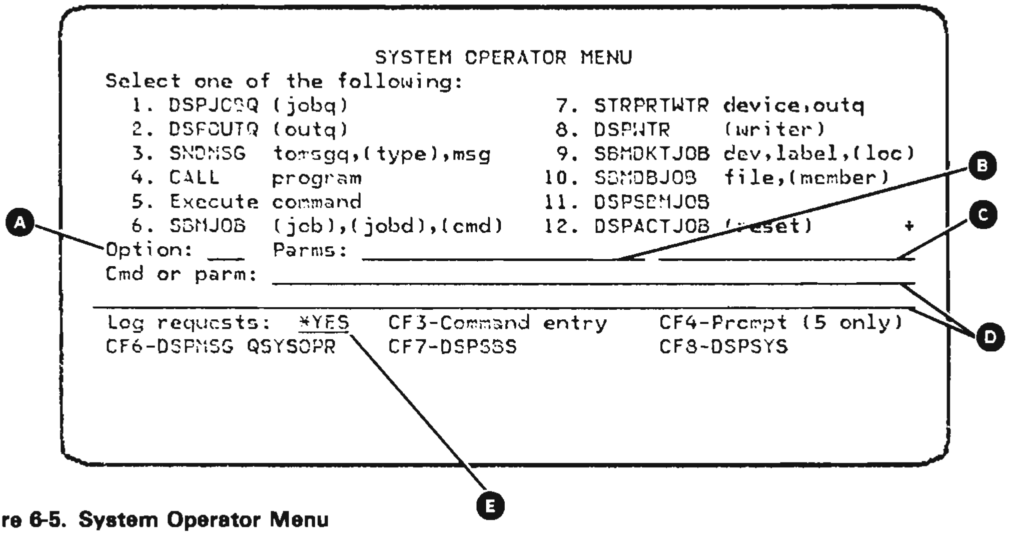
The menu is displayed on two screens when using the system console. To obtain the second display, press the Roll Up key.



```
                         SYSTEM OPERATOR MENU
Select one of the following:
  90. SIGNOFF   (*NOLIST *LIST)






Option: ___ Parms: ______________________ _________________
Cmd or parm: ______________________________________________
___________________________________________________________
Log requests:  *YES   CF3-Command entry     CF4-Prompt (5 only)
CF6-DSPMSG QSYSOPR    CF7-DSPSBS            CF8-DSPSYS
```

The system operator menu provides a shortcut method of executing the commands you are likely to use most frequently in performing your system operator duties. You can also execute other commands by entering them on the menu. Except for command entry (option 5 on the menu), requesting functions through the menu requires fewer system resources than using the general command entry facilities.

You execute a command by entering an option number and any required parameters on the menu or by pressing one of the CF keys indicated on the bottom of the menu. All option numbers except option 5 are associated with a specific command. The command name is shown in uppercase following the option number. Each lowercase word shown following the command name represents a separate parameter of the command.



```
                         SYSTEM OPERATOR MENU
Select one of the following:
  1. DSPJOBQ (jobq)                      7. STRPRTWTR device,outq
  2. DSPOUTQ (outq)                      8. DSPWTR    (writer)
  3. SNDMSG  tomsgq,(type),msg           9. SBMDKTJOB dev,label,(loc)
  4. CALL    program                    10. SBMDBJOB  file,(member)
  5. Execute command                    11. DSPSBMJOB
  6. SBMJOB  (job),(jobd),(cmd)         12. DSPACTJOB (reset)        +
Option: ___  Parms: ______________________  ______________________
Cmd or parm: _______________________________________________________
           _______________________________________________________
 Log requests:  *YES    CF3-Command entry     CF4-Prompt (5 only)
CF6-DSPMSG QSYSOPR      CF7-DSPSBS            CF8-DSPSYS
```

Figure 6-5. System Operator Menu

To use any of the options on the menu, enter the option number and any parameters in the input fields provided; that is:

1. Key the option number into the *Option* field A.

2. Key any parameters into the two *Parms* fields B and C.

3. If you are entering a command (options 5 and 6) or message (option 3), key it into the *Cmd or parm* field D. If your entry is too long to fit on the first line of field D, key it in as if you were entering it on a single line. The cursor will automatically advance to the beginning of the second line of field D when you reach the end of the first line.

4. If you do not want to log requests, enter *NO in the Log requests field E.

5. Press the Enter key.

You can move the cursor to the next input field (for example, from A to B) by pressing the Field Advance key. You can back up to the previous input field by pressing the Field Backspace key. If you want to move the cursor directly from the *Option* field to the *Cmd or parm* field, press the New Line key.

If you enter information into an input field that is not used for the option you selected, the information is ignored. For example, if you select option 1, the system will act only on the value you enter in the first *Parms* field B. Any information you entered in the second *Parms* field C or the *Cmd or parm* D field will be ignored.

The prompting functions described under *Command Selection and Prompting* in this chapter are also available for use from the system operator menu. If you press the prompt key (CF4) after keying in a 5 (for option 5) and the Cmd or Parm field is blank, you will receive a separate command grouping menu that you can use to select a command. If you press the prompt key after selecting option 5 and keying in a command name, you will receive a prompt display that shows all the parameters and default values for the specific command. Any parameters you key in before pressing the prompt key will be shown on the prompt. After it is executed, the prompted command will be returned to the *Cmd or parm* line of the menu. If the command is too long to fit in the field, the last three characters of the field will be replaced with three periods. The command cannot be reexecuted with the periods present because it is a partial command.

The following chart shows the parameters associated with the option you select in field Ⓐ. Parameters enclosed in parentheses are optional. The default for optional parameters is underlined.

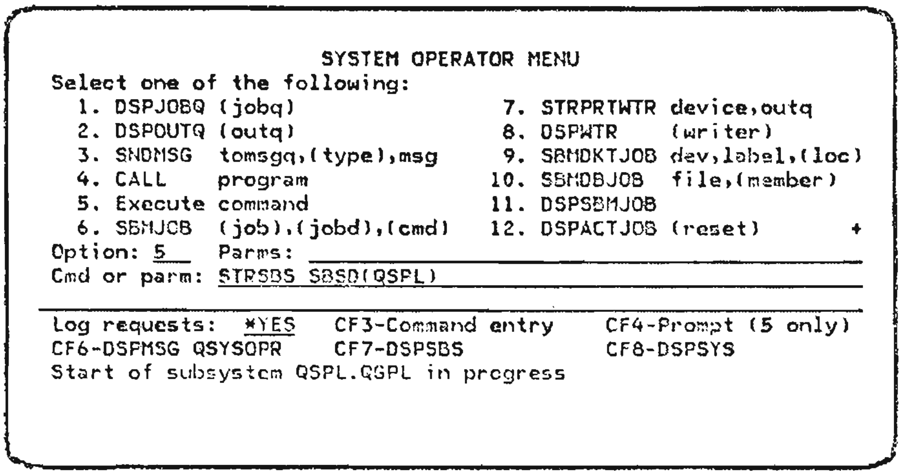
Option Field Ⓐ	**Associated Command**	**First Parms Field Ⓑ**	**Second Parms Field Ⓒ**	**Cmd or Parm Field Ⓓ**
1	DSPJOBQ	(*ALL) or (job name)		
2	DSPOUTQ	(*ALL) or (outq name)		
3	SNDMSG	tomsgq name	(*INFO) or (*INQ)	msg
4	CALL	program name		
5	Any Command			cmd
6	SBMJOB	(*JOBD) or (job name)	(QBATCH) or (job name)	(*JOBD) or (cmd)
7	STRPRTWTR	device name	outq name	
8	DSPWTR	(*ALL) or (writer name)		
9	SBMDKTJOB	device name	label	(*M1) or (loc)
10	SBMDBJOB	file name	(*FIRST) or (member name)	
11	DSPSBMJOB			
12	DSPACTJOB	(*NO) or (*YES)		
90	SIGNOFF	(*NOLIST) or (*LIST)		

The following command function keys can be used from the system operator menu.

Command Keys Supported on the System Operator Menu

Key	Function	Description
CF1	Return	Returns to display from which QPPRMENU was requested. If automatically displayed based on user profile, it is reinvoked. All current values, library lists, and overrides are lost. **Note:** You should be aware that if the CF1 key is pressed while the system operator menu is being displayed, the system resets the delivery mode for message queues. This could result in important messages not being displayed for the system operator until the delivery mode is changed to *BREAK.
CF3	Command entry	Requests the command entry display.
CF4 (after selecting option 5)	Prompt key	Requests prompting assistance. Pressing CF4 after selecting option 5 on the system operator menu, but without making any other entry, results in a menu display of command categories. Pressing CF4 after selecting option 5, and entering at least a command name, results in a prompt display for the parameters of the selected command.
CF6	Display messages in QSYSOPR	Requests a display of all messages currently in the system operator message queue.
CF7	Display subsystems	Requests a display of all jobs in all subsystems currently active in the system.
CF8	Display system	Requests a display of general information about all subsystems that are currently active in the system or being terminated.
Help	Second-level display	Requests the second-level message display for the message on the line where the cursor is positioned.

When you press the Enter key, the system attempts to execute the command based on the information you entered. Any messages regarding the status of your request are shown at the bottom of the menu, such as



```
                         SYSTEM OPERATOR MENU
Select one of the following:
  1. DSPJOBQ (jobq)                  7. STRPRTWTR device,outq
  2. DSPOUTQ (outq)                  8. DSPWTR    (writer)
  3. SNDMSG  tomsgq,(type),msg       9. SBMDKTJOB dev,label,(loc)
  4. CALL    program                10. SBMDBJOB  file,(member)
  5. Execute command                11. DSPSBMJOB
  6. SBMJOB  (job),(jobd),(cmd)     12. DSPACTJOB (reset)        +
Option: 5_   Parms: ______________________________________________
Cmd or parm: STRSBS SBSD(QSPL)____________________________________
__________________________________________________________________
Log requests:  *YES   CF3-Command entry    CF4-Prompt (5 only)
CF6-DSPMSG QSYSOPR    CF7-DSPSBS           CF8-DSPSYS
Start of subsystem QSPL.QGPL in progress
```

There will not always be a message if the command is successfully executed; however, there *will* be a message if the command cannot be executed because of an error.

Some requests will result in multiple messages. If there are more messages than can be displayed on your screen at one time, a + appears to the right of the bottom message. You obtain the additional messages by pressing the Roll Up key. If you are not sure how to respond to a particular message, position the cursor on the line containing the message and press the Help key. You will receive a separate display that contains additional information about the message (see *Handling Error Messages* in Chapter 11).

Whether the command you requested was sucessfully executed or not, your entry in the *Option* field is cleared, but any entries in the *Parms* and *Cmd or parm* fields are not cleared. You can either correct your previously entered command or enter a different command. The procedure is the same:

1. Key the appropriate option number into the *Option* field.

2. Make any needed changes to your previous entries in the *Parms* and *Cmd or parm* fields. Remember that you do not have to change the information in an input field if the field does not apply to the option you selected.

3. Press the Enter key again.

The following summarizes the use of each option and CF key indicated on the system operator menu.

Option 1: Entering *1* in the *Option* field (in Figure 6-5) requests a display of all job queues, or a display of all jobs on a specific job queue if you also enter a job queue name in the first *Parms* field (see *Job Queue Displays* in Chapter 10).

Option 2: Entering *2* in the *Option* field requests a display of all output queues, or a display of all spooled output files from jobs on a specific output queue if you also enter an output queue name in the first *Parms* field (see *Output Queue Displays* in Chapter 10).

Option 3: Entering *3* in the *Option* field allows you to send a message to other work stations. In the first *Parms* field, you must specifiy the name of the work station where the message is to be sent. You can also specify a message type in the second *Parms* field. For example, you may want to enter *INQ to specify an inquiry message that requires a reply. You enter the actual message in the *Cmd or parm* field (see *Sending Messages* in Chapter 11).

Option 4: Entering *4* in the *Option* field calls a program by using the CALL command. You must enter the name of the program in the first *Parms* field. When you call a program using this option, you cannot pass parameters to the program (see *Calling Programs* in Chapter 12). If you need to pass parameters to the program, you must select option 5 and enter the complete CALL command, including the PARM parameter. Note that when you call a program, you will not be able to select another function on the menu until the program is completed.

Option 5: Entering 5 in the *Option* field allows you to enter any CL command that you are authorized to use. You key the command into the *Cmd or parm* field.

The prompting functions described under *Command Selection and Prompting* in this chapter are also available for use from the system operator menu. If you press the prompt key (CF4) after keying in a 5 (for option 5), you will receive a separate command grouping menu that you can use to select a command. If you press the prompt key after selecting option 5 and keying in a partial command, you will receive a prompt display that shows all the parameters and default values for the specific command. Any parameters you key in before pressing the prompt key will be shown on the prompt. After it is executed, the command will be returned to the *Cmd or parm* line of the menu. If the command is too long to fit in the field, the last three characters of the field will be replaced with three periods.

Option 6: Entering *6* in the *Option* field submits a job for batch processing by using the SBMJOB command. If you choose, you can enter a job name in the first *Parms* field and/or a job description name in the second *Parms* field. When submitting a job using this option, you may need to enter the command that defines the actual processing of the job. You enter this command in the *Cmd or parm* field. If multiple commands are needed to process the job, the commands can be compiled into a program that you call by entering the CALL command in the *Cmd or parm* field. For example, to submit a job that is to be processed by the program PAYPGM, you would select this option and then enter the CALL command in the form

```
CALL PAYPGM
```

If you want to use prompting for the command on Option 6, start with Option 5, then press the CF4 key to obtain the prompt, fill in the desired parameters, then press the CF1 key. The CF1 key will return you to the system operator menu and the prompted command will appear on the Cmd or parm line. Then use Option 6 to submit the job.

(See the discussion of job operations in Chapter 12, in particular the section on *Submitting Batch Jobs*.)

Option 7: Entering *7* in the *Option* field starts a printer writer that causes output to be produced on a printer from a specific output queue. You must enter the name of the device in the first *Parms* field and the name of the output queue in the second *Parms* field (see *Starting a Printer Writer* in Chapter 10).

Option 8: Entering *8* in the *Option* field displays a writer. This can be used to display the current status of all writers or to display detailed information about a particular writer. From this display you can hold, release, or cancel a writer, or display the output queue assigned to the writer (see *Writer Displays* in Chapter 10).

Option 9: Entering *9* in the *Option* field submits a batch job from a diskette file specified in the second *Parms* field. The location of the diskette file can be specified in the *Cmd or parm* field. The default is *M1 (see *Submitting Diskette Jobs* in Chapter 10).

Option 10: Entering *10* in the *Option* field submits a batch job from the data base file specified in the first *Parms* field. The member name can be specified in the second *Parms* field (see *Submitting Data Base Jobs* in Chapter 10).

Option 11: Entering *11* in the *Option* field displays the status of all jobs submitted at a work station, in a job, or under a user profile. All jobs submitted by the SBMDKTJOB, SBMDBJOB, SBMCRDJOB or the SBMJOB commands that are still in the system are displayed. Jobs that were submitted with the DSPSBMJOB(*NO) parameter on the Submit Jobs commands are not displayed.

Option 12: Entering *12* in the *Option* field displays performance and status information about all active jobs in the system. If you specify *YES in the first *Parms* field, the statistics are reset and a new measurement period begins. If you do not enter this parameter, the statistics displayed are for the current time period.

Option 90: Entering *90* in the *Option* field signs you off the system as the system operator. If you want to list the job log, enter *LIST in the first *Parms* field.

Log requests: This field allows you to log all requests made from the system operator menu. The default is *YES, which causes all requests to be logged to the job log. If you do not want to log requests from the system operator menu, enter *NO. Logged requests will be printed if *LIST is specified with the SIGNOFF option, or they can be viewed while at the console by using the CF3 key to obtain the command entry display and then using the CF7 key to display low-level messages.

CF3: There may be occasions where the command you want to enter is too long to fit in the input field provided on the system operator menu. In that event, you can press the CF3 key to obtain the command entry display (see *Command Entry Display* in this chapter). The command entry display also allows you to duplicate a previous command. Note that the execution of a command you have entered on the command entry display will return you to the command entry display, not the system operator menu. However, if you press the CF1 key while the command entry display is on your screen, you will return to the system operator menu.

The CF3 key can also be used to view commands and their messages entered previously from the system operator menu.

CF6: Pressing the CF6 key results in a separate display of the messages in the system operator message queue, known as QSYSOPR (see *Receiving and Displaying Messages* in Chapter 11).

CF7: Pressing the CF7 key results in a separate subsystem display that shows the job name, user name, and job number of all jobs in all subsystems that are currently active in the system (see *Subsystem Display* in Chapter 8).

CF8: Pressing the CF8 key results in a separate system display that provides general information about all subsystems currently active in the system or being terminated (see *System Display* in Chapter 8).

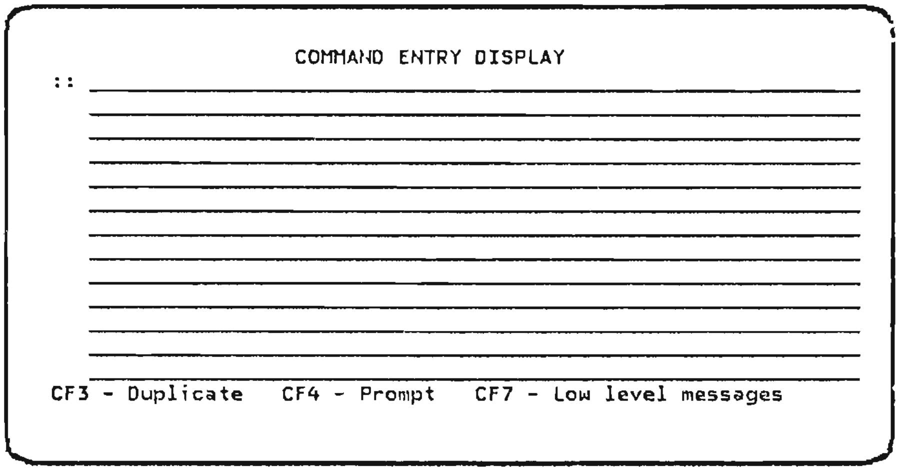
If you enter a request on the system operator menu that results in a different display, you can return to the system operator menu by pressing the CF1 key.

USING 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 using a *command entry display*. If you are not sure what commands are available or what values to use, however, you can obtain assistance from the system by requesting *menu* and *prompt* displays.

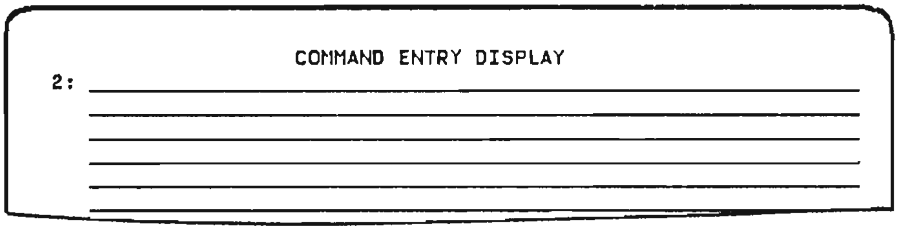
Command Entry Display

The command entry display (below) is the general medium for entering CL commands into the system. Unlike most other displays, it is a rolling display that provides, in effect, a continuous pad on which to enter commands. You enter one command at a time; the system processes the command and indicates its response before you enter the next command.



```
                    COMMAND ENTRY DISPLAY
 :: ________________________________________________

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

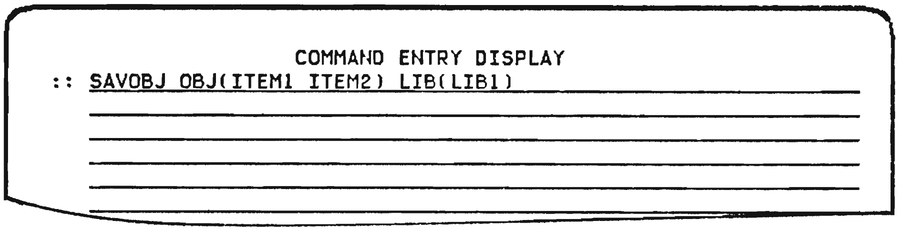
The command entry display can appear in a somewhat different form; the :: symbol is replaced by the symbol 2:



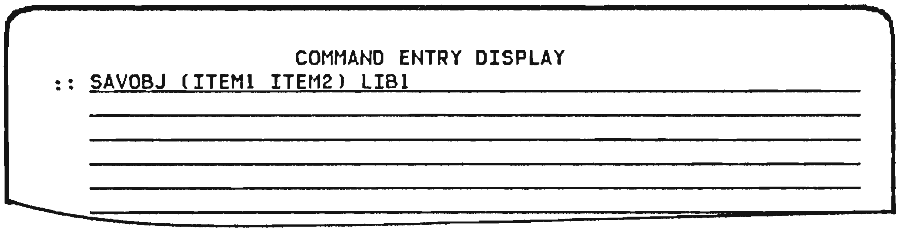
```
                    COMMAND ENTRY DISPLAY
 2: ________________________________________________
```

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

You key your input in the underlined area following the entry symbol :: or 2:. 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 parameter form:



```
                        COMMAND ENTRY DISPLAY
:: SAVOBJ OBJ(ITEM1 ITEM2) LIB(LIB1)
```



```
                        COMMAND ENTRY DISPLAY
:: SAVOBJ (ITEM1 ITEM2) LIB1
```

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 permitted.) If you choose, you can break your text at a convenient point where blanks are allowed (such as between two parameters) before reaching the end of a line. In that case, press the New Line key to move the cursor to the next line.

When you press the Enter 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. The entry symbol :: or 2: is displayed again on the line below the last response message to show where you make your next entry, and the underline below your entered command disappears to indicate that the line containing the command is no longer an input field.

If you enter a command that is in error, the system will respond with one or more messages that indicate the type of error. To successfully execute a command that was entered with errors, you must reenter it correctly in the input area provided below the error messages. You can also ignore the incorrect command and enter a different command in the new input area.

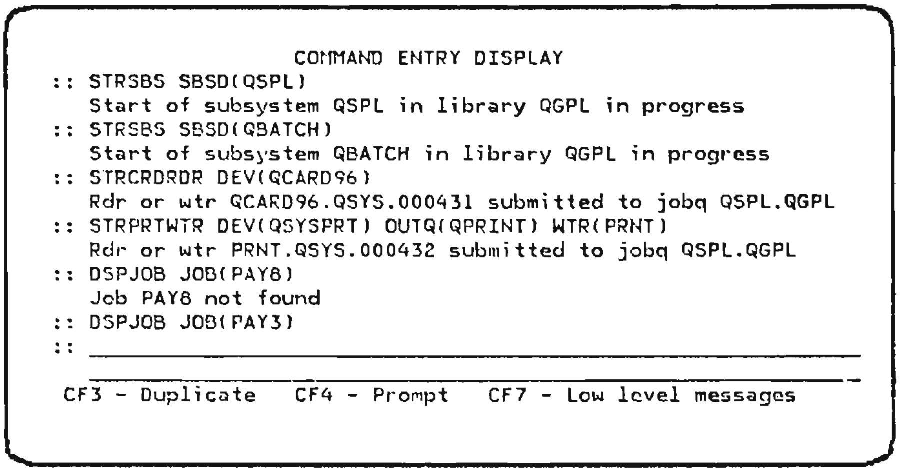
If you are not sure what to do about an error message, position the cursor on the same line as the message and press the Help key. You will receive a separate message display that contains a description of how to correct the error (see *Handling Error Messages* in Chapter 11). To return to the command entry display after examining the message display, you press the Enter key.

You can reenter a command on the command entry display in either of two ways:

- Key the entire command in the correct form into the new input area provided, and press the Enter key again.
- Use the duplication function provided through the CF3 key; that is
  - Press the CF3 key. The command you last entered will be duplicated into the new input area.
  - Key in the correct characters on top of the incorrect parts of the command.
  - Press the Enter key to execute the command.

If you are not sure how to correct a command, you can request prompting assistance for the command. You do this by placing the known part of the command in the current input area and pressing the CF4 key. You can also, of course, request prompting assistance in entering a new command (see the following discussion of *Command Selection and Prompting*).

As you continue to enter commands and receive responses, the entry symbol :: or 2: continues to appear on the next available line below, so that the screen is gradually filled with your entries and system responses (below).



```
                          COMMAND ENTRY DISPLAY
 :: STRSBS SBSD(QSPL)
    Start of subsystem QSPL in library QGPL in progress
 :: STRSBS SBSD(QBATCH)
    Start of subsystem QBATCH in library QGPL in progress
 :: STRCRDRDR DEV(QCARD96)
    Rdr or wtr QCARD96.QSYS.000431 submitted to jobq QSPL.QGPL
 :: STRPRTWTR DEV(QSYSPRT) OUTQ(QPRINT) WTR(PRNT)
    Rdr or wtr PRNT.QSYS.000432 submitted to jobq QSPL.QGPL
 :: DSPJOB JOB(PAY8)
    Job PAY8 not found
 :: DSPJOB JOB(PAY3)
 :: ________________________________________________________________
   _________________________________________________________________
  CF3 - Duplicate    CF4 - Prompt    CF7 - Low level messages
```

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 are appearing at the bottom.

As needed, you can also move the display up or down using the Roll keys. Lines that have been moved off the screen, either automatically or by the Roll Up key, are stored in the job log for your interactive job, unless you have changed your interactive job to filter out some or all of the information on those lines (see discussion of LOG parameter under *Changing a Job* in Chapter 12). By pressing the Roll Down key, you can review any information that has not been filtered out. Regardless of how far you have rolled the display, you can return immediately to the input entry area by pressing the Enter key.

Any command you have entered that you can still view on the screen or roll onto the screen using the Roll keys can be duplicated into the input entry area. You use the CF3 key as you would to correct a command:

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

2. Press the CF3 key.

The command then appears in the current input field following the :: entry symbol. You can press the Enter key to execute that same command again, or you can first modify it by keying in new characters over some of those shown and then press the Enter 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.

If a command you enter results in a different display or displays, you can return to the command entry display by pressing the Enter key or the CF1 key.

Command Selection and Prompting

When you are using the system operator menu or the command entry display, you can request system assistance 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 guides you in entering suitable parameters that lead to the execution of the command.

To obtain assistance, you make a prompt request by pressing the CF4 key. This key is called the *prompt key*.

The relationship of the command entry display and the system operator menu to menus and prompts obtained through a prompt request is summarized in Figure 6-6. Note that:

- If you press the prompt key without entering a command name, you receive a menu of command categories.

- If you key in a valid command name and then press the prompt key, you receive a prompt for that command.

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

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

This relationship exists for the system operator menu only if you select option 5 to enter a command.

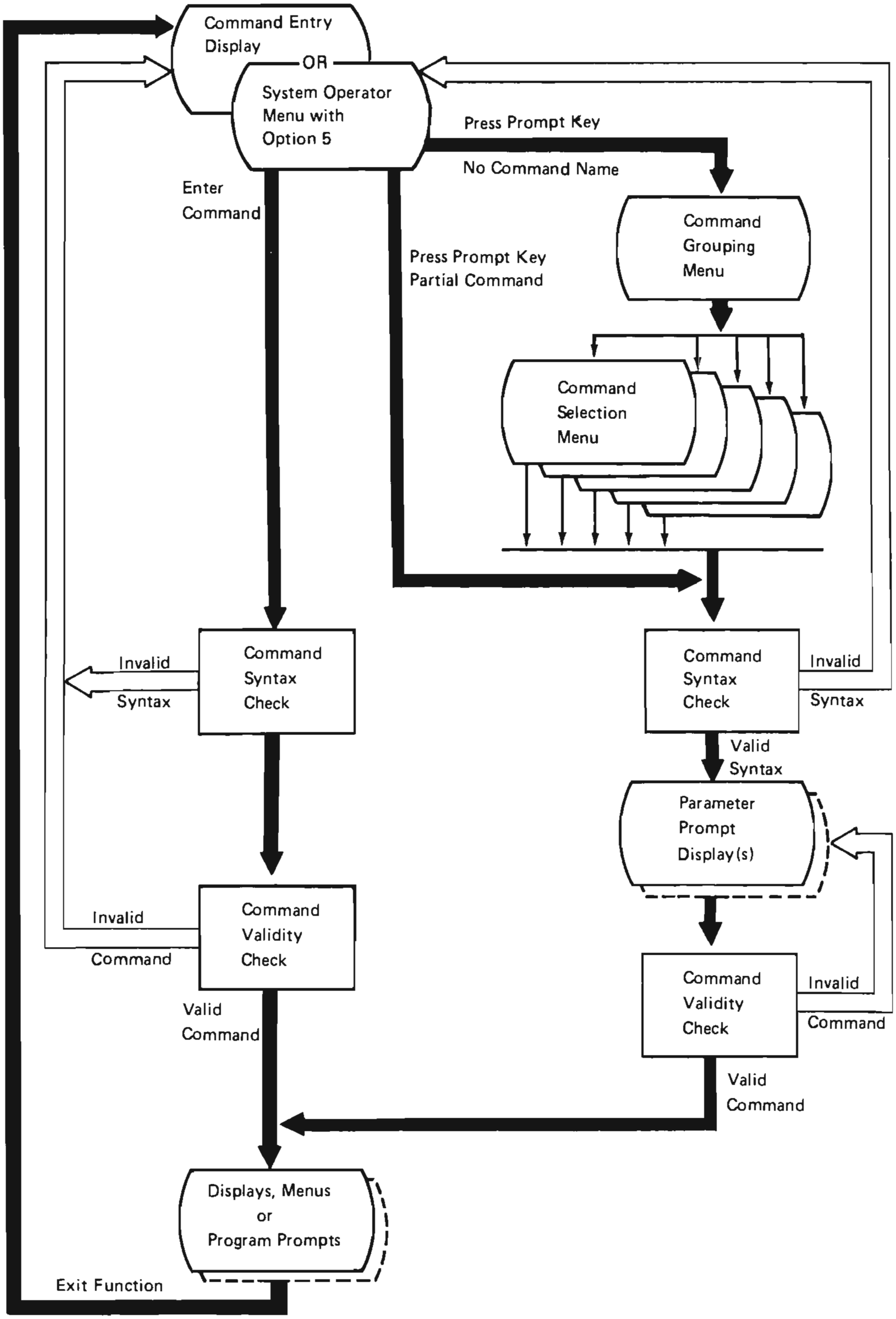


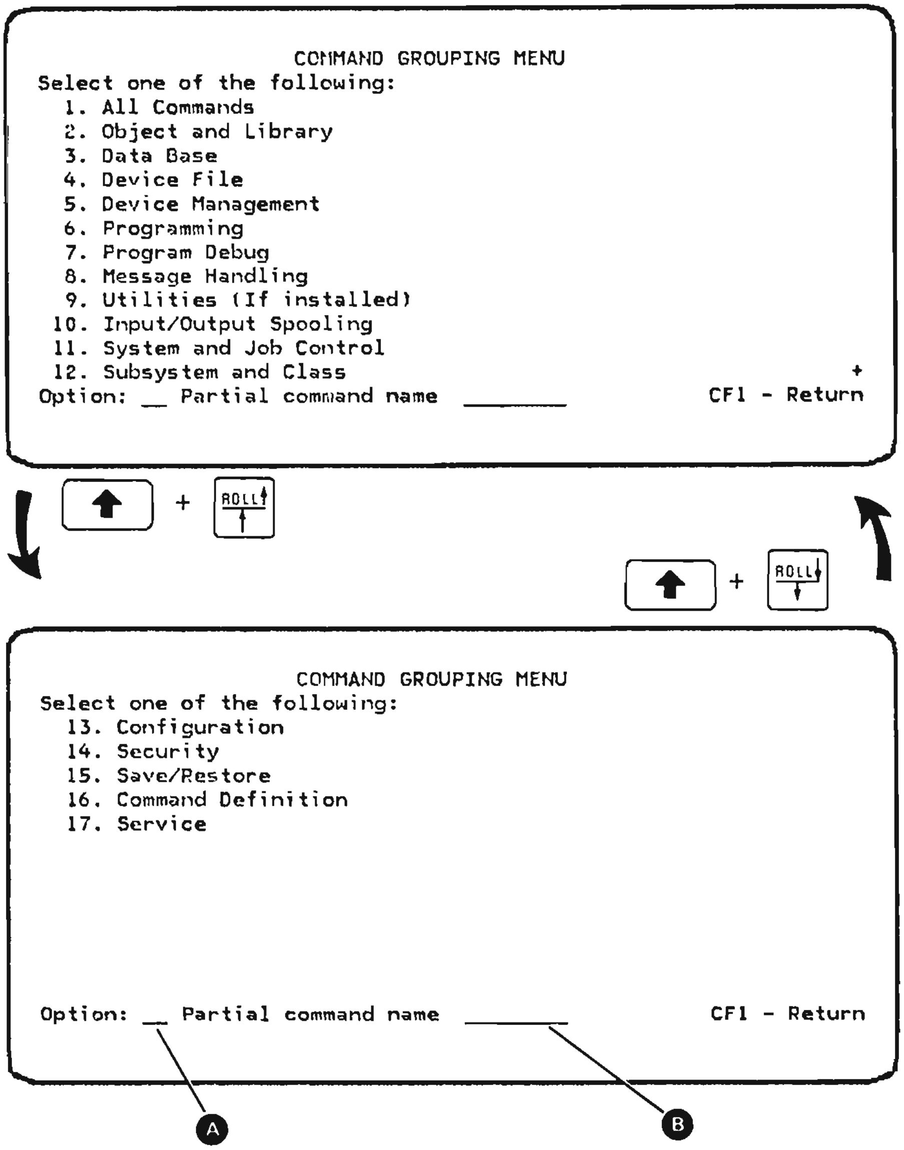
Figure 6-6. Relationship of Displays

Command Selection

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

If you press the prompt key without making an entry on the command entry display, or after keying in the option number 5 on the system operator menu, the system displays a menu (below) that lists functional categories of commands.

On the system console screen, the command grouping menu is displayed in two parts. You initially receive the part shown at the top. To obtain the second part, press the Roll Up key. You can return to the first part by pressing the Roll Down key. On a 5251 or 5252 work station, the entire menu is displayed on the screen at one time.



```
                              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                                                +
Option: __ Partial command name ________                     CF1 - Return
```

```
                              COMMAND GROUPING MENU
Select one of the following:
  13. Configuration
  14. Security
  15. Save/Restore
  16. Command Definition
  17. Service









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

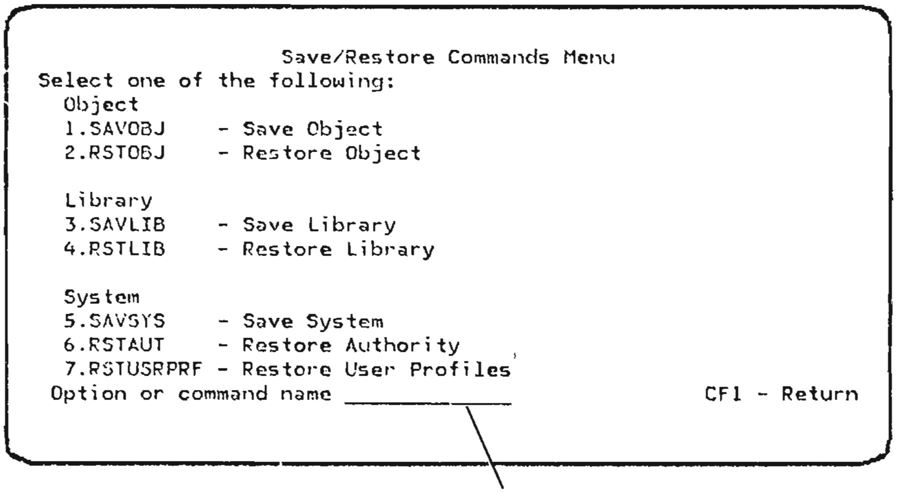
To make a selection from this menu:

1. Key the number of one of the sixteen options into the *Option* field Ⓐ.

2. If you want to restrict the commands to only a certain type within the group you selected in step 1, key a partial command name (such as DSP for display or SAV for save) into the *Partial command name* field Ⓑ (this step is optional).

3. Press the Enter key.

The system will respond with a command selection menu providing a list of commands that depends on what you entered on the command grouping menu. If you selected one of the sixteen command categories, but made no entry in the *Partial command name* field, the resulting command selection menu will display all commands in that category, such as shown below. If you also entered characters in the *Partial command name* field, the resulting menu will display only the commands of the selected group whose name begins with the characters.



```
                      Save/Restore Commands Menu
Select one of the following:
  Object
  1.SAVOBJ    - Save Object
  2.RSTOBJ    - Restore Object

  Library
  3.SAVLIB    - Save Library
  4.RSTLIB    - Restore Library

  System
  5.SAVSYS    - Save System
  6.RSTAUT    - Restore Authority
  7.RSTUSRPRF - Restore User Profiles
 Option or command name _____________               CF1 - Return
```

Enter name or option number
of selected command here.

Using the command selection menu, you can choose the command you need. To make a selection.

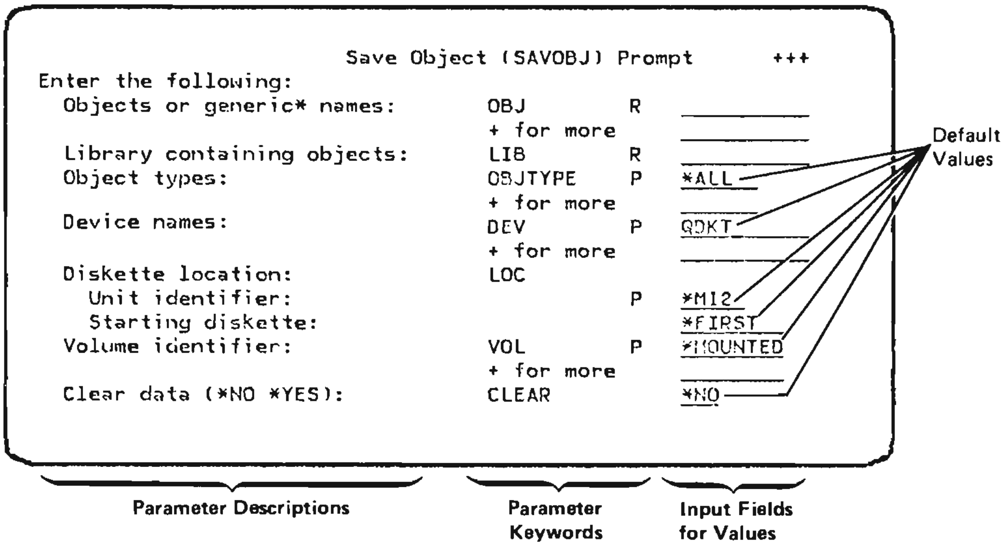
1 Key the selected command name or option number into the input field at the bottom of the menu.

2. Press the Enter key.

The system will then display a prompt for the parameters of the command you selected.

Command Parameter Prompting

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



```
                        Save Object (SAVOBJ) Prompt        +++
Enter the following:
  Objects or generic* names:          OBJ         R    __________
                                      + for more       __________
  Library containing objects:         LIB         R    __________
  Object types:                       OBJTYPE     P    *ALL______
                                      + for more       __________
  Device names:                       DEV         P    QDKT______
                                      + for more       __________
  Diskette location:                  LOC
    Unit identifier:                              P    *M12
    Starting diskette:                                 *FIRST
  Volume identifier:                  VOL         P    *MOUNTED
                                      + for more       ________
  Clear data (*NO *YES):              CLEAR            *NO
```

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 key (CF4).

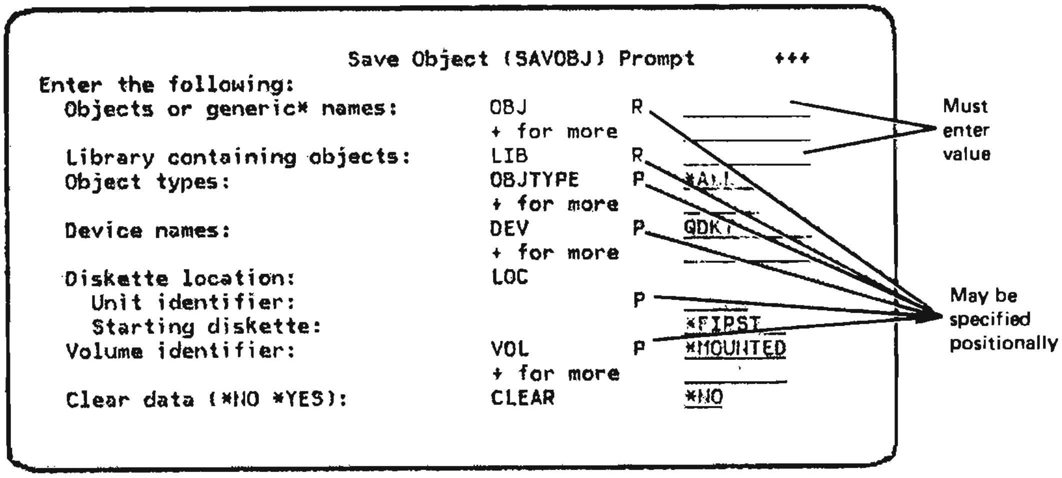
You can also use this procedure to request a prompt from the system operator menu if you select option 5. When you fill in the prompt to meet your needs and press the Enter key, the command is executed as if you entered it directly on the system operator menu or the command entry display.

In using a prompt, follow these guidelines:

Format of Input Fields: There is a separate input field for each value of each parameter. Each input field is displayed on a separate line; an underline shows its location. You can move to the field below by pressing the Field Advance key or the New Line key, or move to the field above by pressing the Field Backspace key.

Determining Required Parameters: A letter R beside a parameter keyword means that the parameter is required and you must enter at least one value for the parameter.

Determining Positional Parameters: A letter P or R beside a parameter keyword means that the parameter may be specified in positional form.



```
                             Save Object (SAVOBJ) Prompt       +++
Enter the following:
  Objects or generic* names:        OBJ         R   __________
                                    + for more      __________
  Library containing objects:       LIB         R   __________
  Object types:                     OBJTYPE     P   *ALL______
                                    + for more      __________
  Device names:                     DEV         P   QDKT______
                                    + for more      __________
  Diskette location:                LOC
    Unit identifier:                            P   ________
    Starting diskette:                              *FIRST__
  Volume identifier:                VOL         P   *MOUNTED
                                    + for more      ________
  Clear data (*NO *YES):            CLEAR           *NO
```

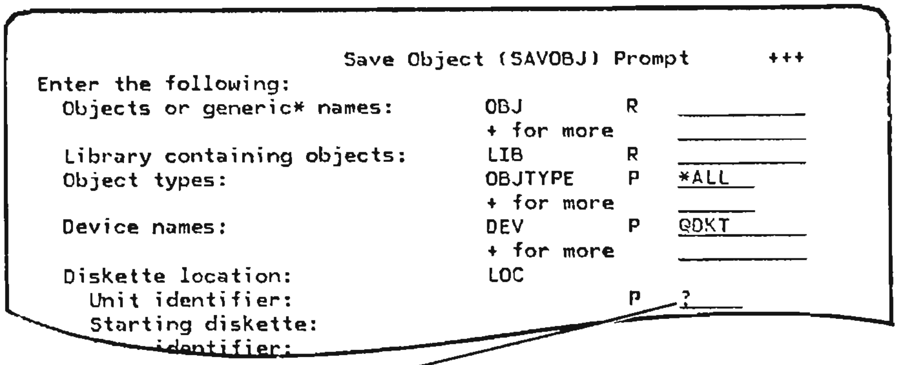
Handling Default Values: If a default value is shown in an input field, you can accept it as it is or change it by keying in another value on top of the one shown. The default value will be used for the parameter unless you change it.

Restoring a Default Value: You can return to the default value originally shown in an input field by blanking out the value that is currently in the field. The default value will be restored automatically when you press the Enter key. On the system console, you blank out a field by using the Spacebar. On a 5251/5252 work station, you can use either the Spacebar or the Field Exit key. You would use this function, for example, when you have replaced the default value in an input field and want to go back to it, or when you have placed values in two fields that are mutually exclusive and want to remove one of the values (in this case the default value is no value).

Determining Permissible Values: If you are not sure what values you can enter in an input field, you can request a separate display that lists the permissible values for the input field. You request the permissible values display by entering a ? in the input field; that is

1. Key ? into the input field. If a value is shown in the field, key in the ? over the first character of the value and press the Spacebar to insert a blank in the character position following the ?; you do not have to blank out any remaining characters in the field.

2. Press the Enter key.

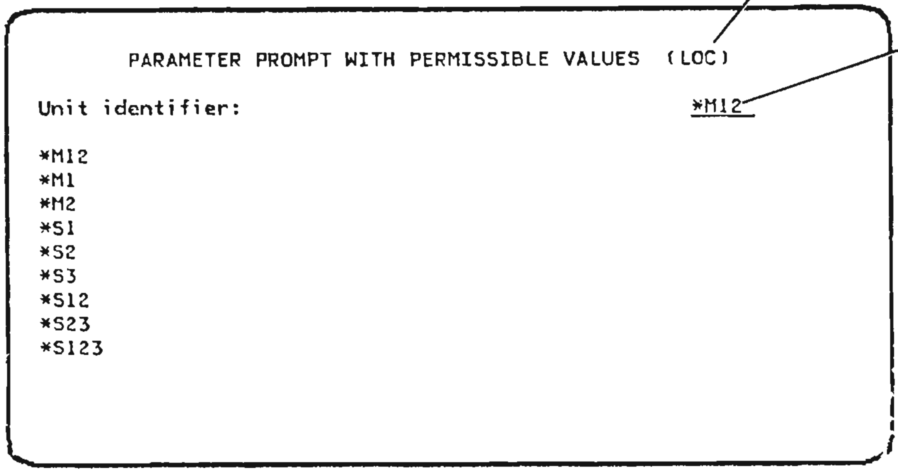


```
                        Save Object (SAVOBJ) Prompt        +++
Enter the following:
  Objects or generic* names:          OBJ       R   __________
                                      + for more    __________
  Library containing objects:         LIB       R   __________
  Object types:                       OBJTYPE   P   *ALL______
                                      + for more    __________
  Device names:                       DEV       P   QDKT______
                                      + for more    __________
  Diskette location:                  LOC
    Unit identifier:                            P   ?_____
    Starting diskette:
      identifier:
```

If you enter ? here

You receive this display:

Keyword



```
        PARAMETER PROMPT WITH PERMISSIBLE VALUES  (LOC)

Unit identifier:                                          *M12_

*M12
*M1
*M2
*S1
*S2
*S3
*S12
*S23
*S123
```

Input Field With Default

Valid Values

You return to the prompt by pressing the Enter key. If you key a value into the input field provided on the permissible values display before you press the Enter key, that value appears on the prompt when you return to it.

Handling Multiple Prompt Displays: If the symbol

+++

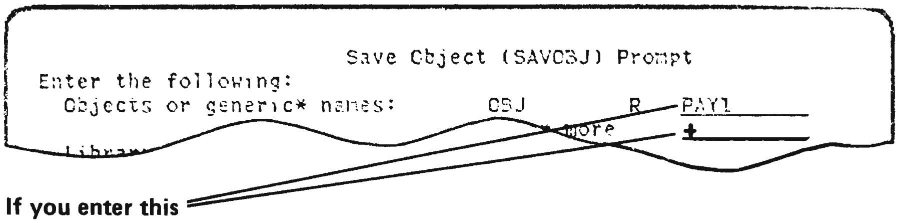
appears in the upper right corner of a prompt display, the command contains more parameters than can be shown on your screen at one time; one or more additional prompt displays follow the display currently shown on your screen. After completing the current display, you press the Enter key to advance to the next display. You can return to the previous display, if needed, by pressing the CF2 key. Note, however, that an entry error will prevent the display from changing when your press the Enter key (see *Effect of Errors* later in this chapter).

When the +++ symbol disappears, you have reached the final display of the prompt. If you press the Enter key while this final display is on your screen, the system will attempt to execute the command based on the information contained in all of the displays of the prompt. If you press the CF16 key, the system will attempt to execute the command when *any* part of the prompt is being displayed.

Adding More Values: If the phrase

+ *for more*

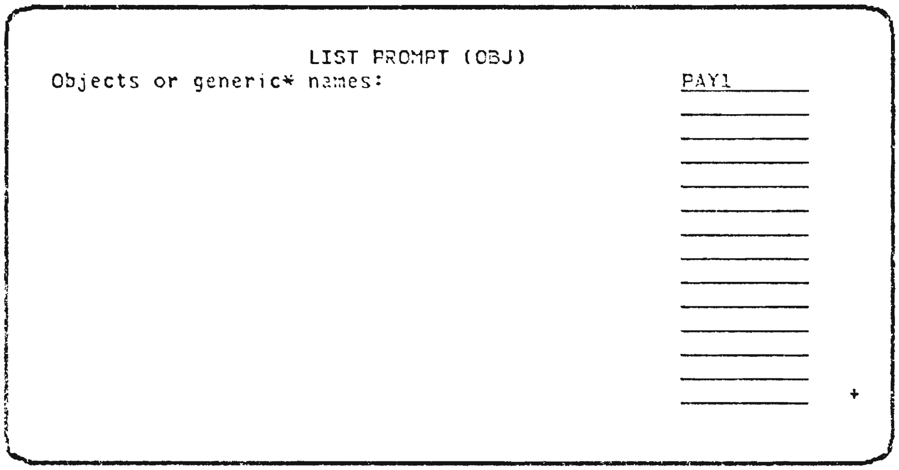
appears next to an input field, entering a + in that field results in a separate list prompt on which you can enter additional values for the parameter (a *list* of values). You can expand the list prompt (to the maximum number of values allowed for the parameter) by using the Roll Up key. When you complete the list prompt, pressing the Enter key returns you to the original prompt display, which is modified to include the additional values that you entered on the list prompt.



```
                         Save Object (SAVOBJ) Prompt
Enter the following:
  Objects or generic* names:        OBJ         R   PAY1______
                                             + more +_________
  Librar
```

If you enter this

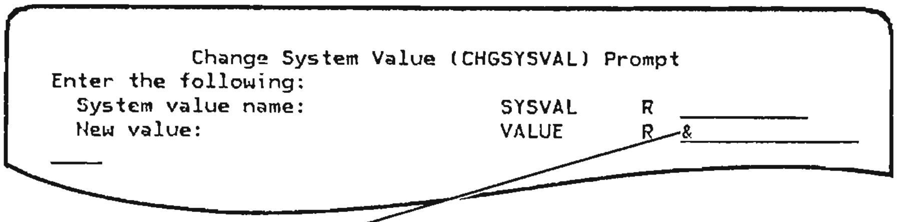
You receive this display:



```
                        LIST PROMPT (OBJ)
  Objects or generic* names:                        PAY1______
                                                    __________
                                                    __________
                                                    __________
                                                    __________
                                                    __________
                                                    __________
                                                    __________
                                                    __________
                                                    __________
                                                    __________
                                                    __________
                                                    __________
                                                    __________    +
```

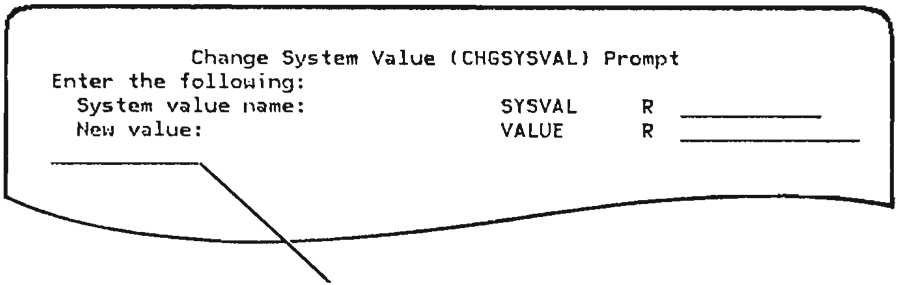
If you are entering a + in an input field that already contains a value, key in the + over the first character of the value and press the Spacebar to insert a blank in the character position following the +; then press the Enter key.

Expanding an Input Field: Entering an & symbol in the input field allowed for the value of a parameter results in an expanded input field that provides additional space for entering the value. The input field is expanded to the next larger length of 11, 17, 25, 32, 50, 80, 132, 256, or 512 characters. On the permissible values screen, an input field will be expanded no larger than 80 characters.



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

If you enter & here



```
              Change System Value (CHGSYSVAL) Prompt
Enter the following:
  System value name:                  SYSVAL     R ___________
  New value:                          VALUE      R ______________
_______________
```

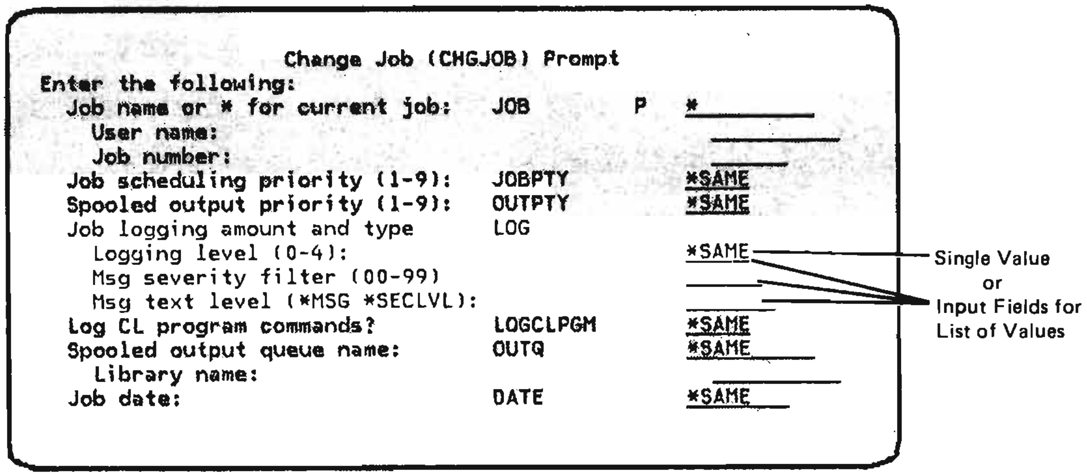
You receive an expanded input field

If you are entering an & in an input field that already contains a value, key in the & over the first character of the value and press the Spacebar to insert a blank in the character position following the &; then press the Enter key.

Input fields are usually the length of the maximum size value allowed; however, it may be necessary under some conditions to expand the field to insert apostrophes at the beginning and end of a quoted value, such as

'ABCDEF'

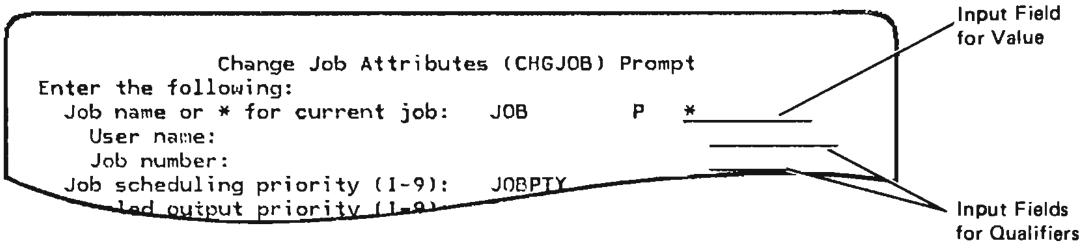
Handling Parameters That Accept a Mixed List of Values: Some commands allow you to enter either a single value or a *mixed list* of separately defined values for a parameter. For example, in the LOG parameter of the Change Job command, you can either use the single value *SAME (the default value) or enter a list of values that separately define the message logging level, the message severity, and the message text level.



```
                     Change Job (CHGJOB) Prompt
 Enter the following:
   Job name or * for current job:   JOB        P  *_________
     User name:                                     __________
     Job number:                                    ______
   Job scheduling priority (1-9):   JOBPTY        *SAME
   Spooled output priority (1-9):   OUTPTY        *SAME
   Job logging amount and type      LOG
     Logging level (0-4):                         *SAME__
     Msg severity filter (00-99)                  ______
     Msg text level (*MSG *SECLVL):               ________
   Log CL program commands?         LOGCLPGM      *SAME
   Spooled output queue name:       OUTQ          *SAME_____
     Library name:                                  __________
   Job date:                        DATE          *SAME___
```

If you enter a list of separately defined values, these values replace the single value. Similarly, if you enter a single value, it replaces the list of separately defined values. If you request the permissible values display, the single values are listed separately on the display.

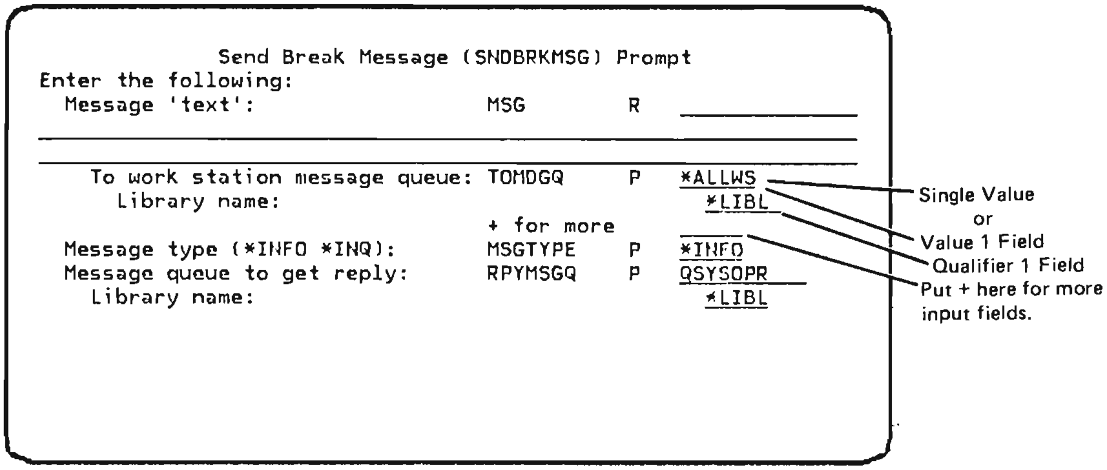
Handling Qualified Values: Various commands allow you to enter qualified object names such as OBJ1.LIB1 or qualified job names such as JOB1.USER1.000123. On the prompt for a command that allows a qualified object name, the input field for the library qualifier is indented below the input field for the object. Similarly, on a prompt for a command that allows a qualfied job name, there is a separate, indented input field for both the user name and the job number below the input field for the job name:



```
              Change Job Attributes (CHGJOB) Prompt
 Enter the following:
   Job name or * for current job:   JOB        P  *_________
     User name:                                     __________
     Job number:                                    ______
   Job scheduling priority (1-9):   JOBPTY
     led output priority (1-9)
```

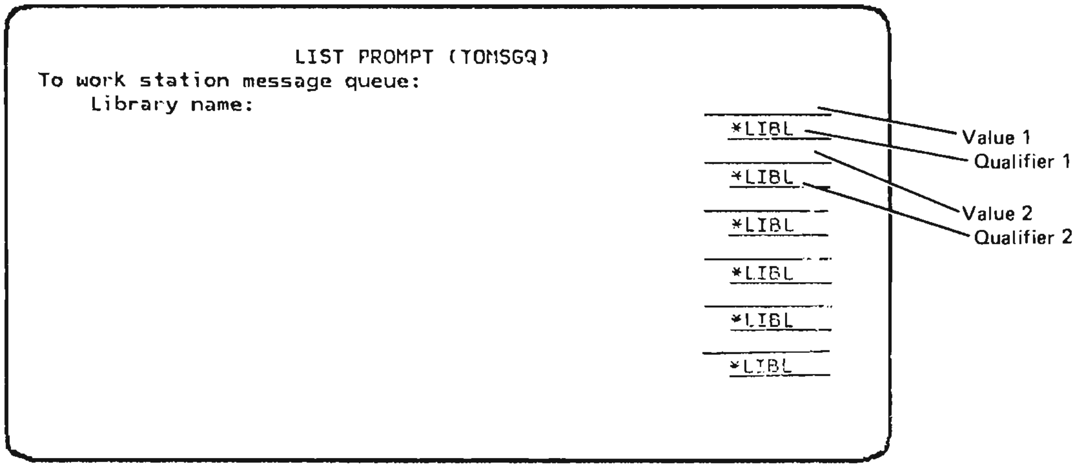
When using a prompt of this type, you should take care that you do not misinterpret the indented qualifier input field as an input field for another value.

In some commands that you will use, you can enter either a single value or multiple specific values, each of which has a library qualifier (a list of qualified values). For example, in the TOMSGQ parameter of the Send Break Message command, you can specify that the message is to be sent either to all work station message queues (*ALLWS) or to one or more specific work station message queues. Optionally, you can specify the name of the library in which each specific message queue is located (the system searches the libraries in the job's library list for the message queue if you do not specify a library name).



```
                Send Break Message (SNDBRKMSG) Prompt
Enter the following:
  Message 'text':                   MSG        R    ______________
__________________________________________________________________
__________________________________________________________________
    To work station message queue: TOMDGQ     P    *ALLWS
      Library name:                                  *LIBL
                                   + for more      ______
  Message type (*INFO *INQ):       MSGTYPE    P    *INFO
  Message queue to get reply:      RPYMSGQ    P    QSYSOPR
    Library name:                                  *LIBL
```

To enter more than one specific value, you enter a + in the special operator field provided below the first specific value and its qualifier. You then receive a list prompt that displays additional sets of input fields for entering a list of specific values and their qualifiers (compare with the previous discussion of *Adding More Values*).



```
                         LIST PROMPT (TOMSGQ)
To work station message queue:
    Library name:
                                                  __________
                                                    *LIBL
                                                  __________
                                                    *LIBL
                                                  __________
                                                    *LIBL
                                                  __________
                                                    *LIBL
                                                  __________
                                                    *LIBL
                                                  __________
                                                    *LIBL
```

If you enter one or more specific values in the input fields provided, those values are used instead of the single value.

Effect of Errors: The system will perform specific types of validity checking as you enter a command through a prompt. If an invalid command is detected, you will receive an error message at the bottom of the current prompt display, so that you can correct the error and successfully enter the command. For multiple prompt displays, as indicated by the +++ symbol at the top right of the screen, the values you enter on each display are checked separately. If an error is detected in the values you key in or if you fail to key in a value for a required parameter on one of the displays, you will not advance to the next display when you press the Enter key. Instead, the same display will be shown again with one or more messages that indicate the errors. By pressing the Enter key again, you can advance to the next display without correcting the errors, 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 remain on each display until you correct the errors and press the Enter key again.

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

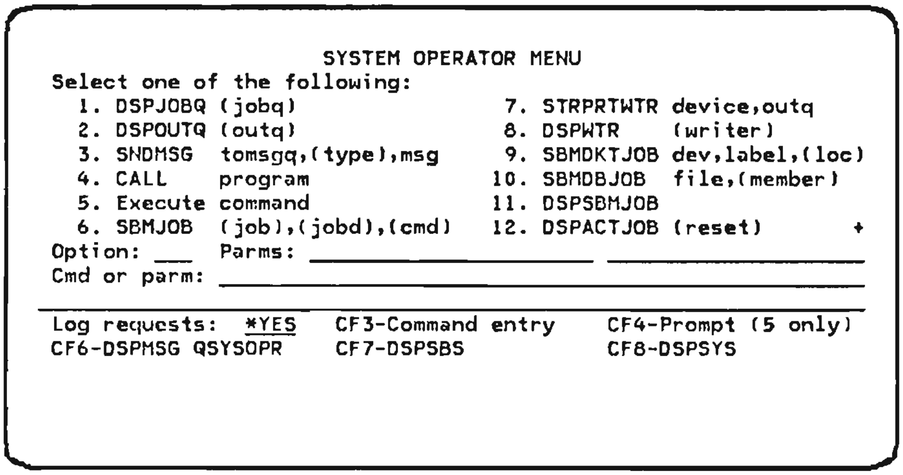
Canceling Prompting: At any time before you have successfully completed the prompting process and pressed the Enter key to execute the command, you can cancel prompting by pressing the CF1 key. You then return to the display from which you requested prompting (system operator menu or command entry display). If you entered a partial command, the command as modified by prompting will be returned to the *Msg or cmd* line on the system operator menu.

Other Prompting Functions: Various other prompting capabilities are available to you, in particular through use of the command function keys. Those capabilities are described in Appendix A, which summarizes command function key assignments for CPF.

For details on all prompting functions, refer to the *Work Station User's Guide*.

Example of Command Entry from the System Operator Menu

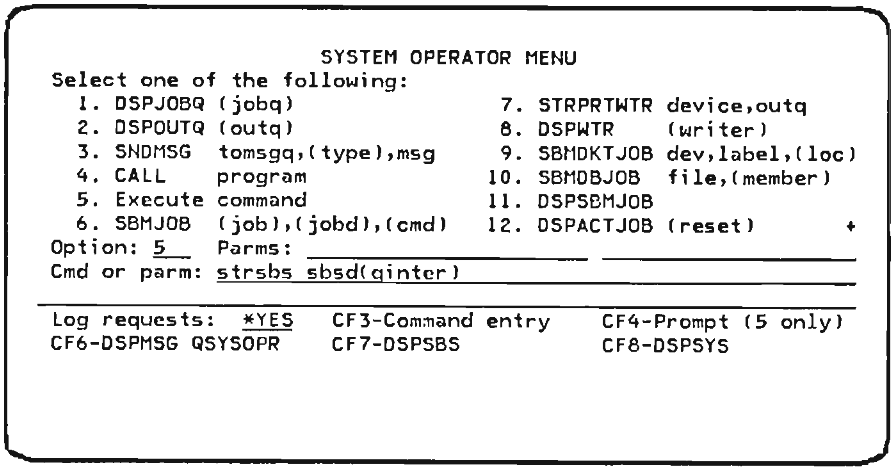
You have prepared the system for operation and have received the system operator menu on the system console screen.



```
                          SYSTEM OPERATOR MENU
Select one of the following:
  1. DSPJOBQ (jobq)                     7. STRPRTWTR device,outq
  2. DSPOUTQ (outq)                     8. DSPWTR    (writer)
  3. SNDMSG  tomsgq,(type),msg          9. SBMDKTJOB dev,label,(loc)
  4. CALL    program                   10. SBMDBJOB  file,(member)
  5. Execute command                   11. DSPSBMJOB
  6. SBMJOB  (job),(jobd),(cmd)        12. DSPACTJOB (reset)        +
Option: ___  Parms: ______________________  ____________________
Cmd or parm: _____________________________________________________
__________________________________________________________________
Log requests:  *YES   CF3-Command entry     CF4-Prompt (5 only)
CF6-DSPMSG QSYSOPR    CF7-DSPSBS            CF8-DSPSYS
```

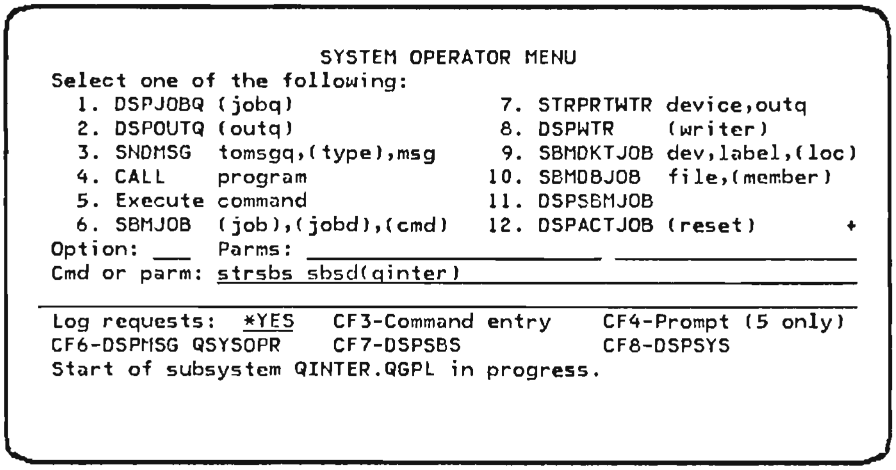
1. Now you want to start the interactive subsystem:
   a. Key a 5 in the *Option* field
   b. Key STRSBS SBSD(QINTER) in the command field as shown below.

   Note: All alphabetic entries can be either uppercase or lowercase.



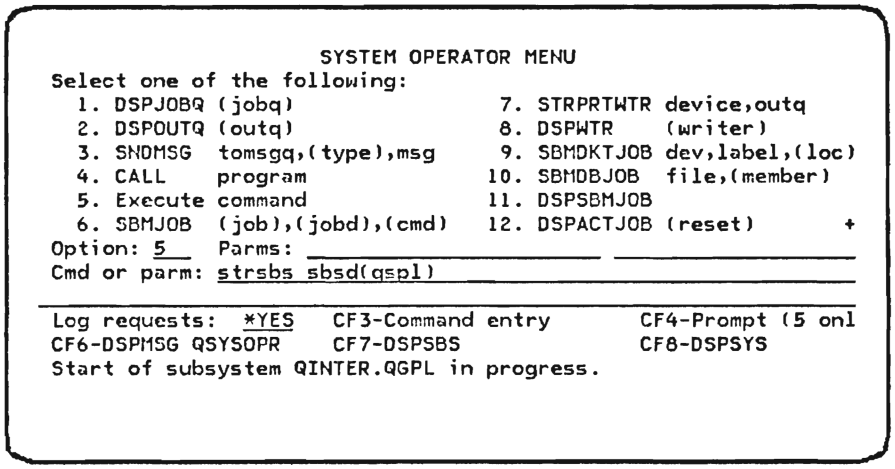
```
                        SYSTEM OPERATOR MENU
Select one of the following:
  1. DSPJOBQ (jobq)                  7. STRPRTWTR device,outq
  2. DSPOUTQ (outq)                  8. DSPWTR    (writer)
  3. SNDMSG  tomsgq,(type),msg       9. SBMDKTJOB dev,label,(loc)
  4. CALL    program                10. SBMDBJOB  file,(member)
  5. Execute command                11. DSPSBMJOB
  6. SBMJOB  (job),(jobd),(cmd)     12. DSPACTJOB (reset)        +
Option: 5_  Parms: _______________  _______________
Cmd or parm: strsbs sbsd(qinter)

Log requests:  *YES  CF3-Command entry    CF4-Prompt (5 only)
CF6-DSPMSG QSYSOPR   CF7-DSPSBS           CF8-DSPSYS
```

   c. Press the Enter key. The STRSBS command is executed and you receive a message that the interactive subsystem is being started:



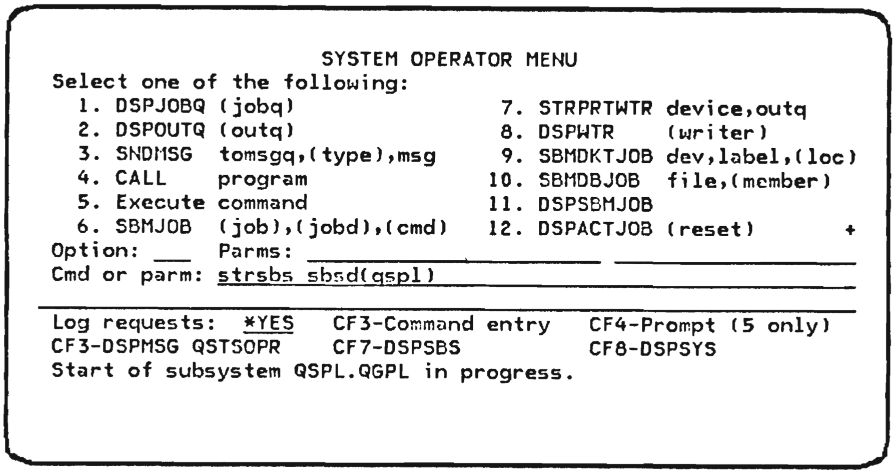
```
                        SYSTEM OPERATOR MENU
Select one of the following:
  1. DSPJOBQ (jobq)                  7. STRPRTWTR device,outq
  2. DSPOUTQ (outq)                  8. DSPWTR    (writer)
  3. SNDMSG  tomsgq,(type),msg       9. SBMDKTJOB dev,label,(loc)
  4. CALL    program                10. SBMDBJOB  file,(member)
  5. Execute command                11. DSPSBMJOB
  6. SBMJOB  (job),(jobd),(cmd)     12. DSPACTJOB (reset)        +
Option: ___ Parms: _______________  _______________
Cmd or parm: strsbs sbsd(qinter)

Log requests:  *YES  CF3-Command entry    CF4-Prompt (5 only)
CF6-DSPMSG QSYSOPR   CF7-DSPSBS           CF8-DSPSYS
Start of subsystem QINTER.QGPL in progress.
```

2. Next you want to start the spooling subsystem:
   a. Key a 5 in the *Option* field.
   b. Key STRSBS SBSD(QSPL) in the command field as shown below.



```
                         SYSTEM OPERATOR MENU
 Select one of the following:
   1. DSPJOBQ (jobq)                    7. STRPRTWTR device,outq
   2. DSPOUTQ (outq)                    8. DSPWTR    (writer)
   3. SNDMSG  tomsgq,(type),msg         9. SBMDKTJOB dev,label,(loc)
   4. CALL    program                  10. SBMDBJOB  file,(member)
   5. Execute command                  11. DSPSBMJOB
   6. SBMJOB  (job),(jobd),(cmd)       12. DSPACTJOB (reset)        +
 Option: 5__  Parms: ______________________ __________________
 Cmd or parm: strsbs sbsd(qspl)_______________________________
________________________________________________________________
 Log requests:  *YES   CF3-Command entry        CF4-Prompt (5 onl
 CF6-DSPMSG QSYSOPR    CF7-DSPSBS               CF8-DSPSYS
 Start of subsystem QINTER.QGPL in progress.
```

   c. Press the Enter key. The command is executed and you receive a message that the spooling subsystem is being started:

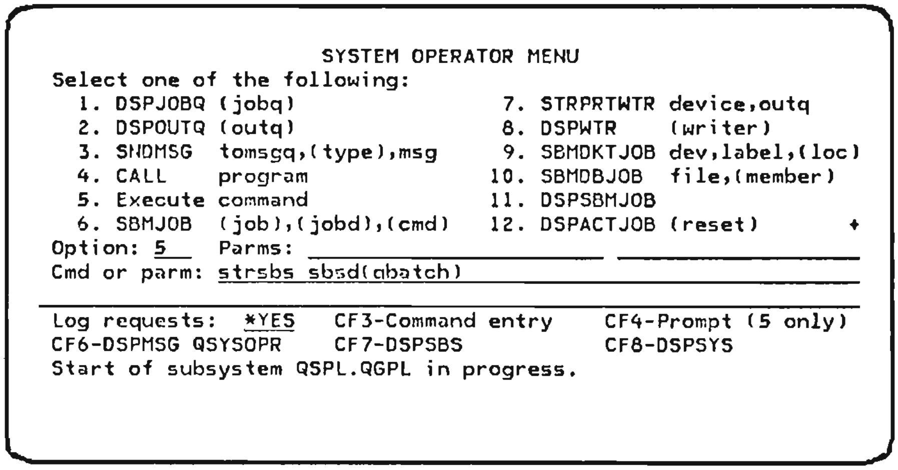


```
                         SYSTEM OPERATOR MENU
 Select one of the following:
   1. DSPJOBQ (jobq)                    7. STRPRTWTR device,outq
   2. DSPOUTQ (outq)                    8. DSPWTR    (writer)
   3. SNDMSG  tomsgq,(type),msg         9. SBMDKTJOB dev,label,(loc)
   4. CALL    program                  10. SBMDBJOB  file,(member)
   5. Execute command                  11. DSPSBMJOB
   6. SBMJOB  (job),(jobd),(cmd)       12. DSPACTJOB (reset)        +
 Option: ___  Parms: ______________________ __________________
 Cmd or parm: strsbs sbsd(qspl)_______________________________
________________________________________________________________
 Log requests:  *YES   CF3-Command entry   CF4-Prompt (5 only)
 CF3-DSPMSG QSTSOPR    CF7-DSPSBS          CF8-DSPSYS
 Start of subsystem QSPL.QGPL in progress.
```

3. Next you want to start the batch subsystem:

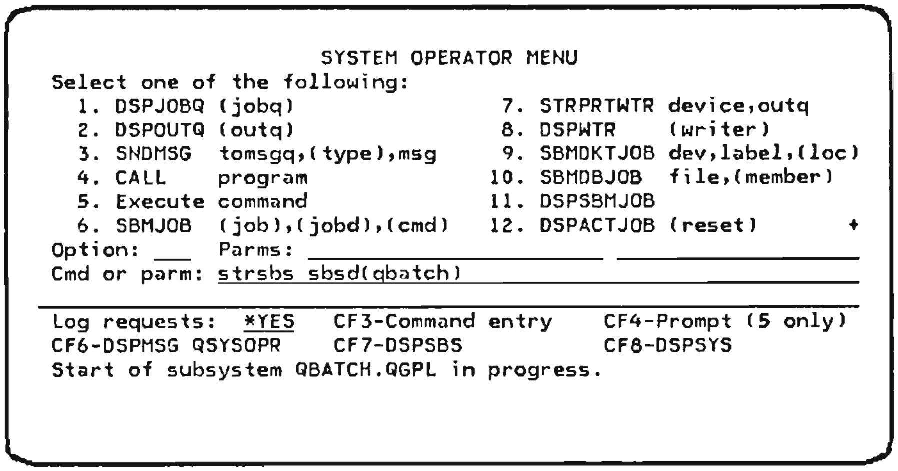
   a. Key a 5 in the *Option* field.

   b. Key STRSBS SBSD(QBATCH) in the command field as shown below.



```
                        SYSTEM OPERATOR MENU
Select one of the following:
  1. DSPJOBQ (jobq)                   7. STRPRTWTR device,outq
  2. DSPOUTQ (outq)                   8. DSPWTR    (writer)
  3. SNDMSG  tomsgq,(type),msg        9. SBMDKTJOB dev,label,(loc)
  4. CALL    program                 10. SBMDBJOB  file,(member)
  5. Execute command                 11. DSPSBMJOB
  6. SBMJOB  (job),(jobd),(cmd)      12. DSPACTJOB (reset)       +
Option: 5__  Parms: _______________________ ___________________
Cmd or parm: strsbs sbsd(qbatch)___________________________________
_________________________________________________________________
Log requests:  *YES   CF3-Command entry    CF4-Prompt (5 only)
CF6-DSPMSG QSYSOPR    CF7-DSPSBS           CF8-DSPSYS
Start of subsystem QSPL.QGPL in progress.
```

   c. Press the Enter key. The command is executed and you receive a message that the batch subsystem is being started.

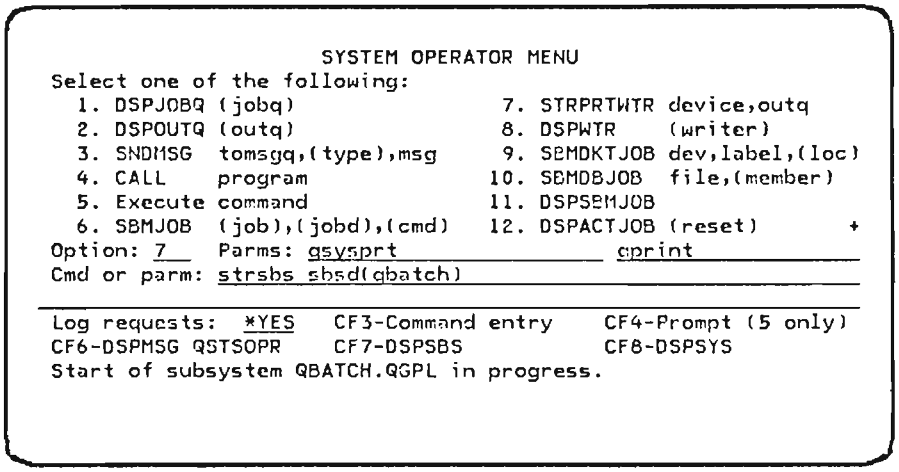


```
                        SYSTEM OPERATOR MENU
Select one of the following:
  1. DSPJOBQ (jobq)                   7. STRPRTWTR device,outq
  2. DSPOUTQ (outq)                   8. DSPWTR    (writer)
  3. SNDMSG  tomsgq,(type),msg        9. SBMDKTJOB dev,label,(loc)
  4. CALL    program                 10. SBMDBJOB  file,(member)
  5. Execute command                 11. DSPSBMJOB
  6. SBMJOB  (job),(jobd),(cmd)      12. DSPACTJOB (reset)       +
Option: ___  Parms: _______________________ ___________________
Cmd or parm: strsbs sbsd(qbatch)___________________________________
_________________________________________________________________
Log requests:  *YES   CF3-Command entry    CF4-Prompt (5 only)
CF6-DSPMSG QSYSOPR    CF7-DSPSBS           CF8-DSPSYS
Start of subsystem QBATCH.QGPL in progress.
```

4. You determine from the job setup instructions that there will be jobs printed. To print jobs on the system printer (QSYSPRT) from the output queue named QPRINT:

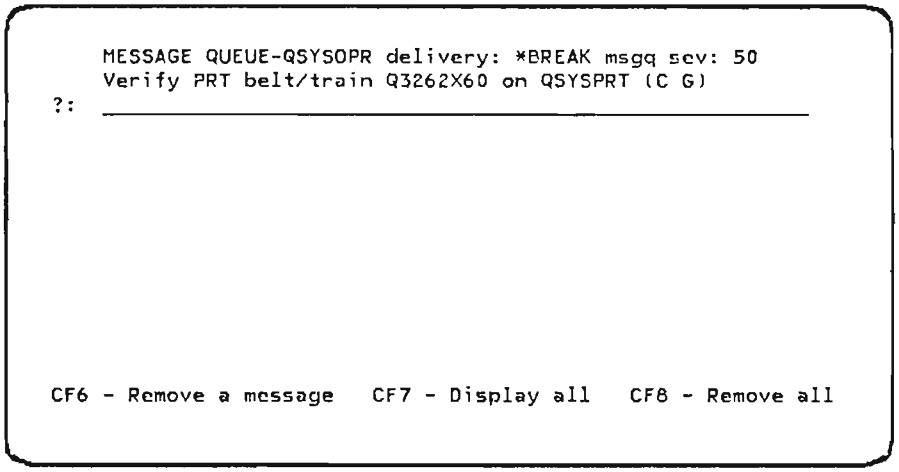
   a. Key a 7 in the *Option* field.

   b. Key in QSYSPRT and QPRINT in the Parms field as shown below.



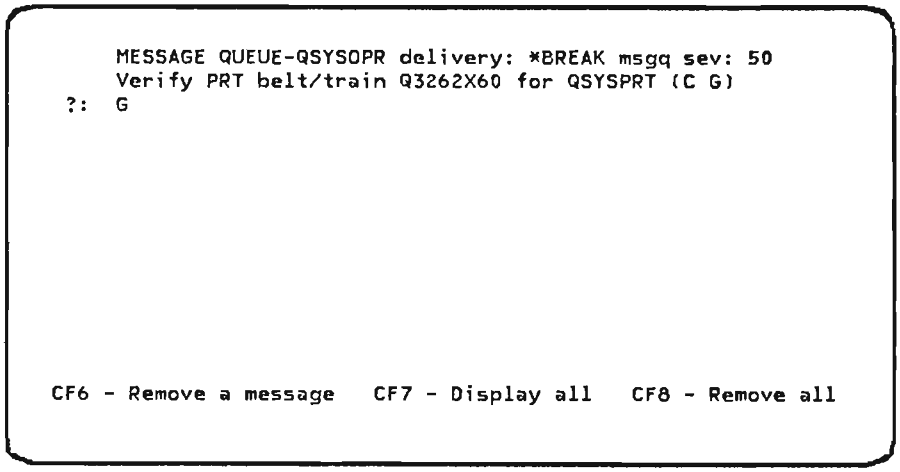
```
                    SYSTEM OPERATOR MENU
Select one of the following:
  1. DSPJOBQ (jobq)                 7. STRPRTWTR device,outq
  2. DSPOUTQ (outq)                 8. DSPWTR    (writer)
  3. SNDMSG  tomsgq,(type),msg      9. SBMDKTJOB dev,label,(loc)
  4. CALL    program               10. SBMDBJOB  file,(member)
  5. Execute command               11. DSPSBMJOB
  6. SBMJOB  (job),(jobd),(cmd)    12. DSPACTJOB (reset)       +
Option: 7    Parms: qsysprt                 qprint
Cmd or parm: strsbs sbsd(qbatch)

Log requests:  *YES   CF3-Command entry    CF4-Prompt (5 only)
CF6-DSPMSG QSTSOPR    CF7-DSPSBS           CF8-DSPSYS
Start of subsystem QBATCH.QGPL in progress.
```

   c. Press the Enter key. The command is executed and a message is sent to the system operator message queue that asks you to verify that the correct print belt/train is installed.



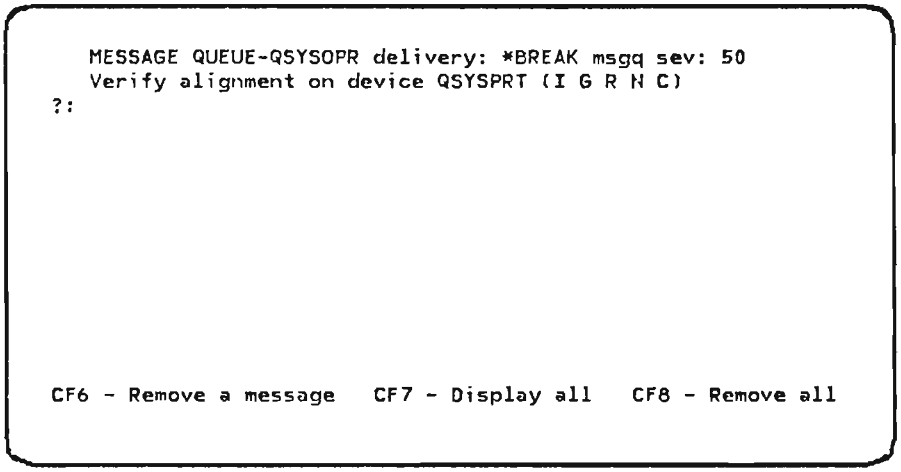
```
    MESSAGE QUEUE-QSYSOPR delivery: *BREAK msgq sev: 50
    Verify PRT belt/train Q3262X60 on QSYSPRT (C G)
?:



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

d. After you have verified that the correct print belt is installed, you key in G, which means go, in the message response field as shown below.



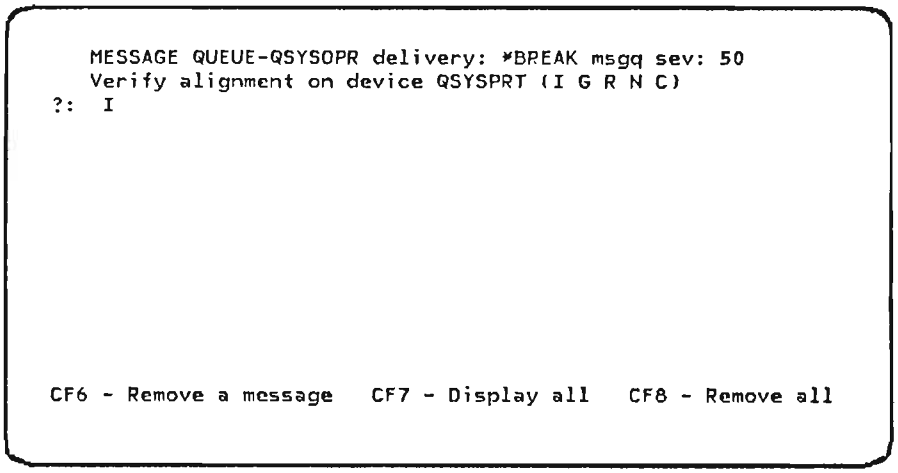
```
    MESSAGE QUEUE-QSYSOPR delivery: *BREAK msgq sev: 50
    Verify PRT belt/train Q3262X60 for QSYSPRT (C G)
?:  G









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

e. Press the Enter key. The system responds with a message that allows you to align the forms in the system printer.



```
   MESSAGE QUEUE-QSYSOPR delivery: *BREAK msgq sev: 50
   Verify alignment on device QSYSPRT (I G R N C)
?:









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

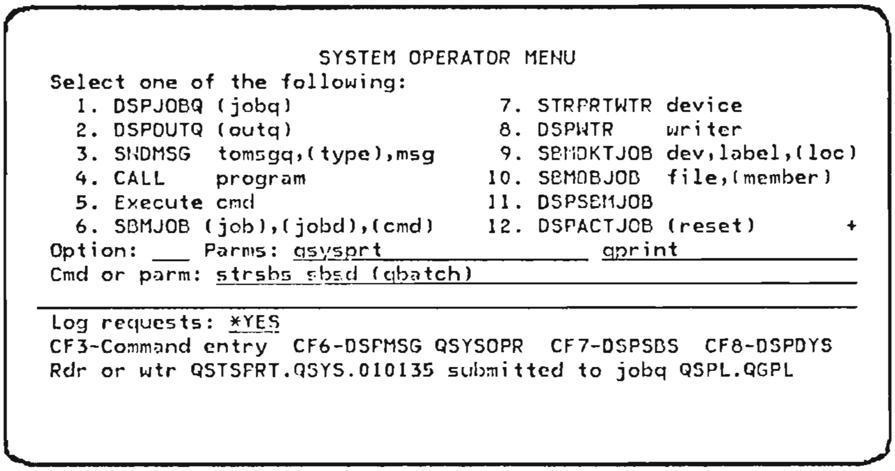
f. After you have verified that the printer forms are aligned properly, key in an I, which means ignore, in the message response field.



```
   MESSAGE QUEUE-QSYSOPR delivery: *BREAK msgq sev: 50
   Verify alignment on device QSYSPRT (I G R N C)
?:  I








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

g. Press the Enter key.

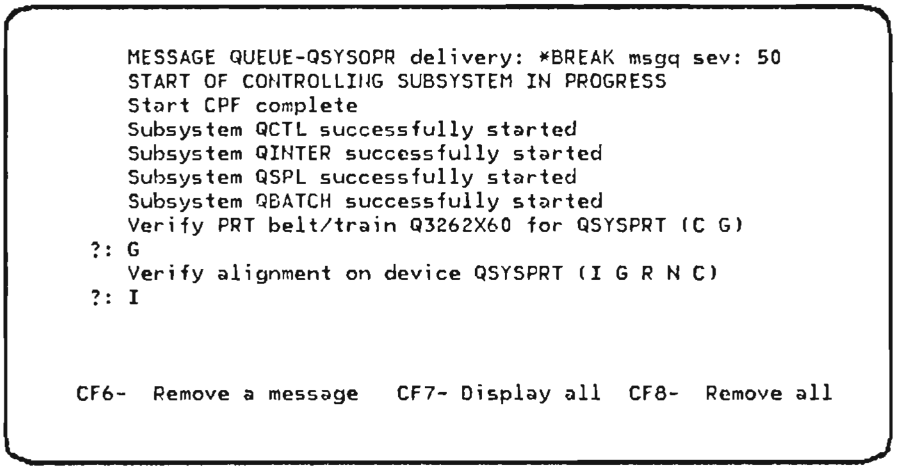
5. After the printer writer is started, the system operator menu is returned.



```
                      SYSTEM OPERATOR MENU
Select one of the following:
  1. DSPJOBQ (jobq)                   7. STRPRTWTR device
  2. DSPOUTQ (outq)                   8. DSPWTR    writer
  3. SNDMSG  tomsgq,(type),msg        9. SBMDKTJOB dev,label,(loc)
  4. CALL    program                 10. SBMDBJOB  file,(member)
  5. Execute cmd                     11. DSPSBMJOB
  6. SBMJOB (job),(jobd),(cmd)       12. DSPACTJOB (reset)       +
Option: ___ Parms: qsysprt________________ qprint __________
Cmd or parm: strsbs sbsd (qbatch)_____________________________
 _____________________________________________________________
Log requests: *YES
CF3-Command entry  CF6-DSPMSG QSYSOPR  CF7-DSPSBS  CF8-DSPDYS
Rdr or wtr QSTSPRT.QSYS.010135 submitted to jobq QSPL.QGPL
```

At this point, you may want to display the messages in the system operator message queue. To do this press the CF6 key.

The system operator message queue is displayed as shown below

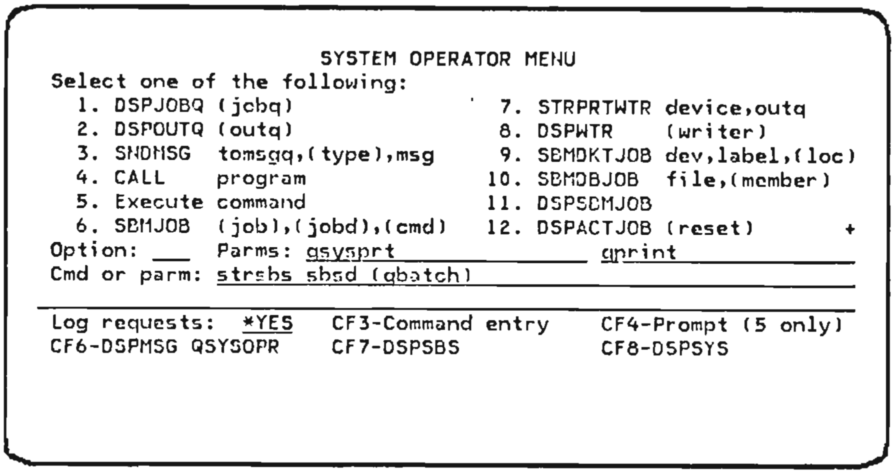


```
      MESSAGE QUEUE-QSYSOPR delivery: *BREAK msgq sev: 50
      START OF CONTROLLING SUBSYSTEM IN PROGRESS
      Start CPF complete
      Subsystem QCTL successfully started
      Subsystem QINTER successfully started
      Subsystem QSPL successfully started
      Subsystem QBATCH successfully started
      Verify PRT belt/train Q3262X60 for QSYSPRT (C G)
   ?: G
      Verify alignment on device QSYSPRT (I G R N C)
   ?: I



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

After you have viewed the messages in the system operator message queue, press the Enter key to return to the system operator menu.

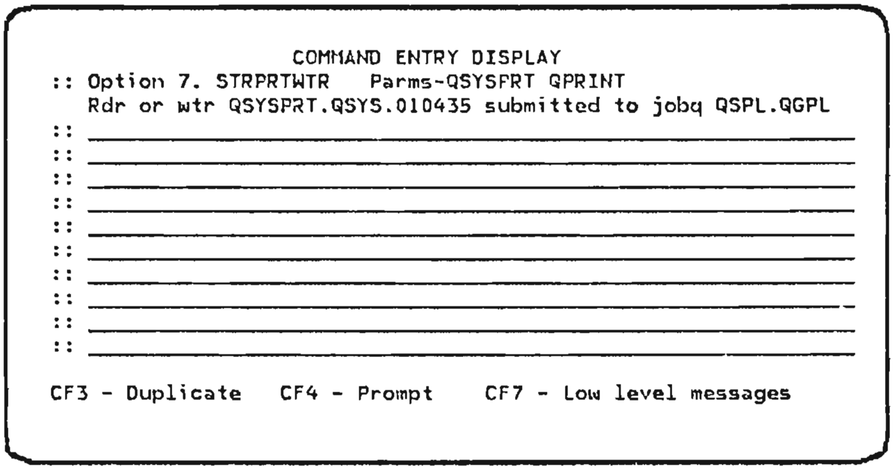
6. At this point you may want to see the commands that you have entered. To do this:

   a. Press the CF3 key.



```
                     SYSTEM OPERATOR MENU
Select one of the following:
  1. DSPJOBQ (jcbq)                 7. STRPRTWTR device,outq
  2. DSPOUTQ (outq)                 8. DSPWTR    (writer)
  3. SNDMSG  tomsgq,(type),msg      9. SBMDKTJOB dev,label,(loc)
  4. CALL    program               10. SBMDBJOB  file,(member)
  5. Execute command               11. DSPSBMJOB
  6. SBMJOB  (job),(jobd),(cmd)    12. DSPACTJOB (reset)        +
Option: ___ Parms: qsysprt_______________ qprint__________
Cmd or parm: strsbs sbsd (qbatch)_____________________________
_______________________________________________________________
Log requests:  *YES   CF3-Command entry    CF4-Prompt (5 only)
CF6-DSPMSG QSYSOPR    CF7-DSPSBS           CF8-DSPSYS
```

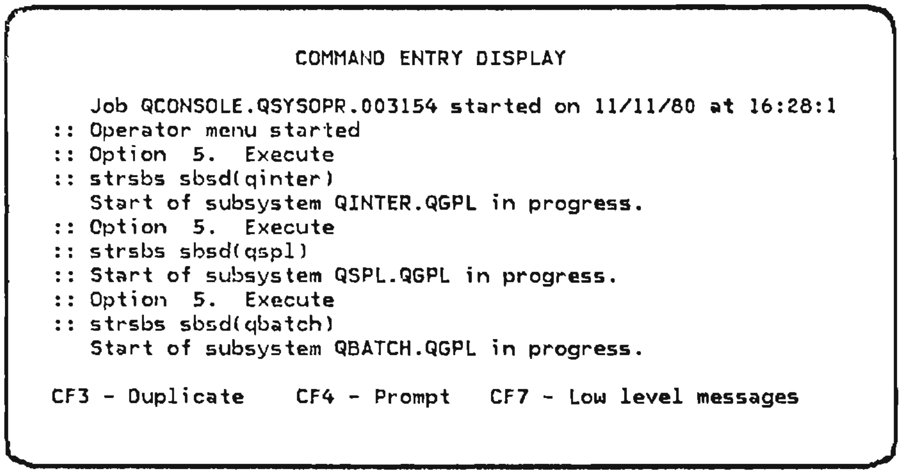
The last request from the system operator menu is shown along with any associated messages.



```
                    COMMAND ENTRY DISPLAY
:: Option 7. STRPRTWTR   Parms-QSYSPRT QPRINT
   Rdr or wtr QSYSPRT.QSYS.010435 submitted to jobq QSPL.QGPL
:: ____________________________________________________________
:: ____________________________________________________________
:: ____________________________________________________________
:: ____________________________________________________________
:: ____________________________________________________________
:: ____________________________________________________________
:: ____________________________________________________________
:: ____________________________________________________________
:: ____________________________________________________________
:: ____________________________________________________________

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

b. Press the Roll Down key.

The requests that you entered and associated messages will be displayed as shown below:



```
                    COMMAND ENTRY DISPLAY

   Job QCONSOLE.QSYSOPR.003154 started on 11/11/80 at 16:28:1
:: Operator menu started
:: Option  5.  Execute
:: strsbs sbsd(qinter)
   Start of subsystem QINTER.QGPL in progress.
:: Option  5.  Execute
:: strsbs sbsd(qspl)
:: Start of subsystem QSPL.QGPL in progress.
:: Option  5.  Execute
:: strsbs sbsd(qbatch)
   Start of subsystem QBATCH.QGPL in progress.

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

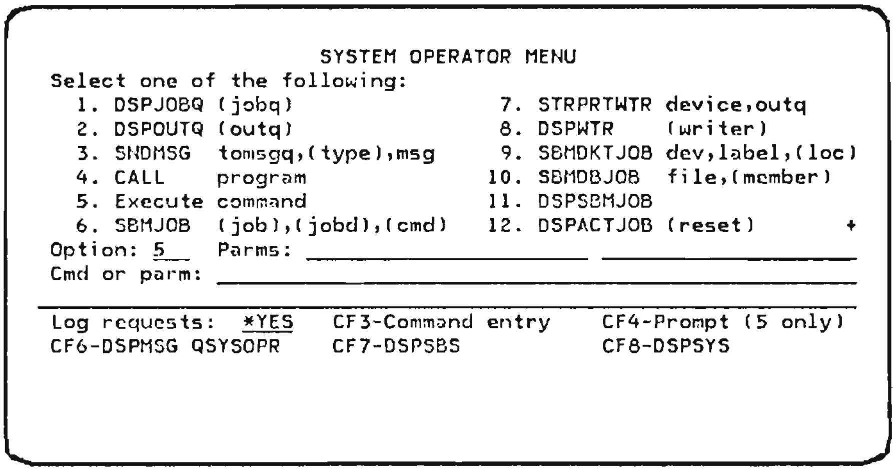
The CF3 key can be used to duplicate requests.

The CF7 key will show low-level messages associated with the current level of messages displayed.

Example of Finding and Executing a Command from System Operator Menu

1. You want to enter a command to save specific files (that is, transfer a copy of them to diskette), but you do not know which command to use. You need to find the correct command, so you request prompting as follows:

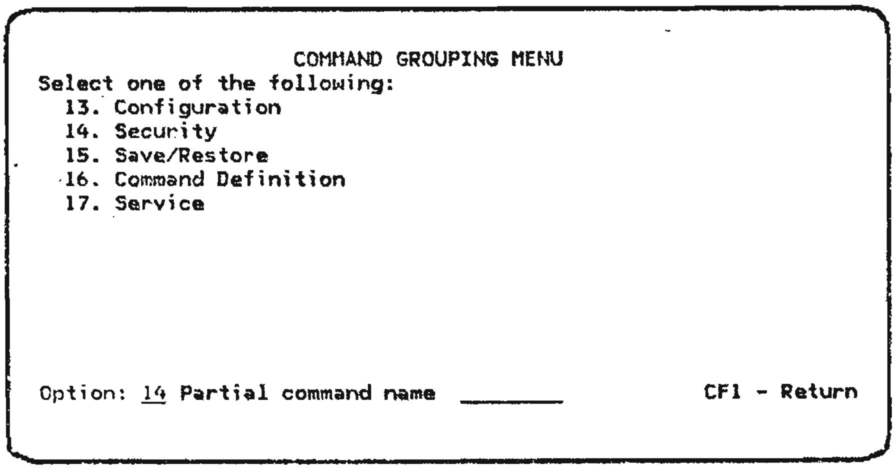
   a. Key a 5 into the *Option* field of the system operator menu to select option 5 (execute command).



```
                         SYSTEM OPERATOR MENU
Select one of the following:
  1. DSPJOBQ (jobq)                  7. STRPRTWTR device,outq
  2. DSPOUTQ (outq)                  8. DSPWTR    (writer)
  3. SNDMSG  tomsgq,(type),msg       9. SBMDKTJOB dev,label,(loc)
  4. CALL    program                10. SBMDBJOB  file,(member)
  5. Execute command                11. DSPSBMJOB
  6. SBMJOB  (job),(jobd),(cmd)     12. DSPACTJOB (reset)         +
Option: 5__  Parms: ______________________  __________________
Cmd or parm: _____________________________________________________
___________________________________________________________________
Log requests:  *YES   CF3-Command entry     CF4-Prompt (5 only)
CF6-DSPMSG QSYSOPR    CF7-DSPSBS            CF8-DSPSYS
```

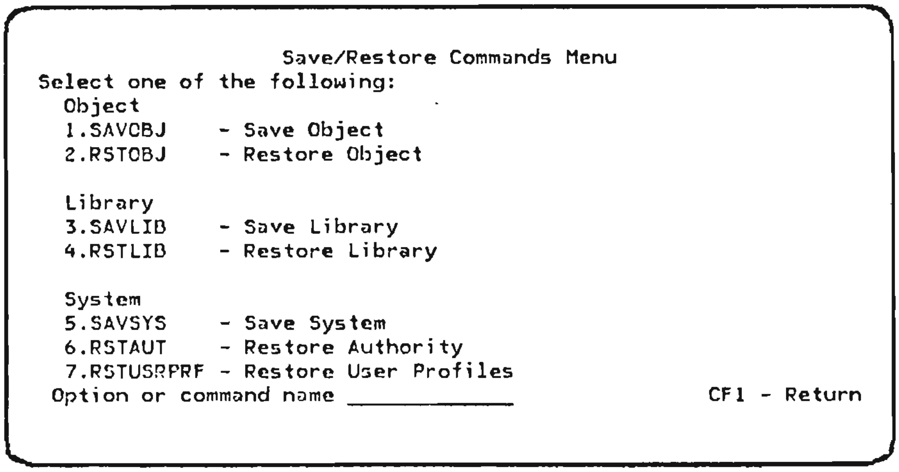
   b. Press the prompt key (CF4) without entering anything in any of the other input fields. The system displays the command grouping menu. The cursor is automatically positioned for input in the *Option* field.

   c. You press the Roll Up key to get the second part of the menu and select and key in option 15 (Save/Restore):



```
                         COMMAND GROUPING MENU
Select one of the following:
  13. Configuration
  14. Security
  15. Save/Restore
  16. Command Definition
  17. Service





Option: 14 Partial command name ________          CF1 - Return
```

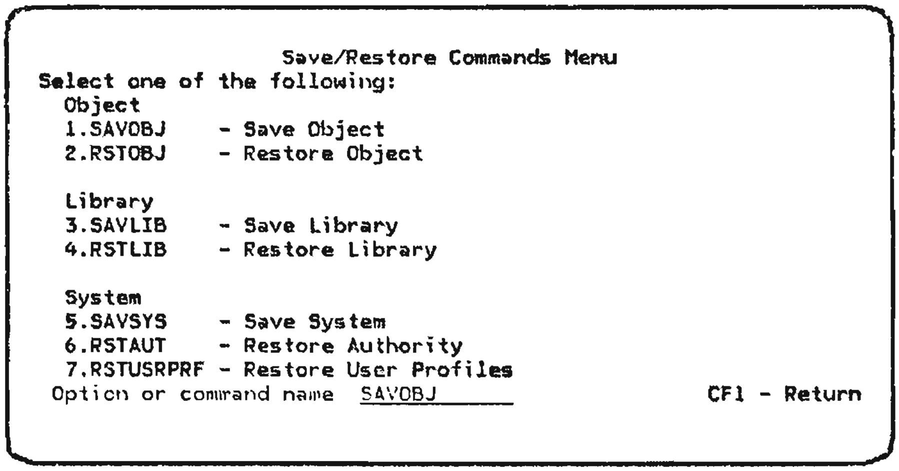
d. Press the Enter key. The system responds with a menu of all save/restore commands:



```
                         Save/Restore Commands Menu
Select one of the following:
  Object
  1.SAVOBJ    - Save Object
  2.RSTOBJ    - Restore Object

  Library
  3.SAVLIB    - Save Library
  4.RSTLIB    - Restore Library

  System
  5.SAVSYS    - Save System
  6.RSTAUT    - Restore Authority
  7.RSTUSRPRF - Restore User Profiles
 Option or command name ____________              CF1 - Return
```

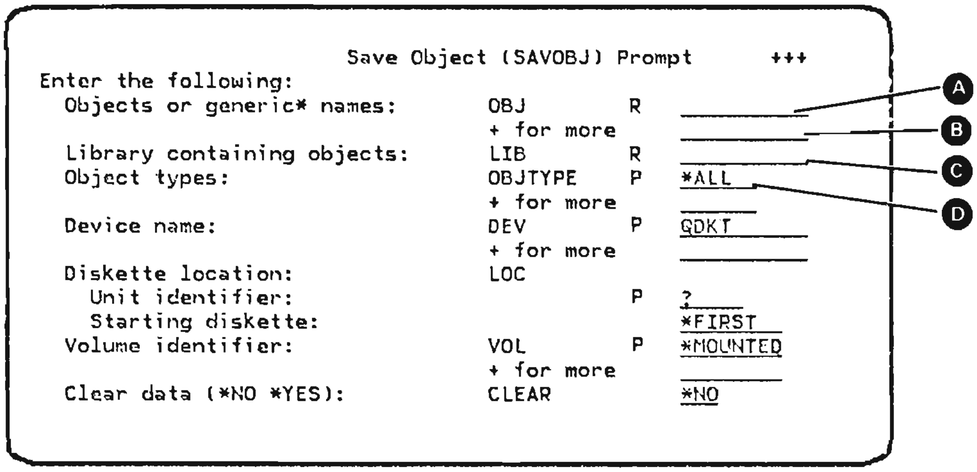
From this menu you see that the command you want is SAVOBJ (because files are objects).

e. To enter the command name SAVOBJ at the bottom of the menu, key in SAVOBJ or 1 for option 1.



```
                         Save/Restore Commands Menu
Select one of the following:
  Object
  1.SAVOBJ    - Save Object
  2.RSTOBJ    - Restore Object

  Library
  3.SAVLIB    - Save Library
  4.RSTLIB    - Restore Library

  System
  5.SAVSYS    - Save System
  6.RSTAUT    - Restore Authority
  7.RSTUSRPRF - Restore User Profiles
 Option or command name  SAVOBJ                   CF1 - Return
```

f. Press the Enter key. The system displays the save object prompt, which lists all the parameters of the SAVOBJ command.



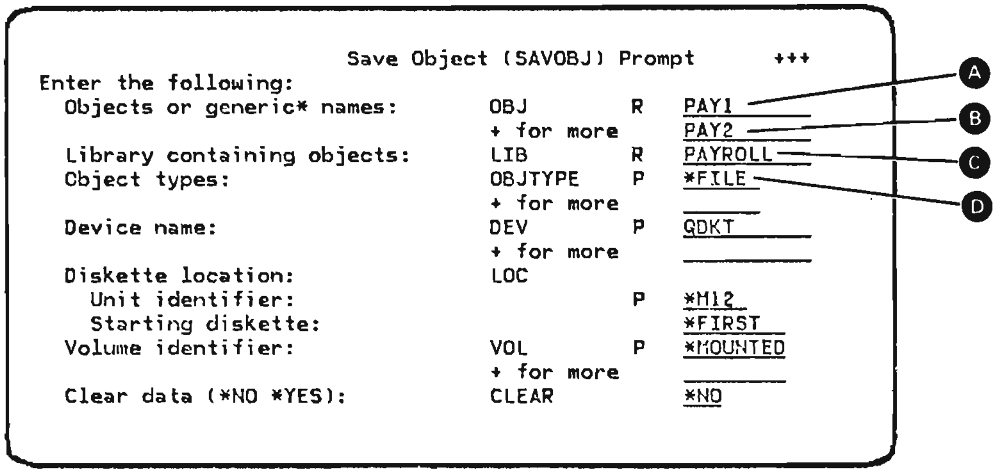
```
                         Save Object (SAVOBJ) Prompt      +++
Enter the following:
  Objects or generic* names:       OBJ        R   __________
                                   + for more     __________
  Library containing objects:      LIB        R   __________
  Object types:                    OBJTYPE    P   *ALL
                                   + for more     ______
  Device name:                     DEV        P   QDKT
                                   + for more     __________
  Diskette location:               LOC
    Unit identifier:                          P   ?
    Starting diskette:                            *FIRST
  Volume identifier:               VOL        P   *MOUNTED
                                   + for more     ________
  Clear data (*NO *YES):           CLEAR          *NO
```

2. The lines for the OBJ and LIB parameters are blank and there is an R next to the keyword for each, which means you must enter the name of at least one object and the name of the library where the object is located. All of the other parameters have defaults, which you can either accept as shown or change.

   In this case, you want to save two files PAY1 and PAY2 that are located in the library PAYROLL.

   a. Key PAY1 into the first OBJ field Ⓐ.
   b. Press the Field Advance key to move the cursor to the second OBJ field Ⓑ.
   c. Key in PAY2.
   d. Press the Field Advance key to move the cursor to the LIB field Ⓒ.
   e. Key in PAYROLL.
   f. Press the Field Advance key to move the cursor to the first OBJTYPE field Ⓓ.

g. Change field D to specify only files (which excludes other object types) by keying in *FILE on top of the default value *ALL, thus replacing the default value. The prompt now looks like this:



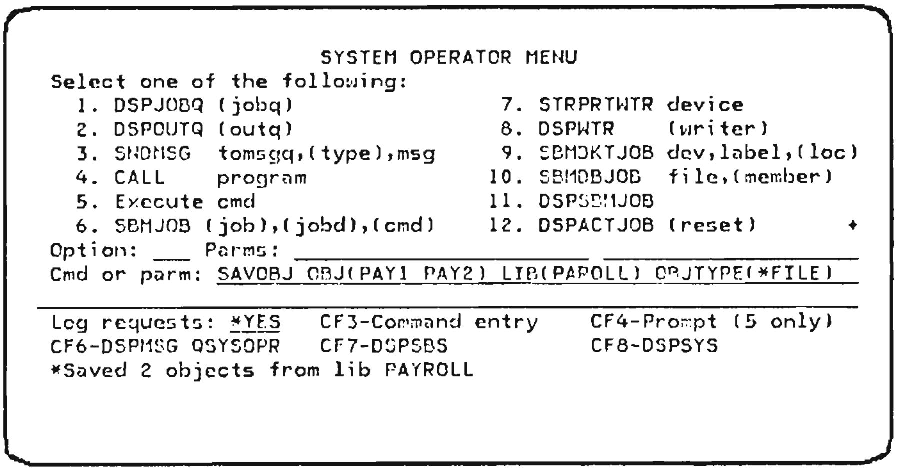
```
                        Save Object (SAVOBJ) Prompt       +++
Enter the following:
  Objects or generic* names:      OBJ         R   PAY1
                                  + for more      PAY2
  Library containing objects:     LIB         R   PAYROLL
  Object types:                   OBJTYPE     P   *FILE
                                  + for more
  Device name:                    DEV         P   QDKT
                                  + for more
  Diskette location:              LOC
    Unit identifier:                          P   *M12
    Starting diskette:                            *FIRST
  Volume identifier:              VOL         P   *MOUNTED
                                  + for more
  Clear data (*NO *YES):          CLEAR           *NO
```

You want the saved objects to be written on the currently mounted diskettes in the diskette magazine drive (QDKT), so you do not change the default values shown for the DEV, LOC, and VOL parameters. (Press the Enter key to view the second prompt screen.) Likewise, you do not change the default value shown for the STG parameter because you want to keep the storage currently occupied by the files. The ENDOPT parameter on the second screen applies only to magnetic tape, so you ignore it.

3. Now you have tailored the command to your needs. To execute it, you press the Enter key. If you want to execute the command as a job by using the submit job command, press the CF1 key. The prompted command will be returned on the Msg or Cmd line and you can execute it by using option 6.

   If any errors were detected in the values you supplied, the prompt would be redisplayed with a message on the bottom line indicating the error. In this case, there were no errors, so the command is successfully executed.

   After the command is executed, you are returned to the system operator menu display, which shows the command you entered and a message indicating that your request was completed:

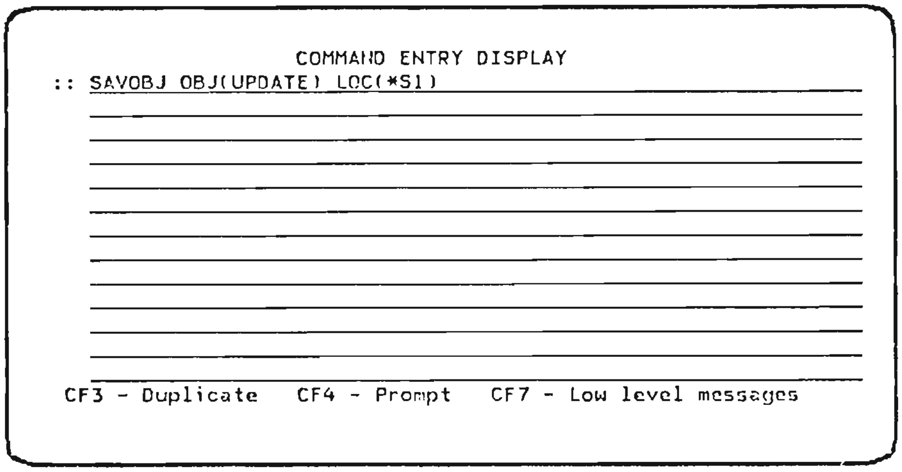


```
                         SYSTEM OPERATOR MENU
Select one of the following:
  1. DSPJOBQ (jobq)                   7. STRPRTWTR device
  2. DSPOUTQ (outq)                   8. DSPWTR    (writer)
  3. SNDMSG  tomsgq,(type),msg        9. SBMDKTJOB dev,label,(loc)
  4. CALL    program                 10. SBMDBJOB  file,(member)
  5. Execute cmd                     11. DSPSBMJOB
  6. SBMJOB (job),(jobd),(cmd)       12. DSPACTJOB (reset)        +
Option: ___ Parms: _______________________ ____________________
Cmd or parm: SAVOBJ OBJ(PAY1 PAY2) LIB(PAPOLL) OBJTYPE(*FILE)
_________________________________________________________________
Log requests: *YES   CF3-Command entry     CF4-Prompt (5 only)
CF6-DSPMSG QSYSOPR   CF7-DSPSBS            CF8-DSPSYS
*Saved 2 objects from lib PAYROLL
```

Example of Executing a Command from Partial Entry from the Command Entry Display

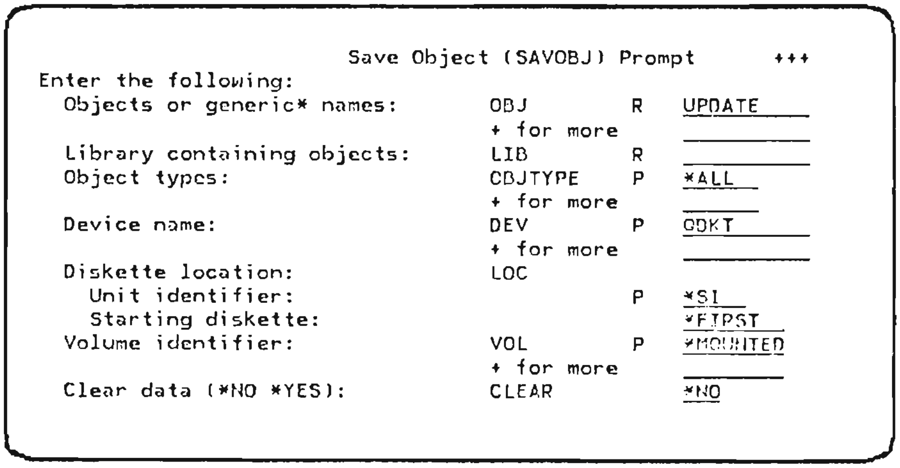
You want to save three short programs (objects) on the diskette that has been placed in I/O slot 1 (*S1) of the diskette magazine drive. You know how to enter some of the parameters but are uncertain of the others.

1. Because you want to display the details on each object saved, you should enter the save commands from the command entry display.
   a. Press the CF3 key from the system operator menu to obtain the command entry display. When the command entry display appears, you do the following:
   b. Key in the part of the command that you know is correct on the command entry display.



```
                          COMMAND ENTRY DISPLAY
 :: SAVOBJ OBJ(UPDATE) LOC(*S1)




















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

   c. Press the prompt key (CF4). The system displays the save object prompt as it did when you selected the SAVOBJ 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 object name UPDATE appears on the first OBJ line, which is normally blank, and the slot number *S1 appears on the first LOC line instead of the default value *M12.

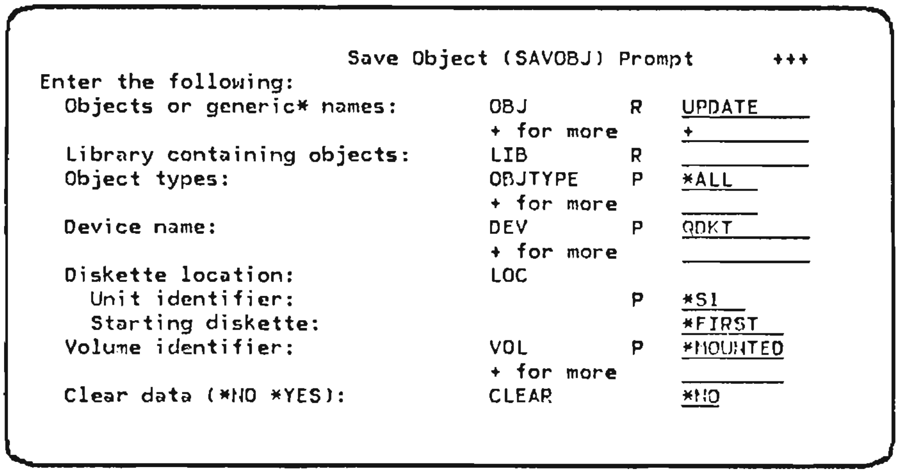


```
                       Save Object (SAVOBJ) Prompt        +++
Enter the following:
  Objects or generic* names:        OBJ          R   UPDATE
                                    + for more
  Library containing objects:       LIB          R
  Object types:                     CBJTYPE      P   *ALL
                                    + for more
  Device name:                      DEV          P   ODKT
                                    + for more
  Diskette location:                LOC
    Unit identifier:                             P   *SI
    Starting diskette:                               *FIPST
  Volume identifier:                VOL          P   *MOUNTED
                                    + for more
  Clear data (*NO *YES):            CLEAR            *NO
```

2. You want to enter two additional object names, but the prompt as displayed allows for only one more. However, the + *for more* statement next to the second line tells you that it is possible to obtain additional input fields for object names. To get those additional input fields:

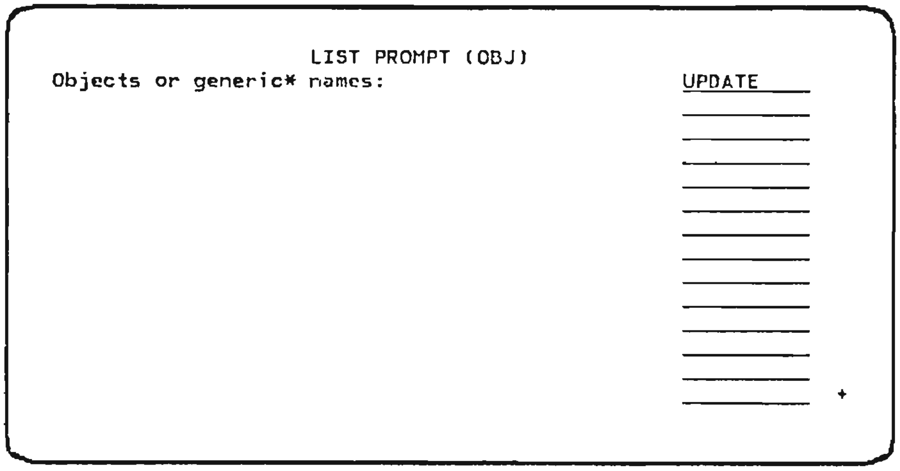
   a. Move the cursor to the second line by pressing the Field Advance key.

   b. Key in a +.



```
                          Save Object (SAVOBJ) Prompt       +++
Enter the following:
  Objects or generic* names:       OBJ        R   UPDATE____
                                   + for more     +_________
  Library containing objects:      LIB        R   __________
  Object types:                    OBJTYPE    P   *ALL__
                                   + for more     ______
  Device name:                     DEV        P   QDKT______
                                   + for more     __________
  Diskette location:               LOC
    Unit identifier:                          P   *S1__
    Starting diskette:                            *FIRST__
  Volume identifier:               VOL        P   *MOUNTED
                                   + for more     ________
  Clear data (*NO *YES):           CLEAR          *NO
```

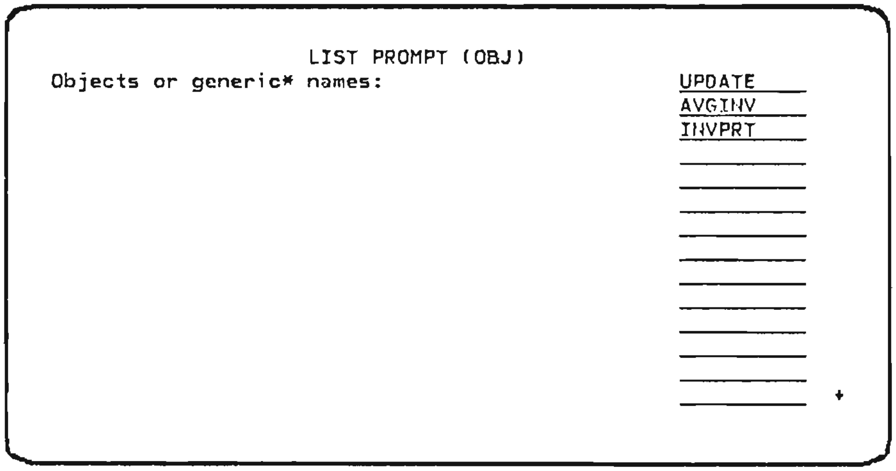
   c. Press the Enter key. The save object prompt is replaced by a separate list prompt.



```
                         LIST PROMPT (OBJ)
 Objects or generic* names:                        UPDATE____
                                                   __________
                                                   __________
                                                   __________
                                                   __________
                                                   __________
                                                   __________
                                                   __________
                                                   __________
                                                   __________
                                                   __________
                                                   __________
                                                   __________
                                                   __________   +
```

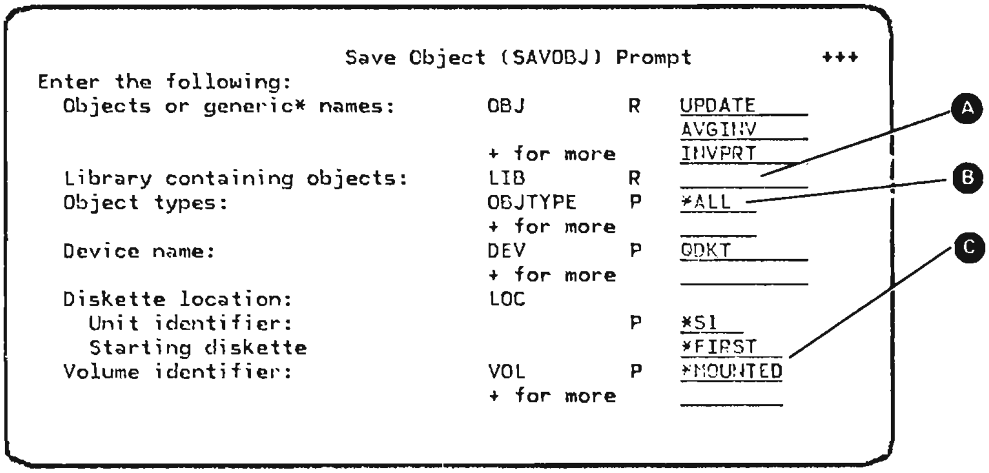
3. Note that the value you already entered appears as the first item of the list. Now you can add the two additional object names:
   a. Press the Field Advance key to move the cursor to the second line.
   b. Key in the first name, AVGINV.
   c. Press the Field Advance key to move the cursor to the next line.
   d. Key in the second name, INVPRT.

   The list prompt now looks like this:



```
                    LIST PROMPT (OBJ)
Objects or generic* names:                        UPDATE____
                                                  AVGINV____
                                                  INVPRT____
                                                  __________
                                                  __________
                                                  __________
                                                  __________
                                                  __________
                                                  __________
                                                  __________
                                                  __________
                                                  __________
                                                  __________ +
```

e. Press the Enter key. When you press the Enter key, the save object prompt reappears on your screen, but in a modified form that includes the values you added. When the additional object names are included in the prompt, however, there are too many lines of information to fit on your screen at one time. Therefore, the prompt is split into two separate displays. The first of the two displays appears on your screen; the +++ symbol in the upper right corner indicates there is another display to follow.



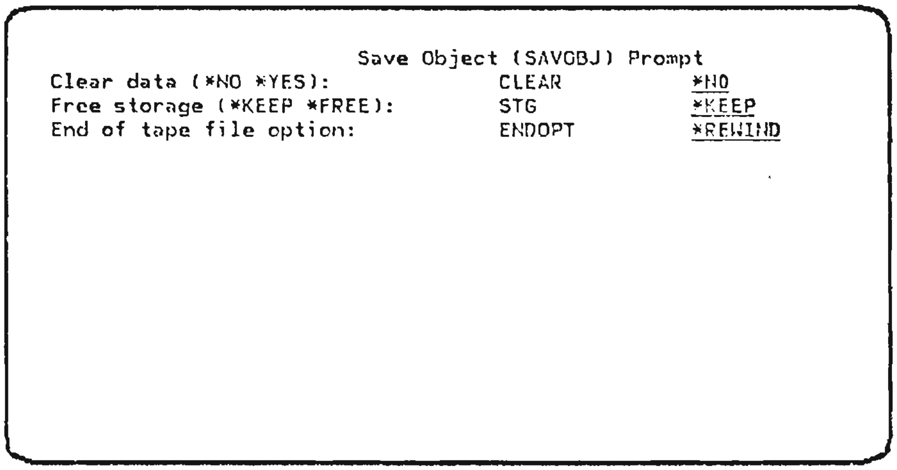
```
                           Save Object (SAVOBJ) Prompt              +++
Enter the following:
  Objects or generic* names:        OBJ         R   UPDATE
                                                    AVGINV
                                    + for more      INVPRT
  Library containing objects:       LIB         R   __________
  Object types:                     OBJTYPE     P   *ALL
                                    + for more      ______
  Device name:                      DEV         P   QDKT
                                    + for more      __________
  Diskette location:                LOC
    Unit identifier:                            P   *S1
    Starting diskette                               *FIRST
  Volume identifier:                VOL         P   *MOUNTED
                                    + for more      ________
```

You see from the first display of the modified prompt that you still must specify (in field A) the name of the library that the three programs (objects) are located in (there is an R beside the LIB keyword). There may be objects other than programs with the names you entered; therefore, you want to specify an object type *PGM in field B so that only programs will be saved. You also see that any diskette currently mounted in the slot will be used unless you specify a volume identifier in field C.

4. So you do the following:
   a. Move the cursor to field Ⓐ using the Field Advance key.
   b. Key in RECRDS.
   c. Press the Field Advance key to move the cursor to field Ⓑ.
   d. Key in *PGM on top of the default value *ALL.
   e. Move the cursor to field Ⓒ using the Field Advance key.
   f. Key in the volume identifier, RCD2A, on top of the first five characters (*MOUN) of the default value *MOUNTED; then press the Spacebar three times to erase the remaining three characters (TED).

5. You do not notice that you inadvertently keyed in the library name incorrectly as RECRDS rather than RECORDS.

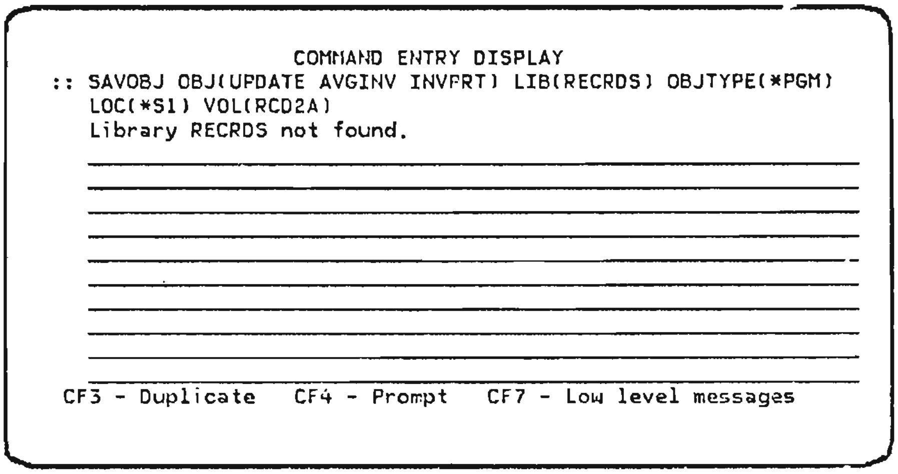
   When you press the Enter key, you receive the second display of the prompt.



```
                             Save Object (SAVOBJ) Prompt
Clear data (*NO *YES):                    CLEAR           *NO
Free storage (*KEEP *FREE):               STG             *KEEP
End of tape file option:                  ENDOPT          *REWIND
```

6. The second display contains three parameters that do not apply to this operation, so you leave it as it is and press the Enter key to execute the command.

   When the system tries to execute the command, it cannot find a library name that matches the name you specified. Therefore, the system redisplays the command on the command entry display, including all the changes you made on the prompt. Below the modified command is a message indicating why the command could not be executed.

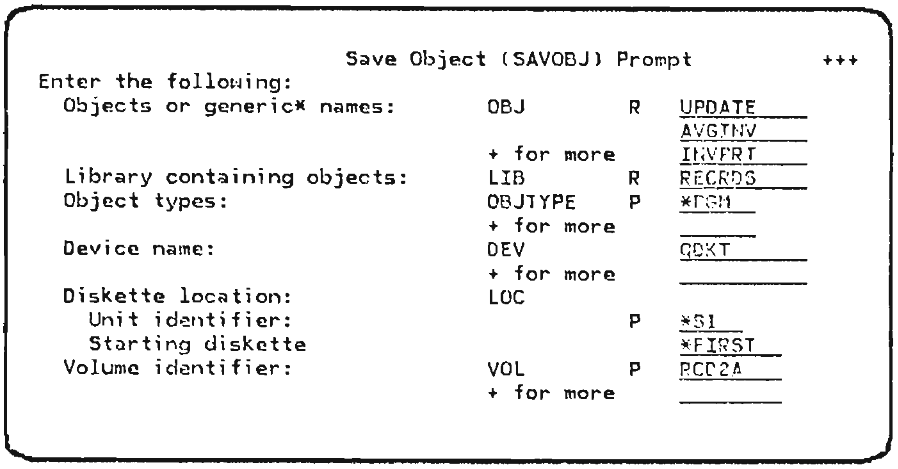


```
                         COMMAND ENTRY DISPLAY
:: SAVOBJ OBJ(UPDATE AVGINV INVFRT) LIB(RECRDS) OBJTYPE(*PGM)
   LOC(*S1) VOL(RCD2A)
   Library RECRDS not found.
   ________________________________________________
   ________________________________________________
   ________________________________________________
   ________________________________________________
   ________________________________________________
   ________________________________________________
   ________________________________________________
   ________________________________________________
   ________________________________________________
   ________________________________________________
 CF3 - Duplicate   CF4 - Prompt   CF7 - Low level messages
```

   You could correct the command on the command entry display, but you want to go back to the prompt.

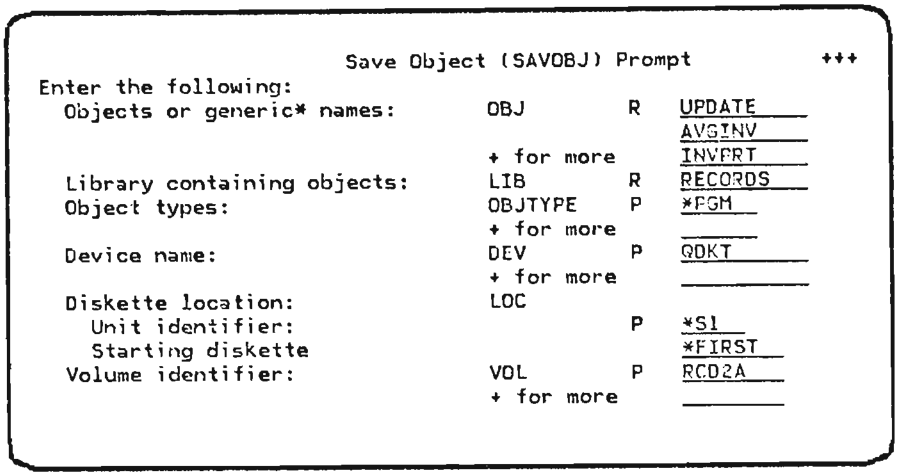
7. To obtain the prompt as you had modified it, you do the following:
   a. Press the CF3 key to duplicate the modified command into the current input area on the command entry display.
   b. Press the prompt key (CF4) to request the prompt for the command in the input area.

   You then receive the prompt with all of the values as you changed them:



```
                           Save Object (SAVOBJ) Prompt            +++
Enter the following:
  Objects or generic* names:       OBJ        R   UPDATE____
                                                  AVGINV____
                                   + for more     INVFRT____
  Library containing objects:      LIB        R   RECRDS____
  Object types:                    OBJTYPE    P   *PGM__
                                   + for more     ______
  Device name:                     DEV            QDKT______
                                   + for more     __________
  Diskette location:               LOC
    Unit identifier:                          P   *S1_
    Starting diskette                             *FIRST_
  Volume identifier:               VOL        P   RCD2A__
                                   + for more     ________
```

8. You correct the error by keying in *ORDS* on top of the *RDS*. The first display of the prompt now looks like this:



```
                        Save Object (SAVOBJ) Prompt                 +++
Enter the following:
  Objects or generic* names:      OBJ          R    UPDATE____
                                                    AVGINV____
                                  + for more        INVPRT____
  Library containing objects:     LIB          R    RECORDS___
  Object types:                   OBJTYPE      P    *PGM__
                                  + for more        ______
  Device name:                    DEV          P    QDKT______
                                  + for more        __________
  Diskette location:              LOC
    Unit identifier:                           P    *S1__
    Starting diskette                               *FIRST__
  Volume identifier:              VOL          P    RCD2A___
                                  + for more        ________
```

If you press the Enter key, you will receive the second display of the prompt and will have to press the Enter key again to execute the command.

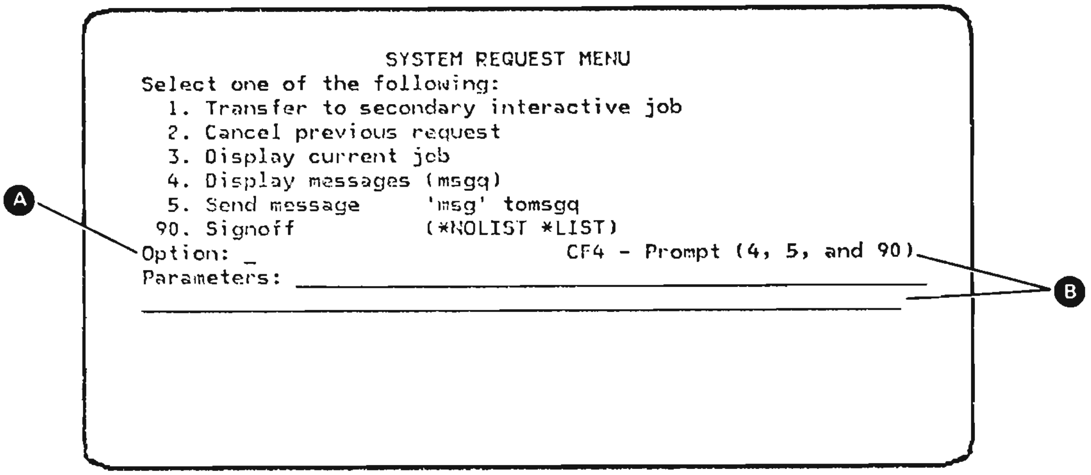
9. You know that you do not want to change the second display of the prompt, so you press the CF16 key instead of the Enter key.

   The second display is bypassed, and the command is executed, based only on the information you have entered on the first display of the prompt. You then receive the command entry display, which shows your entered command along with the message that indicates your request was successfully completed.

10. To get the details on each object saved, you press the CF7 key from the command entry display.

USING THE SYSTEM REQUEST MENU

Each time you enter a system request by pressing the Sys Req key on the system console, your current interactive job is suspended at the point where you left it and you receive the system request menu (below). When you start the system, this menu becomes available to you as soon as the start CPF process is completed (see *Starting CPF* in Chapter 8). During system operations, the menu is available to anyone who creates an interactive job on the system by signing on at a work station. The menu remains available until the interactive job is terminated, such as when you or a work station user sign off or you power down the system.



```
                    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: ________________________________________________
__________________________________________________________
```

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. The first option allows you to create a second interactive job. Each of the remaining options is associated with a specific command. You select an option by keying the option number into the *Option* field Ⓐ. When you select any of the options 4 through 6, you are, in effect, executing the associated command. Therefore, 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* in this chapter). Default parameter values are provided for parameters that you choose not to enter. The default values that are provided for options 4 through 6 are indicated in parentheses next to the description of the option. For option 5, the two values that you *must* enter are indicated next to the option description. Except for option 5, the option number is all you need to enter to execute the associated function.

After you have selected any of the options 4 through 6, you can request the prompt for the command associated with the option by pressing the CF4 key. If you specify parameter values in the *Parameters* field before requesting the prompt, those values will be shown on the prompt (see *Command Parameter Prompting* in this chapter).

When you press the Enter key, the function you specified in the *Option* and *Parameters* fields is executed. However, if your entry resulted in an error, you will receive an error message at the bottom of your screen. The function will not be executed unless you key in the correct information on top of your previous input and press the Enter key again.

To return to your interactive job from the system request menu, press the CF1 key. Your job will then be resumed at the point where you left it, and you will receive the display that was on your screen when you entered the system request by pressing the Sys Req key.

Following is a description of each option.

Option 1: This option allows you to 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 system operator menu or 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 secondary job is suspended and you return to your primary job. From that point on, each time you request the 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.

Note: If you use this option to exit your interactive job 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 is completed. The resources being held are not available for other uses. Therefore, unless there is a problem that requires an immediate exit from the job, you should use this option only when the system has completed your previous request and is waiting for you to enter a new request.

If the controlling subsystem is in the restricted state, Option 1 is not available for the single interactive job that is active in that subsystem. The controlling subsystem enters the restricted state following these conditions:

- Terminate CPF (TRMCPF)
- Terminate all subsystems (TRMSBS *ALL)
- Terminate subsystem of the controlling subsystem (TRMSBS)
- A CE signing on at the start CPF prompt
- IMPL where storage recovery is required

The controlling subsystem remains in the restricted state until restarted by the STRSBS command (or STRSBS option under CSM).

Option 2: This option allows you to cancel the request that was active when you entered the system request. You could use this option, for example, to cancel a command that you do not want to complete. After the request has been canceled, you will return to your interactive job and receive the display from which you entered the request that was canceled (for example, the system operator menu).

Note: Locks obtained by the user on external objects through the use of the Allocate Object (ALCOBJ) command are *not* unlocked (deallocated) by the cancel request.

Option 3: This option allows you to use the Display Job command to display the status of the interactive job from which you made the system request. When you enter 3 in the *Option* field, you receive the first of several available job displays for the job (see *Specific Job Information* in Chapter 12). When you press the Enter key after viewing the displays, you return to the system request menu.

Option 4: This option allows you to use the Display Messages command to display the messages in a message queue. You receive a separate message display (see *Displaying Messages from a Queue* in Chapter 11). If you enter only *4* in the *Option* field, you receive a display of messages in the message queue associated with the work station your are using. If you also specify the name of a message queue in the *Parameters* field, you receive a display of the messages in the specified message queue. For example, to display the messages in the system operator message queue, you would enter QSYSOPR in the *Parameters* field. You can also enter any other parameters of the Display Messages command. When you press the Enter key afer examining the messages, you return to the system request menu.

Option 5: This option allows you to use the Send Message command to send a message to a specified message queue. After entering *5* in the *Option* field, you enter the message in the *Parameters* field, followed by the name of the message queue where the message is to be sent. For example, to send the message

```
Time to sign off
```

to the WRKSTN3 message queue, you enter in the *Parameters* field

```
'Time to sign off' WRKSTN3
```

Note that the message must be enclosed in apostrophes. You can also enter any other parameters of the Send Message command in the *Parameters* field (see *Sending Nonurgent Messages* in Chapter 11).

Option 90: This option allows you to use the SIGNOFF command to end the interactive job from which you made the system request. If you enter only *90* in the *Option* field, the job will be ended without the job log for the job being printed (see *Work Station Sign-Off* in this chapter). To print the job log, you must also specify *LIST in the *Parameters* field. Signing off from a job cancels any processing being done in the job. If you sign off your interactive job from the system request menu, the system interprets the sign-off as an abnormal termination of the job.

WORK STATION SIGN-OFF

You end your interactive job at a work station when you sign off. Under some conditions, system power-down will not be completed until you and all other work station users sign off (see *Powering Down the System* in Chapter 8). There may also be times when you want to sign off for other reasons, such as to prevent an unauthorized person from using system operator commands. If the system is still available when you sign off, the sign-on prompt is redisplayed for the next user.

You sign off by entering the Sign Off (SIGNOFF) command or option 90 on the system operator menu. The form of the command you use depends on how your work station is connected to System/38 and whether you want to print the job log for your interactive job. The command has two parameters.

LOG Parameter: Each job in the system, including your interactive job, has an associated job log for retaining job information (see *Job Logs* in Chapter 12). You use this parameter when you want a printed copy of the job log for the job you are signing off.

- If you do not enter this parameter, the job log is not printed.
- If you specify LOG(*LIST) your job log is placed on an output queue for printing. Note, however, that if you power down the system without signing off, your job log is not printed until the next time the system is started.

DROP Parameter: You use this parameter if your work station is connected to the system by a switched communication line.

- Do not enter this parameter if you are using the system console or a 5251 or 5252 connected to the system through a cable or a nonswitched line (the parameter is ignored).

- If you are using a remote work station connected to the system through a switched line (5251 Model 2 or 12, or another 5251 or a 5252 connected to a Model 2 or 12) and you do not enter this parameter, DROP (*DEVD) is assumed and the disposition of the communication line is based on the work station device description stored in the system.

- If your work station is connected to the system through a switched line and you specify DROP(*YES) the communication line is disconnected, provided that no other work station on the line is signed on.

- If your work station is connected to the system through a switched line and you specify DROP(*NO) the communication 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.

Examples:

1. To sign off the system console without printing the job log for your interactive job, enter

   ```
   SIGNOFF
   ```

2. To sign off the system console and print the job log for your interactive job, enter

   ```
   SIGNOFF LOG(*LIST)
   ```

3. To sign off a work station connected by a switched line and print your job log, but retain the line connection, enter

   ```
   SIGNOFF LOG(*LIST) DROP(*NO)
   ```

Chapter 7. System Operator Commands

The CPF control language commands that you are likely to use in operating System/38 are listed in this chapter. The use of these commands is discussed in the following chapters.

SECURITY CONSIDERATIONS

Depending on the security restrictions of your system, you may or may not be allowed to use all the commands listed. You also may not be allowed to specify particular objects in those commands. If you attempt to enter a command and the system responds with a message stating that you are not authorized to use the command, such as

```
Not authorized to use command RSTAUT in library QSYS
```

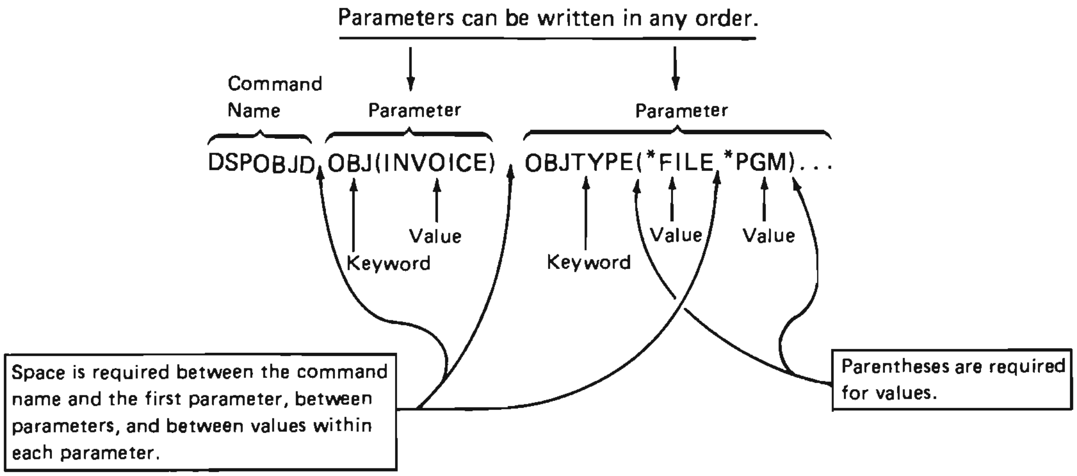
contact the security officer for your system.

The security officer controls your user profile, stored in the system, that defines which commands you can use.

ENTERING COMMANDS

Commands can be entered in keyword-with-value form or positionally. The two forms are shown below.

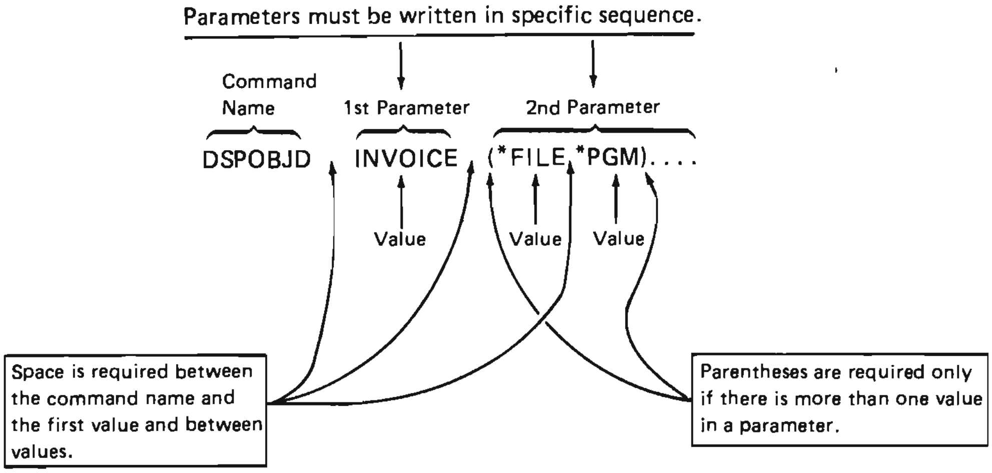
KEYWORD-WITH-VALUE FORM


Parameters can be written in any order.
Command Name
Parameter
Parameter
DSPOBJD OBJ(INVOICE) 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.

Parentheses are required for values.

POSITIONAL FORM


Parameters must be written in specific sequence.
Command Name
1st Parameter
2nd Parameter
DSPOBJD INVOICE (*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.

USING SYNTAX DIAGRAMS

Two typical syntax diagrams are shown on the next page (Figure 7-1). The following discussion, keyed to the two examples, summarizes how to interpret and use the syntax diagrams that appear in the *IBM System/38 Programming Reference Summary*, SC21-7734. The letters indicate specific features of the diagrams noted in the discussion. The numbers indicate specific parameters and the parameter sequence.

Command Name

The command name Ⓐ appears first on the left side of the diagram. In the two sample diagrams of Figure 7-1, the command names are SAVLIB and SNDBRKMSG.

Parameter Order

All parameters of the command are shown in the correct sequence for positional coding. The order goes from left to right on each line and continues on the fo'lowing line. That is, the proper order is ❶, ❷, ❸, and ❹ in the top diagram and, similarly, from ❶ through ❹ in the bottom diagram of Figure 7-1. The ⟨P⟩ after the fourth parameter in the top diagram means that you should not code positionally after the fourth parameter.

When coding parameters in the positional form, you must enter them in the order shown in the diagram. If you choose not to include a parameter and other parameters are to be coded in the positions following the omitted parameter, then you must enter *N to represent the omitted parameter.

The number of parameters which can be specified positionally is limited. The limit for each command is shown by a P on the command prompts and a ⟨P⟩ in the *Programming Reference Summary*.

No parentheses are shown in the diagrams, but parentheses must enclose the parameter value whenever it is preceded by the keyword (keyword-with-value coding) or when multiple values are included in one parameter (a list of values).

Parameter Keywords

For each parameter, the keyword that would be used if the command were coded in the keyword-with-value form is shown first, followed by the value or values. In the top diagram of Figure 7-1, for example, the keyword of parameter ❶ is LIB, the keyworo of parameter ❸ is LOC, and the keyword of parameter ❺ is STG.

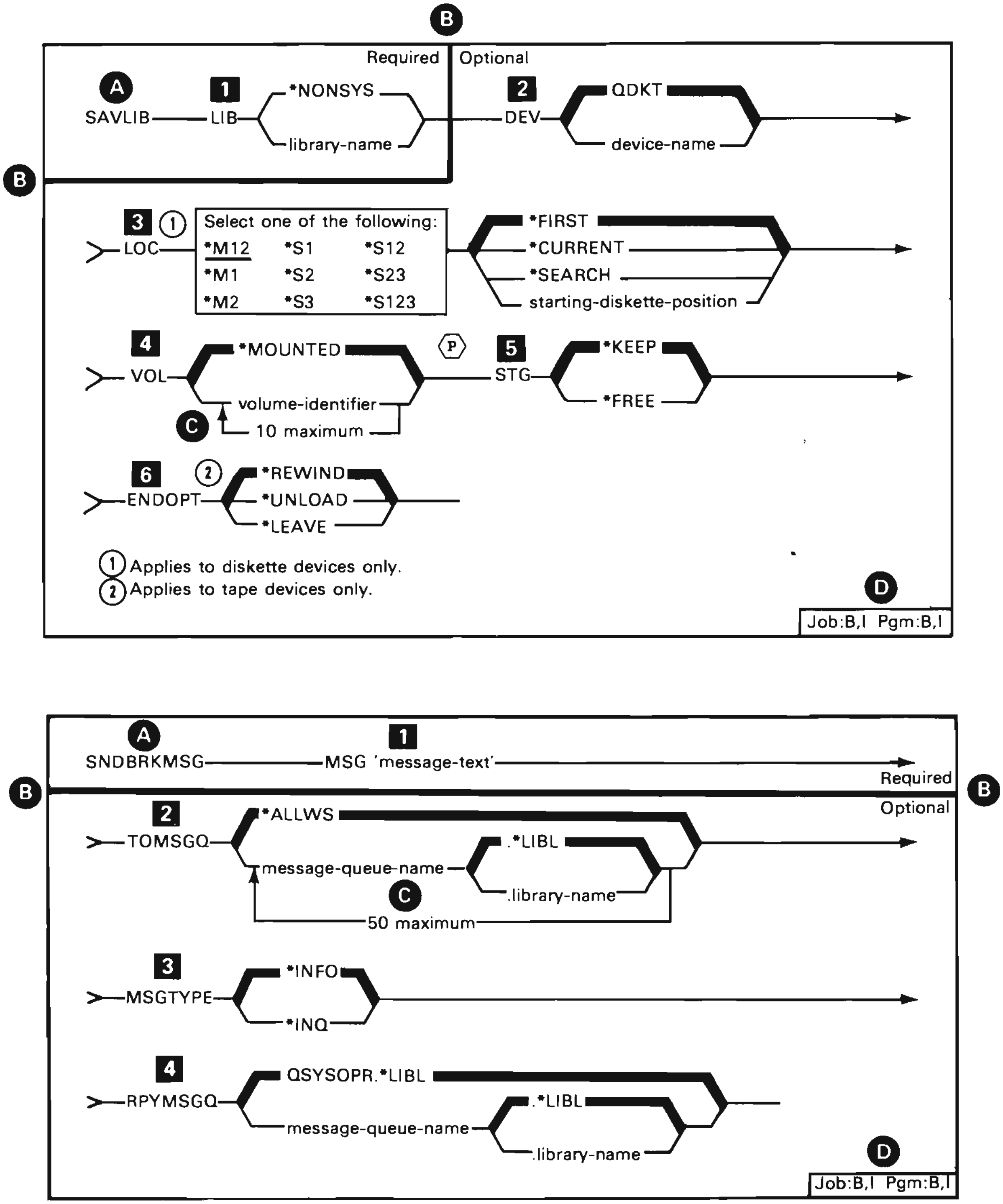


Figure 7-1. Sample Syntax Diagrams

Required and Optional Parameters

All required parameters, if any, appear before the optional parameters. The required parameters, with their keywords and values, are separated from the optional parameters by a heavy dividing line Ⓑ. The required parameter area is identified by the word *Required* above or to the left of the dividing line. The optional parameter area is identified by the word *Optional* below or to the right of the dividing line. If there are no required parameters, no dividing line is used. If there are required parameters but no optional parameters, *Required* is shown at the bottom of the diagram.

Predefined Values/System-Supplied Objects

Predefined values and system-supplied objects are shown exactly as they must be coded.

Predefined values, all of which begin with an asterisk, are values that have a special meaning to the system when entered in a command. In the top diagram of Figure 7-1, for example, the value *NONSYS in parameter 1 and all values in parameter 6 are predefined values.

The names of system-supplied objects, that is, objects supplied with system programming when it is installed, all begin with the letter Q. The value QSYSOPR in parameter 4 on the bottom diagram in Figure 7-1 is the name of a system-supplied object.

User-Defined Values

Where you can enter various values of a certain type rather than a specific predefined value, the type of value is indicated by descriptive words in lower case, such as in parameter 1 of both diagrams. When you enter the parameter, you substitute an actual value for the descriptive words. For example, you might code parameter 1 in the top diagram of Figure 7-1 as

```
LIB(PAYROLL)
```

You have substituted *PAYROLL* for the descriptive words *library-name.*

Quoted Values

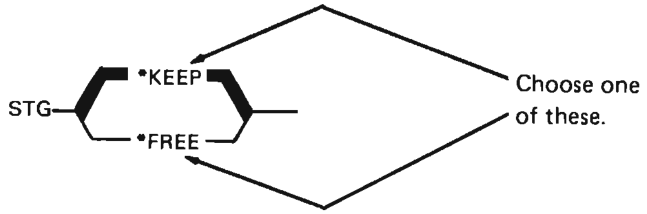
Some commands, such as those used to send messages, allow you to specify a string of characters as the value. To indicate that those characters are to be reproduced exactly as you entered them, including blanks and special characters, you must enclose the value in apostrophes. Characters enclosed in apostrophes are called a quoted string.

Where such quoted strings are allowed, as in parameter **1** of the bottom diagram of Figure 7-1, the apostrophes are shown around the descriptive words for the value to remind you to enclose that value in apostrophes. For example you might code parameter **1** of the bottom diagram as

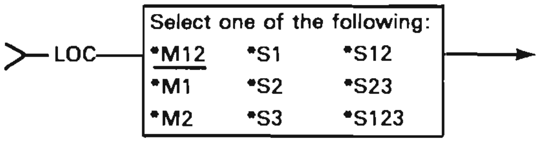
MSG('System going down at 7:00')

Choice of Values

Where you have a choice of alternative values for a parameter, those alternatives are shown either as parallel branches in the diagram

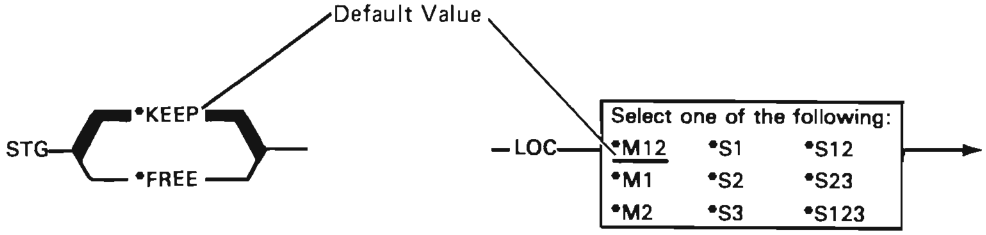


or enclosed in a box from which you are instructed to select one.



Default Values

Optional parameters have default values that are used if you do not include the parameters when entering the command. These default values are indicated by a heavy branch line or an underline. The default is always the top or first value in the group shown.



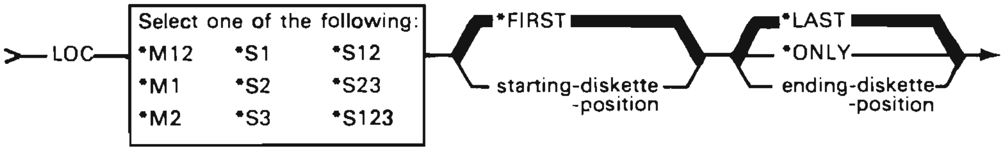
Simple List of Values

Some commands allow you to enter more than one value of the same type (a simple list) in a parameter. This is indicated in the diagram by an arrow looping back beneath the value and a number on the arrow that indicates the maximum number of values (as in Ⓒ on the two sample diagrams of Figure 7-1). When you code the actual values in the command, you separate them by a space, such as

```
OBJ(PAY1 PAY2 PAY3)
```

Mixed List of Values

A few system operator commands (not shown in Figure 7-1) allow you to enter one or more sets of separately defined values (a mixed list) in a parameter. Such a parameter has a form:

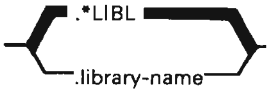

LOC
Select one of the following:
*M12
*S1
*S12
*M1
*S2
*S23
*M2
*S3
*S123
*FIRST
starting-diskette -position
*LAST
*ONLY
ending-diskette -position

The elements of the mixed list must be entered in the order indicated on the syntax diagram. If you used the first or top value in each of the three elements of the example shown here, the entered parameter would look like this:

```
LOC(*M12 *FIRST *LAST)
```

Qualified Object Names

Where an object name can be used as a value of a parameter, you generally have the option of qualifying the object name by the name of the library in which it is located. That option is indicated in the form



as in parameters 2 and 4 of the bottom diagram of Figure 7-1. If you entered a qualified message queue name for parameter 2, it might look like this:

```
TOMSGQ(WSTN3.QGPL)
```

If you do not enter the library qualifier, the default value *LIBL is used and the system searches through a list of libraries for the job until it finds the object you specified.

Qualified Job Names

Parameters whose value is a job name (not shown in the two examples of Figure 7-1) allow you to optionally add either a user name or both a user name and a job number to the job name (a qualified job name). Such a parameter is shown in a syntax diagram in the form:

```
JOB job-name[.user-name[.job-number]]
```

This indicates that you can code the parameter in three different ways. If the job name was PAY, the user name was DEPT25, and the job number was 000358, the three ways would be:

```
JOB(PAY)
JOB(PAY.DEPT25)
JOB(PAY.DEPT25.000358)
```

Whether you specify just the job name or the job name plus one or both of its qualifiers depends on how much information you need to supply for the system to find the correct job. If there is no other job by the same name, you can specify only the simple job name (PAY). If there is more than one job by the same name, you will need to specify the job name plus the name of the user who submitted it (PAY.DEPT25). If more than one job by the same name was submitted by the same user, you will also need to specify the job number assigned to the job when it entered the system (PAY.DEPT25.000358).

Batch and Interactive Entry Codes

The box insert in the lower right corner of each syntax diagram (Ⓓ in Figure 7-1) contains the entry codes that specify where the command can be entered. Each code has one of the following meanings:

Code	**Representing**	**Meaning**
Job:B	Batch job	Valid in batch jobs, external to compiled CL program
Job:I	Interactive job	Valid for interactive entry, external to compiled CL program
Job:B,I	Batch and interactive jobs	Valid for batch and interactive entry, external to compiled CL program
Pgm:B	Program, batch	Valid in compiled CL program that is called from batch entry
Pgm:I	Program, interactive	Valid in compiled CL program that is called from interactive entry
Pgm:B,I	Program, batch and interactive	Valid in compiled CL program that is called from batch or interactive entry

COMMAND LIST

The following is a list of commands that are used to manage the operation of the system. The list is arranged in alphabetical order by command name. For information on how these commands are used, refer to the discussion for the particular command in subsequent chapters of this manual (see Index). For additional details on the commands themselves or other commands not described in this manual, refer to the *CL Reference Manual.*

You can also receive a prompt for each command by entering the command name and pressing the CF4 key.

Commands

ANSLIN (Answer Line)
APYPGMCHG (Apply Programming Change)

CALL (Call)
CHGJOB (Change Job)
CHGMSGQ (Change Message Queue)
CHGOUTQ (Change Output Queue)
CHGSBSD (Change Subsystem Description)
CHGSPLFA (Change Spooled File Attributes)
CHGSYSVAL (Change System Value)
CLNPRT (Clean Print)
CLRDKT (Clear Diskette)
CLRJOBQ (Clear Job Queue)
CLROUTQ (Clear Output Queue)
CNLJOB (Cancel Job)
CNLRDR (Cancel Reader)
CNLSPLF (Cancel Spooled File)
CNLWTR (Cancel Writer)
CPYF (Copy File)
CPYSPLF (Copy Spooled File)

DLTDKTLBL (Delete Diskette Label)
| DSPACTJOB (Display Active Jobs)
DSPCTLSTS (Display Control Unit Status)
DSPDEVCFG (Display Device Configuration)
DSPDEVSTS (Display Device Status)
DSPDKT (Display Diskette)
DSPJOB (Display Job)
DSPJOBQ (Display Job Queue)
DSPLINSTS (Display Line Status)
DSPLOG (Display Log)
DSPMSG (Display Messages)
DSPOBJD (Display Object Description)
DSPOUTQ (Display Output Queue)
| DSPPGMCHG (Display Programming Change)
DSPRDR (Display Reader)
DSPSBS (Display Subsystem)
DSPSBMJOB (Display Submitted Job)
DSPSPLF (Display Spooled File)
DSPSPLFA (Display Spooled File Attributes)
DSPSYS (Display System)
DSPSYSSTS (Display System Status)
DSPSYSVAL (Display System Value)
DSPTAP (Display Tape)
DSPWTR (Display Writer)
DUPDKT (Duplicate Diskette)

HLDJOB (Hold Job)
HLDJOBQ (Hold Job Queue)
HLDOUTQ (Hold Output Queue)
HLDRDR (Hold Reader)
HLDSPLF (Hold Spooled File)
HLDWTR (Hold Writer)

INZDKT (Initialize Diskette)
INZTAP (Initialize Tape)

LODPGMCHG (Load Programming Change)

PWRCTLU (Power Control Unit)
PWRDEV (Power Device)
PWRDWNSYS (Power Down System)

RGZPFM (Reorganize Physical File Member)
RLSJOB (Release Job)
RLSJOBQ (Release Job Queue)
RLSOUTQ (Release Output Queue)
RLSRDR (Release Reader)
RLSSPLF (Release Spooled File)
RLSWTR (Release Writer)
RMVPGMCHG (Remove Programming Change)
RNMDKT (Rename Diskette)
RSTAUT (Restore Authority)
RSTLIB (Restore Library)
RSTOBJ (Restore Object)
RSTUSRPRF (Restore User Profiles)

SAVLIB (Save Library)
SAVOBJ (Save Object)
SAVSYS (Save System)
SBMCRDJOB (Submit Card Jobs)
SBMDBJOB (Submit Data Base Jobs)
SBMDKTJOB (Submit Diskette Jobs)
SBMJOB (Submit Job)
SIGNOFF (Sign Off)
SNDBRKMSG (Send Break Message)
SNDMSG (Send Message)
STRCRDRDR (Start Card Reader)
STRCRDWTR (Start Card Writer)
STRDBRDR (Start Data Base Reader)
STRDKTRDR (Start Diskette Reader)
STRDKTWTR (Start Diskette Writer)
STRPRTWTR (Start Printer Writer)
STRSBS (Start Subsystem)

TRMCPF (Terminate Control Program Facility)
TRMSBS (Terminate Subsystem)

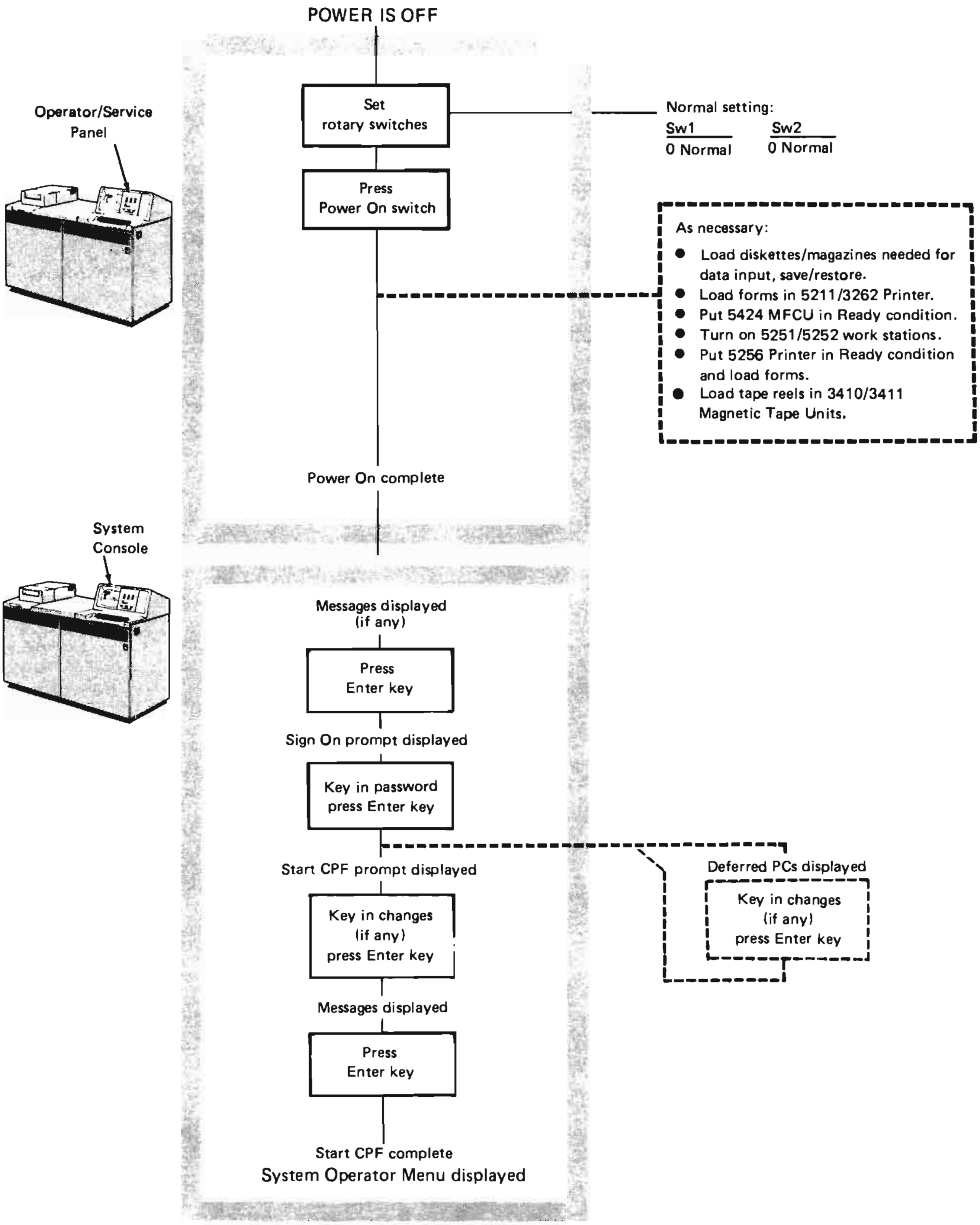
VRYCTLU (Vary Control Unit)
VRYDEV (Vary Device)
VRYLIN (Vary Line)

Chapter 8. General System Operations

This chapter describes general operating procedures, such as starting the system and preparing it for normal operations, determining the operating status of the system, and shutting down the system. Subsequent chapters provide details of specific operations in device control, spooling, message handling, job control, and save/restore.

PREPARING SYSTEM/38 FOR OPERATION

To prepare the system for normal operations, you initiate the power-on sequence using the operator/service panel (see Chapter 1) and then use the system console (see Chapter 2) to start the Control Program Facility (CPF) through which you and other users communicate with the system. A typical sequence for starting the system is shown in Figure 8-1. A discussion of each step in the sequence follows.



As necessary:

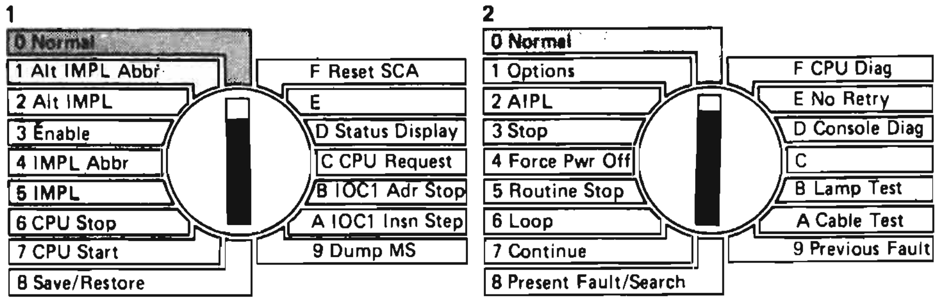
- Load diskettes/magazines needed for data input, save/restore.
- Load forms in 5211/3262 Printer.
- Put 5424 MFCU in Ready condition.
- Turn on 5251/5252 work stations.
- Put 5256 Printer in Ready condition and load forms.
- Load tape reels in 3410/3411 Magnetic Tape Units.

Figure 8-1. Typical System/38 Starting Sequence

Initiating the Power-On Sequence

To start the power-on sequence:

1. Set the two rotary switches on the operator/service panel:



   You use the setting shown here when starting the system to resume normal, everyday operations or when starting the system for the first time after it has been installed. For situations other than normal start-up, refer to *Variations of the Start-Up Procedure* in this chapter.

2. Press the Power On switch on the operator/service panel. If your system has the Power Key Lock feature, you must unlock the Power On switch first; insert the key and turn it clockwise until it stops.

The system performs the IMPL (initial microprogram load) sequence. During the IMPL sequence, the microcode that controls fundamental machine operations is loaded from auxiliary storage. In addition, the hardware undergoes extensive diagnostic testing to ensure that the system is operating properly and all machine components are intialized (set to a known starting condition). After the microcode is loaded, the process begins that loads the Control Program Facility (CPF) or equivalent programming from auxiliary storage. The part of the sequence that loads CPF is referred to as the IPL (initial program load).

When the IMPL sequence is completed, the system is ready to begin normal operations, and a message display or the sign-on prompt appears on the system console screen (see *Preparing System/38 for Operation* in Chapter 8).

Note: If your system has the Auto-IMPL feature and the switch is set to auto-IMPL, the system will automatically restart after a power interruption and perform the same IMPL sequence described here.

Indications of Normal Power-On Sequence

Once the power-on sequence has been initiated, power is supplied to each device connected to the system and the system automatically performs all actions necessary to check system hardware, initialize system components, and load system programming. The amount of time required to complete the sequence depends on the configuration of your system and whether any cleanup operations are needed. A normal power-on sequence will generally be completed in a few minutes.

As long as the power-on sequence is proceeding normally, some of the condition indicators on the operator/service panel will flicker on and off or come on briefly (see *Lights* in Chapter 1). However, if any of these lights remain on for more than a minute, and the Manual indicator is on, the system has encountered a failure that it cannot correct and the power-on sequence has been suspended at that point. If this occurs, see the *Problem Determination Guide*.

When the power-on sequence is completed, the system console screen (see Chapter 2) displays messages, if they are of high enough severity, that were sent to the system operator message queue regarding the startup. (Message queues and message displays are described in Chapter 11.) If there are no messages with high enough severity to be displayed, the sign-on prompt appears. If additional messages of high enough severity are generated during the remaining steps in starting the system, you will see the message display again each time another message arrives in the system operator message queue.

After reviewing the messages, press the Enter key to continue.

Signing On

When you press the Enter key to indicate you are finished with the message display, that display is replaced by the sign-on prompt (Figure 8-2). This indicates that the system is ready to establish an interactive session with you (an interactive job).

To sign on:

1. Key in your password. The characters *do not* appear on the screen as you key them in.

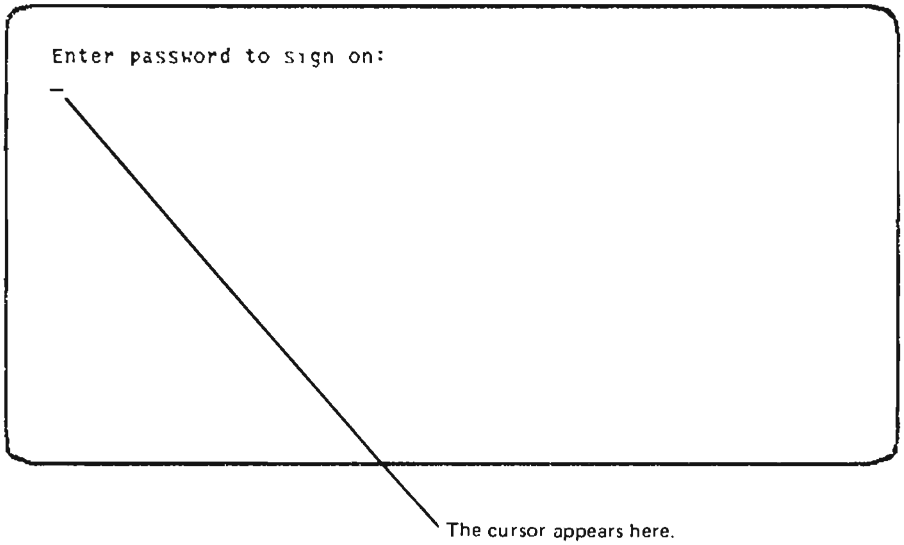
2. Press the Enter key.

Your password directs the system to your user profile stored internally. The user profile tells the system you are authorized to perform system operator functions.

IBM provides a standard user profile for the system operator. This user profile, identified as QSYSOPR, gives you the authority to perform job control and save/restore operations. To use this user profile, you sign on by entering the password

SYSOPR

Depending on the security restrictions of your system, you may be instructed to use a different password.



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

The cursor appears here.

Figure 8-2. Sign-On Prompt

Handling Deferred Programming Changes

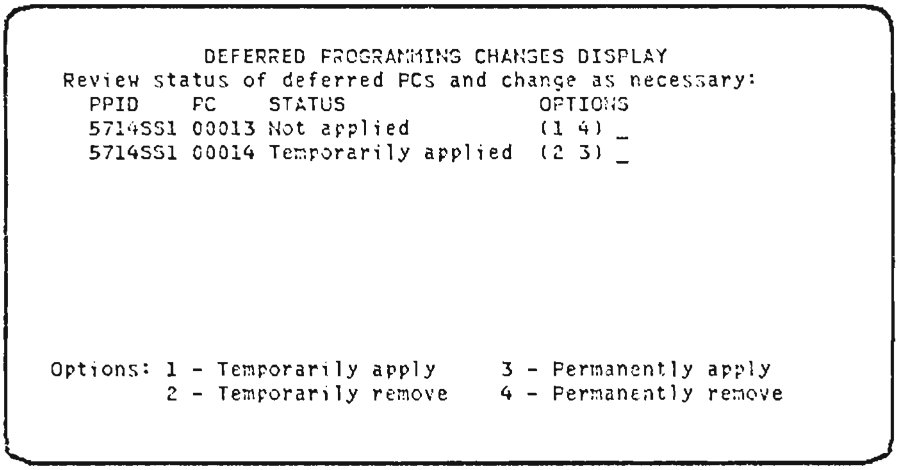
Periodically, programming changes (PCs) are provided for application to the licensed programs in your system. If any of those PCs had to be deferred because the program product was active, you receive a deferred programming changes display (Figure 8-3) after you sign on.

The deferred program change display appears only if there are deferred programming changes. If it does appear, you can respond to it in one of two ways:

- Key in one of the indicated options and press the Enter key.

- Press only the Enter key.

For details on using this display, refer to *Applying a Deferred Programming Change* in Chapter 14.



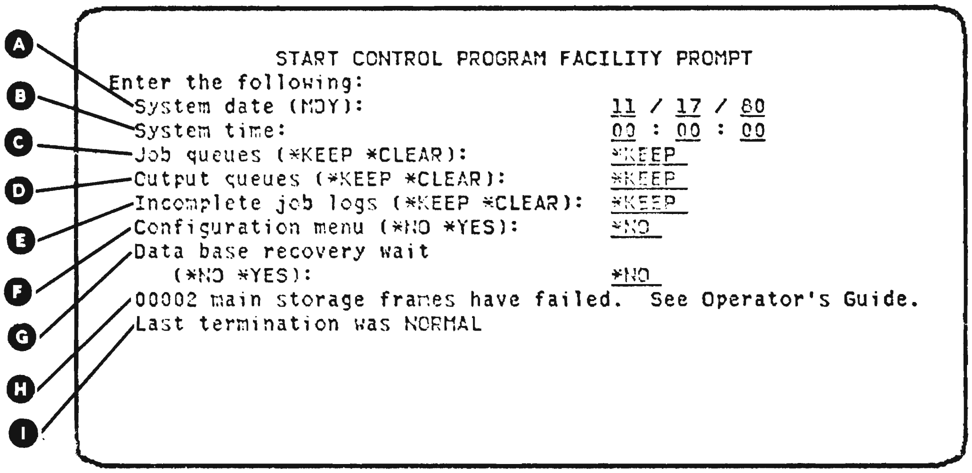
```
                DEFERRED PROGRAMMING CHANGES DISPLAY
     Review status of deferred PCs and change as necessary:
       PPID     PC     STATUS                 OPTIONS
       5714SS1 00013 Not applied              (1 4) _
       5714SS1 00014 Temporarily applied      (2 3) _










    Options: 1 - Temporarily apply      3 - Permanently apply
             2 - Temporarily remove     4 - Permanently remove
```

Figure 8-3. Deferred Programming Changes Display

Starting CPF

When you have signed on and completed the programming changes display, if any, the system responds with the start Control Program Facility (start CPF) prompt (Figure 8-4). As with all prompts, the default values shown on the display are used by the system unless you change them.

CAUTION
Because of the need for recovery capability during IMPL, the options on the start CPF prompt have no authority restrictions. Any person permitted to sign on at the console during IMPL can use any of the options on the Start CPF prompt. For example, information could be lost if output queues, job queues, or incomplete job logs are inadvertently cleared.



```
                START CONTROL PROGRAM FACILITY PROMPT
Enter the following:
  System date (MDY):                       11 / 17 / 80
  System time:                             00 : 00 : 00
  Job queues (*KEEP *CLEAR):               *KEEP
  Output queues (*KEEP *CLEAR):            *KEEP
  Incomplete job logs (*KEEP *CLEAR):      *KEEP
  Configuration menu (*NO *YES):           *NO
  Data base recovery wait
     (*NO *YES):                           *NO
  00002 main storage frames have failed.  See Operator's Guide.
  Last termination was NORMAL
```

Figure 8-4. Start CPF Prompt

The following discussion, keyed to each item of the prompt, explains the meanings of the items and what to do about them.

Ⓐ *System date*: The date shown is the system estimate of the current date. If the date is wrong, you can correct it by keying in new numbers on top of those shown. Because it is possible to display the date different ways, the format in which the date is displayed is also indicated. In the example shown here (Figure 8-4), the format is month-day-year (MDY). The format is controlled by the system value QDATFMT.

Ⓑ *System time*: The time shown is the system estimate of the current time. You can key in another value on top of the value shown. For example, you may want to change the time to match the time of day.

Note: The system estimate of the current date and time is based on the status of the last termination of the system (see item Ⓘ). If the last termination was normal, the estimated time and date are the start (00:00:00 hours, minutes, and seconds) of the next day (the date of the system termination plus one day). If the last termination was abnormal, the system date and time are based on the last system date and time values saved by the machine plus a small time increment. The addition of the small time increment may or may not cause the system date to change.

C *Job queues*: Allowing the default value (*KEEP) to remain as shown retains jobs contained on job queues except jobs that are on the queue as the result of a Transfer Job (TFRJOB) command, thus making those jobs available for processing. If you key in *CLEAR on top of the default value (*KEEP), the job queues will be cleared of all jobs. The start CPF job also cleans up any jobs that have been left in the system because of timing, locks, or other conditions that prevented complete reclaim of job resources.

D *Output queues*: If the default value (*KEEP) is retained as shown, output from previous jobs on output queues is kept for future use. Keying in *CLEAR on top of the default value *KEEP removes all entries from the output queues, thus removing all spooled output from the system.

If *CLEAR is specified and the IMPL does not complete, you must specify *CLEAR at the next IMPL to complete cleanup that may have been started during the IMPL that failed.

E *Incomplete job logs*: If the last system termination was abnormal, you have the option of not printing the job logs for the jobs that were active or transferring when the termination occurred (see *Job Logs* in Chapter 12). This option also applies to the start CPF job. If you do not change the default value (*KEEP), the job logs will be printed. If you key in *CLEAR, the job logs will not be printed. You might be instructed to specify *CLEAR, for example, if the terminated jobs would have large job logs.

F *Configuration menu*: Normally, you will want to retain the existing system configuration, and therefore should not change the default value (*NO) that is shown. If you key in *YES in place of *NO, a configuration menu will be displayed after you complete the start CPF prompt; that is, when you press the Enter key (see *Configuration Menu* in this chapter).

G *Data base recovery wait*: If the previous system power down was abnormal, clean-up activities may be required on files that were being used at the time of the termination. This option allows you to override the clean-up activities for all files created with RECOVER(*AFTSTRCPF) as if they had been specified RECOVER(*STRCPF). It does not affect clean-up activities for files created with RECOVER(*STRCPF) or RECOVER(*NO).

If you do not change the default value (*NO) shown, those clean-up activities will be performed while other system processing is taking place. However, the files involved will not be available for use until the clean-up activities are completed. If you key in *YES in place of *NO, system processing will not be allowed to start until the clean-up activities are completed.

The actual default value shown on the prompt is determined by the system value QDBRCVYWT (see *System Values* in this chapter). Your system was shipped from IBM with QDBRCVYWT set at '0', so that the default value shown on the prompt will be *NO, as in Figure 8-3. However, if QDBRCVYWT was set to '1' the last time the system was active (or at some previous time), *YES will appear as the default value. If *YES is shown, you will have to key in *NO in place of it if you do not want to wait for data base clean-up to be completed.

H *Main storage failure*: This line will appear on the prompt only if one or more frames have failed in main storage; that is, if they cannot be used for storing information. The system can still be used, with little or no noticeable effect on performance; however, you should contact your service representative and report how many of the main storage frames have failed, as indicated on the prompt. If the number of main storage frame failures exceeds a value specified in the system value QBADPGFRM (see *System Values* in this chapter), the start CPF process is terminated and some of the condition indicators on the operator/service panel light up. See the *Problem Determination Guide* for the actions you take when the condition indicators come on.

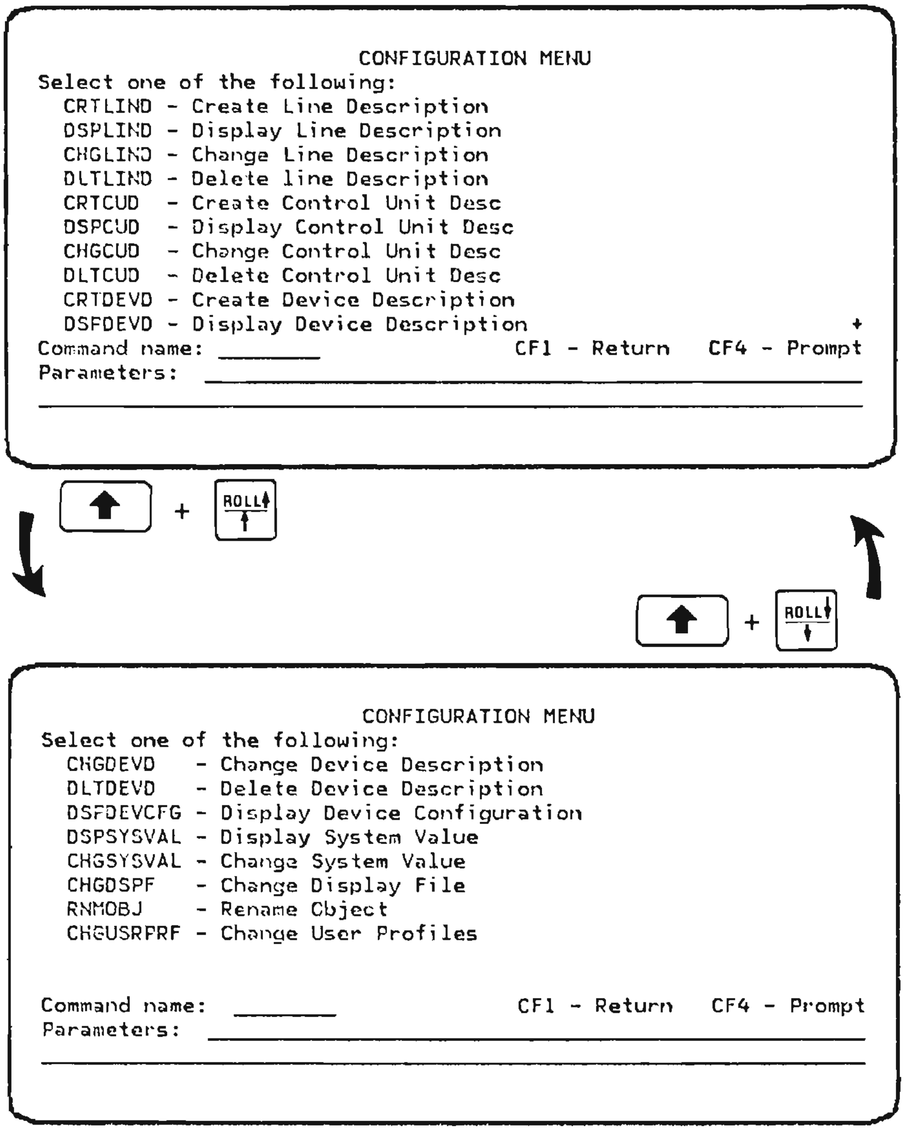
I *Last termination*: This line is for your information only; no response from you is needed. NORMAL means that the previous system termination was achieved through completion of a Power Down System (PWRDWNSYS) command. If there is an ABNORMAL indication (such as a termination resulting from a power interruption or a system failure), residual processing may be required after the start CPF sequence is completed, resulting in a delay in processing user jobs. Another indication of an ABNORMAL prior termination is a system time showing something other than 00:00:00.

When you have completed your changes (if any) to the start CPF prompt, press the Enter key.

As soon as the start CPF process (including any configuration changes) is completed, the system operator menu normally appears on the system console screen. However, if there are any messages relating to the start CPF process, you will first see a message display similar to the display you received before you signed on. When you press the Enter key, you then see the system operator menu.

Configuration Menu

When you enter *YES on the *Configuration menu* line of the start CPF prompt (Figure 8-4), you receive a configuration menu (Figure 8-5) that allows you to select commands for displaying, creating, or changing various elements of the system configuration.



```
                         CONFIGURATION MENU
Select one of the following:
  CRTLIND - Create Line Description
  DSPLIND - Display Line Description
  CHGLIND - Change Line Description
  DLTLIND - Delete line Description
  CRTCUD  - Create Control Unit Desc
  DSPCUD  - Display Control Unit Desc
  CHGCUD  - Change Control Unit Desc
  DLTCUD  - Delete Control Unit Desc
  CRTDEVD - Create Device Description
  DSPDEVD - Display Device Description                               +
Command name: ________                    CF1 - Return   CF4 - Prompt
Parameters:  ____________________________________________________
_________________________________________________________________
```

```
                         CONFIGURATION MENU
Select one of the following:
  CHGDEVD   - Change Device Description
  DLTDEVD   - Delete Device Description
  DSPDEVCFG - Display Device Configuration
  DSPSYSVAL - Display System Value
  CHGSYSVAL - Change System Value
  CHGDSPF   - Change Display File
  RNMOBJ    - Rename Object
  CHGUSRPRF - Change User Profiles


Command name:  ________                   CF1 - Return   CF4 - Prompt
Parameters:  ____________________________________________________
_________________________________________________________________
```

Figure 8-5. Configuration Menu

Because the menu contains too much information to display at one time on your screen, it is divided into two parts. To obtain the next part, press the Roll Up key; to return to the previous part, press the Roll Down key.

You can either enter a complete command on the menu or request a prompt to assist you in entering a command. To enter a complete command, use the following procedure:

1. Key one of the command names shown on the menu into the *Command name* field 🅐.

2. Key the parameters of the command into the *Parameters* field 🅑.

3. Press the Enter key.

When the command is executed, input fields 🅐 and 🅑 will be cleared for your next entry.

To request a prompt from the menu, use the following procedure:

1. Key one of the command names into the *Command name* field 🅐.

2. If you know any of the parameters you want to use, key them into the *Parameters* field 🅑. This step is not required.

3. Press the prompt key (CF4).

Note: If the commands reference an object that is not in QSYS, you must use a qualified object name.

You then receive a prompt display for the selected command. Any parameters you entered on the configuration menu are shown on the prompt. When you complete the prompt and press the Enter key, the command is executed and the configuration menu reappears on your screen.

If your entry results in an error, you will receive an error message on the bottom line of your display (configuration menu or prompt). You correct the error by keying in the correct information and pressing the Enter key again.

To exit the configuration menu, you press the CF1 key. The start CPF process is then complete, and you receive your basic working display, such as the system operator menu.

You will not normally use the configuration menu. However, there may be occasions when you will use the menu to change a system value (see the subsequent discussion of *System Values* in this chapter). The advantage of changing a system value through this menu is that, because CPF has not yet been started, the change becomes effective for particular system values (such as QCTLSBSD) as soon as the current start CPF process is completed. If you change those values after CPF is started, the change does not become effective until the *next* time CPF is started. This is not true for all system values, however.

The procedures for using the configuration menu are summarized in Figure 8-6.

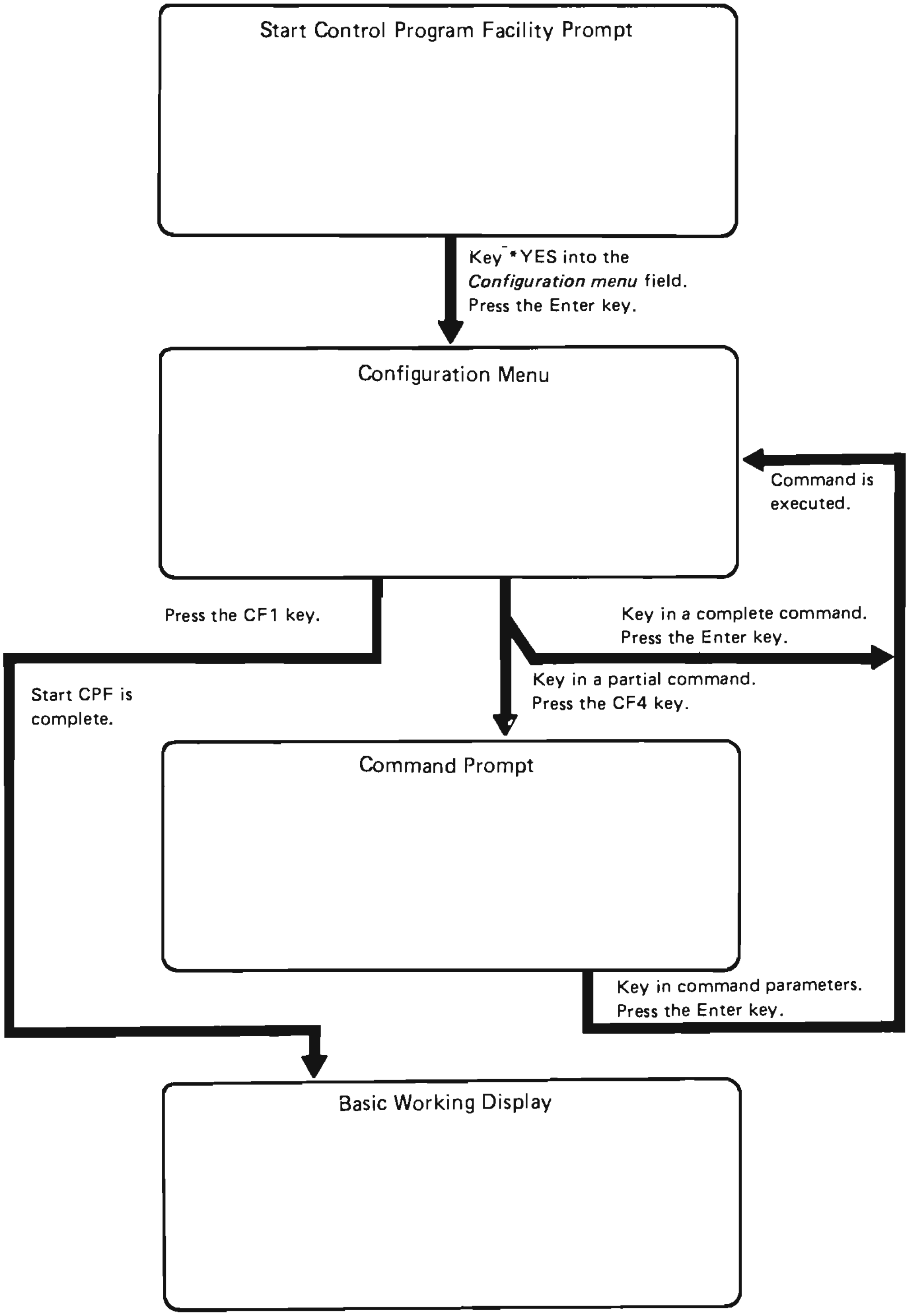

Start Control Program Facility Prompt
Key *YES into the *Configuration menu* field. Press the Enter key.
Configuration Menu
Command is executed.
Press the CF1 key.
Key in a complete command. Press the Enter key.
Start CPF is complete.
Key in a partial command. Press the CF4 key.
Command Prompt
Key in command parameters. Press the Enter key.
Basic Working Display

Figure 8-6. Using the Configuration Menu

VARIATIONS OF THE STARTUP PROCEDURE

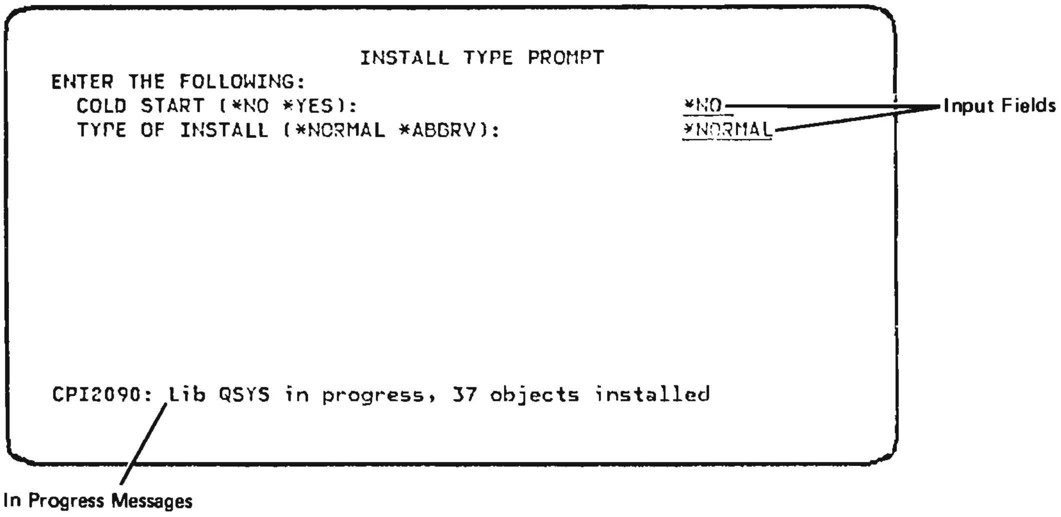
If you are starting the system under other than normal conditions, the procedure you follow may be somewhat different or completely different from the procedure just described.

Installing CPF

If you are installing or reinstalling CPF, you set the rotary switches as shown in *Settings to Install CPF* in Chapter 1. This sequence involves copying information into the system from diskettes or tapes. You must place the diskette magazines containing the CPF information into the diskette magazine drive, or for tape, place the CPF installation diskette into Magazine 1 Position 1 and mount the first CPF tape on the first tape unit *before* pressing the Power On switch.

The procedure you follow from that point on is similar to the procedure previously described for normal startup. Before you receive the sign-on prompt, however, two preliminary displays regarding details of the installation appear on the 'system console screen. You must respond to each.

The first display prompts you for the type of installation to be performed. It contains two input fields, each of which has a default value. You do not change the default values shown unless you are reinstalling CPF to correct a problem of damaged objects in the system (see *Damaged Objects* in the *System/38 Problem Determination Procedures.*



```
                         INSTALL TYPE PROMPT
ENTER THE FOLLOWING:
  COLD START (*NO *YES):                                  *NO
  TYPE OF INSTALL (*NORMAL *ABGRV):                       *NORMAL









CPI2090: Lib QSYS in progress, 37 objects installed
```

During installation, status messages (CPIxxxx) appear on the message line of the installation screens to inform you of the progress of the installation. You do not have to respond to these messages.

The field labeled *COLD START* refers to damage to noninstalled objects; that is, objects such as the system operator message queue that are not brought into auxiliary storage from diskette or tape but are created after the CPF is installed.

- If you do not change the default value (*NO), the system does not delete noninstalled objects unless the they are found to be damaged.

- If you change the default value to *YES, all noninstalled objects are deleted and recreated as a group. This is known as a *cold start*.

A cold start is a means of correcting damage that results in an error condition the system is not prepared to handle. Because objects are deleted during a cold start, any user data accessed through those objects is lost. For example, during a cold start, the CPF job table is destroyed and recreated, thus deleting all jobs on job queues and all output files on output queues.

The field labeled *TYPE OF INSTALL* allows you to specify an abbreviated form of CPF installation in combination with a cold start.

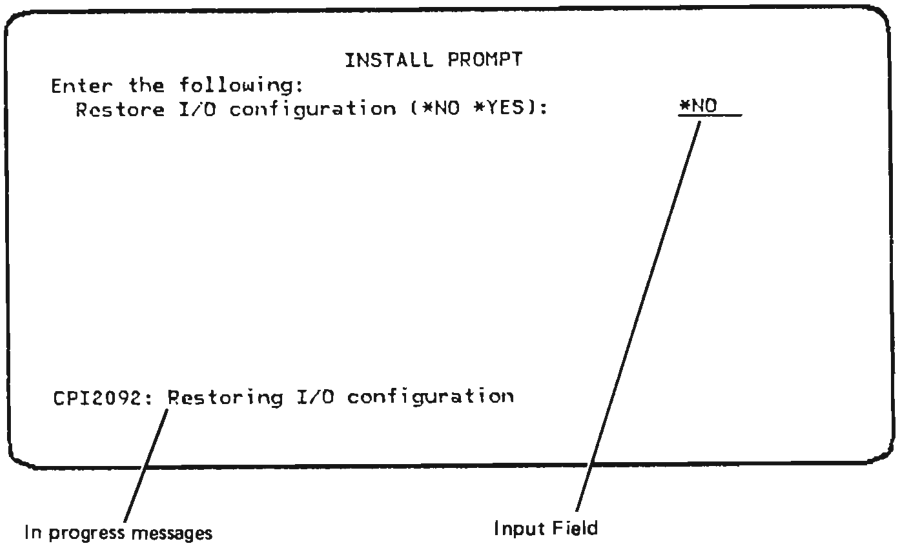
- If you do not change the default value (*NORMAL), CPF objects are brought into auxiliary storage from diskette or tape, as would be done during a normal installation or reinstallation of CPF. The CPF objects brought in replace the same CPF objects, if any, that were previously in auxiliary storage.

- If you change the default value to *ABBRV, CPF objects are not brought in from diskette or tape; therefore, the installed CPF objects in auxiliary storage are not replaced. This abbreviated install process requires much less time than a normal installation of CPF.

If you are doing a normal installation of CPF, you should not change either of the values shown on the prompt. To do a cold start, you specify *YES in the *COLD START* field and *ABBRV in the *TYPE OF INSTALL* field. If you need to reinstall CPF as well as do a cold start, you specify *YES in the *COLD START* field but do not change the *TYPE OF INSTALL* field. For guidelines on when to use these procedures, see the *Problem Determination Guide*.

When you are installing CPF from tape, inquiry type messages will be displayed that request that you mount and ready the tape. After mounting and readying the tape on the first magnetic tape drive, take the default response by pressing the Enter key to continue the CPF installation.

When you press the Enter key, you receive a second install prompt on which you must indicate whether the saved input/output configuration is to be restored.



```
                           INSTALL PROMPT
Enter the following:
  Restore I/O configuration (*NO *YES):             *NO



CPI2092: Restoring I/O configuration
```

You specify *YES in the input field only when you are reinstalling a version of CPF that you have saved using the save/restore function. The *YES option destroys all line, control unit, and device descriptions except the device descriptions for the diskette magazine drive and console. If the install is from tape, the tape control unit description and the tape device description used for installation are not destroyed. The system then restores all the line, control unit, and device descriptions that were on your system when you saved CPF with the save/restore function.

When you press the Enter key after reviewing (and changing if necessary) the second display, you receive the sign-on prompt (or message display). The remainder of the startup sequence is the same as described previously in this chapter.

Note: If auxiliary storage is initialized, all system and user information stored on it is destroyed. You restore vital system information to auxiliary storage by reinstalling CPF. After CPF is started, you can restore user information by entering the Restore Library (RSTLIB) command with the SAVLIB(*NONSYS) option (see *Restoring the System* in Chapter 13). During the start CPF process, QSYS is the only system library and QGPL and QTEMP are the only user libraries on auxiliary storage. If libraries other than these three are specified in the system value QUSRLIBL or QSYSLIBL, they will be ignored because the system will not be able to find the libraries. The configuration menu may be used during the start CPF process to change QUSRLIBL to *QGPL* and *QTEMP* and QSYSLIBL to *QSYS* (see *Changing System Values* in this chapter). After CPF has been started and normal system operations have begun, you (or the system security officer) can change QUSRLIBL and QSYSLIBL back to their original value.

Activating the Service Monitor Mode

If you are starting the system to use the dedicated service monitor, you set the rotary switches as shown in *Settings to Activate the Service Monitor Mode* in Chapter 1. As when installing CPF, this sequence involves copying information (microcode) into the system from diskettes. Therefore, you must place the magazines containing the microcode into the diskette magazine drive *before* pressing the Power On switch.

When the power-on sequence is completed, you do not receive the sign-on prompt or any of the displays that precede or follow the sign-on prompt during a normal startup. Instead, you receive a display on which you can enter numeric codes to perform functions such as reloading the system microcode, dumping the contents of virtual storage onto diskettes, and printing the information dumped onto diskettes. The procedures are described in the *Problem Determination Guide.*

To prepare the system for normal operations after using the service monitor mode, you must reset the rotary switches and perform a normal IMPL.

Auto IMPL

If your system has the auto-IMPL feature and there is a power interruption, the system will automatically restart and perform the IMPL sequence. The sign-on that normally precedes the start CPF process is bypassed, and default values are used for the start CPF prompt. The controlling subsystem becomes active automatically, as in a normal startup, and the sign-on prompt is displayed at the system console (see *Subsystems* in this chapter). When you sign on, you receive your basic working display, such as the system operator menu.

During the initial auto-IMPL sequence, the Power On switch is not lighted. If you press the Power On switch during that period, the normal startup sequence is used instead of the auto-IMPL sequence.

If the last IMPL was an auto-IMPL, the system value QAUTOIMPL will be set at '1'; otherwise, it will be '0' (see *Displaying System Values* in this chapter).

Note: When the system is restarted by an auto-IMPL, messages to the system operator are not automatically displayed on the system console screen (see *The System Operator Message Queue* in Chapter 11). To have the messages displayed automatically, as they are when you start the system, you should set the system operator message queue to the break mode (see *Changing a Message Queue* in Chapter 11).

Startup Without the System Console

There is a system value called QSCPFCONS that allows the system to be started if it is not possible to sign on the system console because of device problems (see *System Values* later in this chapter). If the value of QSCPFCONS is set to '1' and the system detects that the console is not operating properly, while the system is being started, defaults are used for the start CPF prompt and sign-on is deferred as in the auto-IMPL sequence. You can then sign on another work station and control system operations from that work station. However, you can only use a work station that is under control of the controlling subsystem.

The controlling subsystem QCTL that is initially provided for the system (see *Subsystems* in this chapter) allows only the system console to be used in beginning system operations. Therefore, before QSCPFCONS is set to '1', an alternative controlling subsystem needs to be defined (see *Controlling Subystem (QCTL)* in this chapter) that includes at least one other work station in addition to the system console.

Note: When the system console is bypassed, messages to the system operator that would normally be displayed at the system console are *not* automatically displayed at the alternative work station (see *The System Operator Message Queue* in Chapter 11). You must request the messages by using the Display Messages (DSPMSG) command. You can have the messages displayed automatically at the work station by setting the system operator message queue to the break mode (see *Changing a Message Queue* in Chapter 11).

Emergency Startup Procedure

Not all console failures can be detected by the system. In particular, some display failures are not found during IMPL. The result is that IMPL can complete with a blank console display, with no display in the operator panel condition indicators, and when the system appears to have stopped.

As a way of completing startup in an emergency when service is not immediately available, you can try the following procedure. The controlling subsystem and the system value QSCPFCONS must be set as described under *Startup Without the System Console* for this procedure to work.

1. If the IMPL stops with a blank screen and there is no indication on the operator panel condition indicators, check the alternative console work station specified in the controlling subsystem description. If a sign-on prompt is displayed, you may proceed normally using the alternative console.

2. If no sign-on prompt is displayed on the alternative console, attempt to enter your password at the system console keyboard. Make sure that you enter your password correctly because it will probably not be displayed.

3. If the system stops processing during startup, press the Enter key to allow startup to continue.

4. Continue to check the alternative console for a sign-on display. When the sign-on prompt appears, sign on with your password.

5. After you sign on at the alternative console, cancel all interactive jobs associated with the system console to free the system operator message queue. Then allocate the system operator message queue to the alternative console.

6. Check the system operator message queue for messages and respond appropriately.

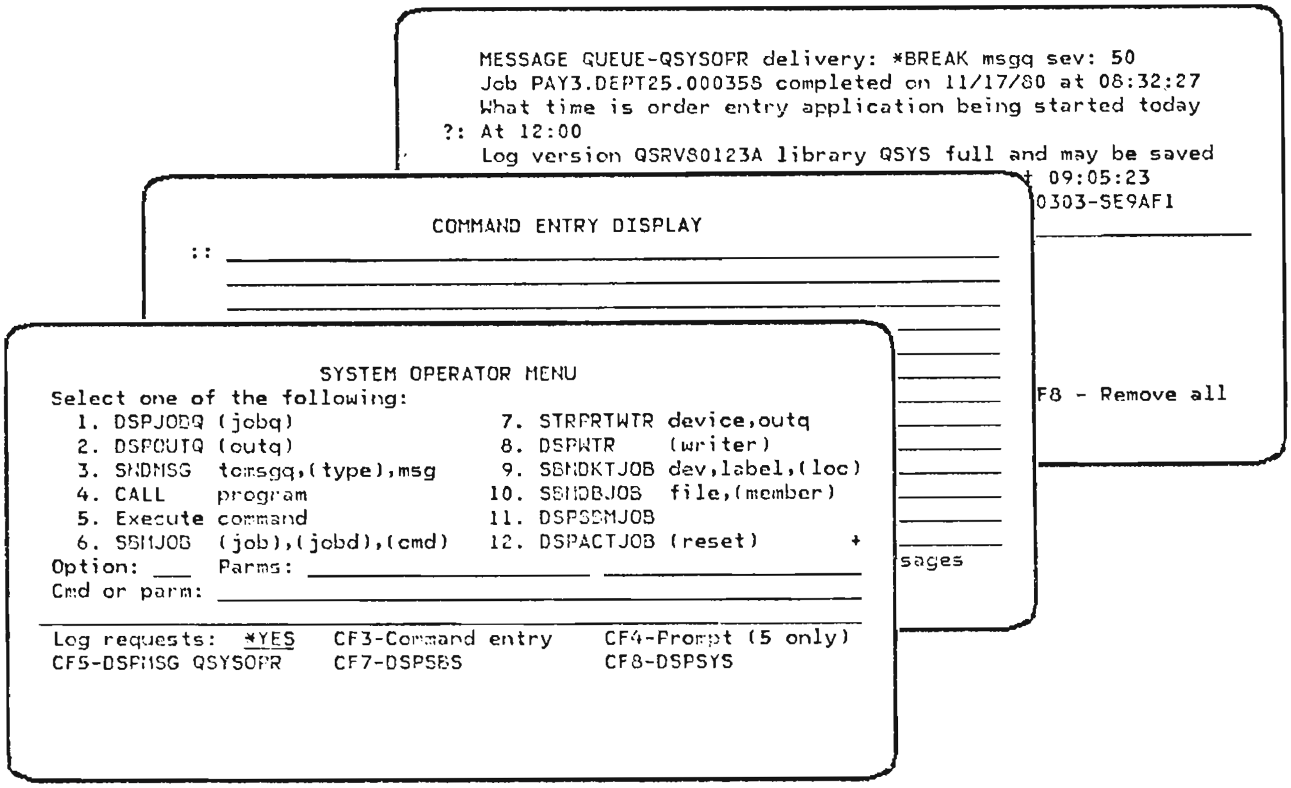
7. Correct the system time and date. This can be done by changing the system values QTIME and QDATE.

If the above procedure is not successful, call your service representative.

OPERATING ENVIRONMENT

After you have prepared the system for operation, you control it by entering commands and responding to messages at the system console or another work station. Your primary tools for controlling the system at a work station are the system operator menu, the command entry display, and a display of messages sent to your message queue (Figure 8-7).

You enter commands using the system operator menu or the command entry display. When you need assistance in finding and executing a command, you can request command selection menus and command parameter prompts from either the command entry display or the system operator menu. The command entry process is described in Chapter 6.



```
    MESSAGE QUEUE-QSYSOPR delivery: *BREAK msgq sev: 50
    Job PAY3.DEPT25.000358 completed on 11/17/80 at 08:32:27
    What time is order entry application being started today
 ?: At 12:00
    Log version QSRV80123A library QSYS full and may be saved
                                                t 09:05:23
                                                0303-SE9AF1

                                                F8 - Remove all
```

```
                    COMMAND ENTRY DISPLAY
 :: ____________________________________________________

                                                sages
```

```
                    SYSTEM OPERATOR MENU
Select one of the following:
  1. DSPJOBQ (jobq)                 7. STRPRTWTR device,outq
  2. DSPOUTQ (outq)                 8. DSPWTR    (writer)
  3. SNDMSG  tomsgq,(type),msg      9. SBMDKTJOB dev,label,(loc)
  4. CALL    program               10. SBMDBJOB  file,(member)
  5. Execute command               11. DSPSBMJOB
  6. SBMJOB  (job),(jobd),(cmd)    12. DSPACTJOB (reset)      +
Option: ___  Parms: __________________  __________________
Cmd or parm: _____________________________________________

Log requests:  *YES   CF3-Command entry    CF4-Prompt (5 only)
CF5-DSPMSG QSYSOPR    CF7-DSPSBS           CF8-DSPSYS
```

Figure 8-7. Your Interactive Tools

Work Stations and the System Operator Message Queue

Messages regarding system conditions are sent to a special repository known as the system operator message queue (QSYSOPR). If intervention is needed on a device, such as changing forms in a printer or diskettes in the diskette magazine drive, the system sends a message to QSYSOPR. If a user at a work station needs information about a system condition, that user can also send a message to QSYSOPR. When you request the messages in QSYSOPR, you receive a separate message display. This is the same display you receive before signing on when you start the system.

The system operator message queue, QSYSOPR, can be set to various delivery modes that determine how you receive messages. When you are starting the system, QSYSOPR is cleared of all old messages and set to the break mode so that you receive important messages automatically at the system console. (You can still obtain the old messages from the system history log, QHST.) If you want to control the system from another work station, you can reallocate (reassign) QSYSOPR so that the messages are automatically displayed at the other work station. This work station then becomes, in effect, the controlling work station instead of the system console.

Several work stations can be used to control various phases of system operations at the same time. However, the messages from QSYSOPR can be delivered automatically to only one of those work stations. QSYSOPR can be changed so that all authorized users can display the messages in it, but none of the users will be notified when an important message arrives. If a condition occurs that requires your intervention, the system will send a message to QSYSOPR and wait for a reply. System resources will be tied up until you or someone else replies to the message.

Messages, the system operator message queue, and message delivery modes are described in Chapter 11.

Controlling Batch Jobs

Depending on the needs of users of your system, your duties may include submitting jobs for batch processing. You can submit a batch job directly from a work station or by using spooling functions.

Batch jobs submitted from input devices other than work stations are handled through spooling programs that read the information into the system. Similarly, output from all types of jobs (even interactive jobs at work stations) that is to be produced on output devices other than work stations can be handled through spooling support that writes the information to the device. To submit a job to the system from a device or produce output files from a job on a device, you start the reader or writer program associated with the device.

Jobs submitted for batch processing, either from a work station or through a spooling reader, are placed on a job queue to wait for their turn to be processed (Figure 8-8). While they are on the queue, their associated data is stored on the job message queue as requests and in spooled inline files. As each job is processed, any resulting data to be produced on an output device is stored in spooled output files and the files are placed on output queues along with other jobs having similar output requirements. The order in which jobs are taken from job queues to be processed is determined by the scheduling priority specified for the job when it was submitted and the time of submission. Similarly, the order in which spooled files from a job are produced on an output device is determined by the output priority specified for the job and the time the job was entered into the system.

You can control job operations on the system by selectively holding and then releasing or canceling one or more of the following:

- An individual job
- Individual spooled output files for a job
- A reader transferring one or more jobs from an input device to a job queue
- A writer transferring one or more spooled files from an output queue to an output device
- A job queue containing all jobs for a particular kind of processing
- An output queue containing all spooled output files for a particular kind of output

You can also affect a job by moving its files to another output queue and/or changing its priority.

All of these functions, from submitting a job to controlling how it is handled on the system, are performed by entering commands.

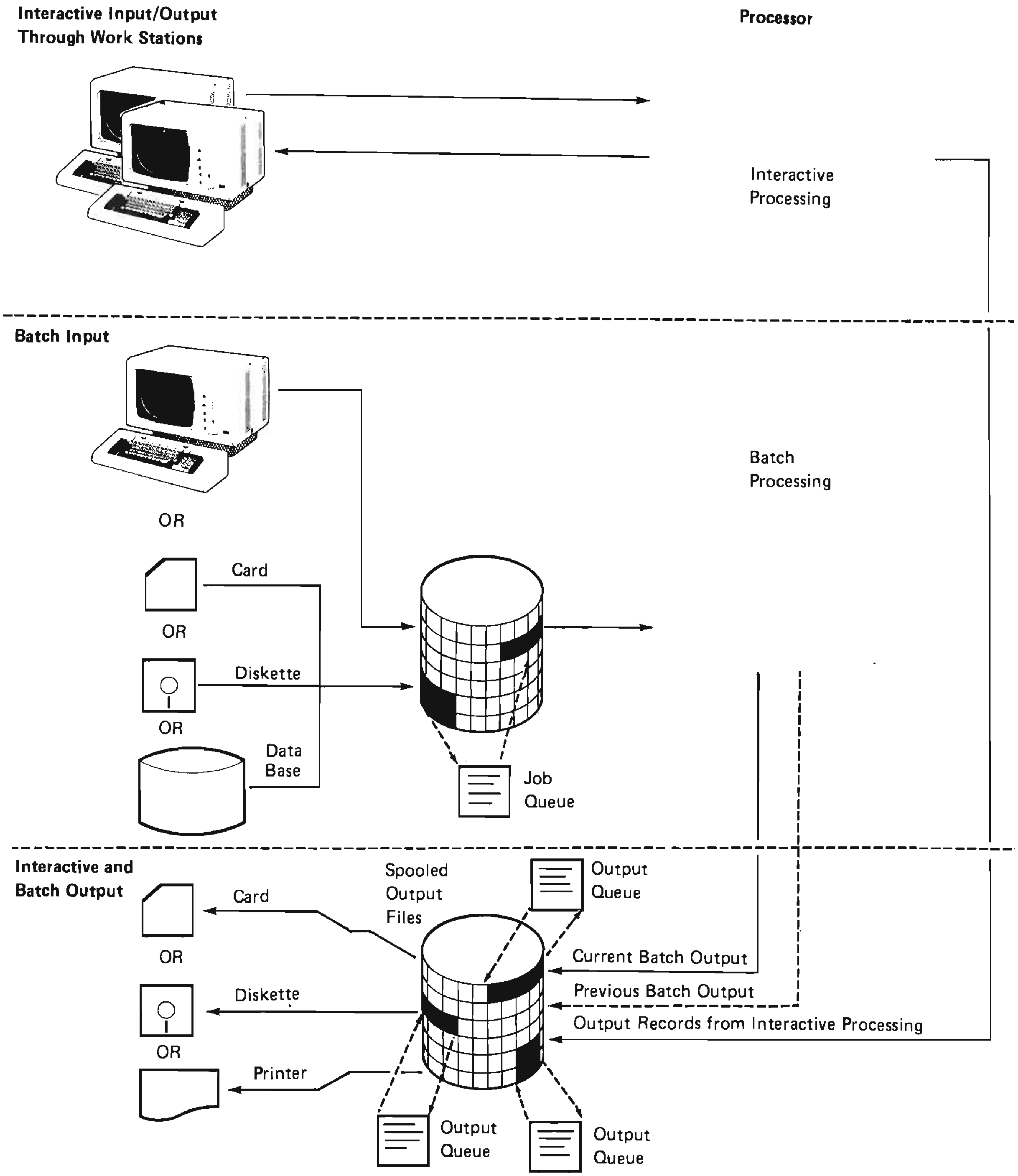


Figure 8-8. Input and Output for Interactive and Batch Processing

Grouping Commands to Simplify Your Job

Your duties may include the repeated use of the same commands, such as various spooling commands, or commands containing complex parameters that are time-consuming to enter. Those commands could be grouped into sets that are entered as a unit by one of the following methods:

- As an autostart job that is defined to start automatically when the subsystem is started

- As a batch job that you submit once at the beginning of a workday, or when needed

- As a CPF control language program that you call when needed by entering a single command

If you find that you are entering the same set of commands or complex commands repeatedly, you may want to suggest to your data processing manager that those commands be grouped by one of these methods.

Advantage of Batch Jobs

Generally, submitting work as a batch job is more efficient in terms of your time than calling a program, unless the program is short. When a batch job is submitted, it is processed independently of your interactive job, and therefore you can continue your work at the work station without waiting for the results. When you call a program, on the other hand, that program takes control of your interactive job. You cannot do other work, such as enter a command, until that program is completed.

Scheduling Jobs on the System

Even though the System/38 will process multiple jobs at the same time, there will still be scheduling restrictions imposed by users and impromptu work demands that dictate how your jobs are entered and processed on the system. Good operating practice will allow you to meet daily schedules, help provide time for impromptu work, and make the best use of system resources.

One important point in good operating practice is to not let a large number of active jobs accumulate on the system unnecessarily. A large number of jobs makes scheduling more difficult and may degrade the performance of certain functions.

SUBSYSTEMS

All work submitted to the system is processed under control of a *subsystem*. Each subsystem provides a specialized environment for handling a certain type of work or function, such as:

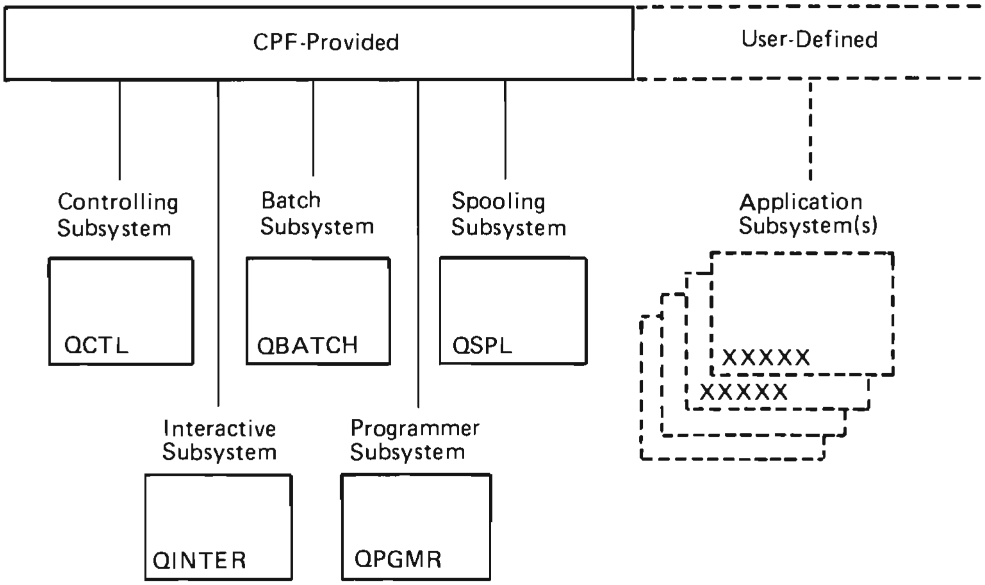
- Interactive processing where the system responds to each individual request of users at work stations.
- Batch processing where the system executes a set of instructions and data and provides information on the result.
- Spooling functions where information is transferred into the system from an input device or transferred out of the system to an output device.

Each subsystem is defined in the system by a *subsystem description* that specifies the environment the subsystem is to provide. You identify a subsystem to the system by supplying the name of its subsystem description. A subsystem description is a CPF object and is therefore located in a library like other objects.

A subsystem description is created, changed, or deleted by a CL command. This work is usually done by a programmer; however, you may occasionally need to change a particular characteristic of a subsystem.

Types of Subsystems

The CPF programming supplied by IBM includes subsystem descriptions that define five separate subsystems for handling the basic types of processing in the system. Your system may also have one or more application subsystems tailored for specific work environments.



The IBM-supplied QCTL subsystem description for the controlling subsystem is located in the system library (QSYS). All other IBM-supplied subsystem descriptions are located in the general purpose library (QGPL).

Controlling Subsystem (QCTL)

When the system is started, the controlling subsystem is started automatically to provide an interactive environment for communicating with the system. You are using the interactive environment provided by the controlling subsystem when you receive displays at the system console and use those displays to enter requests for actions by the system.

The controlling subsystem defined by the subsystem description QCTL that is supplied by IBM allows system operations to be started only through the system console, not other work stations. However, after the system has been started and one or more interactive subsystems other than QCTL have been started, any work station that is allocated (assigned) to the other subsystems can be placed under control of QCTL. You do this by signing on the work station and then entering the Transfer Job (TFRJOB) command as follows:

```
TFRJOB  JOBQ(QCTL)
```

In some situations it is advantageous to include at least one work station in the controlling subsystem description in addition to the system console. When the system is started, this additional work station then becomes immediately available as an alternative means of communicating with the system. For example, this work station could be used to operate the system if the system console is not functional (see the previous discussion of *Startup Without the System Console* in this chapter).

A programmer can create a subsystem description for an alternative controlling subsystem by using the procedures described in the *CPF Programmer's Guide*. To establish the new subsystem as the controlling subsystem, you use the Change System Value (CHGSYSVAL) command to specify the name of the new subsystem description in the system value QCTLSBSD (see *Changing System Values* in this chapter). For example, if the new subsystem description is QCTL2 and it is located in the library QGPL, you enter the command

```
CHGSYSVAL  SYSVAL(QCTLSBSD)  VALUE('QCTL2  QGPL')
```

The change does not become effective until the next IMPL.

Batch Subsystem (QBATCH)

The batch subsystem, as defined by the subsystem description QBATCH, is used for processing batch jobs in the system. The subsystem must be started before batch jobs can be processed; when it is terminated, batch processing stops.

Jobs can be submitted for batch processing when the QBATCH subsystem is not active, but the jobs will not be processed until the subsystem is started.

Spooling Subsystem (QSPL)

The spooling subsystem, as defined by the subsystem description QSPL, provides the processing environment for two types of system programs:

- Programs called readers that read input streams into the system from input devices or the data base

- Programs called writers that produce job output files on output devices

The subsystem must be started before a reader or writer can be used; when the subsystem is terminated, active readers and writers are terminated.

Requests to start the reader and writer programs can be submitted when the QSPL subsystem is not active, but the readers and writers will not be started until the subsystem is started. Details on the use of readers and writers are given in Chapter 10.

Interactive Subsystem (QINTER)

The interactive subsystem, as defined by the subsystem description QINTER, provides a general interactive environment for system users who communicate with the system through work stations. When the QINTER subsystem is started, all work stations that are not already allocated to another subsystem become available for use. This means that when the work station power is turned on (Power switch set to On) and any necessary communication link is established with the system, the sign-on prompt or another initial display appears on the work station screen.

As long as the subsystem remains active, a work station allocated to it will redisplay the sign-on prompt or another initial display each time a user signs off.

Programmer Subsystem (QPGMR)

The programmer subsystem, as defined in the subsystem description QPGMR, provides an interactive environment for online programming at work stations. After the QPGMR subsystem is started, a work station is allocated to the subsystem only when a user signs on the work station and enters the Transfer Job (TFRJOB) command in a form such as

```
TFRJOB JOBQ(QPGMR)
```

When the user signs off or the QPGMR subsystem is terminated, the work station returns to the control of the subsystem (such as QINTER) to which it was initially allocated.

Only users with a programmer or security officer user profile are allowed to transfer into the programmer subsystem.

Application Subsystems

If your system has additional subsystems designed for special working environments, each must be started and terminated as needed. The procedures for your system should indicate the conditions under which these subsystems are to be started and terminated.

Changing Subsystem Characteristics

As operating conditions change, you may need to change the basic capacity of a subsystem. You do this by using the Change Subsystem Description (CHGSBSD) command. The command can be entered while a subsystem is active, but particular parameters or values can be used only while the subsystem is inactive. Those parameters are noted in the following discussion.

When you enter the CHGSBSD command, you must indicate a specific subsystem description name in the SBSD parameter. There are also three optional parameters that allow you to change:

- Subsystem storage pools (POOLS parameter)

- Maximum number of jobs that can be concurrently active in the subsystem (MAXJOBS parameter)

- Descriptive text (TEXT parameter)

If you leave out any of these three parameters, that part of the subsystem description will remain the same as before (the default value for all three parameters is *SAME). If you do enter any of the parameters, the following describes how they are used.

POOLS Parameter: A subsystem can have up to 10 subsystem storage pools, each of which is used for executing specific jobs. Jobs executing at the same time in one storage pool compete for the use of the storage assigned to that pool, but do not compete for storage with jobs executing in other storage pools. Each storage pool has a specified activity level, which limits the number of jobs that can execute at the same time in the storage pool.

The POOLS parameter allows you to specify *one* of the following for each pool identifier:

- Identified storage pool is to have a specific storage size (in K-bytes) and specific activity level.

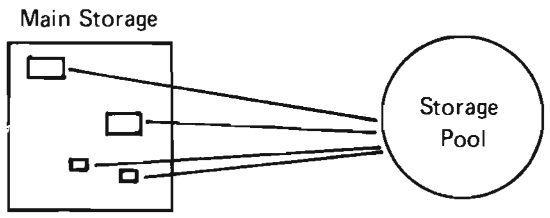
- Identified storage pool is to use the shared system storage pool (*BASE).

- Identified storage pool is to have no storage (*NOSTG) and no activity level assigned to it.

- Identified storage pool is to be removed (*RMV) from the subsystem description.

You can change the size and activity level of a storage pool while the subsystem is active, but you *cannot* add or remove a storage pool while the subsystem is active.

A Note on Storage Pools: When a subsystem is started, each of the storage pools defined in the subsystem description is assigned to a *system storage pool*. There are 16 system storage pools, each of which may consist of storage segments from different areas of main storage.



One of the 16 system storage pools is reserved for machine operations. A second, identified as the shared system storage pool (*BASE), contains all main storage that is not otherwise allocated. *BASE can be shared by several subsystems at one time. Each of the remaining 14 system storage pools can be assigned to only one subsystem, and to only one storage pool defined within that subsystem, at a time. There is a possibility, therefore, that a subsystem will be temporarily unable to obtain its defined storage pools because all of the system storage pools are currently assigned to other active subsystems. Jobs attempting to use defined storage pools for which storage could not be obtained will use the *BASE storage pool instead. When a subsystem is terminated, the system storage pools assigned to it become available for use by other subsystems.

MAXJOBS Parameter: This parameter allows you to set a specific limit on the number of jobs that can be processed in the subsystem at one time, or to specify that there is to be no limit (*NOMAX). For example, if you set a limit on the number of jobs in the QINTER subsystem, no additional jobs would be started as long as the number of existing jobs remains at or exceeds that limit. This could mean that a user is not allowed to sign on a work station until another user has signed off a work station.

TEXT Parameter: When a subsystem description is created, explanatory text about the subsystem may be included in the description. If you change a subsystem description, you may want to specify new text to replace any existing text or indicate that there is to be no text in the description by specifying *BLANK. The text cannot be changed while the subsystem is active.

Examples:

1. To change the QINTER subsystem description so that storage pool 2 is given a storage size of 225 K-bytes with an activity level of 3 and storage pool 3 uses the shared system pool (*BASE), enter

   ```
   CHGSBSD SBSD(QINTER) POOLS((2 225 3) (3 *BASE))
   ```

2. To change the subsystem description ORDER in library LIB6 so that storage pool 3 is removed from the subsystem and the subsystem is allowed to support a maximum of five jobs at one time, enter

   ```
   CHGSBSD SBSD(ORDER.LIB6) POOLS((3 *RMV)) MAXJOBS(5)
   ```

   Note that you cannot enter this command while the ORDER subsystem is active because you are removing a storage pool.

Starting and Terminating Subsystems

All subsystems must be started before jobs under their control can be processed. The controlling subsystem is started automatically when the system is started. You must start all other subsystems, unless your system is set up so that one or more other subsystems are started automatically through an autostart job in the controlling subsystem (see *Work Entries* in Chapter 12). When you terminate a subsystem, work submitted through that subsystem will no longer be processed.

The point in system operations where you start and terminate specific subsystems depends on the types of work being done on your system. Generally, if your system has a mixture of interactive and batch work, the interactive (QINTER), spooling (QSPL), and batch (QBATCH) subsystems should be started as soon as the system is started and should remain active as long as the system is active. If you find that a subsystem is seldom used, you may want to leave it inactive until it is needed. All active subsystems, including the controlling subsystem, are automatically terminated when the system is powered down by the Power Down System (PWRDWNSYS) command.

You can use the ability to start and terminate subsystems as a means of controlling the system workload. For example, if you expect a heavy interactive workload during a part of the day, you may want to temporarily terminate the batch subsystem to release the storage pools assigned to it or to eliminate its processor load.

Starting a Subsystem

To start a subsystem, enter the Start Subsystem (STRSBS) command and specify the name of its subsystem description in the SBSD parameter. You also have the option of adding a library qualifier to specify the library where the subsystem description is to be found.

Examples:

1. To start the batch subsystem, enter

```
STRSBS SBSD(QBATCH)
```

2. To start a subsystem named ORDER, whose description is located in LIB6, enter

```
STRSBS SBSD(ORDER.LIB6)
```

Terminating a Subsystem

To terminate a subsystem, enter the Terminate Subsystem (TRMSBS) command and indicate either *ALL or a specific subsystem name in the SBS parameter. Generally, you can terminate a subsystem from a work station that is allocated (assigned) to any subsystem. However, you can terminate the controlling subsystem or all subsystems only if you enter the TRMSBS command from a work station that is allocated to the controlling subsystem. In addition, the work station must have been allocated to the controlling subsystem at sign-on; that is, the command is not accepted if your work station was allocated to another subsystem when you signed on and you later transferred into the controlling subsystem by entering the Transfer Job (TFRJOB) command (see the previous discussion of the *Controlling Subsystem* in this chapter). These restrictions are intended to prevent the controlling subsystem from being terminated in such a way that no work station can communicate with the system.

If you indicate *ALL for the SBS parameter, all subsystems, including the controlling subsystem, will be terminated, except that *your* interactive job will remain active; that is, the interactive sessions will be ended at all work stations except the work station where you entered the command (see *Terminating All Subsystems* in this chapter).

The OPTION and DELAY parameters determine the conditions under which the subsystems are terminated:

- If you do not enter either parameter, all jobs being processed are allowed to complete their current phase of processing (called a routing step) before they are terminated and the subsystem specified in the SBS parameter is terminated. Interactive jobs that are on a job queue due to a TFRJOB command will be canceled as part of subsystem termination. If an IMPL occurs while either a batch or interactive job is on a job queue due to TFRJOB, that job will be removed from the job queue during IMPL and its job log will be produced.

- If you do not enter the OPTION parameter, but enter the DELAY parameter with a specific number as the value, such as DELAY(60) the jobs being processed are allowed that number of seconds to complete before they are canceled and the subsystem terminated.

- If you do not enter the DELAY parameter, but instead enter the OPTION parameter in the form OPTION(*IMMED) the jobs being processed are canceled and the subsystem terminated as soon as the cancellation process is completed.

CAUTION
Use OPTION(*IMMED) only after you have tried unsuccessfully to terminate the subsystem with a time delay (DELAY parameter). An immediate termination could leave files in a partially updated state.

Examples:

1. To terminate the batch subsystem in a controlled manner, allowing all currently running batch jobs to complete their current phase of processing, enter

   ```
   TRMSBS SBS(QBATCH)
   ```

   Any jobs on the queue that had not been started will remain on the queue, but will not be processed until the subsystem is started again.

2. To terminate a subsystem named ORDER in a controlled manner, allowing subsystem jobs 120 seconds to terminate themselves before they are canceled, enter

   ```
   TRMSBS SBS(ORDER) DELAY(120)
   ```

SYSTEM VALUES

System values are variable quantities that affect the operation of CPF. Generally, they set limits on specific system functions or define the way the system will react to specific conditions. You can display or change a system value using a command. Each system value has a name to identify it in a command.

For information on a specific system value, refer to *CPF Programmer's Guide.*

Displaying System Values

You can determine the current value of a particular system value by requesting the system value display.

Obtaining the Display

To obtain the system value display, enter the Display System Value (DSPSYSVAL) command and specify the name of the desired system value in the SYSVAL parameter. The only other parameter is the OUTPUT parameter, which allows two options:

- If you do not enter the OUTPUT parameter, the information is displayed on your work station screen.
- If you specify OUTPUT(*LIST) the information is spooled and placed on an output queue for printing on the system printer.

Examples:

1. To display the system value QBASPOOL on your work station screen, enter

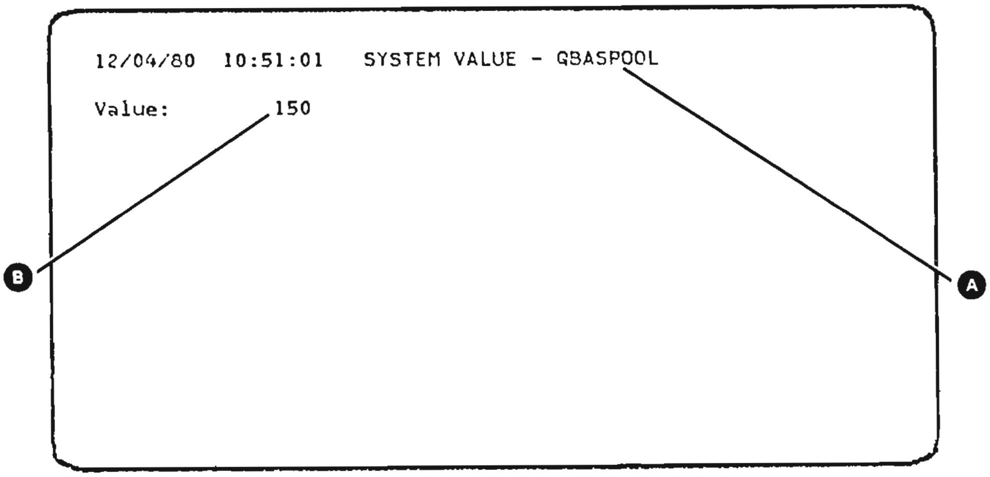
```
DSPSYSVAL  SYSVAL(QBASPOOL)
```

2. To print out the system value QTIME, enter

```
DSPSYSVAL  SYSVAL(QTIME)  OUTPUT(*LIST)
```

Interpreting the Display

When you enter the DSPSYSVAL command, you receive a display of the form:



```
12/04/80  10:51:01   SYSTEM VALUE - QBASPOOL

Value:          150
```

A *Name*: This is the name of the system value you requested.

B *Value*: This is the current value.

Pressing the Enter key returns you to the display from which you requested the system value display.

Changing System Values

You change a system value either through the configuration menu when you start CPF or by entering the Change System Value (CHGSYSVAL) command after CPF has been started.

Using the CHGSYSVAL Command

The CHGSYSVAL command has two parameters that must be included when the command is used. When you enter the command:

1. Specify the name of the system value in the SYSVAL parameter.

2. Specify the new value in the VALUE parameter. If the new value consists of more than one item separated by a space (blank), you must enclose the value in apostrophes, such as

   ```
   VALUE('QCTL2 QGPL')
   ```

   You must also enclose the value in apostrophes if the system value is defined as a character and you want to specify a number, such as

   ```
   VALUE('12')
   ```

   If you are not sure whether a system value is defined as a character or a number, refer to the description of system values in the *CPF Programmer's Guide.*

Examples:

1. To change the system value QMONTH (which is a character value) to 12, enter

   ```
   CHGSYSVAL SYSVAL(QMONTH) VALUE('12')
   ```

   Note in this example that QMONTH is only part of the system value QDATE. The remainder of QDATE (QYEAR and QDAY) will remain unchanged.

2. To change the system value QBASPOOL (a numeric value) such that the minimum size of the *BASE storage pool is increased to 200 K-bytes, enter

   ```
   CHGSYSVAL  SYSVAL(QBASPOOL)  VALUE(200)
   ```

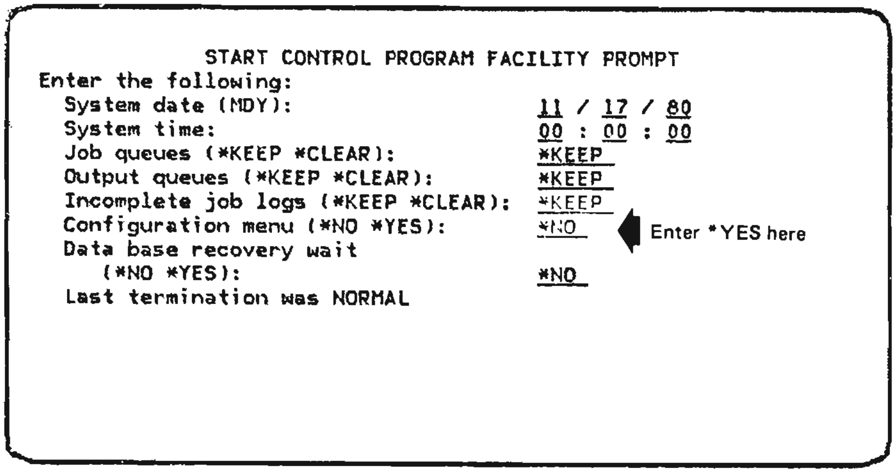
3. To change the system value QCTLSBSD so that QCTL2, which is located in library QGPL, becomes the controlling subsystem instead of QCTL the next time the system is started, enter

   ```
   CHGSYSVAL SYSVAL(QCTLSBSD) VALUE('QCTL2 QGPL')
   ```

Using the Configuration Menu

During the start CPF process, you can change a system value using the configuration menu. The procedure is:

1. Enter *YES on the *Configuration menu* line of the start CPF prompt.

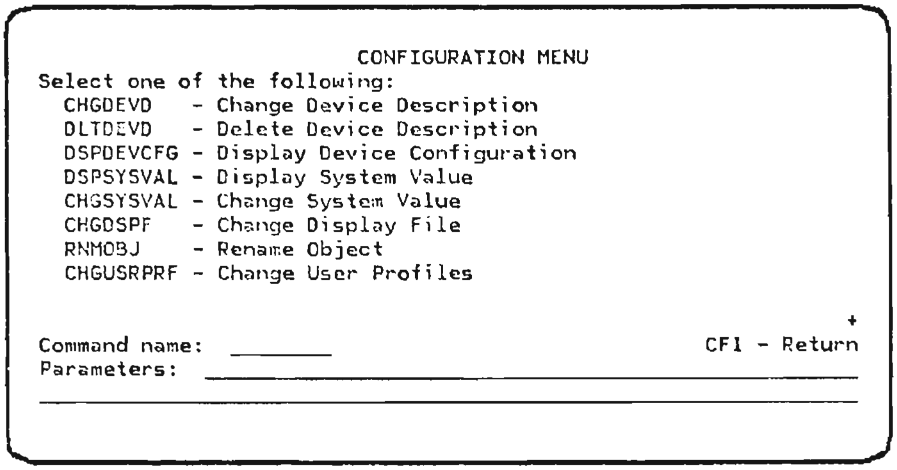


```
                 START CONTROL PROGRAM FACILITY PROMPT
Enter the following:
  System date (MDY):                        11 / 17 / 80
  System time:                              00 : 00 : 00
  Job queues (*KEEP *CLEAR):                *KEEP
  Output queues (*KEEP *CLEAR):             *KEEP
  Incomplete job logs (*KEEP *CLEAR):       *KEEP
  Configuration menu (*NO *YES):            *NO
  Data base recovery wait
     (*NO *YES):                            *NO
  Last termination was NORMAL
```

Enter *YES here

The system responds with the configuration menu.

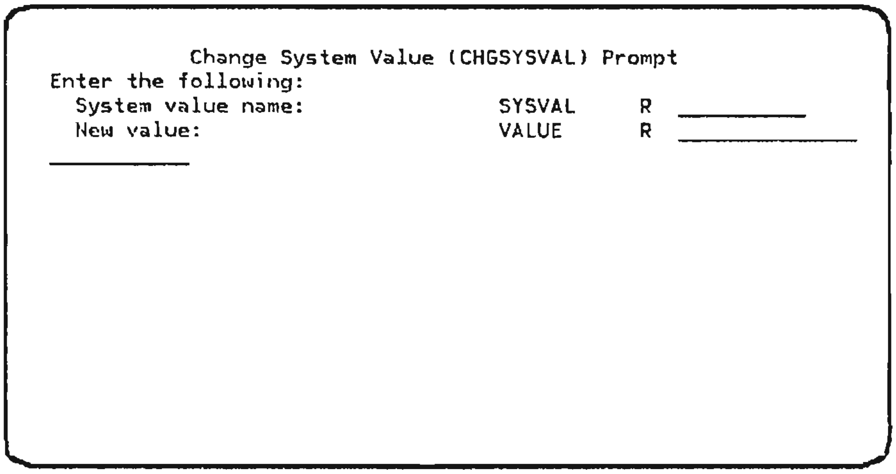
2. Key in the complete CHGSYSVAL command and press the Enter key or just key in CHGSYSVAL and press the CF4 key.



```
                         CONFIGURATION MENU
Select one of the following:
  CHGDEVD    - Change Device Description
  DLTDEVD    - Delete Device Description
  DSPDEVCFG  - Display Device Configuration
  DSPSYSVAL  - Display System Value
  CHGSYSVAL  - Change System Value
  CHGDSPF    - Change Display File
  RNMOBJ     - Rename Object
  CHGUSRPRF  - Change User Profiles

                                                                 +
Command name:  ________                                CF1 - Return
Parameters:  ______________________________________________________
___________________________________________________________________
```

You can use the input fields provided at the bottom of any part of the menu to enter the complete command.

If you press the CF4 key, the system responds with the prompt for the CHGSYSVAL command.



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

3. Key in the system value name and the new value where indicated by the underlined input fields on the prompt, and press the Enter key to execute the command.

If you decide you do not want to change the system value, you can press the CF1 key at any time prior to pressing the Enter key to execute the CHGSYSVAL command. You will return to the configuration menu without the system value being changed.

OBTAINING INFORMATION ABOUT THE SYSTEM

Much of the information you need about system operations is sent to your work station in the form of messages (see Chapter 11). If you need additional information, there are a variety of display commands you can use to obtain that information. The following topics summarize the types of information you are likely to need and the commands you use to obtain the information.

Auxiliary Storage Usage/Current System Conditions/Subsystem Conditions: Use the DSPSYS, DSPSYSSTS, or DSPSBS command (see *Determining System and Subsystem Status* in this chapter).

Previous System Conditions: Use the DSPLOG command (see *System Logs* in this chapter).

Configuration of System Equipment: Use the DSPDEVCFG command (see *Determining System Configuration* in Chapter 9).

Jobs/Files on Spooling Queues: Use the DSPJOBQ or DSPOUTQ command (see *Determlning Queue Status* in Chapter 10).

Information on Individual Jobs: Use the DSPJOB command (see *Determining Job Status* in Chapter 12) or the DSPSPLFA command (see *Determining Output Characteristics* in Chapter 10).

Diskette Contents: Use the DSPDKT command (see *Determining Diskette Contents* in Chapter 13).

Tape Contents: Use the DSPTAP command (see *Determining Tape Contents* in Chapter 13).

Readers and Writers: Use DSPRDR or DSPWTR commands (see *Determining Reader/Writer Status* in Chapter 10).

Communications Status: Use the DSPLINSTS, DSPCUDSTS, and DSPDEVSTS commands (see *Handling Communications Lines* in Chapter 9).

DETERMINING SYSTEM AND SUBSYSTEM STATUS

There are three general displays that provide basic information about the condition of the system and subsystems in the system. Each is requested by a command.

System Display

The system display shows which subsystems are active or in the process of terminating, and the amount of system storage each is using.

Obtaining the Display

To obtain the system display, press the CF8 key from the system operator menu or enter the Display System (DSPSYS) command. The command has only one parameter, the OUTPUT parameter, which gives you two options:

1. If you enter

   ```
   DSPSYS
   ```

   the display appears on your work station screen.

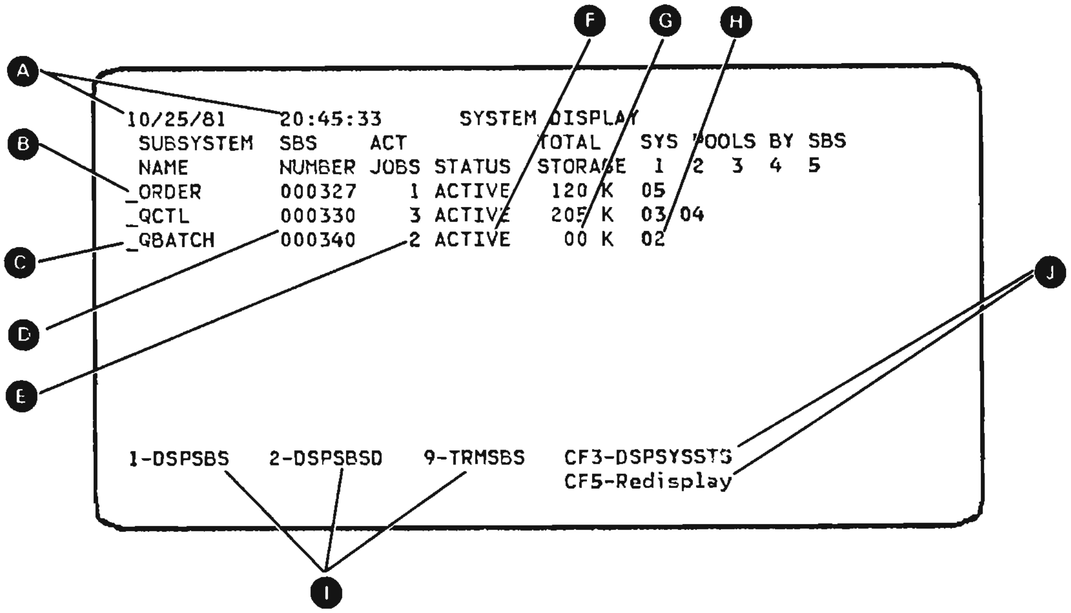
2. If you enter

   ```
   DSPSYS OUTPUT(*LIST)
   ```

   the information is spooled and placed on an output queue for printing on the system printer.

Interpreting the Display

When you enter the DSPSYS command, you receive a display of the form:



```
10/25/81      20:45:33      SYSTEM DISPLAY
 SUBSYSTEM    SBS    ACT                 TOTAL    SYS  POOLS BY SBS
 NAME         NUMBER JOBS STATUS         STORAGE   1   2   3   4   5
_ORDER        000327    1 ACTIVE          120 K   05
_QCTL         000330    3 ACTIVE          205 K   03  04
_QBATCH       000340    2 ACTIVE           00 K   02




1-DSPSBS   2-DSPSBSD   9-TRMSBS     CF3-DSPSYSSTS
                                    CF5-Redisplay
```

A *Date/Time*: The date and time shown are the job date and time when you requested the display.

B *Input Field*: A one-character input field appears in front of each subsystem name. The options described under I can be entered here.

C *Subsystem*: All subsystems currently active in the system are listed. If a subsystem name does not appear, you know it either has not been started since the system was started or was terminated.

D *SBSNBR*: This is the number that is assigned to each subsystem when it is started. This number is significant only for servicing purposes.

E *Jobs*: This is the number of jobs currently being processed in each subsystem. If two interactive jobs were created at a work station (see *Using the System Request Menu* in Chapter 6), the job that is currently suspended is not included in this count.

F *Status*: The status shown will be either ACTIVE, meaning that the subsystem is currently active, or TRM, meaning that the subsystem is in the process of terminating, or RSTD, meaning that the controlling subsystem (QCTL) is in the restricted state. The controlling subsystem may be brought into a *restricted* state by:

- Terminating the controlling subsystem with the TRMSBS SBS(*QCTL) command

- Terminating all subsystems with the TRMSBS SBS(*ALL) command

- Terminating CPF with the TRMCPF command

- Initiating an IMPL under the QCE user profile (Concurrent Service Monitor)

- Failing to be allocated sufficient storage during IMPL

  During the restricted state, the controlling subsystem is the only active subsystem, with only one active work station (which can be the system console.

  If you sign off the active work station, the sign-on prompt will appear; if you sign on, the controlling subsystem remains in the restricted state.

  To exit the restricted state, you must restart the controlling subsystem.

G *Tot Stg*: The number shown is the amount of storage (in K-bytes) obtained by each subsystem. This number does not include the storage in *BASE, however, if the subsystem is using that storage pool. Therefore, if a subsystem description specifies only *BASE in the POOLS parameter, a 0 will appear here.

H *System Stg Pools*: The numbers shown identify system storage pools assigned to each subsystem. These numbers are *not* the pool identifiers of subsystem storage pools that appear in a subsystem description. In the example display shown, system storage pool 2 (*BASE) is assigned to the QBATCH subsystem, pools 3 and 4 are assigned to the QCTL subsystem, and pool 5 is assigned to the ORDER subsystem. Information on system storage pools is shown on the system status display.

🅘 *Options*: The values that can be specified in the input field to perform the following:

1 DSPSBS: Displays information about a subsystem.

2 DSPSBSD: Displays the subsystem description.

9 TRMSBS: Terminates a subsystem.

🅙 *CF Keys*:

CF1: Return to system operator display

CF3: Display system status (DSPSYSSTS)

CF5: Redisplay with current values

Pressing the Enter key returns you to the display from which you requested the system display.

System Status Display

The system status display provides information on the status of the system and system storage pools.

Obtaining the Display

To obtain the system status display, enter the Display System Status (DSPSYSSTS) command. The OUTPUT parameter, the only parameter of the command, gives you two options:

1. If you enter

   DSPSYSSTS

   the display appears on your work station screen.

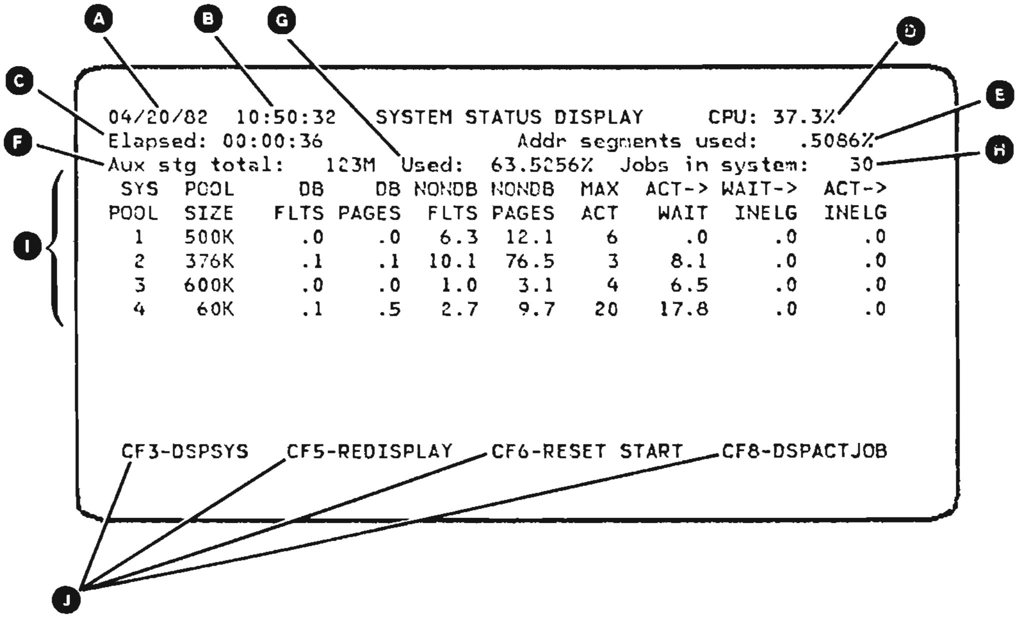
2. If you enter

   DSPSYSSTS OUTPUT(*LIST)

   the information is spooled and placed on an output queue for printing on the system printer.

Interpreting the Display

When you enter the DSPSYSSTS command, you receive a display of the form:



```
04/20/82  10:50:32   SYSTEM STATUS DISPLAY      CPU: 37.3%
Elapsed: 00:00:36                    Addr segments used:   .5086%
Aux stg total:    123M  Used:  63.5256%  Jobs in system:    30
 SYS  POOL      DB     DB NONDB NONDB  MAX  ACT-> WAIT->  ACT->
POOL  SIZE    FLTS PAGES  FLTS PAGES  ACT   WAIT  INELG  INELG
  1   500K      .0    .0   6.3  12.1    6     .0     .0     .0
  2   376K      .1    .1  10.1  76.5    3    8.1     .0     .0
  3   600K      .0    .0   1.0   3.1    4    6.5     .0     .0
  4    60K      .1    .5   2.7   9.7   20   17.8     .0     .0




 CF3-DSPSYS   CF5-REDISPLAY   CF6-RESET START   CF8-DSPACTJOB
```

In addition to general information on system status, this display provides statistics on the utilization of the processor and the transfer of data from auxiliary storage to main storage. Data is transferred between auxiliary storage (which contains the data base) and main storage in units of 512 bytes, known as *pages*. The items identified as pages on the display refer to these units of 512 bytes.

Ⓐ *Date*: The date shown is the current system date.

Ⓑ *Time*: The time shown is the current system time.

Ⓒ *Elapsed*: The value shown is the elapsed time between the measurement start time Ⓑ and the current system time shown.

Ⓓ *CPU Use*: The value is the percentage of time that the CPU was used during the measurement interval.

Ⓔ *Addr segments used*: The percentage of generated addresses that have been used is shown here. When 95% of the addresses have been used the system will notify you at IMPL time. However, address regeneration can be done any time you feel it is necessary. (See *Address Regeneration* in Chapter 1.)

Ⓕ *Aux stg total*: This is the total auxiliary storage on the system.

Ⓖ *Used*: The value shown is the percentage of the total auxiliary storage that is currently in use. Use this display to monitor auxiliary storage usage. If the percentage of auxiliary storage used seems higher than normal, you should consider the following causes:

- History logs have been allowed to accumulate because they were not saved.

- Excessive number of output files have accumulated on output queues.

- The system has not been IMPLed for a long period and temporary objects no longer needed have accumulated on the system.

The system value QAUXSTGTH contains the auxiliary storage threshold value. When the value of QAUXSTGTH is exceeded, a message will be sent to the QSYSOPR message queue. If you do not reduce the storage used or raise the threshold value within 60 minutes, the message is sent again.

CAUTION
If you choose to raise the threshold value you should follow the guidelines in the *CPF Programmer's Guide*.

Ⓗ *Jobs in System*: The total number of user and system jobs currently in the system is shown. The total includes:

- All jobs on job queues waiting to be processed
- All jobs currently active (being processed)
- All jobs with output on output queues to be produced

Ⓘ *Storage Pool Statistics*: The remainder of the display provides statistics on the utilization of each of the 16 *system* storage pools that currently has storage allocated to it. The information is grouped under column headings as follows:

- Sys Pool: The pool identifier for the system storage pool. The pool identifiers in this display are the same as those given in the system display. The system display identifies which subsystem (if any) the storage pools are currently assigned to. These identifiers are *not* the same as those specified in the subsystem description (see *Subsystems* in this chapter).
- Pool Size: The amount of storage, in K-bytes, in the storage pool.
- DB Faults: The rate, in page faults per second, of page faults against pages containing either data base data or access paths. (An access path is the means by which CPF provides a logical organization to the data in a data base file.)
- DB Pages: The rate, in pages per second, at which data base pages are brought into the storage pool.
- Non-DB Faults: The rate, in page faults per second, of page faults against pages other than those designated as data base pages.
- Non-DB Pages: The rate, in pages per second, at which non-data base pages are brought into the storage pool.
- Max Act: The maximum activity level; that is, maximum number of jobs that can be active in the pool at any time.
- Act->Wait: The rate, in transitions per minute, of transitions of jobs from an active state to a waiting state.
- Wait->Inelg: The rate, in transitions per minute, of transitions of jobs from a waiting state to an ineligible state.
- Act->Inelg: The rate, in transitions per minute, of transitions of jobs from an active state to an ineligible state.

J *CF Keys:*

CF3: Display system

CF5: Redisplay with updated values

CF6: Start a new time period to monitor elapsed time and CPU use

CF8: Display all active jobs on the system

You can update the statistics on this display by pressing the CF5 key; this causes the measurement interval to be extended, and more data is gathered during that time. You can reset the elapsed time and initiate a new measurement interval by pressing the CF6 key. Using the CF3 key, you can go directly to the system display; by again pressing the CF3 key, you can reshow the system status display with updated statistics.

The data is gathered and stored in counters. Because these counters can overflow (wrap around), it is advisable to use the CF6 key to perform occasional resets (setting the counters to zero). Otherwise, if wraparound does occur in a counter, it may appear that very few faults have occurred in the time interval, instead of many.

Also, note that the data used to mark the start of the measurement interval is established when you first request the display in your interactive job; that is, the first time you request the display after signing on. Therefore, the statistics that appear the first time you request the display are not significant. To obtain accurate statistics, allow several minutes to elapse and request the display again.

If there are more statistics to be displayed than can be shown on your screen at one time, a + will appear in the lower right corner of the display. To obtain the additional information, press the Roll Up key. You can return to the display from which you requested the system status display by pressing the Enter key.

For information on using the statistics in this display to evaluate system performance, refer to the *CPF Programmer's Guide.*

Subsystem Display

The subsystem display shows the status of jobs in all active subsystems or a specified subsystem, including jobs on spooling queues.

Obtaining the Display

To obtain the subsystem display, enter the Display Subsystem (DSPSBS) command. The command has three parameters, each of which offers two options. When you press the CF7 key on the system operator menu, you are using this command in the form

```
DSPSBS SBS(*ALL)
```

SBS Parameter: This parameter determines what subsystems are displayed.

- If you do not enter the parameter (which means you accept the default value *ALL), information is provided on all subsystems and jobs on spooling queues.

- If you enter the parameter and specify a subsystem name, such as SBS(QBATCH) information is provided only for that subsystem.

USER Parameter: This parameter determines whether information is provided on the jobs of all users or a specific user.

- If you do not enter the parameter USER(*ALL) is assumed, information is provided on all user jobs in the subsystem or subsystems specified in the SBS parameter.

- If you enter the parameter and specify a user name, such as USER(QSYSOPR) information is provided only on jobs for that user.

OUTPUT Parameter: This parameter determines how the information is provided.

- If you do not enter the parameter, the information is displayed on your work station screen.
- If you specify OUTPUT(*LIST) the information is spooled and placed on an output queue for printing on the system printer.

Examples:

1. To display at your work station the status of all jobs on the system, enter

   ```
   DSPSBS
   ```

2. To display at your work station the status of the controlling subsystem QCTL, enter

   ```
   DSPSBS SBS(QCTL)
   ```

3. To display at your work station the status of the jobs for the user DEPT25 in the batch subsystem, enter

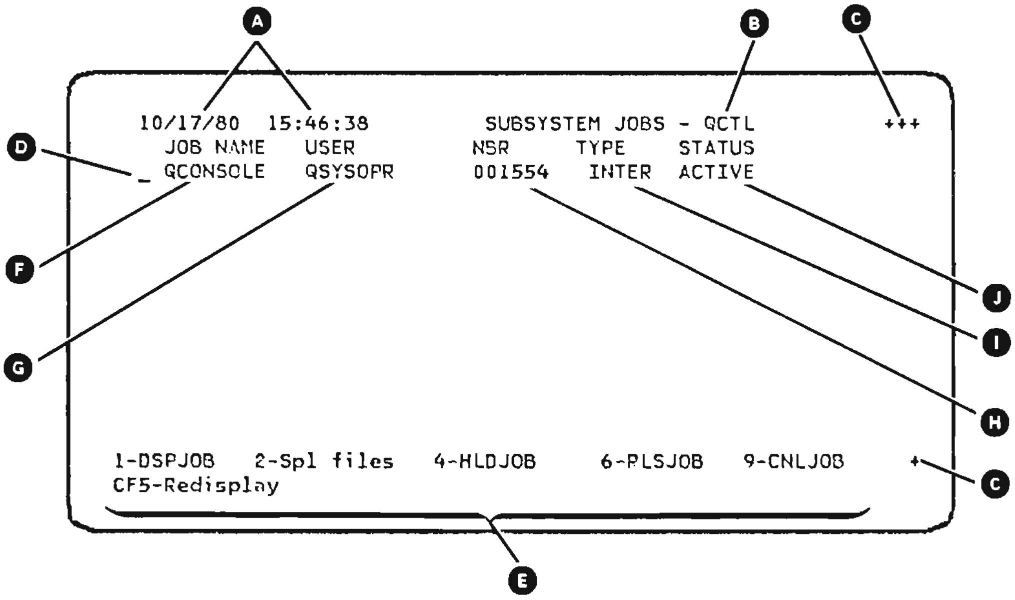
   ```
   DSPSBS SBS(QBATCH) USER(DEPT25)
   ```

4. To print out the status of all jobs for the users who signed on with the QPGMR user profile, enter

   ```
   DSPSBS USER(QPGMR) OUTPUT(*LIST)
   ```

Interpreting the Display

When you enter the DSPSBS command, you receive a display of the form:



```
    10/17/80  15:46:38         SUBSYSTEM JOBS - QCTL          +++
      JOB NAME   USER         NSR      TYPE    STATUS
    _ QCONSOLE   QSYSOPR      001554   INTER  ACTIVE




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

A *Date/Time*: The date and time shown are the job date and system time when you requested the display.

B *Subsystem Name*: The name of the subsystem being displayed. If all subsystems are being displayed, each subsystem will have its own display.

C *Additional Display Indicator*: +++ appears if there are more subystems to be displayed; press the Enter key to display more subsystems. + appears at the lower right corner if there are more jobs than will fit on the display; press the Roll key to display additional jobs.

D *Input Field*: The options shown under E are entered here.

E *Options:* The values can be specified in the input field to perform the following:

1 DSPJOB: Displays information about the job.

2 Spl files: Displays information about the spooled files that exist for the job.

4 HLDJOB: Holds the designated job but do not hold spool files produced by that job.

6 RLSJOB: Releases the designated job. The job must have been previously held.

9 CNLJOB: Cancels the job, but do not cancel the spool files produced by that job.

F *Job Name:* The name shown depends on whether the job is an interactive job, a batch job, an autostart job, or a spooling function. (An autostart job is one that starts automatically when a subsystem starts.)

- Interactive Jobs: The name of the work station on which the interactive job was created when a user signed on. If two interactive jobs were created at the same work station (see *Using the System Request Menu* in Chapter 6), two separate entries with the same work station name will be shown on the display.

- Batch Jobs: The name by which the job was identified to the system when it was submitted, such as the name specified in the Job command that indicates the beginning of a batch input stream, or the name specified in the Submit Job command that places a job on a job queue (see Chapter 12).

- Autostart Jobs: The name provided when the autostart job entry was defined in the subsystem description (see *Job Characteristics* in Chapter 12).

- Spooling Readers and Writers: The name of the reader or writer. This may have been defaulted to the device name when the reader or writer was started.

G *User*: The name shown is the name of the user who submitted the job. It depends on the type of job.

- Interactive Jobs: The user profile name associated with the password of the user who signed on the work station.

- Batch Jobs: The name defined in the Job or Submit Job command when the job was submitted.

- Autostart Job: The name provided when the autostart job entry was defined in the subsystem description.

- Spooling Readers and Writers: This will always be QSYS because all readers and writers run under QSYS profile.

H *Nbr*: The number shown is the job number that is assigned to each job when it enters the system.

I *Type*: This is the type of job: INTER (Interactive), BATCH, AUTO (autostart job), RDR, and WTR

J *Status:* The status of the job is shown in two columns. The first column indicates:

- ACTIVE: The job is active.

- JOBQ: The job is on a job queue waiting to be processed.

- OUTQ: The output from a job is on an output queue waiting to be produced.

- TFRJOB: The job has been initiated but is currently residing on a job queue as a result of a TFRJOB command.

- SYSREQ: The interactive job was suspended through the use of the system request key.

- CANCEL: The job has been canceled because of cancel job, or terminate subsystem immediately.

The second column indicates:

- HELD: The job is being held. The HELD for an OUTQ job means that output is held.

- Blank: The job is not being held.

Both columns can indicate:

- JOBLOG PENDING: If the job log has not yet been written, this indication will appear until the job log can be written.

- FIN: The job has not properly completed its termination. It will be removed the next time the system is started or may be removed by the CNLJOB command.

If you specified SBS(*ALL) or used the CF7 key from the system operator menu, pressing the Enter key will display the next subsystem. If all subsystems have been displayed, pressing the Enter key returns you to the display from which you requested the subsystem display.

Pressing the CF5 key causes the display to be updated with current information, just as if the command were reissued.

SYSTEM LOGS

In addition to determining the current status of the system, you will frequently need to determine what happened on the system previously. System logs contain that information. There are three system logs:

- *History Log (QHST)*: The history log provides a high-level audit trail of activities performed on the system. You can use it to review system activities and obtain information for use in recovering from system failures.

- *Service Log (QSRV)*: The service log provides a record of errors detected in system programming. Service personnel can use this information when servicing the system.

- *Programming Change Log (QCHG)*: The programming change log provides a record of all programming changes made on the system. You can use it to determine what programming changes were made and when.

Log Versions

How Information is Logged

As each event occurs that is to be recorded in a log, information about that event is sent to the log in the form of a message. The messages are then written to a data base file in the order they arrive. These messages thus form a chronological history of significant system events.

Because messages are continually being logged, the system logs could eventually become so large that they occupy a considerable amount of system storage. The total history, service, and programming change log files, therefore, are divided into smaller files called *versions*. Each version is restricted to a specific number of records by a system value. The system values that specify the maximum number of records of the history, service, and programming change logs are QHSTLOGSIZ, QSRVLOGSIZ, and QCHGLOGSIZ, respectively.

When a version of a log reaches its maximum size:

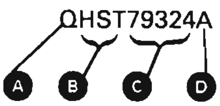
- A new version of the log file is automatically created and logging continues with entries going into the new version.

- You receive a message that the log version is full and should be saved.

To prevent the previous versions of the system logs from accumulating in the data base and occupying needed storage, you should save them by using the Save Object (SAVOBJ) command and then delete them. (If you do not have the authority to delete files, have someone delete the logs for you.) This makes the space in the data base occupied by the logged information available for other use. If the information is needed again, you can return it to the data base by using the Restore Object (RSTOBJ) command. These commands are described in Chapter 13. A system log version can be saved on a single diskette if the size of the version has not been increased since the system was shipped from IBM. (See example 2 under *Saving Objects* in Chapter 13.)

Identifying Log Versions

When each log version is created, it is given a unique identifying name, such as



A Indicates a system file

B Indicates the log type, either HST (history), SRV (service), or CHG (programming change)

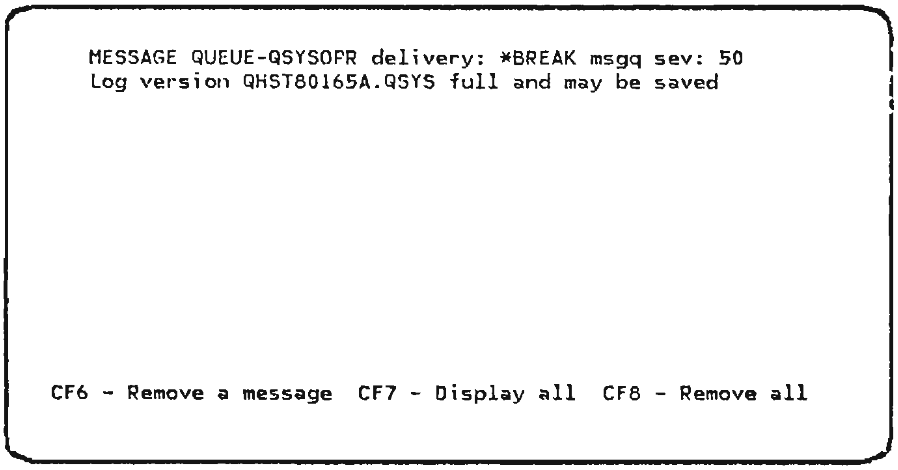
C Indicates the Julian date on which the log version was created. In the example shown, the log version was created on the three hundred and twenty-fourth day of 1979.

D Indicates a sequential character on the Julian date (in case more than one version was created on the same date). The character could be any number from 0 through 9 or any letter from A through Z.

You can use this name when you refer to a log version in a command, such as when you are saving it or restoring it.

Saving Log Versions

Each time a system log version becomes full, a message notifying you that it is full is sent to the system operator message queue (see *Message Queues* in Chapter 11). The following is an example of such a message as it might appear on your work station screen:



```
    MESSAGE QUEUE-QSYSOPR delivery: *BREAK msgq sev: 50
    Log version QHST80165A.QSYS full and may be saved










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

When you receive a message that a log version is full, you should use the system save/restore facilities to save the version on diskette or tape and remove it from the data base so that it does not occupy storage needed for other purposes. Because the log version file is an object, you can save it by using the Save Object (SAVOBJ) command. The object name that you specify (OBJ parameter of the command) is the name of the log version.

For example, if you wanted to save the history log QHST7934A on a diskette name HIST1, in slot 1, you would use the following command.

```
SAVOBJ OBJ(QHST7934A) DEV(QDKT) LOC(*S1)
   VOL(HIST1)
```

If you need to refer to the log information after it has been removed from the system by the Save Object command, you can use the *Restore Object* command to return the log version to the system see *Restoring Objects* in Chapter 13. Periodically, old versions of the system logs should be purged from the system by using the Delete File command.

Note: If a Save System (SAVSYS) command is issued, and some of the system logs have been saved with their storage freed, a message will be issued. The message can be ignored if the only objects not saved are system logs. This message can be avoided by deleting the system logs after saving them.

Because the message notifying you of a full log version is sent only once, and could be unintentionally removed before you see it, you should periodically check for this type of message in the system history log. You can have the system search for all messages of this type that were issued by specifying the message identifier of the message (CPF2456) in the Display Log (DSPLOG) command (see the following discussion of *Displaying System Logs*). For example, to print a list of all messages issued during July, 1980, that indicate a system log is full, you might enter the following command:

```
DSPLOG PERIOD((*AVAIL 070180) (*AVAIL 073180)) MSGID(CPF2456)
   OUTPUT(*LIST)
```

If needed, you can also obtain the names of all system log versions by using the Display Object Description (DSPOBJD) command. If you enter

```
DSPOBJD OBJ(QHST*.QSYS) OBJTYPE(*FILE)
```

the descriptions of all history logs that are on the system are displayed. This display indicates if the object was saved.

Displaying System Logs

You can obtain the information in a system history, service, or programming change log by requesting the system log display.

Obtaining the Display

To obtain the system log display, enter the Display Log (DSPLOG) command. If you do not enter any of the parameters, you will receive a display of all messages in the active version of the history log for the current date; that is, the version in which messages are currently being logged. By selectively entering certain parameters, you can restrict the information displayed to a specific time or period of time, or have it printed.

LOG Parameter: This parameter determines the type of log you receive.

- If you do not enter the parameter, LOG(*QHST) is assumed and you receive the history log.
- If you specify LOG(QSRV) you receive the service log.
- If you specify LOG(QCHG) you receive the programming change log.

PERIOD Parameter: This parameter determines the period for which the log is to be displayed.

- If you do not enter the parameter, PERIOD(*AVAIL *CURRENT) is assumed and all available log entries for the current date are displayed.

- If you want to see log entries for a date other than the current date or for a specific period of time, you have several options:
  - You can specify a particular starting or ending date, or starting or ending time, or a combination of these. For example, you could specify the time 11:00 a.m. and date November 17, 1980, as

    ```
    (110000  111780)

       or

    ('11:00:00'  '11/17/80')
    ```

    Note that you must enclose the values in apostrophes if you use the special characters : and /.
  - You can specify *BEGIN for a starting date to have the display begin at the first day of the log.
  - You can specify *END for the ending date to have the display end at the latest day of the log.
  - You can specify *CURRENT (the default value for the date) in combination with a particular starting or ending time, such as

    ```
    ('11:00:00'  *CURRENT)
    ```

    to have the display begin or end at the specified time on the current date.
  - You can specify *AVAIL (the default value for the time) in combination with a particular starting or ending date, such as

    ```
    (*AVAIL  '11/17/80')
    ```

    to have the display begin or end with all available entries on the specified date.

If you restrict the log messages to a particular period by specifying starting and ending dates and times, the effect of that restriction depends on whether you request the information to be displayed or printed (see the following description of the OUTPUT parameter).

- If you request the log information to be displayed, the display begins with the first message on the specified starting date and time, but the ending date and time are ignored. Any messages that arrived in the log later than the specified starting date and time can be displayed.

- If you request the log information to be printed, the only messages printed are those that arrived in the log after the specified starting date and time but before the specified ending date and time.

If the log version for the specified period is no longer on internal storage, you will receive a message that it is no longer available. Provided you previously saved the version, you can use the Restore Object command to restore it to internal storage and then display it.

JOB Parameter: This parameter determines whether information is provided on all jobs or only on specific jobs.

- If you do not enter the parameter, information on all jobs is shown.

- If you enter the parameter and specify one or more job names (up to a maximum of 5), such as JOB(PAY CKPRT) information relating only to those jobs is shown.

MSGID Parameter: This parameter determines which specific messages are displayed.

- If you do not enter the parameter, MSGID(*ALL) is assumed and all messages for the period and jobs specified are shown.

- If you enter the parameter and specify one or more message identifiers (up to a maximum of 5), such as MSGID(CPF1229 RPG1123) only those specific messages are shown. If you specify only part of the message identifier and use zeros for the remaining digits, such as MSGID(CPF1200) all messages having the part of the message identifier you indicated are shown.

OUTPUT Parameter: This parameter determines how the information is provided.

- If you do not enter the parameter, the information is displayed on your work station screen.

- If you specify OUTPUT(*LIST) the information is spooled and placed on an output queue for printing on the system printer.

Examples:

1. To display at your work station all available entries in the history log, starting on the current date, enter

```
DSPLOG
```

2. To display at your work station information in the history log related to the messages CPF1924 and CPF1932, starting at 9:00 a.m. on November 23, 1979, enter

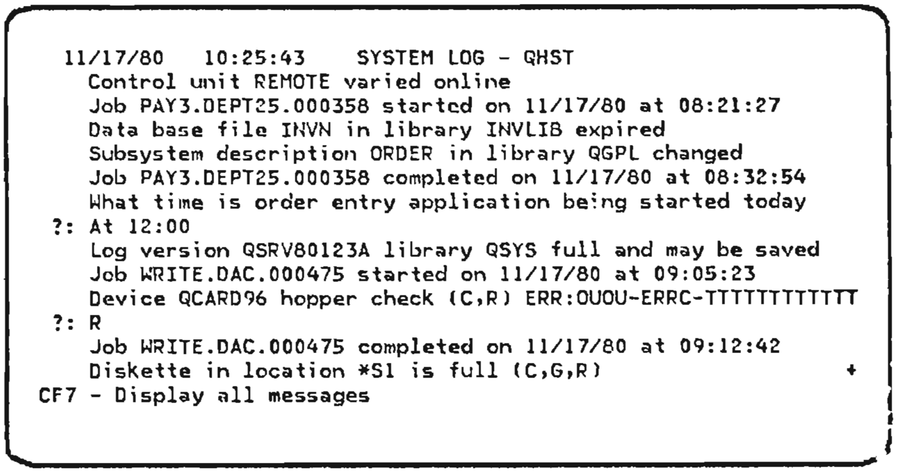
```
DSPLOG PERIOD(('09:00:00' '11/23/79'))
  MSGID(CPF1924 CPF1932)
```

3. To print out all information in the active version of the history log on the job INVPRT during September 1979, enter

```
DSPLOG JOB(INVPRT) PERIOD((*AVAIL 090179) (*AVAIL 093079))
  OUTPUT(*LIST)
```

Interpreting the Display

When you enter the DSPLOG command, you receive a display of the form:



```
  11/17/80   10:25:43    SYSTEM LOG - QHST
    Control unit REMOTE varied online
    Job PAY3.DEPT25.000358 started on 11/17/80 at 08:21:27
    Data base file INVN in library INVLIB expired
    Subsystem description ORDER in library QGPL changed
    Job PAY3.DEPT25.000358 completed on 11/17/80 at 08:32:54
    What time is order entry application being started today
 ?: At 12:00
    Log version QSRV80123A library QSYS full and may be saved
    Job WRITE.DAC.000475 started on 11/17/80 at 09:05:23
    Device QCARD96 hopper check (C,R) ERR:OUOU-ERRC-TTTTTTTTTTTT
 ?: R
    Job WRITE.DAC.000475 completed on 11/17/80 at 09:12:42
    Diskette in location *S1 is full (C,G,R)                     +
CF7 - Display all messages
```

The format of the log display is similar to a regular message display, with a separate message on each line (see *Receiving and Displaying Messages* in Chapter 11). Unless you restrict the information to only a few messages, there will usually be more information than can be displayed on your screen at one time. A + in the lower right corner of your screen indicates that there are additional lines of information to be displayed. You can view the additional lines by pressing the Roll Up key, and return to the previously displayed lines by pressing the Roll Down key.

If you position the cursor on the same line as a message and press the Help key, you receive a separate second-level message display that provides additional information about the selected message. To return to the original log display, press the Enter key.

When you finish examining the system log display, press the Enter key or the CF1 key to return to the display from which you requested the log display.

Using the History Log

At a minimum, you should be able to obtain the following types of information from a system history log:

- Each time CPF was started
  - When was CPF started?
  - Did it follow normal or abnormal system termination?
  - Did the CPF start *manually* (as described at the beginning of this chapter) or *automatically* where the system console was bypassed?
  - What was the status of machine initialization?
  - Was the start CPF process completed?
  - If system termination was abnormal, what batch jobs were lost and what jobs were active?
  - If data base recovery was needed, what cleanup actions were performed on what files?
  - Were any programming changes made or changes removed?

- Each time the system was powered down

- Each time a subsystem was started or terminated
  - When was it started?
  - When was it terminated?

- When changes were made to subsystem descriptions

- Changes in device status (see Chapter 9)
  - When were control unit, device, or line descriptions created, changed, or deleted?
  - When were control units or devices powered on or off?
  - When were control units, devices, or lines varied online or offline?
  - When were edit codes, print images, or tables created or deleted?

- System operator messages (see Chapter 11)
  - What messages were sent to the system operator message queue, such as

    Messages from user programs?
    Messages from work station users?
    Messages to perform some action on a device?
    Messages from spooling readers and writers? (see Chapter 10)
  - What replies did the system operator send to inquiry messages?
  - What inquiry messages were sent by the system operator and what were the replies to those messages?

- Each job processed (see Chapter 12)
  - When did it start?
  - When did it end, and was it completed?
  - If it was canceled, who canceled it (this could be a program as well as a person)?
  - Was it held and later released?
  - Were changes made to the job?

- System save/restore activity (see Chapter 13)
  - What objects or libraries were saved and when?
  - What objects or libraries were restored and when?
  - When were the system library and basic system data saved or restored?

Using the information in the history log, you should be able to reconstruct what happened to each job as a result of actions by programs running the job, or actions by the system operator, or system/subsystem conditions.

POWERING DOWN THE SYSTEM

The normal means of powering down the system is by entering the Power Down System (PWRDWNSYS) command. You *do not* need to set or activate any switches. If necessary, you can also shut off the system power by using the operator/service panel or an emergency Power Off switch on the side of the system unit.

CAUTION
If your system has one or more 3410/3411 tape units, you should ensure that all tape reels are unloaded before powering down the system. Otherwise, data could be lost from the tape. The unload procedure is described in the *3410/3411 Operator's Guide.*

Normal Power Down Using the PWRDWNSYS Command

The PWRDWNSYS command initiates an orderly sequence of events that results in power being shut off throughout the system without loss of data. When the command is executed, the following functions are performed:

- New jobs are prevented from starting.
- Jobs currently being processed are terminated (either canceled or completed). All files are closed.
- Work remaining on the spooling queues is either completed or held on the queues.
- Each subsystem is terminated after the final job being processed in it is terminated.
- CPF is terminated.
- System hardware is sequentially powered down until all power is shut off except the control supply (the Control Supply indicator on the operator/service panel will remain on).

Regardless of what form of the PWRDWNSYS command you use, no new jobs or routing steps are allowed to start. The OPTION and DELAY parameters determine what will happen to jobs currently being processed. Depending on how you enter these two parameters, you can define how the system is powered down.

If a PWRDWNSYS command is issued while an interactive transferring job is on a job queue, the job is canceled as part of subsystem termination.

Because a PWRDWNSYS command inhibits new jobs or routing steps from being initiated by any subsystem, a batch job which is on a job queue due to a TFRJOB command will not be allowed to complete before the system is powered down. The temporary objects associated with a transferring job (such as library lists, QTEMP library and all objects in QTEMP) are destroyed during PWRDWNSYS. This means that on the next IMPL the system is unable to restore the job to its previous state and removes the job from the job queue and produces its job log.

Controlled Power Down with No Time Limit

If you enter

```
PWRDWNSYS
```

you are requesting a controlled power down in which jobs, including your interactive job, are allowed an unlimited amount of time to complete. Therefore, the system power will not be shut off until:

- All batch jobs have been completed or canceled.
- You and all other work station users have signed off.

You should *not* sign off, however, until you have ensured a system operator is not longer needed. Once the power-down sequence has been started, you will not be allowed to sign on again. If some action or response should be required by you on a job after you have signed off, you would have no way of responding, and the system could end up in a wait state.

The following examples show two of the ways to use a controlled power-down with no time limit.

Example 1

In this example, you do not want any more work to be done on the system and you want to confirm that it is powered down before you leave.

Enter the command

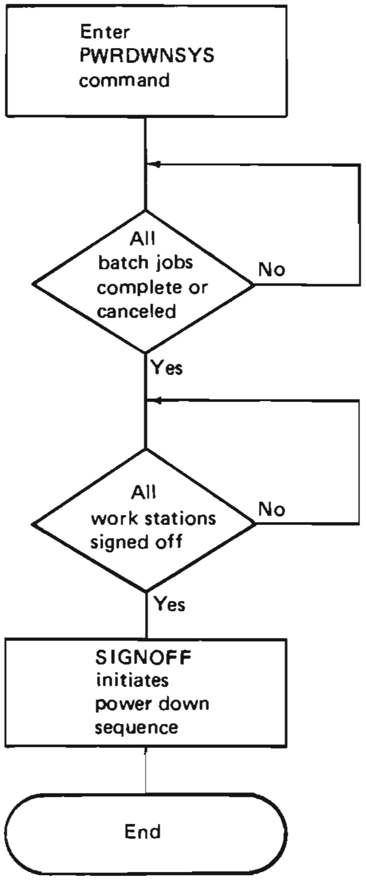
```
PWRDWNSYS
```

Verify that all jobs except your interactive job have completed. You can do this by requesting the subsystem display for all subsystems. If you signed on as the system operator at the system console, your own interactive job will be indicated on the subsystem display.

After all other jobs have been completed, sign off to end your own interactive job; for example, enter

```
SIGNOFF
```

Flowchart


Enter PWRDWNSYS command
All batch jobs complete or canceled
No
Yes
All work stations signed off
No
Yes
SIGNOFF initiates power down sequence
End

Explanation

The PWRDWNSYS command is entered by the system operator, a CL jobstream, or a CL program.

The system operator can wait for batch jobs to complete or cancel them.

The system operator can notify work station users to sign off by sending messages. When other users have signed off the system operator should sign off.

The system operator signing off completes the last active job on the system and the system powers down.

Example 2

In this example, you want to power down the system after all active jobs have completed. You should verify that a system operator is no longer needed to handle devices or respond to messages. You can then enter the PWRDWNSYS command and leave. The system will power down when the last active job has completed.

CAUTION
If the active jobs in the system need any form of operator action, the system will hang and the jobs will not be completed.

Enter the command

```
PWRDWNSYS
```

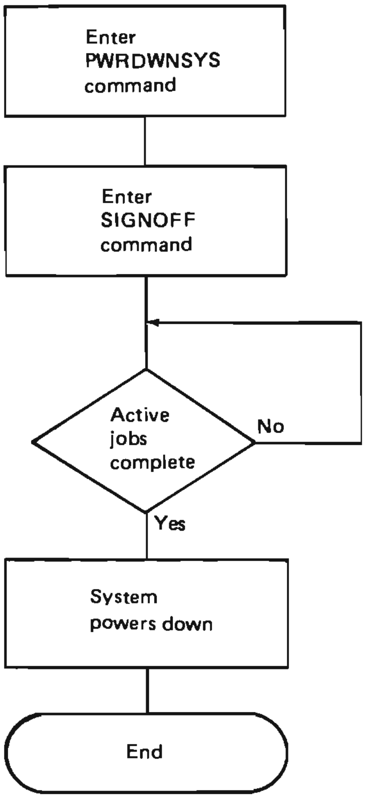
Verify that the active jobs do not need you to handle devices or respond to messages.

Enter the command

```
SIGNOFF
```

Flowchart

Explanation



The PWRDWNSYS command is entered by the system operator, a CL jobstream, or a CL program.

The system operator verifies that the active jobs do not need a system operator present and then signs off and leaves.

Processing continues until all active jobs are complete.

The system powers down after all active jobs are complete.

Controlled Power Down with Time Limit

If you enter the PWRDWNSYS command in a form that specifies a number of seconds in the DELAY parameter, such as

```
PWRDWNSYS DELAY(120)
```

you are setting a time limit on the controlled power down. Any jobs that are not completed at the end of the specified time are automatically canceled. Because your interactive job is canceled along with any other jobs at the end of the time limit, you do not need to sign off.

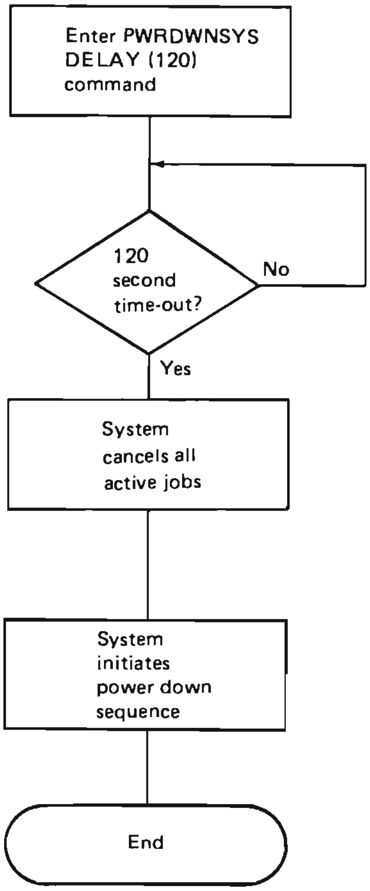
Time

120 seconds

This is the amount of time it takes the system to cancel active jobs and terminate processing. The maximum amount of time is limited by the system value QPWRDWNLMT. If this value is exceeded, the system terminates processing and the condition indicators on the operator service panel display 09FB.

After all jobs have completed or been canceled, it takes approximately 1 minute for each megabyte of storage to complete power down.

Flowchart


Enter PWRDWNSYS DELAY (120) command
120 second time-out?
No
Yes
System cancels all active jobs
System initiates power down sequence
End

Explanation

The PWRDWNSYS command with time delay can be issued from a CL jobstream, a CL program, or the system operator.

The DELAY parameter allows you to enter the number of seconds you want to continue processing before canceling active jobs.

If jobs are canceled because the time limit has run out, the jobs files may be partially updated.

The system initiates the power down sequence after all jobs have completed or canceled.

Immediate Power Down

If you enter the PWRDWNSYS command in the form

```
PWRDWNSYS  OPTION(*IMMED)
```

or simply

```
PWRDWNSYS *IMMED
```

a cancelation process begins immediately on all jobs currently on the system. The system is powered down as soon as all jobs are canceled. Because your interactive job is canceled along with the other jobs, you do not need to (and cannot) sign off to complete the power down.

There may be situations where you will want to use an immediate power down to modify a previously requested controlled power down. For example, if you enter

```
PWRDWNSYS
```

to start the power-down sequence and later decide to power down without allowing particular long-running jobs to complete, you can reenter the command in the form

```
PWRDWNSYS *IMMED
```

to cancel the jobs and complete the power down.

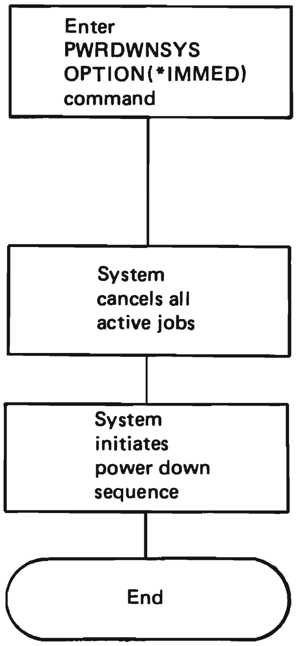
Notes:

1. Even when you put a time limit on the power down (by specifying *IMMED in the OPTION parameter or a delay time in the DELAY parameter) and the specified time expires, the power down takes approximately 1 minute for each megabyte of storage to complete. To prevent the possibility that the system is unable to complete a power down because of an error condition, the system value QPWRDWNLMT defines a maximum time allowed after the *IMMED or DELAY time limit has expired. If the maximum time specified in QPWRDWNLMT is reached, all machine processing is terminated and the condition indicators on the operator/service panel come on. To recover from this condition, see *System/38 Problem Determination Guide*. As shipped from IBM, QPWRDWNLMT is set at 300 seconds (5 minutes).
2. If jobs are canceled as a result of an immediate power down or a controlled power down with a time limit, files being used for the canceled jobs can be left in a partially updated state.

Time | **Flowchart** | **Explanation**

This is the amount of time it takes the system to cancel active jobs and terminate processing. The maximum amount of time is limited by the system value QPWRDWNLMT. If this value is exceeded, the system terminates processing and the condition indicators on the operator service panel display 09FB.

After all jobs have completed or been canceled, it takes approximately 1 minute for each megabyte of storage to complete power down.


Enter PWRDWNSYS OPTION(*IMMED) command
System cancels all active jobs
System initiates power down sequence
End

The PWRDWNSYS *IMMED command can be issued from a CL jobstream, a CL program, or the system operator.

The OPTION(*IMMED) parameter tells the system to stop processing active jobs and cancel all active jobs immediately. The jobs files may be partially updated.

Power Down with Power Warning Feature

When a utility power interruption occurs for a System/38 that has the Power Warning feature and an uninterruptible power supply, the following options can be used:

- Automatic power down using the system value QUPSMSGQ of *NONE or *RESTART. This method uses the quick power-down function that will save volatile storage, ensure data base files and access paths, and power down the system. The quick power down reduces the time required to bring the system back up because data base access paths and machine directories should not have to be rebuilt. For more information on the system value QUPSMSGQ, see the *CPF Programmer's Guide*.
- Power down handled by the system operator.
- Invoke a user-defined job to handle the power down. This option allows you to define a job to handle the power down when the utility power is interrupted.

The Power Warning feature provides the following functions to support the options shown above:

- Sends a message to the system operator message queue (QSYSOPR) and to the system history log (QHST), indicating if the system power source has switched from utility to auxiliary or auxiliary to utility power.
- Sends a message to a user-defined message queue (defined in the system value QUPSMSGQ) indicating if the power warning signal is active or inactive.

Auto-IMPL can be performed when utility power is restored. This is controlled by the RESTART parameter on the PWRDWNSYS command or by the system value QUPSMSGQ (*RESTART).

Automatic Power Down Using the QUPSMSGQ System Value

The automatic power-down option uses the system value QUPSMSGQ of *NONE or *RESTART to signal a quick power down whenever utility power is interrupted. The initial value of QUPSMSGQ when shipped from IBM is *RESTART. If you do not change the system value the result of a power interruption will be quick power down and system restart when power is restored if you have the Auto-IMPL feature and the Auto-IMPL switch is on.

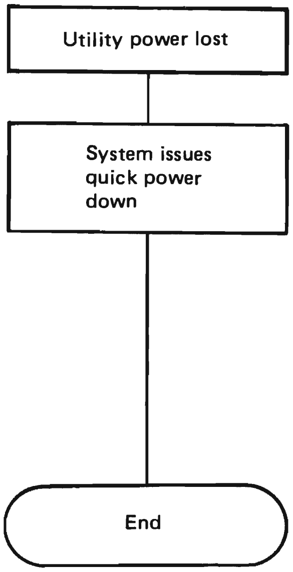
The option requires the minimum amount of planning and a minimum UPS battery capacity. If you are considering this option, you should be aware that when active jobs are canceled, they must be resubmitted and the job files may be partially updated.

This option does not require that a system operator be present.

Time

This is the amount of time that you want the system to be powered by UPS. The quick power down requires approximately 1 minute for each megabyte of main storage.

Flowchart



Explanation

The system issues quick power down as the result of system value QUPSMSGQ *NONE or *RESTART. The quick power down tells the system to stop processing active jobs and cancel all active jobs immediately. The job files may be partially updated. Job logs are written on the next IMPL. The quick power down sequence ensures data base files and access paths, saves volatile storage, and powers down the system.

To bring the system back up after utility power has been restored, use the normal power on sequence described earlier in this chapter.

If you have the Auto-IMPL feature and QUPSMSGQ is set to *RESTART and the Auto-IMPL switch is on, the system will preform IMPL when utility power is restored.

Power Down Handled by the System Operator

If the system value QUPSMSGQ is set to a valid work station message queue or the system operator message queue (QSYSOPR), the power warning message is sent to that queue. The queue that receives the message must be in *BREAK or *NOTIFY delivery mode. If the message queue is not in *BREAK or *NOTIFY mode, CPF will send a message to the system operator message queue and the history log (QHST), then initiate a quick power down.

This option requires the system operator be present to issue the PWRDWNSYS command unless you choose to do a quick power down by changing the message queue delivery status.

If you have the Auto-IMPL feature, you can control the Auto-IMPL by using the *RESTART parameter on the PWRDWNSYS command.

If QUPSMSGQ is set to *RESTART and you don't want the system to come back up, wait until power off is complete and then turn the Auto-IMPL switch off.

Time

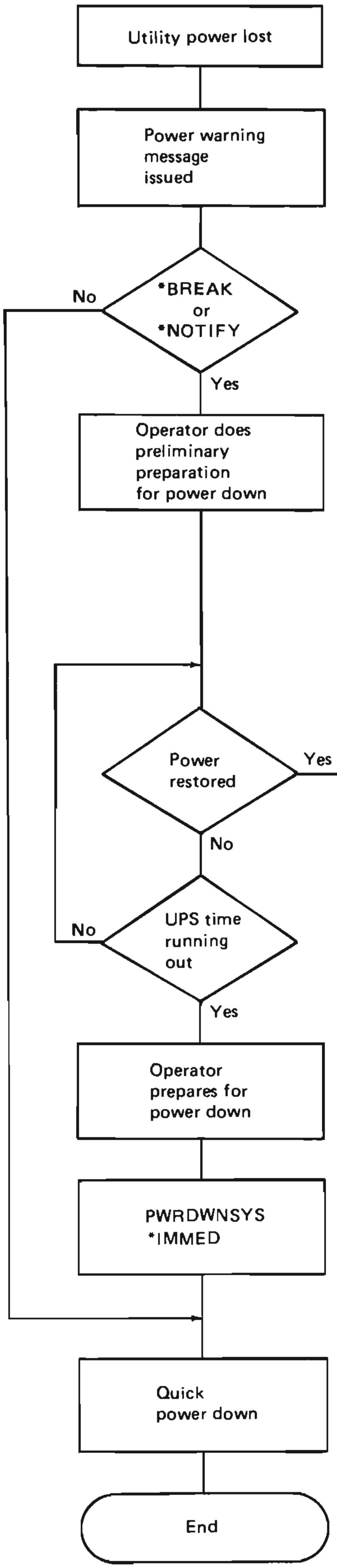
This is the amount of time you want the system to be powered by UPS after a utility power failure. The amount of time that you plan for depends on:

- How long a power interruption you want to ride out without powering down.
- How long it takes the system operator to prepare for power down.

If utility power is restored during this time, you can resume normal processing.

The quick power down requires approximately 1 minute for each megabyte of main storage.

Flowchart



Explanation

The power warning feature sends a message to the message queue defined in QUPSMSGQ.

The receiving message queue must be in *BREAK or *NOTIFY delivery mode.

The system operator receives the warning message. At this point you may want to do the following:

- Try to determine how long utility power will be off.
- Hold long running jobs on the job queue so they will not be partially complete if you have to power off.
- If possible, notify work station users that there may be a power down.

If utility power is restored, you can release held jobs, notify work station users, and proceed with normal processing.

When it becomes evident that you will have to power down, you may want to:

- Notify work station users to sign off.
- Cancel active batch jobs.

Enter the PWRDWNSYS *IMMED command. If you have the Auto-IMPL feature and you want the system to power on when utility power is restored, use the parameter RESTART(*YES) on the PWRDWNSYS command.

The quick power down sequence ensures data base files and access paths, saves volatile storage, and powers down the system.

Power Down Handled by a User-Defined Job

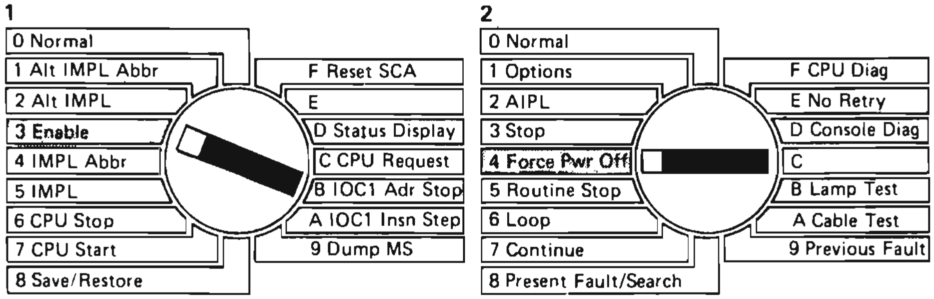
Normal System/38 commands provide the capability to activate a user-defined program whenever a message is sent to a message queue. This capability can be used to design a power failure strategy that will activate a user-defined program whenever a utility power failure message is issued. The user-defined program can then handle whether you want to ride out the power interruption or power down the system. This option does not require that a system operator be present.

Power Down Using the Operator/Service Panel

CAUTION
If the system power is turned off in this manner, the machine performs power-down sequencing of the hardware, but machine processing is interrupted. As a result, any data that was being processed at the time will be lost and any files that were being used will not be closed. Recovery activities may be required when the system is started again (refer to the recovery discussion for *Abnormal Termination* in the *Problem Determination Guide)*.

If you are unable to enter the PWRDWNSYS command, you can power down the system by using the operator/service panel. The procedure is:

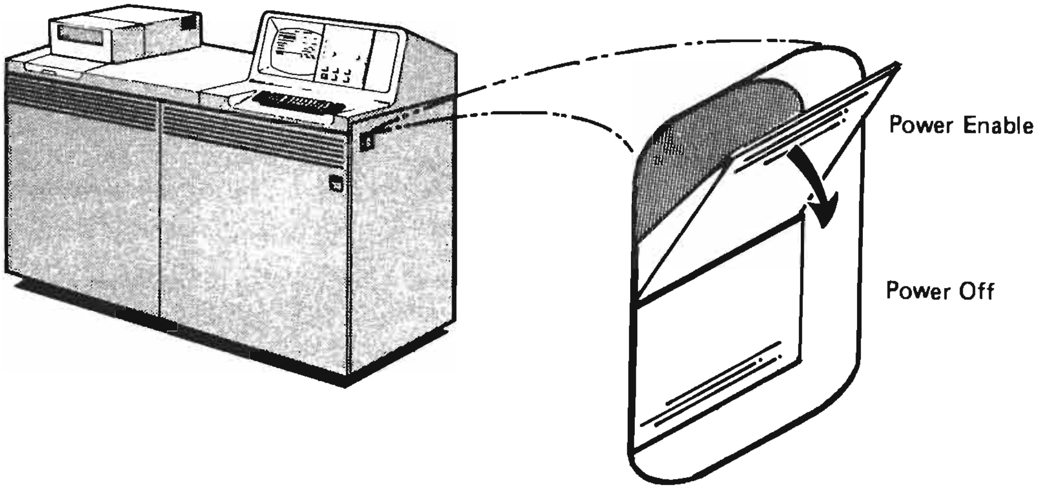
1. Set the rotary switches to



2. Press the Load switch.

Emergency Power Down

An EPO (emergency power-off) switch is located on the side of the 5381 System Unit.



When the EPO switch is moved to the Power Off position, power is instantly cut off to all devices attached to the machine power supply. Processing is instantly interrupted and no power-down sequencing of the hardware is done. Data being processed will be lost, and files being used will not be closed.

You should use the EPO switch only as a last resort when no other means of shutting down the system is possible.

When you use the EPO switch to power down the system, the switch must be reset by your service representative before you can use the system again. After the switch has been reset and the system is restarted, recovery activities will be required before normal operations can be resumed (refer to the recovery discussion for *Abnormal Termination* in the *Problem Determination Guide*).

TERMINATING OPERATIONS WITHOUT POWERING DOWN

Terminating CPF

You can terminate CPF without powering down the system. When you do this, all subsystems except the controlling subsystem are terminated, and all jobs in the controlling subsystem are terminated except an interactive job for the system console. The controlling subsystem is then in a restricted state. Normal operations cannot be performed while the system is in this condition, but you can perform service functions such as taking a standalone dump (see *Standalone Dumps* in the *Problem Determination Guide*). Your service representative may ask you to terminate CPF so that service functions can be performed.

To terminate CPF, enter the Terminate CPF (TRMCPF) command and specify the conditions of the termination in the OPTION and DELAY parameters. These two parameters have the same function as the OPTION and DELAY parameters of the PWRDWNSYS command. If two interactive jobs were created at the system console through the use of the system request menu, one of the jobs must be terminated before the execution of the TRMCPF command can be completed. You terminate one of the jobs by signing off from that job.

You can enter the TRMCPF command from a work station other than the system console if that work station is allocated to the controlling subsystem; however, you will only be able to communicate with the system through the system console after the command is executed.

Examples:

1. To terminate CPF in a controlled manner such that all jobs currently being processed are completed before termination, enter

   ```
   TRMCPF
   ```

2. To terminate CPF immediately, canceling all jobs, enter

   ```
   TRMCPF OPTION(*IMMED)
   ```

   Note: When this option is used, files related to jobs that were canceled may be left in a partially updated state.

Terminating All Subsystems

When you specify *ALL in the SBS parameter of the Terminate Subsystem (TRMSBS) command, such as entering

```
TRMSBS  SBS(*ALL)
```

the effect on the system is similar to entering the TRMCPF command. All jobs in all subsystems are terminated, except the interactive job from which the TRMSBS command was entered. If the TRMSBS command is entered at a work station other than the system console, the single job that remains active is at the other work station instead of the system console. However, you must enter the command at a work station that was allocated to the controlling subsystem at sign-on (for an additional explanation, see *Starting and Terminating Subsystems* in this chapter). If two interactive jobs were created at your work station, the termination is not completed until you sign off one of the jobs.

You use the TRMSBS command to terminate all subsystems when you need to perform a function that can be done only when all other operations have ceased. For example, you must terminate all subsystems before you save or restore all user libraries or before you save the system (see *Saving Information* and *Restoring Information* in Chapter 13). After the save or restore operation is completed, you can resume normal operations by using the STRSBS command to restart the subsystems that are needed. If the save or restore operation is done at the end of a workday, you can enter the PWRDWNSYS command to power down the system without first resuming normal operations.

Interactive jobs that are on a job queue due to a TFRJOB command will be canceled as part of subsystem termination. If an IMPL occurs while either a batch or interactive job is on a job queue due to TFRJOB, that job will be removed from the job queue during IMPL and its job log will be produced.

Chapter 9. Device Control Operations

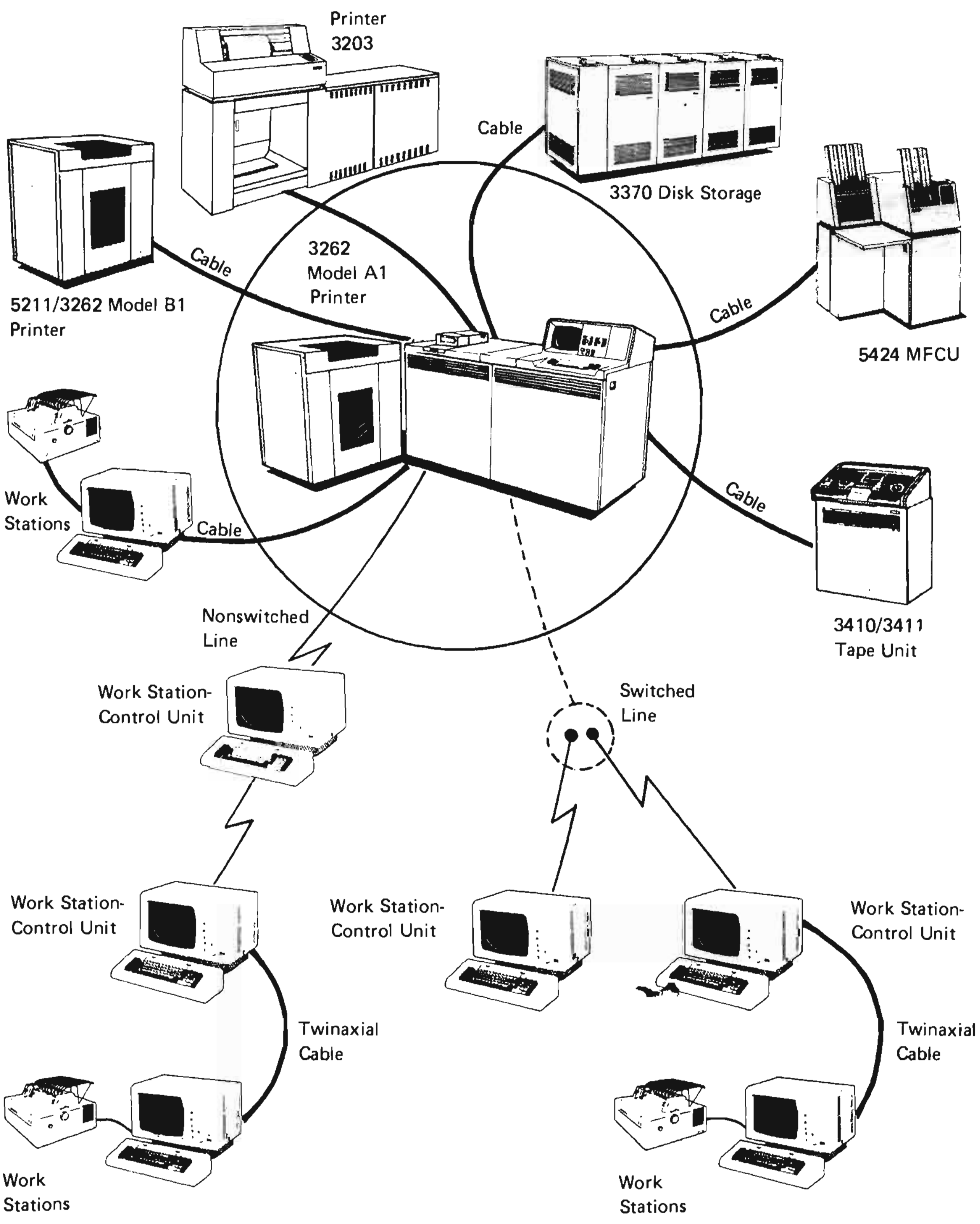
Your duties may include ensuring that the devices connected to the system are operating properly and providing them with the materials needed to run jobs. If a problem occurs with a device, the actions you can take depend on the type of device and the way it is connected to the system.

DEVICES, CONTROL UNITS, AND LINES

Except for the diskette magazine drive, console, and the 3262 Printer Model A1 (if your system has one), all system devices are attached to the system through a cable or a remote communication line, either switched or nonswitched. The 5211, 3203, and 3262 Printers, the 5424 MFCU, the 3410/3411 Magnetic Tape Units, and the 3370 Direct Access Storage Device (see Chapter 4) can only be attached through a cable. The 5251/5252 work stations and the work station printers that comprise the 5250 Information Display System (see Chapter 5) can be either cable-connected (referred to as *local*) or connected through a communication line (referred to as *remote*).

A local work station communicates with the system through a control unit in the 5381 System Unit that is known as the work station controller (WSC). A remote work station communicates with the system through a switched or nonswitched line and remote equipment that includes a control unit and a modem. The control unit and modem may be contained in a 5251 Model 2 or 12 work station/control unit. The line is connected to the system through a local modem, which may be included in the IBM equipment or may be a separate unit.

Some work stations may be connected by cable to a work station/control unit, which in turn is connected to the system through a communication line. Depending on the type of line, there may be more than one control unit connected to a line. There can also be more than one work station connected to a control unit. Thus, your system could have a configuration consisting of multiple lines, each connected to multiple control units to which multiple devices are connected (Figure 9-1).



Work Station-Control Unit – 5251 Models 2 and 12

Work Stations – 5251 Models 1 and 11, 5252, 5256, 5224, 5225

Figure 9-1. Ways Devices Can Be Connected to the System

Each device, control unit, and line is defined to the system by a device, control unit, or line description. To refer to any of them in a command, you use the name of the associated device, control unit, or line description.

When your system was installed, a description was created for each device, control unit, and line. As needed, a programmer can change, delete, or create additional device descriptions, control unit descriptions, or line descriptions using commands.

You can use other commands to turn on or off the power of specific devices or control units, or to vary online or offline any device, control unit, or communication line.

Note: The 3370 Direct Access Storage Device is handled by the system as a part of internal storage rather than as a separate device; therefore, the commands and procedures described in this chapter do not apply to the 3370. If your system has one or more 3370 units, the only action you need to take is to ensure that the switches on each unit are set properly (see *3370 Direct Access Storage Device* in Chapter 4). Setting the 3370 switches incorrectly could cause a system termination.

Power/Vary Guidelines

In using a command to power a device or control unit on or off or to vary a device, control unit, or line online or offline, note that:

- Generally, you would vary a device, control unit, or line offline to prevent it from being used or while it is being serviced. You would power off a device or control unit if it is to be offline for an extended period.

- Some types of error conditions involving the interaction between the system and a device, control unit, or line can be corrected by varying the device, control unit, or line offline and then varying it online again.

- A device that is being used or has been allocated for use cannot be powered off or varied offline.

- The only devices you can power on or off separately using a command are:
  - The 5211 Printer
  - The 3262 Printer (any model)
  - The 3203 Printer
  - The MFCU

  Each 5251/5252 work station and 5256, 5224, and 5225 work station printer has its own power switch that must be used to turn its power on or off.

  You can shut off the power to the diskette magazine drive only by powering down the system.

- The only control unit you can power on and off separately by a command is the control unit in the 3411 Magnetic Tape Unit.

- Any device (except the 3370) can be varied offline and then online again.

- A device or control unit must be varied offline before its power can be shut off by command. It must be powered on before it can be varied online.

- If a device was powered off and then a power on/off switch on the device (such as on the side of the 5211 Printer and the back of the 3262 Printer) was set to Off, that switch must be set to On before the device can be powered on.

- All devices connected to a control unit must be varied offline before the control unit can be varied offline.

- All control units connected to a line must be varied offline before the line can be varied offline.

- A line must be varied online before each control unit connected to it can be varied online.

- A control unit must be varied online before each device connected to it can be varied online.

- If a new device is to be added to an existing control unit, that control unit and all existing devices attached to it must be varied offline and the subsystem terminated before the new device can be added (by a command that creates a device description for the new device). The control unit and the devices attached to it, including the new device, must then be varied online and the subsystem started before those devices can be used again.

- If a new control unit is to be added to a line, that line, any other control units attached to it, and all devices attached to the control units must be varied offline before the new control unit can be added (by a command that creates a control unit description for the new control unit). Before a device can be used again, the device itself, the control unit to which it is attached, and the line to which the control units are attached must be varied online.

Powering Devices On and Off

To shut off the power of a 5211, 3203, 3262 Printer, or the MFCU without shutting down the system, or to power it on after it has been shut off, enter the Power Device (PWRDEV) command. The command has two parameters, both of which you must include.

DEV Parameter: Use this parameter to specify the name of each device to be powered on or off, such as

```
DEV(QCARD96)
```

STATUS Parameter: Use this parameter to specify whether the device is to be powered on or off.

- If you specify STATUS(*ON) the device is powered on.

- If you specify STATUS(*OFF) the device is powered off.

Examples:

1. To turn *off* the power supply of the MFCU, whose device description name is QCARD96, enter

   ```
   PWRDEV DEV(QCARD96) STATUS(*OFF)
   ```

2. To turn *on* the power supply of the MFCU identified as QCARD96 and the system printer named QSYSPRT, enter

   ```
   PWRDEV DEV(QCARD96 QSYSPRT) STATUS(*ON)
   ```

Powering the 3411 Control Unit On and Off

To shut off the power to the 3411 Magnetic Tape Unit, or to power it on after it has been shut off, enter the Power Control Unit (PWRCTLU) command. The command has two parameters that you must include:

CTLU Parameter: Use this parameter to specify the name of the 3411 to be powered on or off, such as

```
CTLU(QTAPE)
```

STATUS Parameter: Use this parameter to specify whether the control unit is to be powered on or off.

- If you specify STATUS(*ON) the control unit is powered on.
- If you specify STATUS(*OFF) the control unit is powered off.

CAUTION
Before powering off the 3411 control unit, you should unload all tapes from the associated tape units to prevent the possibility of data being lost (see the *Unload Procedure* in the *3410/3411 Operator's Guide*).

Examples:

1. To turn off the power of the 3411 that is identified as QTAPE, enter

```
PWRCTLU CTLU(QTAPE) STATUS(*OFF)
```

2. To turn the power of the same control unit on again, enter

```
PWRCTLU CTLU(QTAPE) STATUS(*ON)
```

Varying Devices, Control Units, and Lines Online and Offline

When you vary a device, control unit, or communication line offline, it is removed from the control of the system and cannot be used for system operations. When it is varied online, it is placed in contact with and under the control of the system.

There are separate, but similar, commands to vary a device, control unit, or line. The order in which the commands must be done is shown in examples 3 and 4. The commands are:

- Vary Device (VRYDEV)
- Vary Control Unit (VRYCTLU)
- Vary Line (VRYLIN)

Each command has a parameter that you must enter to specify the device, control unit, or line as follows:

- Use the DEV parameter of the VRYDEV command to specify each device that is to be varied on or off.
- Use the CTLU parameter of the VRYCTLU command to specify each control unit that is to be varied on or off.
- Use the LINE parameter of the VRYLIN command to specify each communication line that is to be varied on or off.

All three commands have the same *STATUS parameter that you must enter to specify whether the equipment is to be varied online or offline.

- If you specify STATUS(*ON) each device, control unit, or line specified in the command is varied online.
- If you specify STATUS(*OFF) each is varied offline.

Note that you use the VRYCTLU command to vary the 3411 tape unit/control unit online or offline, but you use the VRYDEV command to vary the 3410 tape units online or offline. Similarly, you use the VRYCTLU command to vary the 5251 Model 2 or Model 12 display station/control unit online or offline, but you use the VRYDEV command to vary the 5251 Model 1 or Model 11 or the 5252 display station online or offline.

If a device is varied online while more than one subsystem to which it can be allocated is active in the system, there is no way of determining which of the subsystems the device will be allocated to. One way to avoid this uncertainty is to temporarily terminate the subsystems that you do not want the device allocated to, vary the device online, and then restart the subsystems. For example, if the work station WS3 can be allocated to both the QINTER and ORDER subsystems, both of which are active, but you want WS3 to be allocated to QINTER when you vary it online, you could do the following:

1. Request system users working through the ORDER subsystem to temporarily sign off.

2. Terminate the ORDER subsystem by using the TRMSBS command.

3. Vary the work station WS3 online by using the VRYDEV command. It will be allocated to the QINTER subsystem.

4. Restart the ORDER subsystem by using the STRSBS command.

Examples:

1. To temporarily remove the work station named WRKSTN3 from system control so it can be moved, enter

   ```
   VRYDEV  DEV(WRKSTN3)  STATUS(*OFF)
   ```

2. To place the printer QSYSPRT back online after it has been serviced, enter

   ```
   VRYDEV  DEV(QSYSPRT)  STATUS(*ON)
   ```

3. To take offline the communication line SWL2 connected to control units CU3 and CU4, which are connected to the work stations WS4, WS5, WS6, WS7, and WSPRT2, enter in the following sequence.

   Note: These commands must be executed in the exact sequence shown below.

   ```
   a. VRYDEV DEV(WS4  WS5  WS6  WS7  WSPRT2)  STATUS(*OFF)
   b. VRYCTLU CTLU(CU3  CU4)  STATUS(*OFF)
   c. VRYLIN LINE(SWL2)  STATUS(*OFF)
   ```

4. To return to system operation the work stations WS4 and WS5 connected to control unit CU4, which is connected to line SWL2, enter in the following sequence.

   Note: These commands must be executed in the exact sequence shown below.

   ```
   a. VRYLIN  LINE(SWL2)  STATUS(*ON)
   b. VRYCTLU  CTLU(CU4)  STATUS(*ON)
   c. VRYDEV  DEV(WS4  WS5)  STATUS(*ON)
   ```

DETERMINING SYSTEM CONFIGURATION

There will be times when you will need to know the name of a particular device, what devices are connected to a particular control unit, or what control units are connected to a particular line. The work sheets that were used to configure the system originally will provide all the information you need if they were updated with each change of the system configuration. If those sheets are not immediately accessible, however, you can obtain much of the same type of information by requesting system configuration displays.

Obtaining the Displays

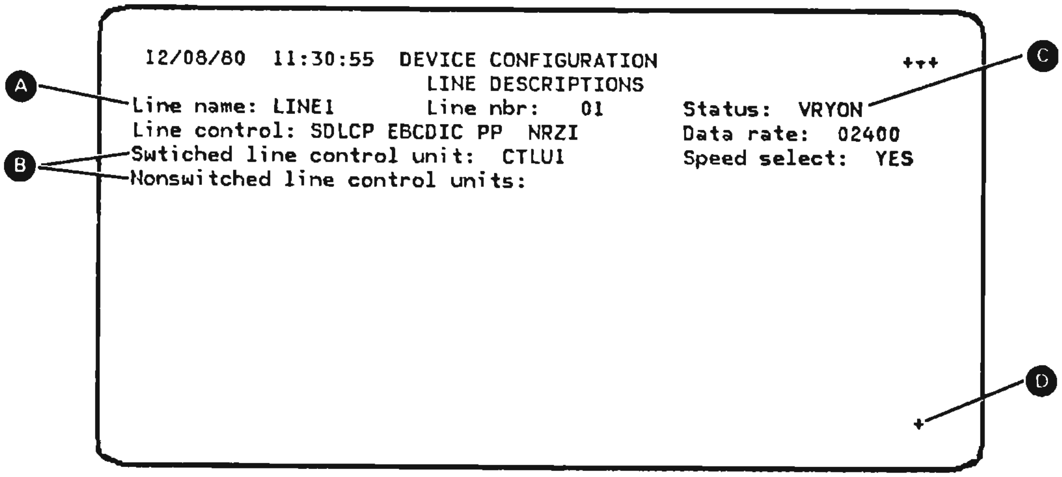
You can obtain all of the configuration displays by entering the Display Device Configuration (DSPDEVCFG) command. When you enter the command, you have two options:

- If you enter DSPDEVCFG the displays appear on the screen of your work station.
- If you enter DSPDEVCFG OUTPUT(*LIST) the same information is written to a spooled output file and placed on an output queue for printing.

Interpreting the Displays

When you enter the DSPDEVCFG command, you receive a set of three displays that show, in sequence, all line, control unit, and device descriptions.

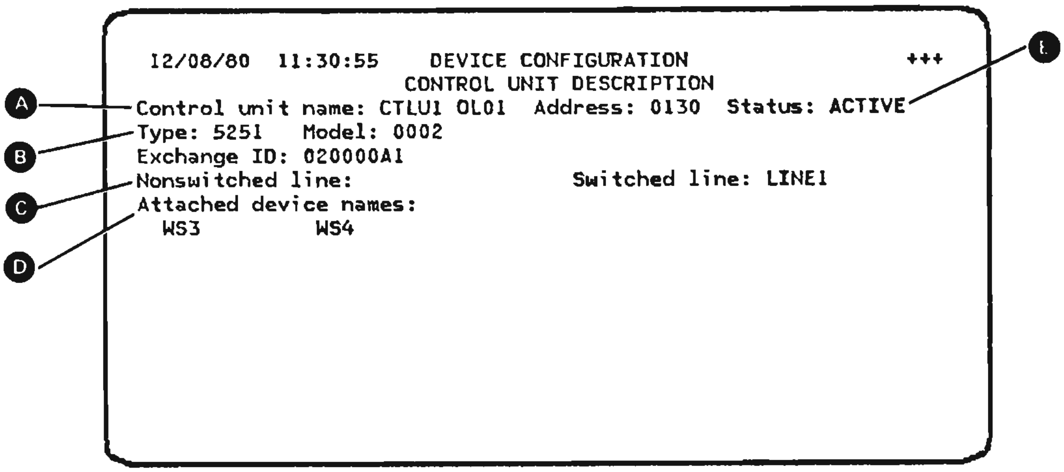
If your system has remote work stations connected to the system through a switched or nonswitched line, the first display you see shows a line description, such as:



```
 12/08/80  11:30:55  DEVICE CONFIGURATION                      +++
                       LINE DESCRIPTIONS
Line name: LINE1       Line nbr:   01       Status:  VRYON
Line control: SDLCP EBCDIC PP  NRZI         Data rate:  02400
Swtiched line control unit:  CTLU1          Speed select:  YES
Nonswitched line control units:




                                                               +
```

The line name A is the name you should use to identify the line in a command. The display lists all the control units B that are attached to a particular line. The current status of the Line C is also shown. System users can communicate with the system through the line if it is varied online (VRYON). If there is a plus sign at the lower right hand corner D of the display, press the Roll Up key to view additonal line descriptions. To review previously displayed line descriptions, press the Roll Down key.

When you have completed viewing the line descriptions, press the Enter key to obtain the second display in the sequence. If your system only has local work stations connected to the system through the WSC, the sequence begins with the second display. This display shows control unit descriptions, such as:



```
 12/08/80  11:30:55    DEVICE CONFIGURATION                    +++
                     CONTROL UNIT DESCRIPTION
Control unit name: CTLU1 OL01  Address: 0130  Status: ACTIVE
Type: 5251   Model: 0002
Exchange ID: 020000A1
Nonswitched line:                    Switched line: LINE1
Attached device names:
  WS3          WS4
```

Of particular interest to you are the following:

A *Control Unit Name*: This is the name you would use when referring to the control unit in a command.

B *Type/Model*: These two fields show the type of control unit and its model number. The possible types and model numbers are:

Type	**Model**	**Description**
5251	0002	Display station/control unit with 960-character screen
	0012	Display station/control unit with 1920-character screen
WSC		Work station controller in 5381 System Unit
3411		Magnetic tape unit/control unit
	0001	– 317.5 mm/sec (12.5 in/sec)
	0002	– 635 mm/sec (25 in/sec)
	0003	– 1270 mm/sec (50 in/sec)
PU2	00XX	Secondary LU1 controller
*BSC	0000	BSC device

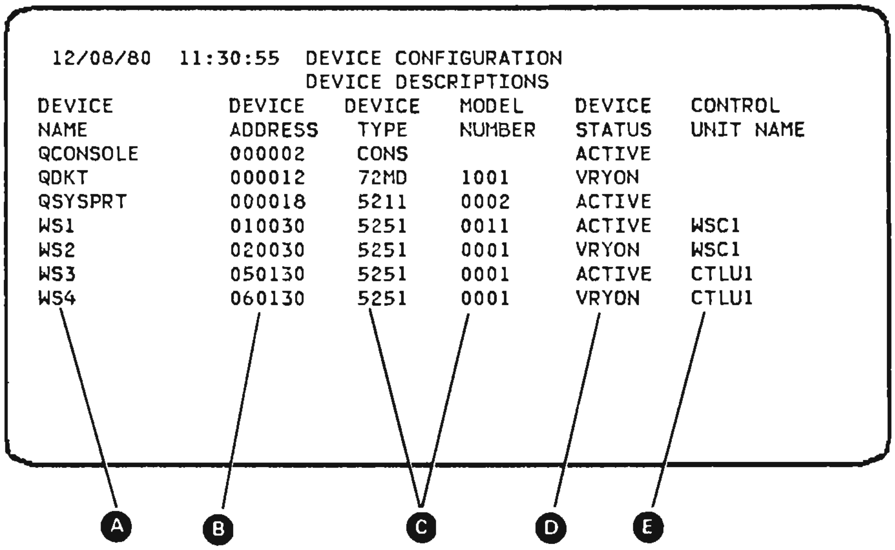
C *Attached Line*: This is the name of the line to which the control unit is attached.

D *Attached Devices*: The names of all of the devices attached to the control unit are listed.

E *Status*: The current status of the control unit is shown. If the status is VRYOFF, the control unit is varied offline and must be varied online before the devices attached to it can be used.

You can view additional control unit descriptions by pressing the Roll Up key. You can review previously displayed control unit descriptions by pressing the Roll Down key. You can also back up to the line description display by pressing the CF2 key.

When you have completed viewing the control unit descriptions, press the Enter key to obtain the final display in the sequence. This display shows device descriptions, such as:



```
 12/08/80  11:30:55  DEVICE CONFIGURATION
                     DEVICE DESCRIPTIONS
DEVICE          DEVICE   DEVICE   MODEL    DEVICE   CONTROL
NAME            ADDRESS   TYPE    NUMBER   STATUS   UNIT NAME
QCONSOLE        000002    CONS             ACTIVE
QDKT            000012    72MD    1001     VRYON
QSYSPRT         000018    5211    0002     ACTIVE
WS1             010030    5251    0011     ACTIVE   WSC1
WS2             020030    5251    0001     VRYON    WSC1
WS3             050130    5251    0001     ACTIVE   CTLU1
WS4             060130    5251    0001     VRYON    CTLU1
```

A *Device Name*: This column shows the device names as you would enter them in a command.

B *Device Address*: This column shows the system address for each device.

C *Device Type/Model Number*: These two columns show the type and model number of each device. The possible types and models are:

Type	Model	Description
3203	0005	System printer (cable-connected)
5211	0002	System printer (cable-connected)
3262	A1	System printer (using system power)
	B1	System printer (cable-connected)
5424	A1	Minimum-speed read/punch/print
	A2	Maximum speed read/punch/print
72MD	1001	Diskette magazine drive
3410	0001	Magnetic tape unit – 317.5 mm/sec (12.5 in/sec)
	0002	– 635 mm/sec (25 in/sec)
	0003	– 1270 mm/sec (50 in/sec)
5251	0001	Display station with 960-character screen
	0011	Display station with 1920-character screen
5252	0001	Dual display station
5256		Work station printer
	0001	– 40 characters/sec
	0002	– 80 characters/sec
	0003	– 120 characters/sec
5224	0001	– 140 LPM
	0002	– 240 LPM
5225	0001	– 280 LPM
	0002	– 400 LPM
	0003	– 490 LPM
	0004	– 560 LPM
PLU1	0000	Secondary LU1 device
*BSC	0000	BSC device

D *Device Status*: This column shows the current status of the device. If the status is VRYON, the device is varied online but not currently active. If the status is VRYOFF, the device has been varied offline and cannot be used unless it is varied online.

E *Control Unit*: This column shows the name of the control unit each device is connected to. For some device types, this column will be blank.

You can view additional device descriptions by pressing the Roll Up key. You can review previously displayed device descriptions by pressing the Roll Down key. You can also back up to the display of control unit descriptions by pressing the CF2 key.

When you have completed viewing the device descriptions, press the Enter key to return to the display from which you requested the configuration displays. Note that you can return immediately to your original display, regardless of which of the three displays you are viewing, by pressing the CF1 key.

OPERATOR/DEVICE INTERACTION

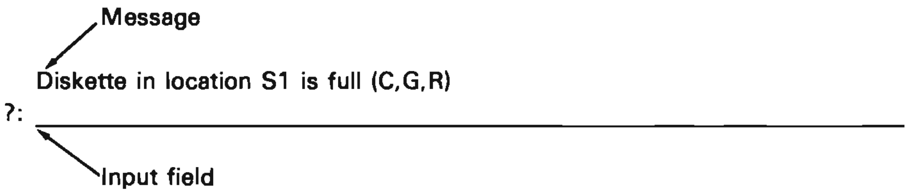
When an action is required on a system device to prepare for system operations or a particular job, the required action should be specified in your operating or job setup instructions. Those instructions should indicate, for example,

- Whether special forms or a different print belt are to be used in the system printer
- Which cards should be placed in which hopper of the MFCU
- Which diskettes or magazines should be placed in which location of the diskette magazine drive
- Which tape reel should be placed on which magnetic tape unit
- If the job is to use spooling, which reader and/or writer to start
- Which commands, if any, to enter

Handling Devices through Messages

If your intervention is required on a device during system operations, the system sends a message to the system operator message queue. For spooling operations, the message is sent to the message queue specified in the command that starts a spooling reader or writer (see Chapter 10). To make effective use of those messages, you must set the message queue to the break or notify delivery mode, so that you receive each message automatically, or at least are notified when each message arrives (see Chapter 11 for a description of messages and message queues). The system operator message queue will be in the break mode if its message delivery mode has not been changed since the system was started.

When you display the messages in the queue, the intervention message is shown on one line of the message display and the next line below contains an input field for your reply, such as



```
  Diskette in location S1 is full (C,G,R)
?: ____________________________________________
```

The type of reply you can enter is indicated following the message; such as *(C,R)* or *(C,G,R)* as in the example shown. If you are not sure what to do or what the reply codes mean, position the cursor on the line containing the message and press the Help key. You then receive a separate display containing basic details of the message plus an expanded description that tells you what to do. For example, if you press the Help key for more details on the message in the example shown, the expanded description would be:

> Insert another diskette and enter R to retry, or enter G to continue processing on next diskette specified in program, or enter C to cancel processing.

In messages of this type, you are given the option of correcting or bypassing the condition and continuing the operation, or stopping the operation. When you enter your reply on the message display, the system operator menu returns to your screen.

The types of conditions you will need to handle depends on the device. A description of typical situations involving devices follows.

Diskette Magazine Drive

The system can detect a variety of diskette conditions, such as

- Diskette has the wrong format for the specified operation

- Diskette is faulty

- Diskette in the location is not the one specified (wrong volume identifier) or does not contain the specified files

- No more room on the diskette

- Additional diskettes or magazines need to be loaded

The procedures for loading and removing diskettes and magazines are described in Chapter 3. The diskette and individual slot positions as specified in commands are shown in Figure 13-1 of Chapter 13. Remember to close the cover of the diskette magazine drive after changing diskettes or magazines. The device will not function properly if the cover is not closed.

Examples:

1. You receive the following message:

   ```
   Diskette at S1 not formatted for basic exchange files (C,R)
   ```

   You are not sure what to do so you press the Help key. You receive the expanded message:

   ```
   Enter C to cancel processing or insert another diskette and enter R to
   retry.
   ```

   You replace the current diskette in I/O slot 1 by a properly formatted diskette and enter *R* to indicate that the operation should be tried again.

2. You receive the same message but you are not sure what to do, so you enter C to cancel the operation.

5211/3262, 3203, 5224, 5225, and 5256 Printers

Line Spacing: For the 5211, 3262, and 3203 printers, a line spacing of 6 or 8 lines per inch is defined by programming (specified as an attribute of a printer device file). For the 5224 and 5225 printers, lines per inch are also programmable and valid values are 4, 6, 8, and 9 lines per inch. For the 5256, line spacing is determined by the setting of a switch on the front of the printer. An inquiry message is sent to the system operator or work station operator each time the device file requires a change in the lines per inch setting.

Forms Type: When a spooling writer is started to produce output on a printer, it is assumed that the standard forms (*STD) are loaded in the printer (see *Writer Operating Considerations* in Chapter 10). If a different type of forms is specified for any of the files to be produced by the writer, a message is issued to load the forms.

No printing is done until you respond to the message that requested you to load the forms. When you respond to the message by entering G (go), printing begins (or resumes). If you enter *H* (hold), the spooled output file is held.

The system cannot detect whether you have loaded the correct forms or not, only that your response to the message has indicated printing can proceed. If you have loaded forms that are not long enough, the printing intended for one form will run over onto the next form. If the forms are not wide enough, some of the printing will be off the forms.

If forms alignment is specified for a printer output file, printing stops after the first line has been printed, and you receive a message to align the forms.

The procedures for loading and aligning forms on each printer are given in the separate operator's guides for the printers (see *About This Manual*). Note that the Ready light on the printer operator's panel must be on before printing can resume. The procedure for putting the printer in a ready condition is also described in the separate operator's guides.

Print Image and Print Belt/Train: For the 5211, 3262, and 3203 printers, several different type styles are available on interchangeable print belts/trains. Each type style is associated with a specific print image stored in the system. When a different print image is to be used, the system sends you a message to load the print belt/train that matches the specified print image.

When you have changed the print belt/train, you respond to the message by entering G (go). Printing does not resume until you respond to the message. You should check that the Ready light on the printer operator's panel is on before responding to the message.

The procedures for changing a print belt/train are given in the separate operator's guides for the 5211, 3262, and 3203 printers.

Printer Ribbon: The system can detect the failure of the printer ribbon to feed properly, but it has no way of detecting print quality. Therefore, you should check the printer periodically to verify that the printing has not become too light. If the printing does become light, you need to change the print ribbon.

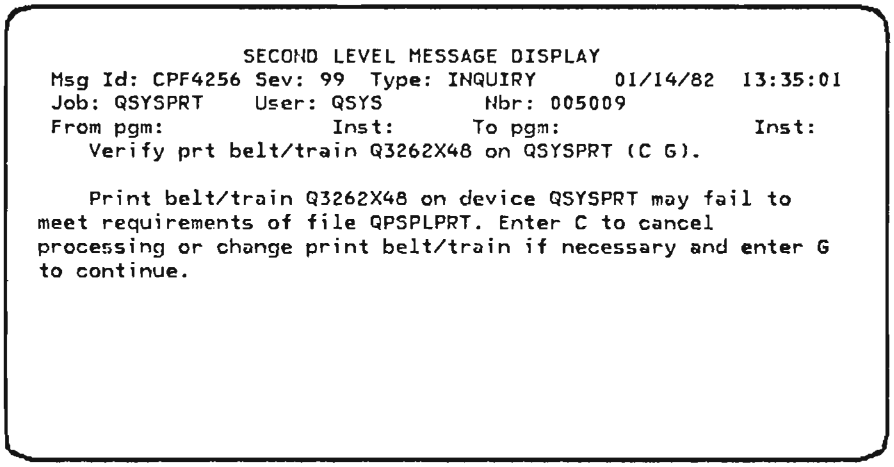
The procedure for changing the ribbon is given in the printer operator's guides.

Examples:

1. You receive the following message:

   Verify prt belt/train EBCDIC2 on QSYSPRT (C,G)

   You are not sure what to do, so you press the Help key to request additional information. You receive the expanded message:



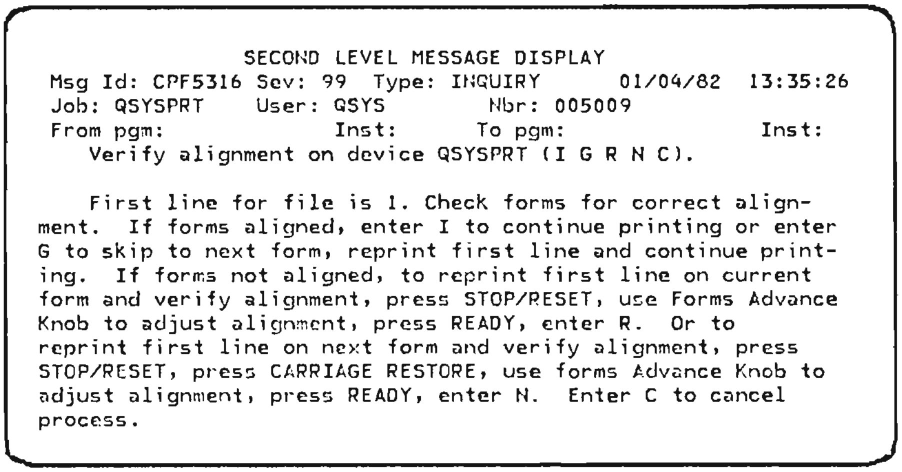
```
                 SECOND LEVEL MESSAGE DISPLAY
 Msg Id: CPF4256 Sev: 99  Type: INQUIRY       01/14/82  13:35:01
 Job: QSYSPRT    User: QSYS        Nbr: 005009
 From pgm:                Inst:      To pgm:               Inst:
    Verify prt belt/train Q3262X48 on QSYSPRT (C G).

    Print belt/train Q3262X48 on device QSYSPRT may fail to
meet requirements of file QPSPLPRT. Enter C to cancel
processing or change print belt/train if necessary and enter G
to continue.
```

   You check the job setup instructions and find that the files to be produced from the job do require a different print belt. You open the printer and replace the current print belt with the specified print belt, following the instructions in the printer operator's guide. Then you enter G (go) at your work station.

2. You receive the following message

   Verify alignment on device QSYSPRT (I G R N C)

   You are not sure what to do, so you press the Help key to request additional information. You receive the expanded message:



```
                   SECOND LEVEL MESSAGE DISPLAY
 Msg Id: CPF5316 Sev: 99  Type: INQUIRY      01/04/82  13:35:26
 Job: QSYSPRT    User: QSYS        Nbr: 005009
 From pgm:              Inst:      To pgm:               Inst:
    Verify alignment on device QSYSPRT (I G R N C).

    First line for file is 1. Check forms for correct align-
ment.  If forms aligned, enter I to continue printing or enter
G to skip to next form, reprint first line and continue print-
ing.  If forms not aligned, to reprint first line on current
form and verify alignment, press STOP/RESET, use Forms Advance
Knob to adjust alignment, press READY, enter R.  Or to
reprint first line on next form and verify alignment, press
STOP/RESET, press CARRIAGE RESTORE, use forms Advance Knob to
adjust alignment, press READY, enter N.  Enter C to cancel
process.
```

Forms Alignment

When forms alignment is required for a printer output file, the first line is printed, and the inquiry message CPF5316 is sent to the system operator message queue asking you to verify the forms alignment and then reply to the message. When the first line is printed, all the skip and space commands that are to be executed before the first line of data are included in the data stream. For example, if a program prints the first line of data on line 6, the commands to skip to line 6 will be executed before the first line of data is printed.

Forms alignment will be requested as follows:

- For nonspooled output
  - When ALIGN(*YES) was specified on the CRTPRTF, CHGPRTF, or OVRPRTF command for the file being printed.

- For spooled output
  - When the file being printed is the first file to print after a spool writer has been started.
  - Any time the lines per inch value changes.
  - Any time the forms length changes.
  - Any time the forms type changes.
  - On a work station printer if invalid control characters have been detected in the previous file.
  - On a work station printer if the CANCEL key has been pressed while the previous file is being printed.
  - On a work station printer if a C reply has been given to the inquiry message for unprintable characters detected in a previous file.

The valid replies to message CPF5316 are I, G, R, N, and C.

- I (ignore): You should use the I reply when the forms are aligned properly and you want to continue printing the entire file.

- G (go): You should use the G reply when the forms are aligned correctly and you want to eject to a new page and print the entire file.

- R (reprint): You should use the R reply when the forms are not aligned correctly and you want to do the alignment on the current form. For the R reply, the first line of data is reprinted on the first form exactly where the paper is positioned. The space and skip commands are not reissued. The forms alignment message is issued again. The following operator actions are required:
  - When the forms alignment message is issued, check to see if the first line is printed in the correct position.
  - Press the Stop/Reset key on the printer.
  - Use the forms advance knob to position the paper to where the first line should print.
  - Use the R reply for the inquiry message.
  - The first line of data will be printed exactly where the paper is positioned and the forms alignment message will be issued again.
  - If the forms are aligned correctly, you can use the I or G reply to continue printing. If the forms are still not positioned correctly, you can repeat the procedure by using the R or N reply.

- N (next): You should use the N reply when the forms are not aligned properly and you want to recheck forms alignment on the next form. After you have aligned the forms, the first line is printed on the next form and the message CPF5316 is issued again. The following operator actions are required:
  - When the forms alignment message is issued, check the forms alignment on the printer.
  - Press the Stop/Reset key on the printer.
  - Press the Carriage Restore key to advance the paper to the next form.
  - If necessary, use the Forms Advance knob to position the form where the first line should print.
  - Press the Ready key on the printer.
  - Reply N to the inquiry message. The first line of data will be printed again on the next form and the forms alignment message will be issued again.
  - Check the forms alignment again. If the forms are aligned properly, you can use the I or G reply to continue printing. If the forms are still not aligned properly, you can continue to use the N or R reply until the forms are aligned properly.

- C (cancel): You should use the C reply if you want to cancel printing of the file. An escape message is signaled to indicate the cancel reply was entered. For spooled data, the writer will not be canceled and the writer will start printing the next file on the output queue. For direct, nonspooled printing, the processing of the file will be canceled.

Error Recovery for Printers

When an error occurs on a system or work station printer, the system sends a message to the appropriate message queue. This message queue can be QSYSOPR, the message queue specified on the CRTDEVD command, or the message queue specified on the STRPRTWTR command. The message can be:

- An information message that gives you additional information about printer status.

- An inquiry message, informing you of a correctable error condition. You are given reply options that tell the system whether to attempt to reprint the data, ignore the error, hold the spooled file, or cancel the file. The second-level text to the message contains the recovery instructions (operator actions) for each reply option in the first-level message. You can obtain the second-level text by using the Help key. The reply options and the operator actions for an inquiry message depend on the type of printer (work station or system) and on the type of output (spooled or direct). See the following sections on inquiry message recovery.

- An escape message for an error in the data stream, such as an invalid printer control character. Usually these messages request the programmer to check the data and resubmit the job.

- An escape message for a noncorrectable error.

Inquiry Message Recovery for Work Statlon Printers (5256, 5224, 5225)

For work station printers, it is important that you follow exactly the recovery actions specified in the inquiry message. Deviation from them could result in improper recovery action by the system. For example, if a forms check occurred during printing of a spooled file, message CPF5344 would be sent to the appropriate message queue. If you want to reprint the last page, the second-level text for the message says to press the Stop switch; remove any obvious causes; press the Cancel switch to clear machine buffers; align the forms to line 1; press the Stop switch to reset the error; press the Start switch; and enter the page number. If you do not follow this sequence and enter the page number first or do not press the Cancel switch, the proper system recovery action will not be performed.

For each inquiry message, you should carefully read the instruction in the second-level text before you perform any recovery action on the printer.

For work station printers, correctable errors have the following options: C, H, I, 1-9999, or R. These options do the following:

- C cancels the printing of the file.

  The error recovery actions for the C option are:

  1. Press the Stop switch and remove or correct any obvious causes.

  2. Press the Cancel switch to clear the machine buffer.

  3. Press the Stop switch to reset the error.

  4. Press the Start switch to ready the printer.

  5. Reply C to the inquiry message.

  Note that even though you do not want to continue printing the file for the C option, it is important to press the Stop and Cancel switches on the printer. If you fail to do this, one or more pages could be printed from the current file and the system may not do the proper cleanup processing in closing the file.

- H holds the current file. The entire file is kept on the output queue and its status is changed to hold. The writer will then begin processing the next file on the output queue.

  The error recovery actions for the H option are:

  1. Press the Stop switch and remove any obvious causes.

  2. Press the Cancel switch to clear machine buffers.

  3. Press the Stop switch to reset the error.

  4. Press the Start switch to ready the printer.

  5. Reply H to the inquiry message.

  Note that even though you do not want to continue printing the file for the H option, it is important to press the Stop and Cancel switches on the printer. If you fail to do this, one or more pages from the current file may be printed and the system may not perform the proper cleanup processing in closing the file.

- I allows you to ignore the error and continue processing the file.

  The error recovery actions for the I option are:

  1. Press the Stop switch to reset the error; remove any obvious causes.

  2. Press the Start switch to ready the printer.

  3. Reply I to the inquiry message.

  4. Printing will continue on the current line of the form.

- 1-9999 allows you to start reprinting the file at a specified page number. This option is valid for a spooled output file only.

  The error recovery actions for the 1-9999 option are:

  1. Press the Stop switch and remove any obvious causes.
  2. Press the Cancel switch to clear machine buffers.
  3. Align the forms to line 1 of a new form.
  4. Press the Stop switch to reset the error.
  5. Press the Start switch.
  6. Determine which page you want the printing to restart and enter the page number as a reply to the inquiry message. If the output file does not contain page numbers you will have to examine the printed output already produced and calculate the page number you want to start with (this can be an approximation for large files).

- R allows reprinting of the page where the error occurred. This option is valid for direct output only. For direct output, the printer gets its data from a system or user program so the entire file is not available for reprinting. However, the printer data management does maintain two print buffers which allows the reprinting of the last page when a recoverable error occurs.

  The error recovery actions for the R options are:

  1. Press the Stop switch and remove any obvious causes.
  2. Press the Cancel switch to clear machine buffers.
  3. Align forms to line 1 of the new form.
  4. Press the Stop switch to reset the error.
  5. Press the Start switch to ready the printer.
  6. Reply R to the inquiry message.

Notes for Work Station Printers

- When you are advancing forms on the 5256 printer, you should always use the following procedure.

  1. Press the Stop switch on the 5256.
  2. Advance the forms with the forms adjustment knob, the Forms Feed switch, or the Line Feed switch.
  3. Press the Start switch to make the 5256 ready.

  If you do not follow this procedure you can get print check errors.

- When an unrecoverable hardware error occurs, an escape message will be signaled to the user program, and then a notify message will be sent to the associated message queue. The notify message (CPF5201) instructs the operator to press the Stop switch and then the Start switch on the device. This action is necessary so that printer data management can complete error recovery. If you fail to follow the instructions in the message, the spool writer or a user program that uses direct output will remain in a wait state and processing will not continue until you ready the printer.

  You should always ensure that the associated message queue is in break mode. Otherwise the notify message will not be displayed and the print writer process may appear to be in a wait state.

Inquiry Message Recovery for System Printers (5221, 3262, 3203)

For system printers, the reply options for inquiry messages are similar to those for work station printers, but the operator actions for readying the printer are quite different.

For system printers, correctable errors have the following options: C, I, G, 1-9999, or R. These options do the following:

- C cancels the printing of the file; the writer will start processing the next file for spooled file related errors or cancel the writer for hardware related errors.

  The error recovery actions for the C option are:

  1. Press the Stop switch to reset the error; remove or correct any obvious causes.
  2. Press the Ready switch to make the printer ready.
  3. Reply C to the inquiry message.

- I allows you to ignore the error and continue printing the file.

  The error recovery actions for the I option are:

  1. Press the Stop switch to reset the error; remove any obvious causes.

  2. Press the Ready switch to make the printer ready.

  3. Reply I to the inquiry message.

- G allows you to continue printing the file. This option is generally available on messages which require you to verify something on the printer, for example correct print belt or forms alignment.

  The error recovery actions for the G option are:

  1. Verify the condition described in the message text.

  2. Reply G to the inquiry message.

- 1-9999 allows you to back up a specified number of pages and start reprinting the file. This option is valid for a spooled output file only.

  The recovery actions for the 1-9999 option are:

  1. Press the Stop switch to reset the error; remove any obvious causes.

  2. Align the forms to line 1 of a new form.

  3. Press the Ready switch to make the printer ready.

  4. Determine how many pages back you want to start reprinting and enter the value on the inquiry message.

- R allows you to reprint the page where the error occurred. This option is valid for direct output only.

  The error recovery actions for the R option are:

  1. Press the Stop switch to reset the error; remove any obvious causes.

  2. Position the forms at line 1 of a new form.

  3. Press the Ready switch to make the printer ready.

  4. Reply R to the inquiry message.

Error Recovery for MFCU

The system can detect a variety of conditions on the MFCU, such as:

- Cards jammed
- Card did not feed properly
- Additional cards needed
- Print ribbon not feeding properly

But, as with the printers, the system cannot detect poor print quality.

The procedures for loading and unloading cards, clearing a card jam, and starting the MFCU are described in the separate MFCU operator's guide (see *About This Manual*).

Examples:

1. You receive the following message:

   ```
   Device QCARD96 hopper check (C,R)
   ```

   You press the Help key to obtain the expanded description:

   File INV2 being processed when hopper check occurred on device QCARD96. Check MFCU. PRI or SEC light tells which hopper failed to feed. Straighten cards in hopper. If necessary, correct any damaged cards. If both hoppers being used, raise cards in nonfailing hopper and press NPRO. If card fed to stacker, place it in nonfailing hopper. Press MFCU Start. Enter R to retry, or clear all error indications on MFCU and enter C to cancel.

   You straighten the cards, replace a damaged card, and press the Start key on the MFCU operator's panel. Then you enter *R* to indicate that the operation should resume.

2. You receive the same message, but when you attempt to replace the cards, you find you are not sure what order they were in. Therefore, you enter *C* to cancel the operation until you can contact the user who submitted the job.

Error Recovery for 3410/3411 Magnetic Tape Units

The system can detect a variety of tape conditions, such as

- Tape unit not ready
- Tape reel currently loaded in the unit is not the volume specified
- BOT (beginning of tape) marker not found
- Tape ran off reel

The procedures for loading and unloading the tape reels and for using the indicators and pushbuttons are summarized in the *3410/3411 Operator's Guide,* G232-0004.

Examples:

1. You receive the following message:

   ```
   Load volume RECORDS onto device QTAPE1 (C,R)
   ```

   You press the Help key to obtain the expanded description:

   ```
   Volume ORDERS currently loaded on device. When correct volume
   loaded, enter R to retry or enter C to cancel processing.
   ```

   You remove the tape reel labeled ORDERS and replace it with the reel labeled RECORDS. Then you enter R to indicate that the operation should proceed.

2. You receive the following message:

   ```
   Ready device QTAPE2 (C,R)
   ```

   You press the Help key to obtain the expanded description:

   ```
   Press the Start key. Enter R to retry or enter C to cancel processing.
   ```

   You press the Start switch (push button) on the tape unit and enter R to indicate that the operation should be retried.

Failing Device Without Message

In some situations, a device may fail in such a way that there is no indication of a failure to the system, and therefore the system will not issue an error message. If a device is not operating properly, but you have not received an error message, you may be able to correct the condition by varying the device offline and then online again (see *Varying Devices, Control Units, and Lines Online and Offline* in this chapter).

If the device still fails to operate properly, see the *Problem Determination Guide.*

Controlling Devices through Commands

There may be occasions when you will want to stop a device even though you have not received a message indicating a problem with it. If input/output operations are being done through spooling, you can use spooling commands to stop or start the devices being used.

For example, you can stop a device by entering a command to hold or cancel the reader or writer program currently using the device. If you have stopped a device by a command to hold the associated reader or writer, you can later restart it by entering a command to release the reader or writer.

Details on spooling commands and examples of their use are given in Chapter 10.

HANDLING COMMUNICATIONS LINES

If work stations are connected to your system through a switched line, the line connection must be established. That connection may be made automatically (by autocall or autoanswer equipment), or your assistance may be required.

If the connection is made automatically, the only indication to you is a message that the connection was successfully established. This message is sent to the system operator message queue, and appears on the message display when you display messages on the queue.

If your assistance is needed in making the connection, you will use one of two procedures. Because these procedures involve responding to a message sent to the system operator message queue, you should make sure that the queue is in the break delivery mode so that you receive the message automatically (see *Message Queues* in Chapter 11). The message queue will be in the break mode if you have not changed it since the system was started.

Communications Statistical Network Analysis Procedure (CSNAP)

As the system operator, you may be responsible for keeping track of the condition of the lines and control units on your communications network. The system keeps track of the errors that occur on each line and control unit. You can print this information periodically to determine the condition of your lines and control units by using the List CSNAP History (LSTCNPHST) command. This command should be used when the line is installed and periodically (probably weekly) thereafter to keep track of the condition of your communications network.

The system usually sends you a message if there is a problem with a line or control unit. However, if no message is sent but there seems to be a problem with a line or control unit, such as the system seems unusually slow, you can use the List CSNAP History command to determine the condition of the line and control unit.

For information on how to use the List CSNAP History command, see the *Problem Determination Guide*.

CL Commands for Determining Communications Status

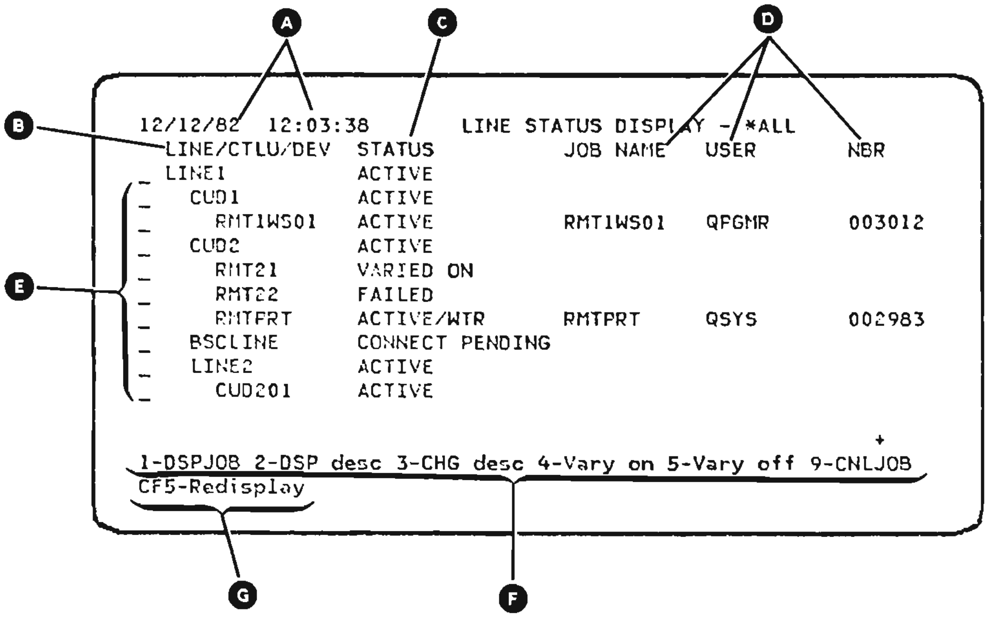
The CL commands used for determining communications status are Display Line Status (DSPLINSTS), Display Control Unit Status (DSPCTLSTS), and Display Device Status (DSPDEVSTS). These commands will aid you in quickly determining the status of lines, control units, and devices that make up your communications network. The default for the first parameter is *ALL, which displays all lines, control units, or devices that make up your network. If you want to view a specific component, use its name in place of the default *ALL. The second parameter allows you to display or print the status.

Display Line Status Command

If you enter

DSPLINSTS

you receive the following display:



```
12/12/82  12:03:38        LINE STATUS DISPLAY - *ALL
 LINE/CTLU/DEV  STATUS              JOB NAME  USER      NBR
_ LINE1          ACTIVE
_   CUD1         ACTIVE
_     RMT1WS01   ACTIVE              RMT1WS01  QPGMR     003012
_   CUD2         ACTIVE
_     RMT21      VARIED ON
_     RMT22      FAILED
_     RMTPRT     ACTIVE/WTR          RMTPRT    QSYS      002983
_   BSCLINE      CONNECT PENDING
_   LINE2        ACTIVE
_     CUD201     ACTIVE
                                                              +
1-DSPJOB 2-DSP desc 3-CHG desc 4-Vary on 5-Vary off 9-CNLJOB
CF5-Redisplay
```

A *Date/Time*: This is the current job date and time.

B *Line/CTLU/Dev*: This is the name of the line, control unit, or device. The line name is first; the control units and devices attached to the line are indented and listed below the line name.

C *Status*: This is the status of the object. One of the following values can appear here:

- ACTIVE: The line, control unit, or device is currently in use. For a display type device, the device is signed on or has been allocated by a batch, auto-start, or interactive job.

- ACTIVE/RDR: A spool reader is using this device.

- ACTIVE/WTR: A spool writer is using this device.

- CONNECT PENDING: A VRYLIN command has been issued for this line, and the system is waiting for an action to be completed, such as a switched connection being made.

- DIAGNOSTIC MODE: The line, control unit, or device is being serviced or has otherwise been set to diagnostic mode.

- FAILED: The line, control unit, or device is in an unusable state and can possibly be made usable again by varying it off then on again. A failed device may still be allocated to a job.

- FAILED/RDR: This device, which is in an unusable state, is still allocated to a spool reader type job.

- FAILED/WTR: This device, which is in an unusable state, is still allocated to a spool writer type job.

- POWERED OFF: The control unit or device is in a varied-off and powered-off state.

- SIGNON DISPLAY: This display type device currently has the *Enter password to signon* screen displayed.

- SYSREQ: The system has requested this display type device and the job associated with this status does not have a lock on the device. SYSREQ status only exists with another status for this device of ACTIVE or SIGNON DISPLAY.

- VARIED OFF: For a control unit or device that can be powered on or off by the PWRCTLU or PWRDEV command, the status is indicated after the control unit or device is powered on. For a line, this indicates that the line is varied off.

- VARIED ON. The line, control unit, or device is varied online, although it may not be physically powered on.

- VARY ON PENDING: A VRYCTLU or VRYDEV command has been issued for this control unit or device, respectively, and the system is waiting for an action to be completed, such as a switched connection being made.

- *DAMAGED: The line, control unit, or device has incurred hard or partial damage and it is not possible to obtain any more information about its status.

- *LOCKED: The line, control unit, or device is allocated to another job with an *EXCL lock, and its attributes cannot be determined at this time.

- UNKNOWN: All of the status bits for the line, control unit, or device have been checked, and none are set. This is an exceptional VMC condition.

Ⓓ *Job Name/User/Number*: This is the job name, user, and number of the job that is using the device. These will be displayed for ACTIVE devices only.

Ⓔ *Input Field*: This is a one-character input field where you can enter the options described in Ⓕ.

Ⓕ *Options*: The values can be entered in the input field to perform the following:

1 DSPJOB: Executes the DSPRDR command for a reader type job, executes the DSPWTR command for a writer type job, executes the DSPJOB command for all other type jobs, or returns *Job...not found* if there is no job associated with the input record.

2 DSP desc: Executes the DSPLIND command for a line, executes the DSPCUD command for a control unit, or executes the DSPDEVD command for a device.

3 CHG desc: Prompts for the CHGLIND command for a line, prompts for the CHGCUD command for a control unit, or prompts for the CHGDEVD command for a device.

4 Vary on: For a line, the line is varied on, all attached control units are varied on, and all attached devices are varied on.

For a control unit, the control unit is varied on and all attached devices are varied on.

For a device, the device is varied on.

5 Vary off: For a line, all attached devices are varied off, if possible, all attached control units are varied off, if possible, and the line is varied off, if possible.

For a control unit, all attached devices are varied off, if possible, and the control unit is varied off, if possible.

For a device, the device is varied off.

Note: As with entering the VRYxx commands individually, there is a noticeable delay when varying off an object that is already varied off.

9 CNLJOB: Executes the CNLRDR command for a reader type job, executes the CNLWTR command for a writer type job, executes the CNLJOB command for all other type jobs, or returns *Job...not found* if there is no job associated with the input record.

When all of the commands have been executed, the display is reshown with the status fields of the objects updated and with any error messages that occurred when the commands were executed.

If the configuration has more elements than can be shown on a single display, the Roll Up key can be used to display them all. Numbers can be placed in the input fields on multiple displays before the Enter key is pressed.

After the commands have executed, if there are more error messages than can fit on that display, a + is shown at the end of the last message. You must position the cursor at the first message and use the Roll Up key to view all of the error messages.

G *CF Keys*:

- CF1: Exit to the basic working display. CF1 causes you to exit from your current display and return to a basic working display. A basic working display is a request entry display like the command entry display or system operator menu display. CF1 is considered a request for *normal but immediate exit*. It requests complete exit from the original primary request from the basic working display even though you may be many layers down in the function or subfunctions.

  New requests for performance of additional functions that are entered at the same time as CF1 are ignored. This allows you to get out even if you are unable to enter valid information on the current display. In summary:

  1. It is always possible to get out via CF1.

  2. CF1 is not a backup key. It is a request to exit the total application, utility, or primary function.

  3. The exit is normal in that all files are closed and cleaned up.

  4. The exit is abnormal in that processing not yet encountered will be bypassed.

- CF2: Return to the previous display. CF2 causes a backup to the previous display. New requests that are entered at the same time that CF2 was pressed are ignored. If the current display was requested from another display or followed automatically from another display when the Enter key was pressed, CF2 causes a return to that previous display.

  If the current display was requested from a menu or selection display, CF2 causes a return to that menu or selection display.

  The continued pressing of CF2 results in a return to the basic working request entry display. CF2 always moves toward the basic working display.

- CF5: Redisplay. CF5 causes the current display to be reshown with the most current data available. This allows the user to, in effect, rerequest the display without exiting and reentering the request for it.

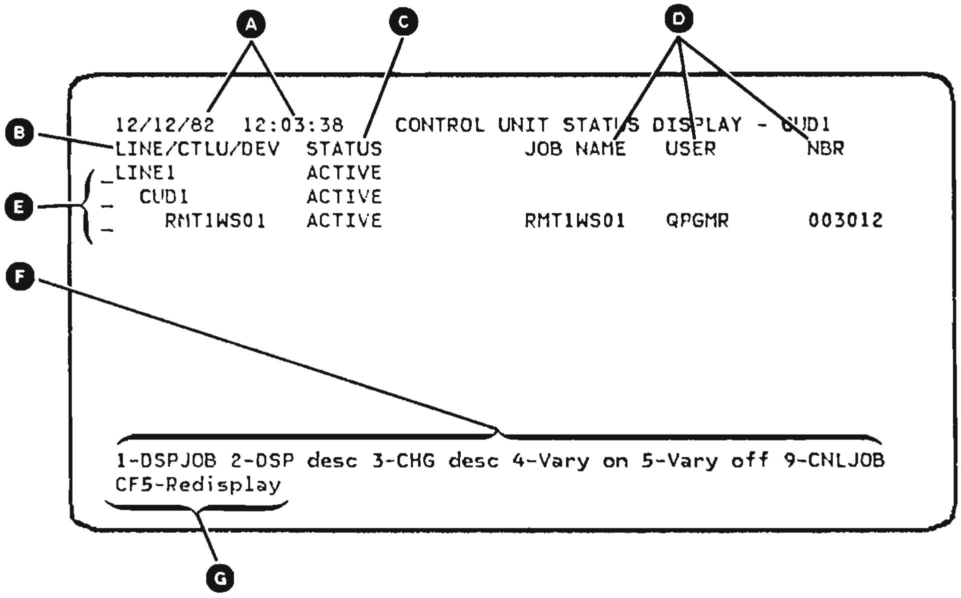
  If the display is rollable, the redisplayed display is positioned on entry one.

Display Control Unit Status

If you enter

DSPCTLSTS CUD1

you receive the following display:



```
12/12/82  12:03:38    CONTROL UNIT STATUS DISPLAY - CUD1
LINE/CTLU/DEV  STATUS            JOB NAME   USER       NBR
_LINE1         ACTIVE
_  CUD1        ACTIVE
_    RMT1WS01  ACTIVE            RMT1WS01   QPGMR      003012




1-DSPJOB 2-DSP desc 3-CHG desc 4-Vary on 5-Vary off 9-CNLJOB
CF5-Redisplay
```

A *Date/Time*: This is the current job date and time.

B *Line/CTLU/Dev*: This is the name of a line, control unit, or device. The line name is displayed first. The control unit and devices attached to the control unit are indented and listed below the line name.

Ⓒ *Status*: This is the status of the object. One of the following values can appear here:

- ACTIVE: The line, control unit, or device is currently in use. For a display type device, the device is signed on or has been allocated by a batch, auto-start, or interactive job.

- ACTIVE/RDR: A spool reader is using this device.

- ACTIVE/WTR. A spool writer is using this device.

- CONNECT PENDING: A VRYLIN command has been issued for this line, and the system is waiting for an action to be completed, such as a switched connection being made.

- DIAGNOSTIC MODE: The line, control unit, or device is being serviced or has otherwise been set to diagnostic mode.

- FAILED: The line, control unit, or device is in an unusable state, and can possibly be made usable again by varying it off then on again. A failed device may still be allocated to a job.

- FAILED/RDR: This device, which is in an unusable state, is still allocated to a spool reader type job.

- FAILED/WTR: This device, which is in an unusable state, is still allocated to a spool writer type job.

- POWERED OFF: The control unit or device is in a varied-off and powered-off state.

- SIGNON DISPLAY: This display type device currently has the *Enter password to signon* screen displayed.

- SYSREQ: The system has requested this display type device. The job associated with this status does not have a lock on the device. SYSREQ status only exists with another status for this device of ACTIVE or SIGNON DISPLAY.

- VARIED OFF: For a control unit or device that can be powered on or off by the PWRCTLU or PWRDEV command, the status is indicated after the control unit or device is powered on. For a line, this indicates that the line is varied off.

- VARIED ON: The line, control unit, or device is varied online, although it may not be physically powered on.

- VARY ON PENDING: A VRYCTLU or VRYDEV command has been issued for this control unit or device, respectively, and the system is waiting for an action to be completed, such as a switched connection being made.

- DAMAGED: The line, control unit, or device has incurred hard or partial damage and it is not possible to obtain any more information about its status.

- LOCKED: The line, control unit, or device is allocated to another job with an *EXCL lock, and its attributes cannot be determined at this time.

- UNKNOWN: All of the status bits for the line, control unit, or device have been checked, and none are set. This is an exceptional VMC condition.

Ⓓ *Job Name/User/Number*: This is the job name, user, and job number of the job that is using the device. These will be displayed for ACTIVE devices only.

Ⓔ *Input field*: This is a one-character input field where you can enter the options described in Ⓕ.

Ⓕ *Options*: These are the options that can be entered in the input field to perform the following:

1 DSPJOB: Executes the DSPRDR command for a reader type job, executes the DSPCUD command for a control unit, or executes the DSPDEVD command for a device.

2 DSP desc: Executes the DSPLIND command for a line, executes the DSPCUD command for a control unit, or executes the DSPDEVD command for a device.

3 CHG desc: Prompts for the CHGLIND command for a line, prompts for the CHGCUD command for a control unit, or prompts for the CHGDEVD command for a device.

4 Vary on: For a line, the line is varied on, all control units are varied on, and all attached devices are varied on.

For a control unit, the control unit is varied on and all attached devices are varied on.

For a device, the device is varied on.

5 Vary off: For a line, all attached devices are varied off, if possible, all attached control units are varied off, if possible, and the line is varied off, if possible.

For a control unit, all attached devices are varied off, if possible, and the control unit is varied off, if possible.

For a device, the device is varied off.

Note: As with entering the VRYxxx commands individually, there is a noticeable delay when varying off an object that is already varied off.

9 CNLJOB: Executes the CNLRDR command for a reader type job or executes the CNLWTR command for a writer type job.

Ⓖ *CF Keys:*

- CF1: Exit to the basic working display. CF1 causes you to exit from your current display and return to a basic working display. A basic working display is a request entry display like the command entry display or system operator menu display. CF1 is considered a request for *normal but immediate exit*. It requests complete exit from the original primary request from the basic working display even though you may be many layers down in the function or subfunctions.

  New requests for performance of additional functions that are entered at the same time as CF1 are ignored, this allows you to get out even if you are unable to enter valid information on the current display. In summary:

  1. It is always possible to get out via CF1.

  2. CF1 is not a backup key. It is a request to exit the total application, utility, or primary function.

  3. The exit is normal in that all files are closed and cleaned up.

  4. The exit is abnormal in that processing not yet encountered will be bypassed.

- CF2: Return to the previous display. CF2 causes a backup to the previous display. New requests that are entered at the same time that CF2 was pressed are ignored. If the current display was requested from another display or followed automatically from another display when the Enter key was pressed, CF2 causes a return to that previous display.

  If the current display was requested from a menu or selection display, CF2 causes a return to that menu or selection display.

  The continued pressing of CF2 results in a return to the basic working request entry display. CF2 always moves toward the basic working display.

- CF5: Redisplay. CF5 causes the current display to be reshown with the most current data available. This allows the user to, in effect, rerequest the display without exiting and reentering the request for it.

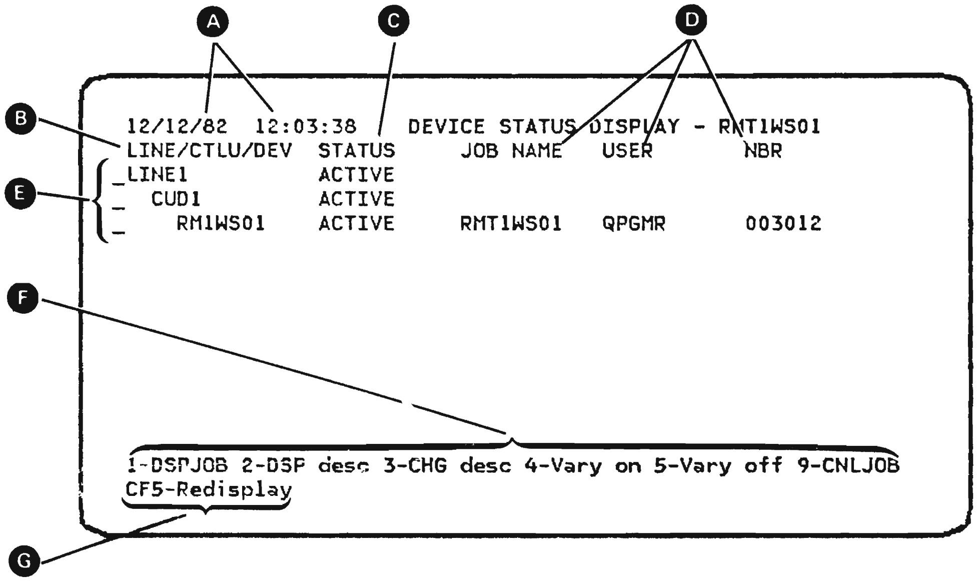
  If the display can be rolled, the redisplayed display is positioned on entry one.

Display Device Status

If you enter

DSPDEVSTS RMT1WS01

you receive the following display:



```
12/12/82  12:03:38    DEVICE STATUS DISPLAY - RMT1WS01
LINE/CTLU/DEV  STATUS      JOB NAME    USER       NBR
_LINE1         ACTIVE
_  CUD1        ACTIVE
_    RM1WS01   ACTIVE      RMT1WS01    QPGMR      003012




1-DSPJOB 2-DSP desc 3-CHG desc 4-Vary on 5-Vary off 9-CNLJOB
CF5-Redisplay
```

A *Date/Time*: This is the current job date and time.

B *Line/CTLU/Dev*: This is the name of a line, control unit, or device. The line name is displayed first, and the control units and devices that are attached to the control units are indented and listed below the line name.

C *Status*: This is the status of the object. One of the following can appear here:

- ACTIVE: The line, control unit, or device is currently in use. For a display type device, the device is signed on or has been allocated by a batch, auto-start, or interactive job.
- ACTIVE/RDR: A spool reader is using this device.
- ACTIVE/WTR. A spool writer is using this device.
- CONNECT PENDING: A VRYLIN command has been issued for this line, and the system is waiting for an action to be completed, such as a switched connection being made.
- DIAGNOSTIC MODE: The line, control unit, or device is being serviced or has otherwise been set to diagnostic mode.
- FAILED: The line, control unit, or device is in an unusable state, and can possibly be made usable again by varying it off then on again. A failed device may still be allocated to a job.
- FAILED/RDR: This device, which is in an unusable state, is still allocated to a spool writer type job.

- FAILED/WTR: This device, which is in an unusable state, is still allocated to a spool writer type job.

- POWERED OFF: The control unit or device is in a varied-off and powered-off state.

- SIGNON DISPLAY: This display type device currently has the *Enter password to signon* screen displayed.

- SYSREQ: This display type device has been system requested, and the job associated with this status does not have a lock on the device. SYSREQ status only exists with another status for this device of ACTIVE or SIGNON DISPLAY.

- VARIED OFF: For a control unit or device that can be powered on or off by the PWRCTLU or PWRDEV command, the status is indicated after the control unit or device is powered on. For a line, this indicates that the line is varied off.

- VARIED ON: The line, control unit, or device is varied online, although it may not be physically powered on.

- VARY ON PENDING: A VRYCTLU or VRYDEV command has been issued for this control unit or device, respectively, and the system is waiting for an action to be completed, such as a switched connection being made.

- *DAMAGED: The line, control unit, or device has incurred hard or partial damage and it is not possible to obtain any more information about its status.

- *LOCKED: The line, control unit, or device is allocated to another job with an *EXCL lock, and its attributes cannot be determined at this time.

- *UNKNOWN: All of the status bits for the line, control unit, or device have been checked, and none are set. This is an exceptional VMC condition.

Ⓓ *Job Name/User/Number*: This is the job name, user, and number of the current job that is using the device. These are displayed for ACTIVE devices only.

Ⓔ *Input Field*: This is a one-character input field where you can enter the options described in Ⓕ.

F *Options*: These are the options that can be entered in the input field to perform the following:

1 DSPJOB: Executes the DSPRDR command for a reader type job, executes the DSPWTR command for a writer type job, executes the DSPJOB command for all other type jobs, or returns *Job...not found* if there is no job associated with the input record.

2 DSP desc: Executes the DSPLIND command for a line, executes the DSPCUD command for a control unit, or executes the DSPDEVD command for a device.

3 CHG desc: Prompts for the CHGLIND command for a line, prompts for the CHGCUD command for a control unit, or prompts for the CHGDEVD command for a device.

4 Vary on: For a line, the line is varied on, all control units are varied on, and all attached devices are varied on.

For a control unit, the control unit is varied on and all attached devices are varied on.

For a device, the device is varied on.

Note: When you enter option 4 for a line or control unit from the DSPDEVSTS display, an attempt will be made to vary on all attached control units and devices. This allows varying on an entire network from the DSPDEVSTS display when DSPDEVSTS device-name is entered, although only a single line, control unit, and device is displayed. Messages indicating actions taken will be displayed at the bottom of the display.

5 Vary off: For a line, all attached devices are varied off, if possible, all attached control units are varied off, if possible, and the line is varied off, if possible.

For a control unit, all attached devices are varied off, if possible, and the control unit is varied off, if possible.

For a device, the device is varied off.

Notes:

1. As with entering the VRYxxx commands individually, there is a noticeable delay when varying off an object that is already varied off.
2. When you enter option 5 for a line or control unit from the DSPDEVSTS display, an attempt will be made to vary off all attached control units and devices. This allows varying off an entire network from the DSPDEVSTS display when DSPDEVSTS device-name is entered, although only a single line, control unit, and device is displayed. Messages indicating actions taken will be displayed at the bottom of the display.

9 CNLJOB: Executes the CNLRDR command for a reader type job or executes the CNLWTR command for a writer type job.

If the configuration has more elements than can be shown on a single display, the Roll Up key can be used to display them all. Numbers can be placed in the input fields on multiple displays before the Enter key is pressed.

When all of the commands have been executed, the display is reshown with the status fields of the objects updated and with any error messages that occurred when the commands were executed. If there are more error messages than can fit on that display, a + is shown at the end of the last message. You must position the cursor at the first message and use the Roll Up key to view all of the error messages.

G *CF Keys*:

- CF1: Exit to the basic working display. CF1 causes you to exit from your current display and return to a basic working display. A basic working display is a request entry display like the command entry display or system operator menu display. CF1 is considered a request for *normal but immediate exit*. It requests complete exit from the original primary request from the basic working display even though you may be many layers down in the function or subfunctions.

  New requests for performance of additional functions that are entered at the same time as CF1 are ignored. This allows you to get out even if you are unable to enter valid information on the current display. In summary:

  1. It is always possible to get out via CF1.
  2. CF1 is not a backup key. It is a request to exit the total application, utility, or primary function.
  3. The exit is normal in that all files are closed and cleaned up.
  4. The exit is abnormal in that processing not yet encountered will be bypassed.

- CF2: Return to the previous display. CF2 causes a backup to the previous display. New requests that are entered at the same time that CF2 was pressed are ignored. If the current display was requested from another display or followed automatically from another display when the Enter key was pressed, CF2 causes a return to that previous display.

  If the current display was requested from a menu or selection display, CF2 causes a return to that menu or selection display.

  The continued pressing of CF2 results in a return to the basic working request entry display. CF2 always moves toward the basic working display.

- CF5: Redisplay. CF5 causes the current display to be reshown with the most current data available. This allows the user to, in effect, rerequest the display without exiting and reentering the request for it.

  If the display can be rolled the redisplayed display is positioned on entry one.

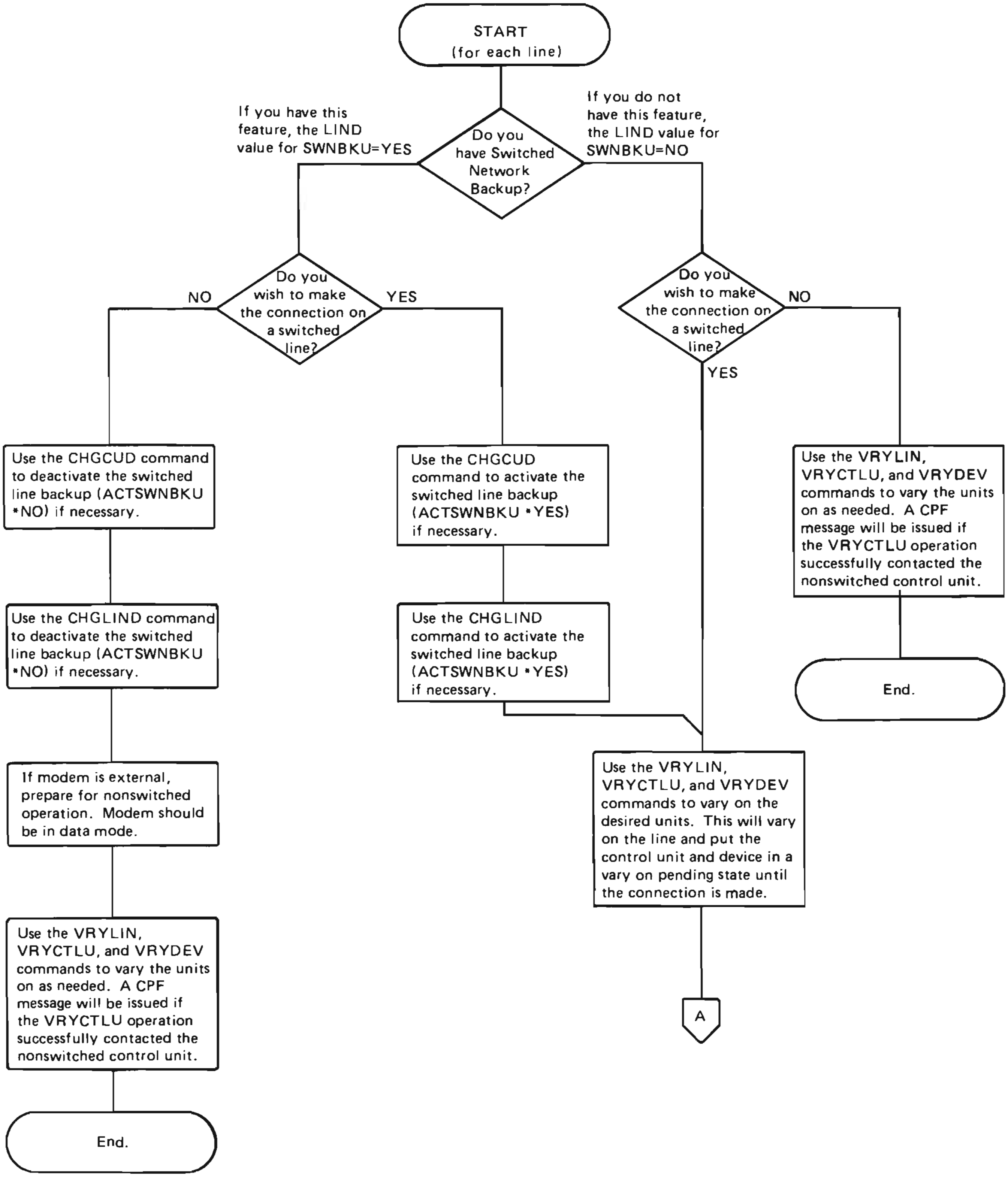
  If the display is one of many different displays for that function, the current display is redisplayed.

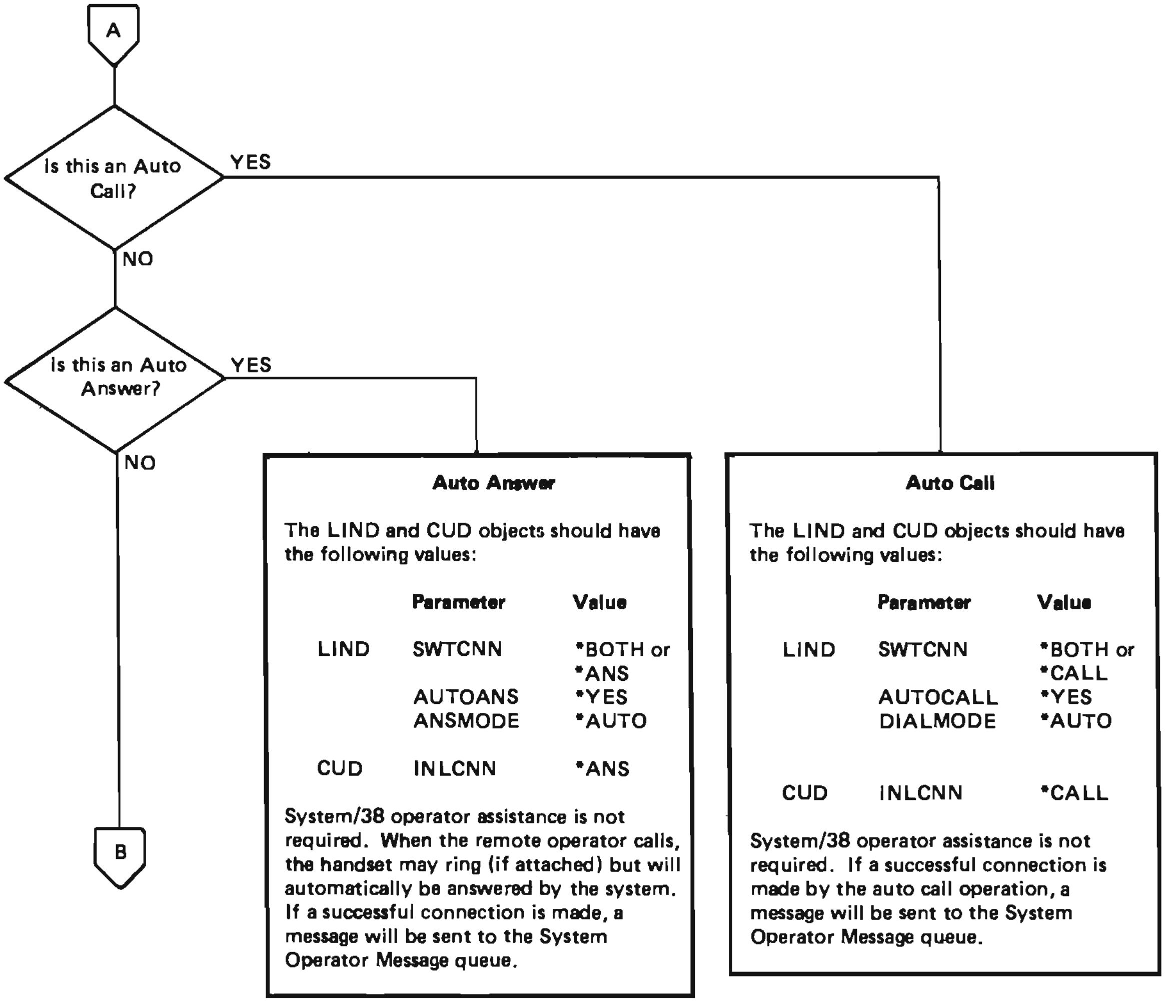
Operator Procedures

The flowchart on the following pages describes the general operating procedures for handling switched and nonswitched communications lines. The procedures for going to data mode and use of the handset depend upon the type of network and modems present on your system.

The flowchart procedures apply to one communication line (LIND object) and one or more controllers (CUDs) associated with the line. The flowchart should be reentered for each line on the system.

Note: Switched or switched backup controllers can be associated with more than one line. This configuration can be detected by using the Communications Display Status commands.


START (for each line)
Do you have Switched Network Backup?
If you have this feature, the LIND value for SWNBKU=YES
If you do not have this feature, the LIND value for SWNBKU=NO
Do you wish to make the connection on a switched line?
NO
YES
Do you wish to make the connection on a switched line?
NO
YES
Use the CHGCUD command to deactivate the switched line backup (ACTSWNBKU *NO) if necessary.
Use the CHGLIND command to deactivate the switched line backup (ACTSWNBKU *NO) if necessary.
If modem is external, prepare for nonswitched operation. Modem should be in data mode.
Use the VRYLIN, VRYCTLU, and VRYDEV commands to vary the units on as needed. A CPF message will be issued if the VRYCTLU operation successfully contacted the nonswitched control unit.
End.
Use the CHGCUD command to activate the switched line backup (ACTSWNBKU *YES) if necessary.
Use the CHGLIND command to activate the switched line backup (ACTSWNBKU *YES) if necessary.
Use the VRYLIN, VRYCTLU, and VRYDEV commands to vary on the desired units. This will vary on the line and put the control unit and device in a vary on pending state until the connection is made.
A
Use the VRYLIN, VRYCTLU, and VRYDEV commands to vary the units on as needed. A CPF message will be issued if the VRYCTLU operation successfully contacted the nonswitched control unit.
End.



Auto Answer

The LIND and CUD objects should have the following values:

	Parameter	Value
LIND	SWTCNN	*BOTH or *ANS
	AUTOANS	*YES
	ANSMODE	*AUTO
CUD	INLCNN	*ANS

System/38 operator assistance is not required. When the remote operator calls, the handset may ring (if attached) but will automatically be answered by the system. If a successful connection is made, a message will be sent to the System Operator Message queue.

Auto Call

The LIND and CUD objects should have the following values:

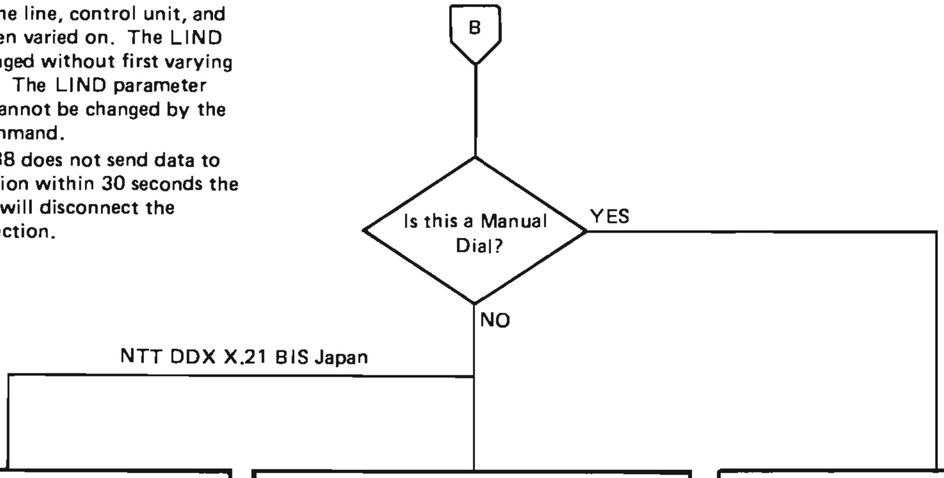
	Parameter	Value
LIND	SWTCNN	*BOTH or *CALL
	AUTOCALL	*YES
	DIALMODE	*AUTO
CUD	INLCNN	*CALL

System/38 operator assistance is not required. If a successful connection is made by the auto call operation, a message will be sent to the System Operator Message queue.

Note: At this point the line, control unit, and device have been varied on. The LIND cannot be changed without first varying everything off. The LIND parameter AUTOCALL cannot be changed by the CHGLIND command. The LIND object must be deleted and recreated.

Notes:

1. At this point the line, control unit, and device have been varied on. The LIND cannot be changed without first varying everything off. The LIND parameter AUTOCALL cannot be changed by the CHGLIND command.
2. If the System/38 does not send data to the remote station within 30 seconds the remote station will disconnect the switched connection.



Manual Answer

The LIND and CUD objects should have the following values:

	Parameter	Value
LIND	SWTCNN	*ANS
	ANSMODE	*MANUAL
CUD	INLCNN	*ANS

If these values must be changed see Note 1.

Procedure:

1. Enter the ANSLIN command with the correct LINE ( ) parameter.
2. CPF message 5956 will be sent to the System Operator message queue instructing you to reply C (cancel) or G (go).
3. Reply G (go) to CPF 5696. Do not press the answer button on the data communications equipment at this time.
4. When an incoming call arrrives, as indicated by a signal from the data communications equipment (DCE), press the answer button on the DCE within 60 seconds.
5. If a successful connection has been made, a message will be sent to the System Operator message queue.

Manual Answer

The LIND and CUD objects should have the following values:

	Parameter	Value
LIND	SWTCNN	*BOTH or *ANS
	ANSMODE	*MANUAL
CUD	INLCNN	*ANS

If these values must be changed see Note 1.

Procedure:

1. When the remote equipment operator calls, the System/38 operator answers the call manually by lifting the handset and going to talk mode.
2. Verify with the remote equipment operator that the correct connection has been made.
3. Enter the ANSLIN command with the correct LINE( ) parameter.
4. CPF message 5956 will be sent to the System Operator Message queue instructing you to go to data mode and reply C (cancel) or G (go).
5. Instruct the remote equipment operator to go to data mode.
6. Reply G (go) to CPF 5956 and go to data mode (see Note 2).
7. If a successful connection has been made, a message will be sent to the System Operator Message queue.

Manual Dial

The LIND and CUD objects should have the following values:

	Parameter	Value
LIND	SWTCNN	*BOTH or *CALL
	DIALMODE	*MANUAL
CUD	INLCNN	*CALL

If these values must be changed see Note 1.

Procedure:

If a program requests a switched remote device, message CPF5954 is sent to the System Operator Message queue.

If a subsystem is started that contains a work station entry for a switched remote device, message CPF5954 will not appear. To make the switched connection, follow the manual dial procedure after the subsystem has been started.

1. If you have a handset, lift the handset and place a call to the remote number.
   a. If the remote number is manually answering, verify that the correct remote location has been reached and instruct the remote equipment operator to go to data mode.
   b. If the remote location is auto answering, you will hear a high pitched answer tone. The remote location will go to data mode automatically after sending the answer tone.
2. If message CPF5954 is displayed, reply G (go) and go to data mode. If message CPF5954 is not displayed, go directly to data mode.
3. If a successful connection is made, a message will be sent to the System Operator Message queue.

Manual Dial and Manual Answer

Manual Dial

You establish the line connection by the following procedure.

1. When you receive a message such as

   ```
   Manually dial 1234567 for control unit CTLU02 line LINE02 (C,G)
   ```

   dial the indicated phone number on the indicated line (in the example message shown, the number is 123-4567 and the line is LINE02).

2. When the call is answered, enter G in the underlined input field, wait 5 seconds and go to data mode.

When the manual dial sequence is completed, another message will be sent to the system operator message queue, indicating whether the connection was successfully established.

Manual Answer

You establish the line connection by the following procedure.

1. When the phone connected to the data line rings, answer it and verify the validity of the caller, if possible.

2. Enter the Answer Line (ANSLIN) command at your work station, specifying the line name in the LINE parameter. For example, if the line name is LINE02, enter

   ```
   ANSLIN LINE(LINE02)
   ```

3. When you receive the message

   ```
   Place modem for line LINE02 in data mode (C,G)
   ```

   Enter G (go) in the underlined input field on the line below the message and place the modem in the data mode. You can also cancel the operation by entering C instead of G. After you enter your reply, the system operator menu returns to your screen.

When the manual answer sequence is completed, another message will be sent to the system operator message queue indicating whether the connection was successfully established.

Manual Answer for NTT DDX X.21 BIS for Japan

1. Enter the ANSLIN command at your work station, specifying the line name in the LINE parameter. For example, if the line name is LINE02, enter

   ```
   ANSLIN LINE(LINE02)
   ```

2. When you receive the message

   ```
   Place modem for LINE02 in data mode (C,G)
   ```

   enter a G in the underlined input field. Do not press the answer button on the data communications equipment at this time.

3. When an incoming call is indicated by a signal from the data communications equipment, press the answer button within 60 seconds.

4. If a successful connection has been made, a message will be sent to the system operator message queue.

FCC Regulations

FCC (Part 68) regulations govern the Operation and Maintenance of IBM 2400 bps and 4800 bps Integrated Modems (Sales Features 5641 and 5741) over the Public Switched Network in the U.S.A. Data protective circuitry in the IBM 2400 bps and the IBM 4800 bps Integrated Modems is registered with the U.S. Federal Communications Commission (FCC). The FCC requires that IBM publish the following information about the operation and maintenance of all devices containing registered circuitry. The FCC also requires that all persons who use and maintain devices containing registered circuitry in the U.S.A. comply with the following FCC rules.

1. To ensure compliance with the regulations regarding the repair of registered circuitry, field repair is limited to replacement of field-replaceable units only. Field replacement of these units will be performed by authorized personnel only. No other field repair or adjustment to the registered circuitry is permitted.

2. If a problem occurs that is determined to be in registered circuitry or to be caused by registered circuitry, the device containing that circuitry must be disconnected from the public switched network immediately by removing its System/38 cable from its network data jack.

   If the telephone company detects a problem on the network that has been caused by System/38, the telephone company has the right to temporarily discontinue service to the device containing that circuitry. Service may be discontinued immediately and remain discontinued until the problem no longer exists.

3. The telephone company may make changes in its communications facilities, equipment, operations, or procedures, where such action is reasonably required in the operation of its business and is not inconsistent with the rules and regulations of the FCC. If these changes can be reasonably expected to cause your equipment to be incompatible with the telephone company's communications facilities, or to require modification or alteration of your equipment, you will be given adequate advance notice, in writing, to allow you an opportunity to react in a manner that allows you to maintain uninterrupted service.

4. You must notify the telephone company upon final disconnection of a device containing registered circuitry, identifying the line from which the device was disconnected.

Chapter 10. Spooling Operations

Through spooling, the system can receive input from multiple sources, and produce output on multiple devices in an efficient manner. Both interactive jobs at a work station and batch jobs share system resources, such as a printer, without any apparent effect on the timely completion of the job functions.

Your role in spooling operations is primarily to start the spooling subsystem and to start the various readers and writers as needed. You start the subsystem and the readers and writers by using commands. These commands could be combined into a CL program that is invoked by a single command. The readers and writers that are regularly used on your system could be defined as an autostart job in the spooling subsystem description so that they are automatically started for you when the subsystem is started. The procedures for creating CL programs and defining autostart jobs are described in the *CPF Programmer's Guide*. The discussions in this chapter assume that you are entering a separate command for each spooling function.

SPOOLING SUBSYSTEM

The spooling subsystem (QSPL) provides the operating environment for readers and writers that transfer jobs into the system for processsing and out of the system after processing. It is one of the basic subsystems supplied with the system.

Jobs can also be submitted to the system by using the Submit Jobs commands (see *Submit Jobs Commands* in Chapter 10 and *Using Submit Jobs Commands* in Chapter 12).

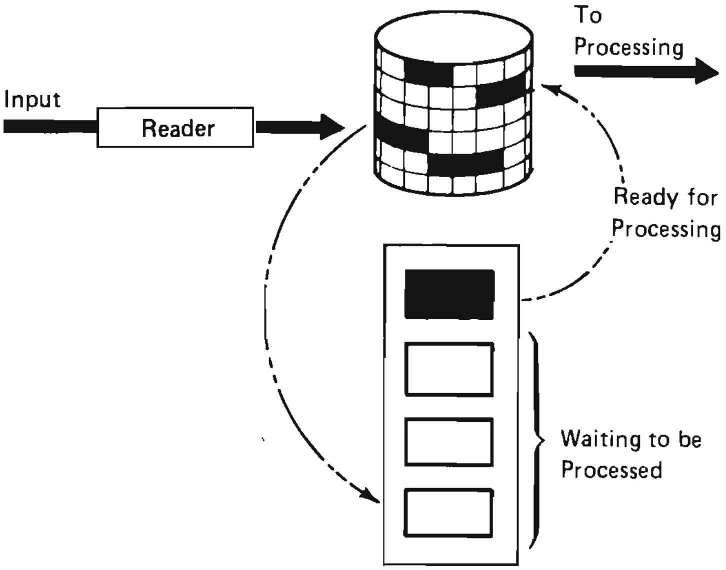
Readers and Job Queues

A *reader* is a CPF program that reads an input stream of job information into the system from an input source. The input stream consists of one or more batch jobs and any related inline data files to be processed. The input stream may be

- A card file (stack of cards) that has been placed in a card device such as the MFCU (see Chapter 4).
- A diskette file on one or more diskettes that have been placed in the diskette magazine drive (see Chapter 4).
- A data base file in internal storage.

More than one reader can be active at the same time, but a particular reader can read from only one input device and each input device must have a unique reader. The maximum number of active readers and writers is defined in the spooling subsystem description.

When a batch job is read from an input source by a reader, the commands in the input stream are stored in the system as requests for the job, the inline data is spooled as inline data files, and an entry for the job is placed on a job queue. The job information remains stored in the system where it was placed by the reader until the job entry is selected from the job queue for processing by a subsystem (see *Subsystems* in Chapter 8).



Job Queue

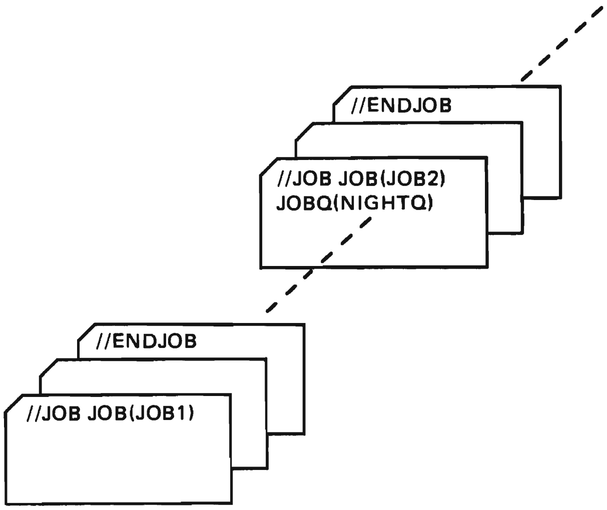
A *job queue* is an ordered list of jobs waiting to be processed by a particular subsystem. Job entries will not be selected from a job queue by a subsystem unless the subsystem is active and the job queue is not held. There is a job queue for each subsystem supplied with the system. The job queue is identified by the name associated with the subsystem; that is

- QBATCH: Job queue for the batch subsystem QBATCH
- QSPL: Job queue for the spooling subsystem QSPL
- QCTL: Job queue for the controlling subsystem QCTL
- QINTER: Job queue for the interactive subsystem QINTER
- QPGMR: Job queue for the programmer subsystem QPGMR

Other job queues may have been created for your system; for example,

- A job queue for long-running jobs.
- A job queue for overnight jobs.
- A job queue for high-priority jobs.

The name of the job queue on which a batch job is to be placed is indicated in the command that is used to submit the job. For jobs to be read in by a spooling reader, this command is the JOB command (//JOB). The JOB command indicates the beginning of each job in the input stream. If a user does not specify a job queue in the JOB command, the default value of *RDR is used, which means that the job is to be placed on the job queue specified in the start reader command. If the default was used in the JOB command and you do not specify a job queue when you enter the start reader command, the job is placed on the QBATCH job queue (the default for the job queue in all start reader commands is QBATCH). In the following example of a card input stream, JOB1 will be placed on the job queue specified in the start reader command because no job queue is specified in the JOB command. JOB2 will be placed on the job queue identified as NIGHTQ because that job queue is specified in the JOB command. Note that the end of each job in the input stream is indicated by the //ENDJOB command.



Specific procedures for using readers to submit batch jobs to the system for processing are given in *Submitting Batch Jobs* in Chapter 12.

Output Queues and Writers

The processing of batch and interactive jobs may result in spooled output records that are to be produced on an output device, such as a printer or a card punch. These output records are stored in *spooled output files* until they can be produced. There could be multiple spooled output files for a job.

When a spooled output file is created, an entry for the file is placed on an *output queue*. Each output queue contains an ordered list of entries for spooled output files. The spooled files for a job can have entries on one or more output queues. Different jobs can have entries on the same output queue. All of the entries on a particular output queue should have a common set of output attributes (characteristics), such as device, print image, forms type, translate table, and lines per inch. This will reduce the amount of intervention required from you and increase the device throughput. Five standard output queues are provided with the system when it is installed:

- QPRINT, for printing on one-part paper

- QPRINT2, for printing on two-part paper

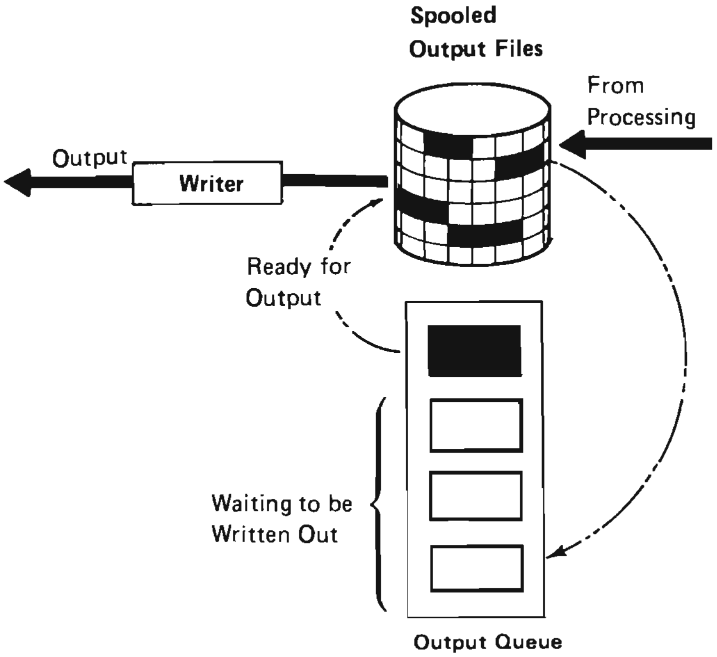
- QPRINTS, for printing on special forms

- QPUNCH, for punching on 96-column cards

- QDKT, for writing on diskette

Other output queues can be created for overnight (low-priority) output or large output files, or to protect the spooled output data.

The spooled output files that have been placed on a particular output queue will remain stored in the system until a *writer* is started to the output queue. A writer is a CPF program that takes spooled output files from an output queue and produces them on an output device.



The writer takes entries one at a time from the output queue, based on their priority (see *Input/Output Priorities* in this chapter). The priority for all spooled output files in a job is specified in the command that is used to submit the job (such as the JOB command), or in the job description (see *Job Descriptions* in Chapter 12).

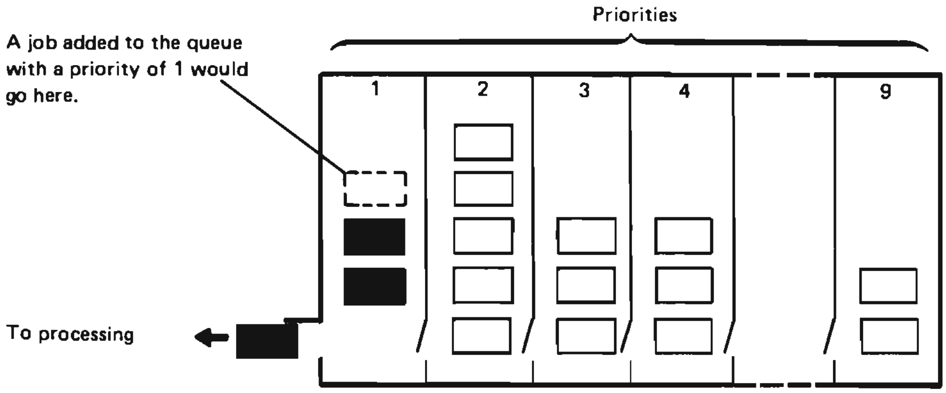
The writer will produce a spooled output file only if its entry on the output queue indicates that it has a ready (RDY) status. You can determine the status of a particular output file by requesting the output queue display (see *Output Queue Displays* in this chapter). If the spooled output file has a ready status, the writer takes the entry from the output queue and produces the specified job and/or file separators, followed by the output data in the file. If the output file does not have a ready status, the writer leaves the entry on the output queue and goes on to the next entry. In most cases the writer will continue to produce output files (preceded by job/file separators) until all entries with a ready status have been taken from the output queue. Based on the AUTOTRM parameter specified on the start writer command, the writer can continue to wait for new files to become available to be written, or the writer may terminate after it has produced one file, or it may terminate after all entries with a ready status have been taken from the output queue. For more details on job/file separators and writer termination, see *Writer Operating Considerations* in this chapter.

Input/Output Priorities

Entries for batch jobs submitted for processing are placed on a job queue and the entries for output files from processed jobs are placed on an output queue. Priorities are specified for each job that determine the order in which it will be taken from the queue. The *job priority* determines the order in which a job is selected from the job queue. The *output priority* determines the order in which a job's output files are selected from the output queue.

Both the job and output priority can be a number from 1 through 9, with the lowest number being the highest priority.

If jobs on a queue have an equal priority, the job with the earlier time stamp will be taken first.


Priorities
A job added to the queue with a priority of 1 would go here.
1
2
3
4
9
To processing

Starting and Terminating the Spooling Subsystem

You must start the spooling subsystem before spooling readers and writers can function. The spooling subsystem does not have to be started to execute the Submit Jobs commands. You start the subsystem by using the Start Subsystem (STRSBS) command; for example, by entering

```
STRSBS SBSD(QSPL.QGPL)
```

You can prevent any readers or writers from executing by terminating the subsystem using the Terminate Subsystem (TRMSBS) command:

```
TRMSBS SBS(QSPL)
```

When the spooling subsystem is not active, job streams cannot be read into the system by a reader, and spooled output files cannot be produced by a writer; however, jobs can still be placed on a job queue by using the Submit Job command or the Submit Jobs commands (see *Submitting Batch Jobs* in Chapter 12). Jobs already on a job queue will be taken from the queue according to their job priority and processed, if their subsystem is active, whether the spooling subsystem is active or not. Similarly, output from jobs being processed will be stored in spooled output files and placed on an output queue without the spooling subsystem being active. The output files will remain on the output queues until the spooling subsystem and the appropriate writer are started. They will then be taken off the queue according to the output priority established for the job and produced on an output device.

Subsystem/Queue Status at Power Down

When the system is powered down (either intentionally by the techniques described in Chapter 8 or unintentionally as the result of a power interruption), the spooling subsystem is terminated. This means that all readers and writers stop operating. The subsystem and the individual readers and writers must be restarted the next time the system is powered on.

Information on the job queues and output queues is not lost when power is shut off. When the system is started again, jobs on a job queue (unless they were on the job queue as the result of a TFRJOB) are still available for processing and files on an output queue can still be produced. When you receive the start CPF prompt as a part of starting the system, you have the option of keeping the information that is on the queues or clearing the information from the queues (see *Starting CPF* in Chapter 8).

Jobs which were on a job queue as the result of a TFRJOB command will be removed from the job queue and their job logs will be produced during IMPL.

Handling Damaged Job and Output Queues

If a job queue or output queue becomes damaged such that it cannot be used, you will be notified by a message sent to the system operator message queue (see *Message Queues* in Chapter 11). The message will come from a system function when a reader, Submit Jobs command, a writer, or a job tries to put entries on or take entries from the damaged queue.

A damaged job or output queue is automatically deleted by the system during the next IMPL and start CPF process. You cannot delete a damaged job or output queue yourself. After the system is restarted, the deleted queue can be recreated by entering the Create Job Queue (CRTJOBQ) or Create Output Queue (CRTOUTQ) command.

Use the following procedure to recover from a damaged job queue or output queue:

1. If possible, warn all work station users that the system must be powered down temporarily and that they should sign off. You can do this by sending a break message to all work stations (see *Sending Break Messages* in Chapter 11).

2. Power down the system by using the PWRDWNSYS command (see *Powering Down the System* in Chapter 8).

3. Set the rotary switches on the operator/service panel for an IMPL Abbr, and press the Power On switch to restart the system.

4. When you receive the start CPF prompt, ensure that *KEEP is specified in both the *Job queue* and *Output queue* fields (see *Starting CPF* in Chapter 8). This step is necessary to ensure that you receive messages on the status of the queues and jobs or output files on the queues.

5. During the start CPF process, you receive messages at the system console that indicate which queue or queues were deleted. After normal system operations have resumed, contact the user who originally created the queue or the appropriate authority (such as the system security officer) and ask him to enter the command to recreate the queue.

Job information that was on a damaged job or output queue will be lost when the queue is deleted. You will also receive messages during the start CPF process that indicate which jobs or files were not completed. You or system users can also determine the status of a particular job by using the Display Job (DSPJOB) command (see *Specific Job Information* in Chapter 12). Jobs that were not processed and jobs whose spooled files were not produced may have to be resubmitted.

DETERMINING QUEUE STATUS

On occasion you may need to determine what jobs are on a job queue or what files are on an output queue, or if a particular job is on a particular queue. To find out, you request job queue and output queue displays.

Job Queue Displays

Job queue displays allow you to determine the overall status of all job queues or a particular job queue. From the display of all job queues you can display a specific job queue, hold a job queue, or release a job queue. From the display of a specific job queue, you can display a job, hold a job, release a job, or cancel a job.

Obtaining the Displays

To obtain a job queue display, enter the Display Job Queue (DSPJOBQ) command. When you select option 1 on the system operator menu, you are using this command. The command has only two parameters:

JOBQ Parameter: This parameter determines whether information is provided on all job queues or on a particular job queue.

- If you do not enter the parameter, JOBQ(*ALL) is assumed and the overall status of all job queues is shown.
- If you enter the parameter and specify a particular job queue name, such as JOBQ(QBATCH.QGPL) the status of each job on that job queue is shown.

Output Parameter: This parameter determines how the information is provided.

- If you do not enter the parameter, the information is displayed on your work station screen.
- If you specify OUTPUT(*LIST) the information is spooled for printing on the system printer and placed on the output queue associated with your interactive job.

Examples:

1. To display at your work station the status of all job queues, enter

   ```
   DSPJOBQ
   ```

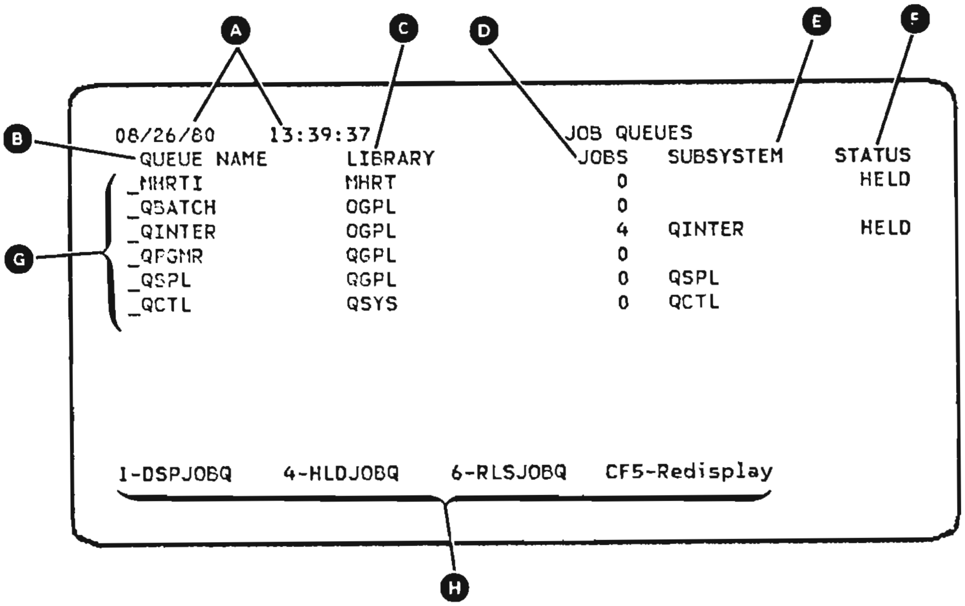
2. To print out the status of jobs on the batch subsystem job queue, enter

   ```
   DSPJOBQ JOBQ(QBATCH.QGPL) OUTPUT(*LIST)
   ```

To display a specific job queue, you can also enter option 1 on the line describing the particular job queue on the display of all job queues.

Interpreting the Display of All Job Queues

When you enter the DSPJOBQ command without the JOBQ parameter, you receive a display of all job queues. From this display, you can display all the jobs on a particular queue, hold a job queue, or release a job queue.



```
08/26/80     13:39:37                     JOB QUEUES
  QUEUE NAME       LIBRARY                  JOBS   SUBSYSTEM     STATUS
 _MHRTI            MHRT                      0                     HELD
 _QBATCH           OGPL                      0
 _QINTER           OGPL                      4     QINTER          HELD
 _QPGMR            QGPL                      0
 _QSPL             QGPL                      0     QSPL
 _QCTL             QSYS                      0     QCTL




1-DSPJOBQ     4-HLDJOBQ     6-RLSJOBQ     CF5-Redisplay
```

🅐 *Date/Time:* The date and the time shown are the job date and the system time when you requested the display.

🅑 *Queue Name:* This is the name by which each listed job queue is identified to the system.

🅒 *Library:* This is the library in which each job queue is located.

🅓 *Jobs:* This is the number of jobs waiting on each queue to be processed. If this is ever blank it is because the queue could not be allocated.

🅔 *Subsystem:* If a subsystem is currently using the job queue, its name appears here.

🅕 *Status:* There are two possible conditions:

- Held: The queue was held by the Hold Job Queue command. No jobs can be processed until the queue is released.

- Blank: If nothing is shown, the queue is not being held and jobs are available for processing.

🅖 *Input Field:* A one-character input field where you can enter the options described in 🅗.

If there are more lines of job queue information than can be shown on your screen at one time, a + appears to the right of the last line currently displayed. You obtain the additional lines by pressing the Roll Up key. If you press the CF5 key while the job queues display is on your screen, the display is reshown with the latest status of all job queues.

🅗 *Options:* These are the valid options that can be entered in the input field to perform the following:

1 DSPJOB: Displays a particular job queue.

4 HLDJOBQ: Holds a particular job queue.

6 RLSJOBQ: Releases a particular job queue.

Using the Job Queue Display

If you request the job queue display for all job queues, you can execute job queue control commands directly from that display. An input field is provided to the left of each job queue shown on the display. At the bottom of the display, three command names are shown, each preceded by a number. By entering one of these numbers in the input field next to a job queue name, you can execute the command associated with the number; the command is executed only for the job queue where you entered the number.

If you key a number into several input fields and then press the Enter key, the commands associated with those numbers are executed one at a time in the same order that the job queues are displayed. Note that if you enter a 1 (DSPJOBQ) for a particular job queue, you receive a separate display for that job queue. If you press the CF1 key while you are viewing a separate job queue display (requested through the DSPJOBQ function on the job queue display), you return directly to your basic working display, bypassing the job queues display. If you specified commands on the all job queues display for job queues that follow the job queue you were displaying, none of those commands are executed.

The following chart summarizes the functions you can perform directly from the all job queue display:

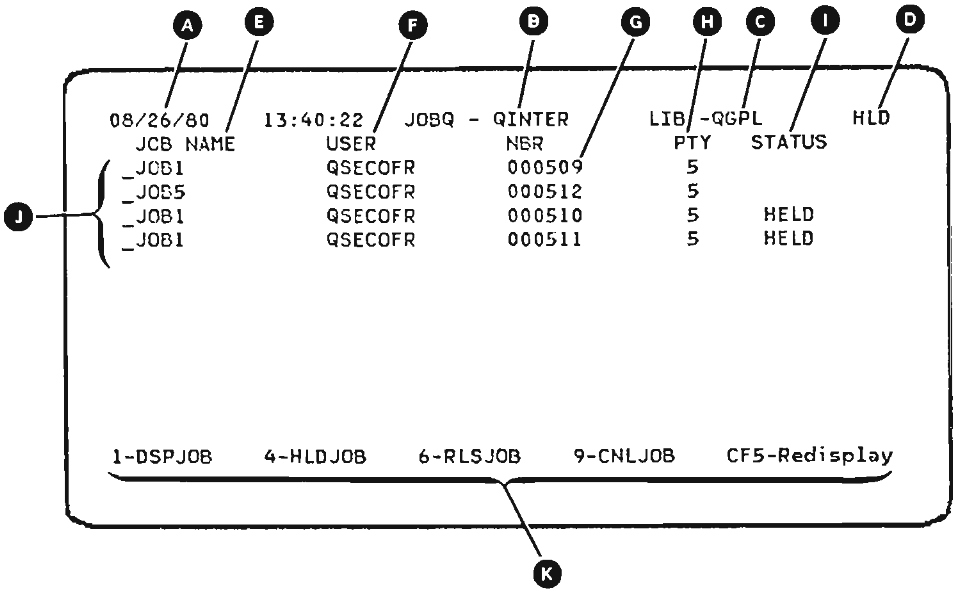
To do this:	Enter this:	For details, see:
Display a job queue (DSPJOBQ command)	1	*Interpreting a Specific Job Queue Display* in this chapter
Hold a job queue (HLDJOBQ command)	4	*Holding a Job Queue* in this chapter
Release a job queue file (RLSJOBQ command)	6	*Releasing a Job Queue* in this chapter

After the commands have been executed, the all job queues display is reshown with the current status of all job queues on the display, including the job queues whose status you changed by executing the HLDJOBQ or RLSJOBQ commands from the display. Any messages related to the execution of the commands from the display are shown at the bottom of the display. If there are more messages than can be shown on your screen at one time, a + appears to the right of the last message currently displayed. You obtain the additional messages by positioning the cursor on any of the messsage lines and pressing the Roll Up key.

When you are finished with the all job queues display, you return to your basic working display (such as the system operator menu) by pressing the Enter key or the CF1 key.

Interpreting a Specific Job Queue Display

When you enter the DSPJOBQ command and specify a job queue name in the JOBQ parameter, you receive a display of jobs on the specified job queue. From this display you can display a job, hold a job, release a job, or cancel a job.



```
08/26/80    13:40:22   JOBQ - QINTER        LIB -QGPL           HLD
  JCB NAME           USER          NBR         PTY   STATUS
 _JOB1             QSECOFR       000509         5
 _JOB5             QSECOFR       000512         5
 _JOB1             QSECOFR       000510         5      HELD
 _JOB1             QSECOFR       000511         5      HELD



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

A *Date/Time:* The date and the time shown are the job date and the system time when you reauested the display.

B *Queue Name:* This is the name of the job queue being displayed.

C *Lib:* This is the name of the library where the job queue is located.

D *Queue Status:* The following indicators can be shown here:

- SBS: A subsystem is active and using this job queue.
- SBS/HLD: A subsystem is active and using this queue but the queue is held.
- HLD: The job queue is held and is not being used by an active subsystem.
- Blank: The queue is not held and is not being used by an active subsystem.

Ⓔ *Job Name:* The name of each job on the queue is shown.

Ⓕ *User*: This is the name of the user profile from the job description used to start the job.

Ⓖ *Nbr:* This is the job number assigned to the job by the system when it entered the system. No other job in the system has this job number (see *Job Numbers* in Chapter 12).

Ⓗ *Pty:* This is the job priority; the lower the number, the higher the priority.

Ⓘ *Status:* There are two possible conditions:

- Held: The job is on the queue and ready to be processed, but was held. It will not be processed until it is released.

- Blank: If nothing is shown, the job is on the queue and will be processed when its turn comes.

Ⓙ *Input Field:* This is a one-character input field where you can enter the options described in Ⓚ.

Ⓚ *Options:* These are the valid options that can be entered in the input field to perform the following:

1 DSPJOB: Displays a particular job.

4 HLDJOB: Holds a particular job.

6 RLSJOB: Releases a particular job.

9 CNLJOB: Cancels a particular job.

If there are more lines of job information than can be shown on your screen at one time, a + appears to the right of the last line currently displayed. You obtain the additional lines by pressing the Roll Up key. If you press the CF5 key while the job queue display is on your screen, the display is reshown with the latest status of all jobs on the job queue.

Using the Job Queue Display

If you request the job queue display for a particular job queue, you can execute job control commands directly from that display. An input field is provided to the left of each job shown on the display. At the bottom of the display, four command names are shown, each preceded by a number. By entering one of these numbers in the input field next to a job name, you can execute the command associated with the number; the command is executed only for the job where you entered the number.

If you key a number into several input fields and then press the Enter key, the commands associated with those numbers are executed one at a time in the same order that the jobs are displayed. Note that if you enter a 1 (DSPJOB) for a particular job, you receive a separate display for that job. If you press the Enter key after viewing the display, you return to the job queue display. If you press the CF1 key while you are viewing a separate job display (requested through the DSPJOB function on the job queue display), you return directly to your basic working display, bypassing the job queue display. If you specified commands on the job queue display for jobs that follow the job you were displaying, none of those commands are executed.

The following chart summarizes the functions you can perform directly from the output queue display:

To do this:	**Enter this:**	**For details, see:**
Display the job (DSPJOB command)	1	DSPJOB command in *CL Reference Manual*
Hold a job (HLDJOB command)	4	*Holding a Job* in this chapter
Release a job (RLSJOB command)	6	*Releasing a Job* in this chapter
Cancel (remove) a job (CNLJOB command)	9	*Canceling a Job* in this chapter

After the commands have been executed, the job queue display is reshown with the current status of all jobs on the display, including the jobs whose status you changed by executing the HLDJOB or RLSJOB commands from the display. Any messages related to the execution of the commands from the display are shown at the bottom of the display. If there are more messages than can be shown on your screen at one time, a + appears to the right of the last message currently displayed. You obtain the additional messages by positioning the cursor on any of the message lines and pressing the Roll Up key.

When you are finished you press the Enter key or the CF1 key to return to the display from which you requested the specific job queue display. If you press the CF1 key while you are viewing a separate job display (requested through the DSPJOB function on the job queue display), you return directly to your basic working display, bypassing the job queue display. If you specified commands on the job queue display for jobs that follow the job you were displaying, none of those commands are executed.

Output Queue Displays

Output queue displays allow you to determine the overall status of all output queues or a particular output queue. From the display of all output queues you can display a specific queue, hold an output queue, or release on output queue. From the display of a specific output queue, you can display, hold, release, or cancel particular spooled output files.

Obtaining the Displays

To obtain an output queue display, enter the Display Output Queue (DSPOUTQ) command. When you select option 2 on the system operator menu, you are using this command. The command has only two parameters:

OUTQ Parameter: This parameter determines whether information is provided on all output queues or a particular output queue.

- If you do not enter the parameter, OUTQ(*ALL) is assumed and the overall status of all output queues is shown.
- If you enter the parameter and specify a particular output queue name, such as OUTQ(QPRINT.QGPL) the status of each file on that output queue is shown.

OUTPUT Parameter: This parameter determines how the information is provided.

- If you do not enter the parameter, the information is displayed on the screen of your work station.
- If you specify OUTPUT(*LIST) the information is spooled for printing on the system printer and placed on the output queue associated with your job.

Examples:

1. To display at your work station the status of all output queues, enter

```
DSPOUTQ
```

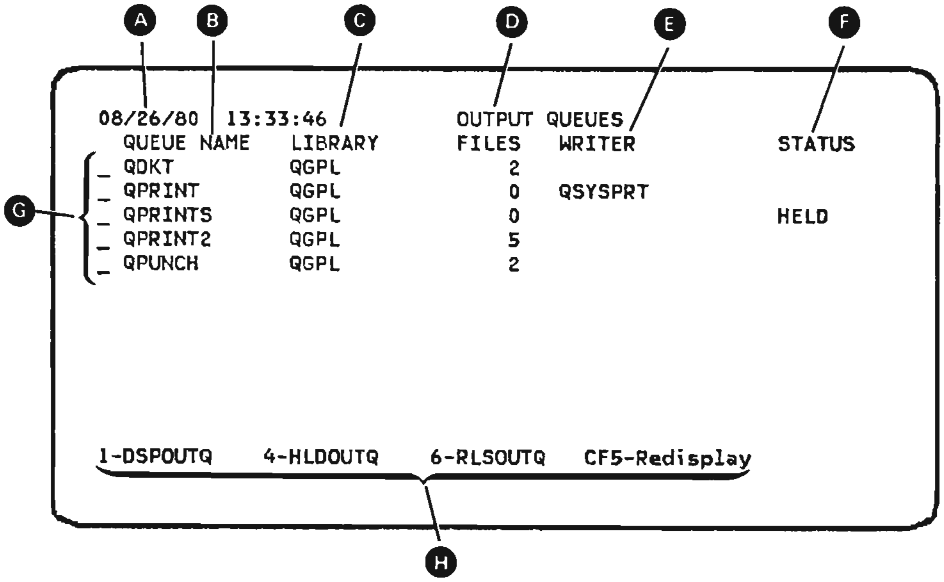
2. To print out the status of files on the QPRINT output queue, enter

```
DSPOUTQ OUTQ(QPRINT) OUTPUT(*LIST)
```

To display a specific output queue, you can also enter option 1 from the display-all-output-queues display.

Interpreting the Display of All Output Queues

When you enter the DSPOUTQ command without the OUTQ parameter, you receive a display of all output queues. From this display you can display a particular output queue, hold an output queue, or release an output queue.



```
08/26/80  13:33:46          OUTPUT QUEUES
  QUEUE NAME    LIBRARY       FILES  WRITER          STATUS
_ QDKT          QGPL              2
_ QPRINT        QGPL              0  QSYSPRT
_ QPRINTS       QGPL              0                  HELD
_ QPRINT2       QGPL              5
_ QPUNCH        QGPL              2





1-DSPOUTQ    4-HLDOUTQ    6-RLSOUTQ   CF5-Redisplay
```

Ⓐ *Date/Time:* The date and the time shown are the job date and the system time when you requested the display.

Ⓑ *Queue Name:* This is the name by which each listed output queue is identified to the system.

Ⓒ *Library:* This is the name of the library in which each listed output queue is located.

Ⓓ *Files:* This is the number of output files on each queue.

Ⓔ *Writer:* If a writer is currently producing output from the queue, its name is shown here.

Ⓕ *Status:* There are two possible conditions:

- Held: The queue was held by the Hold Output Queue command. No output will be produced from the queue until it is released.

- Blank: If nothing is shown, the queue is not being held, and output files can be produced when their turns come.

Ⓖ *Input Field:* This is a one-character input field where you can enter the options described in Ⓗ.

Ⓗ *Options:* These are the options that can be specified in the input field to perform the following:

1 DSPOUTQ: Display a particular output queue.

4 HLDOUTQ: Hold a particular output queue.

6 RLSOUTQ: Release a particular output queue.

Note: The CF5 key can be used to ignore all input values and redisplay the output queues showing the most current values. If there are more lines of output queue information than can be shown on your screen at one time, a + appears to the right of the last line currently displayed. You obtain the additional lines by pressing the Roll Up key. If you press the CF5 key while the output queues display is on your screen, the display is reshown with the latest status of all output queues.

Using the Output Queue Display

If you request the output queue display for all output queues, you can execute output queue commands directly from that display. An input field is provided to the left of each output queue shown on the display. At the bottom of the display, three command names are shown, each preceded by a number. By entering one of these numbers in the input field next to a file name, you can execute the command associated with the number; the command is executed only for the output queue where you entered the number.

If you key a number into several input fields and then press the Enter key, the commands associated with those numbers are executed one at a time in the same order that the files are displayed. Note that if you enter a 1 (DSPOUTQ) for a particular output queue, you receive a separate display for that output queue. If you press the CF1 key while you are viewing a separate output queue display (requested through the DSPOUTQ function on the all queues display), you return directly to your basic working display, bypassing the all output queues display. If you specified commands on the all output queues display for output queues that follow the output queue you were displaying, none of those commands are executed.

The following chart summarizes the functions you can perform directly from the output queue display:

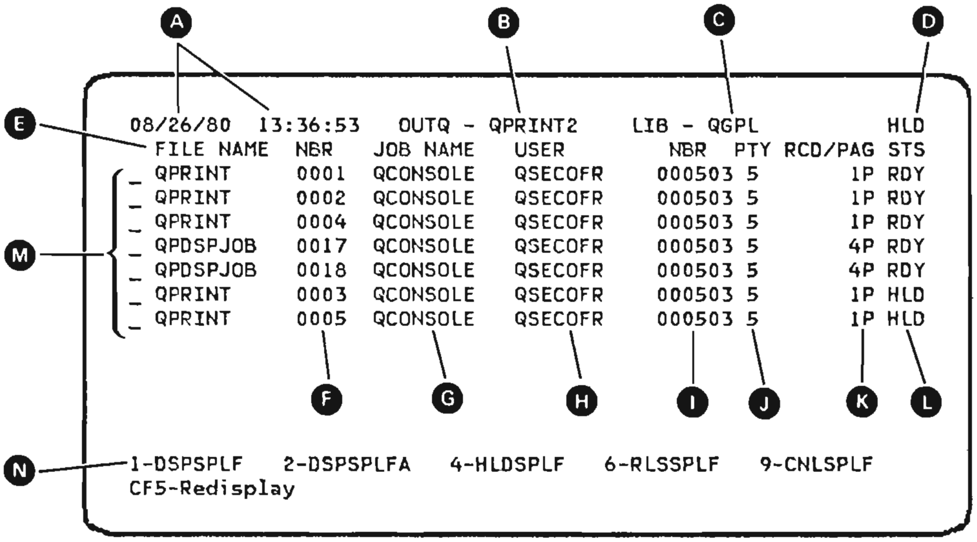
To do this:	Enter this:	For details, see:
Display the output queue (DSPOUTQ command)	1	*Interpreting a Specific Output Queue Display* in this chapter
Hold an output queue (HLDOUTQ command)	4	*Holding an Output Queue* in this chapter
Release an output queue (RLSOUTQ command)	6	*Releasing an Output Queue* in this chapter

After the commands have been executed, the all output queues display is reshown with the current status of all output queues on the display, including the output queues whose status you changed by executing the HLDOUTQ, or RLSOUTQ command from the display. Any messages related to the execution of the commands from the display are shown at the bottom of the display. If there are more messages than can be shown on your screen at one time, a + appears to the right of the last message currently displayed. You obtain the additional messages by positioning the cursor on any of the messsage lines and pressing the Roll Up key.

When you are finished with the all output queues display, you return to your basic working display (such as the system operator menu) by pressing the Enter key or the CF1 key. If you press the CF1 key while you are viewing a separate output queue display (requested through the DSPOUTQ function on the all output queues display), you return directly to your basic working display, bypassing the all output queues display. If you specified commands on the all output queues display for output queues that follow the output queue you were displaying, none of those commands are executed.

Interpreting a Specific Output Queue Display

When you enter the DSPOUTQ command and specify an output queue name in the OUTQ parameter, you receive a display of the output files on the specified output queue. From this display you can display a particular file, display the spooled files attributes, hold a spool file, release a spool file, or cancel a spooled file.



```
08/26/80  13:36:53    OUTQ - QPRINT2    LIB - QGPL          HLD
  FILE NAME  NBR   JOB NAME   USER         NBR   PTY RCD/PAG STS
_ QPRINT     0001  QCONSOLE   QSECOFR      000503 5       1P RDY
_ QPRINT     0002  QCONSOLE   QSECOFR      000503 5       1P RDY
_ QPRINT     0004  QCONSOLE   QSECOFR      000503 5       1P RDY
_ QPDSPJOB   0017  QCONSOLE   QSECOFR      000503 5       4P RDY
_ QPDSPJOB   0018  QCONSOLE   QSECOFR      000503 5       4P RDY
_ QPRINT     0003  QCONSOLE   QSECOFR      000503 5       1P HLD
_ QPRINT     0005  QCONSOLE   QSECOFR      000503 5       1P HLD



1-DSPSPLF   2-DSPSPLFA   4-HLDSPLF   6-RLSSPLF   9-CNLSPLF
CF5-Redisplay
```

A *Date/Time:* The date and the time shown are the job date and the system time when you requested the display.

B *Queue Name:* This is the name of the output queue being displayed.

C *Lib:* This is the name of the library where the output queue is located.

D *Queue Status:* Possible indicators are:

- WTR: A writer is attached to this queue.
- WTR/HLD: A writer has been started but the queue is held.
- HLD: The queue is held and there is no writer started.
- Blank: The queue is not held and there is no writer started.

E *File Name:* This is the name of each spooled output file created for the job. There may be more than one file for the job and more than one file with the same name in a job.

F *Nbr (file):* This is the spooling number assigned to the output file. This number is unique within a job.

G *Job Name:* This is the name of the job that created the file described on this line.

Ⓗ *User :* This is the name of the user profile from the job description used to start the job which created this file.

Ⓘ *Nbr (job):* This is the job number that the system assigned to the job when it entered the system. No other job in the system has the same number.

Ⓙ *Pty:* This is the output priority of output files of the job.

Ⓚ *RCD/PAG:* This is the total number of records or pages on the queue for the file name. If it is a card or diskette file the field will show the number of records with R as the last character. If it is a printer file, the field will show the number of pages followed by P.

Ⓛ *STS:* There are several possible conditions:

- RDY: The output file is complete and ready to be produced by a writer.
- OPN: The file has not been completely processed by a program and is not ready to be selected by a writer.
- CLO: The file has been completely processed by a program, but SCHEDULE(*JOBEND) was specified in the associated device file and the job that produced the file has not yet finished.
- HLD: The file has been held and cannot be produced until it is released. This may be from either HLDSPLF or from HLDJOB SPLFILE(*YES). If a file is held by HLDSPLF, it can be released by using option 6.
- WTR: The file is currently being produced by a writer.

Ⓜ *Input Field:* This is a one-character input field where you can enter the options described in Ⓝ.

Ⓝ *Options:* The values that can be specified in the input field to perform the following:

1 DSPSPLF: Displays the contents of a spooled file. The spooled file display allows you to perform scan operations, window operations, and other expanded functions on the spooled file. You can obtain help on using these functions by pressing the Help key.

2 DSPSPLFA: Displays a spooled file attributes.

4 HLDSPLF: Holds a spooled file on the output queue.

6 RLSSPLF: Releases a spooled file that was previously held.

9 CNLSPLF: Cancels a spooled file.

If there are more lines of output file information than can be shown on your screen at one time, a + appears to the right of the last line currently displayed. You obtain the additional lines by pressing the Roll Up key. If you press the CF5 key while the output queue display is on your screen, the display is reshown with the latest status of all files on the output queue.

Using the Output Queue Display

If you request the output queue display for a particular output queue, you can execute spooled output file commands directly from that display. An input field is provided to the left of each spooled output file shown on the display. At the bottom of the display, five command names are shown, each preceded by a number. By entering one of these numbers in the input field next to a file name, you can execute the command associated with the number; the command is executed only for the file where you entered the number.

If you key a number into several input fields and then press the Enter key, the commands associated with those numbers are executed one at a time in the same order that the files are displayed. Note that if you enter a 1 (DSPSPLF) or a 2 (DSPSPLFA) for a particular file, you receive a separate spooled file display for that file. If you press the CF1 key while you are viewing a separate spooled file display (requested through the DSPSPLF or DSPSPLFA function on the output queue display), your return directly to your basic working display, bypassing the output queue display. If you specified commands on the output queue display for files that follow the file you were displaying, none of those commands are executed.

The following chart summarizes the functions you can perform directly from the output queue display:

To do this:	**Enter this:**	**For details, see:**
Display the data in a spooled output file (DSPSPLF command)	1	DSPSPLF command in *CL Reference Manual*
Display the output characteristics of a spooled output file (DSPSPLFA command)	2	*Determining Output Characteristics* in this chapter
Hold a spooled output file (HLDSPLF command)	4	*Holding a Spooled File* in this chapter
Release a spooled output file (RLSSPLF command)	6	*Releasing a Spooled File* in this chapter
Cancel (remove) a spooled output file (CNLSPLF command)	9	*Canceling a Spooled File* in this chapter

After the commands have been executed, the output queue display is reshown with the current status of all output files on the display, including the files whose status you changed by executing the DSPSPLFA with CF3 option to CHGSPLFA, and files whose status was changed with HLDSPLF, RLSSPLF, or CNLSPLF command from the display. Any messages related to the execution of the commands from the display are shown at the bottom of the display. If there are more messages than can be shown on your screen at one time, a + appears to the right of the last message currently displayed. You obtain the additional messages by positioning the cursor on any of the messsage lines and pressing the Roll Up key.

When you are finished with the output queue display, you press the Enter key to return to the display from which you requested the specific output display.

Example:

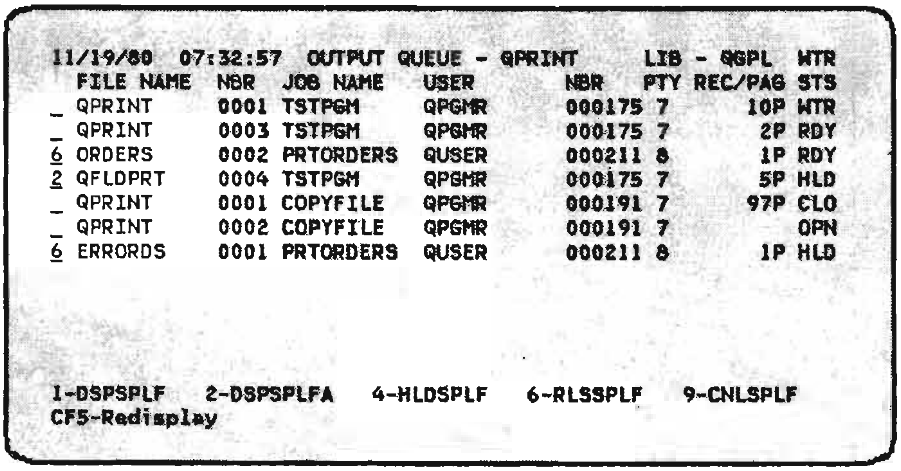
You obtained the display of the QPRINT output queue by selecting option 2 and specifying QPRINT on the system operator menu (see *Using the System Operator Menu* in Chapter 6). From the output queue display, you want to display the output characteristics of the file QFLDPRT from the job TSTPGM, and you want to release all output files from the job PRTORDERS. Therefore, you do the following:

1. Key a 6 (RLSSPLF command) into the input field next to the file name ORDERS. This final step is a mistake, because the output queue display shows that the file ORDERS was not held (its status is RDY, not HLD).

2. Key a 2 (DSPSPLFA command) into the input field next to the file name QFLDPRT.

3. Key a 6 (RLSSPLF command) into the input field next to the file name ERRORDS.

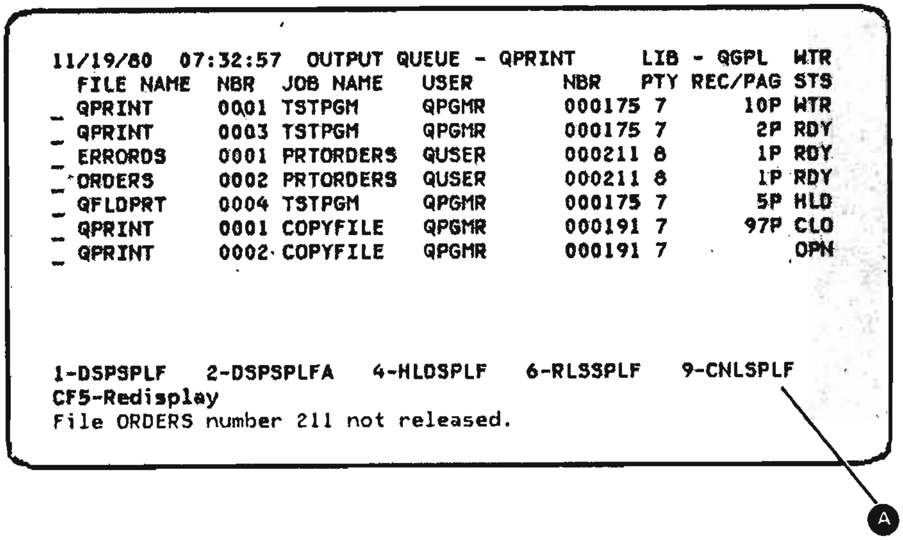
The output queue display now looks like this:



```
11/19/80  07:32:57  OUTPUT QUEUE - QPRINT      LIB - QGPL  WTR
  FILE NAME  NBR  JOB NAME   USER         NBR    PTY REC/PAG STS
_ QPRINT     0001 TSTPGM     QPGMR        000175 7       10P WTR
_ QPRINT     0003 TSTPGM     QPGMR        000175 7        2P RDY
6 ORDERS     0002 PRTORDERS  QUSER        000211 8        1P RDY
2 QFLDPRT    0004 TSTPGM     QPGMR        000175 7        5P HLD
_ QPRINT     0001 COPYFILE   QPGMR        000191 7       97P CLO
_ QPRINT     0002 COPYFILE   QPGMR        000191 7           OPN
6 ERRORDS    0001 PRTORDERS  QUSER        000211 8        1P HLD




1-DSPSPLF   2-DSPSPLFA   4-HLDSPLF   6-RLSSPLF   9-CNLSPLF
CF5-Redisplay
```

When you press the Enter key, the system attempts to release the file ORDERS, the execution of the DSPSPLFA command begins, and you receive the first of two spooled file attribute displays that show the printing characteristics of the file QFLDPRT (for examples of spooled file attribute displays, see *Determining Output Characteristics* in this chapter). After examining the first spooled file attribute display, you press the Enter key and receive the second display. After examining the second display, you press the Enter key again and the system executes the RLSSPLF command for the file ERRORDS.

The output queue display is then reshown with the current status of all files on the output queue, including the file ERRORDS that was released. A message at the bottom of the display indicates why the RLSSPLF command was not executed for the file ORDERS. If there are more messages than will fit on the screen, a plus sign (+) will appear following the displayed message Ⓐ. To display the additional messages, position the cursor on the message line and press the Roll Up key.



```
11/19/80  07:32:57  OUTPUT QUEUE - QPRINT      LIB - QGPL  WTR
  FILE NAME   NBR  JOB NAME    USER          NBR    PTY REC/PAG STS
_ QPRINT      0001 TSTPGM      QPGMR         000175 7       10P WTR
_ QPRINT      0003 TSTPGM      QPGMR         000175 7        2P RDY
_ ERRORDS     0001 PRTORDERS   QUSER         000211 8        1P RDY
_ ORDERS      0002 PRTORDERS   QUSER         000211 8        1P RDY
_ QFLDPRT     0004 TSTPGM      QPGMR         000175 7        5P HLD
_ QPRINT      0001 COPYFILE    QPGMR         000191 7       97P CLO
_ QPRINT      0002 COPYFILE    QPGMR         000191 7           OPN




1-DSPSPLF   2-DSPSPLFA   4-HLDSPLF   6-RLSSPLF   9-CNLSPLF
CF5-Redisplay
File ORDERS number 211 not released.
```

Note that if you had pressed the CF1 key instead of the Enter key when either of the spooled file attribute displays was on your screen, you would have returned to the system operator menu without executing the RLSSPLF command for the file ERRORDS.

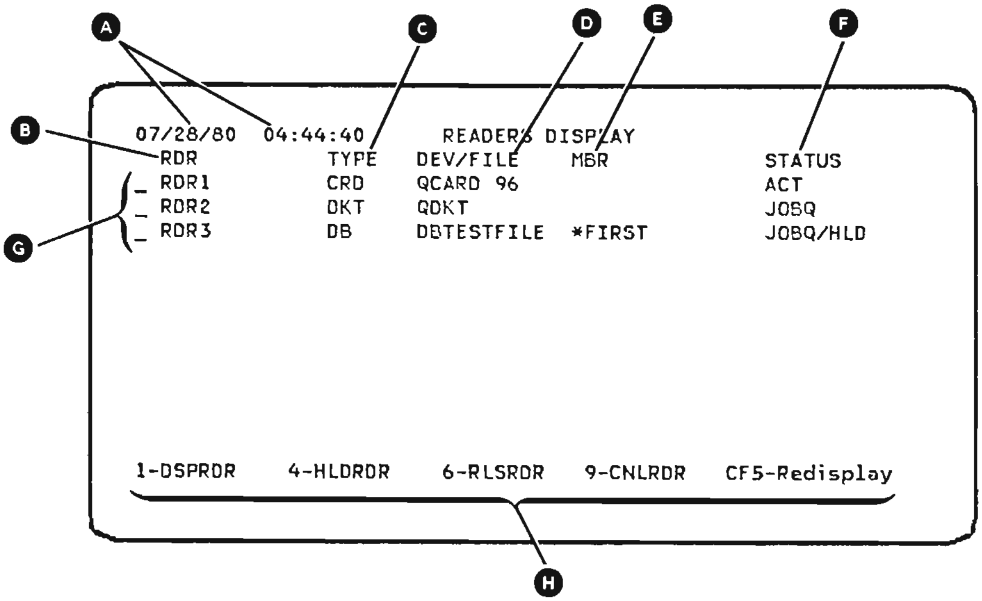
DETERMINING READER/WRITER STATUS

The setup instructions for each job should indicate when a particular reader or writer is needed. If the reader or writer does not automatically terminate when it is no longer needed, you can terminate it yourself by canceling it. The End Input (//ENDINP) command can also be used to terminate a reader (see *Controlling Readers and Writers* in this chapter). While a reader or writer is active, you can suspend its operation by holding it. If a reader or writer was held, you can release it to allow it to continue.

All Readers Display

The Display Reader (DSPRDR) command can be used either to display the current status of all readers or to display detailed information about a particular reader. From this display you can display detailed information for a particular reader, hold a reader, release a reader, or cancel a reader controlled.

Interpreting the Readers Display



```
07/28/80  04:44:40       READERS DISPLAY
 RDR           TYPE   DEV/FILE    MBR             STATUS
_ RDR1         CRD    QCARD 96                    ACT
_ RDR2         DKT    QDKT                        JOBQ
_ RDR3         DB     DBTESTFILE  *FIRST          JOBQ/HLD




1-DSPRDR    4-HLDRDR    6-RLSRDR   9-CNLRDR   CF5-Redisplay
```

Ⓐ *Date/Time:* This is the date and time the reader was first started.

Ⓑ *RDR:* This is the name of the reader from the Start Reader command.

Ⓒ *Type:* The type can be card (CRD), diskette (DKT) or data base (DB).

Ⓓ *DEV/FILE:* This is the name of the device or file from which the reader is to read the input stream.

Ⓔ *MBR:* This is for data base readers only. It is the name of the member that the reader is to process.

Ⓕ *Status:* The status of a reader can be:

- ACT: The reader is active.
- ACT/HELD: The reader was executing and was held.
- JOBQ: The reader is on the job queue.
- JOBQ/HELD: The reader is held on a job queue.

Ⓖ *Input Field:* This is a one-character input field where you can enter the options described in Ⓗ.

Ⓗ *Options:* These are the options that can be specified in the input field to perform the following:

1 DSPRDR: Displays a particular reader.

4 HLDRDR: Holds a particular reader.

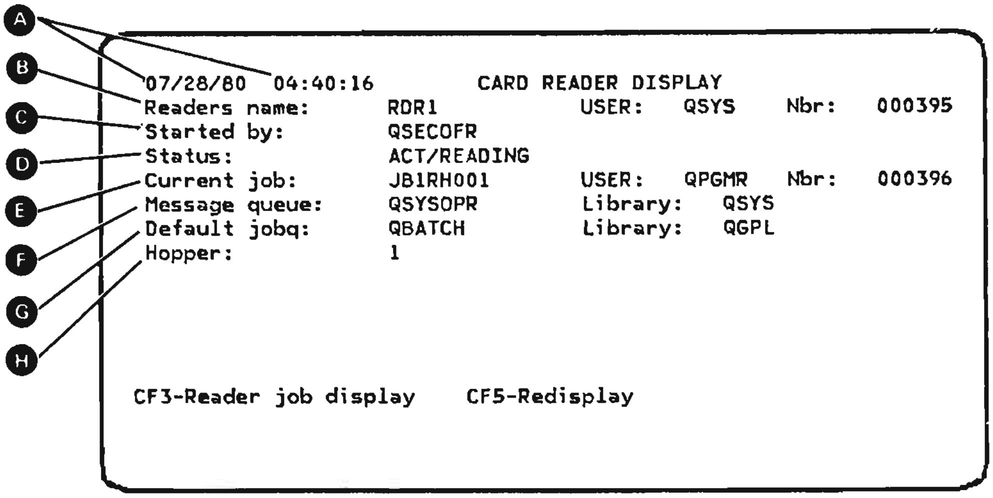
6 RLSRDR: Releases a particular reader.

9 CNLRDR: Cancels a particular reader.

Note: The CF5 key can be used to ignore all input values and redisplay the readers with the status fields updated.

Interpreting a Specific Reader Display

You can obtain the display for a specific reader by using the reader name on the Display Readers (DSPRDR) command or taking option 1 from the all readers display. If the reader is a card reader, you get the following display.



```
07/28/80  04:40:16          CARD READER DISPLAY
Readers name:        RDR1           USER:   QSYS    Nbr:   000395
Started by:         QSECOFR
Status:             ACT/READING
Current job:        JB1RH001        USER:   QPGMR   Nbr:   000396
Message queue:      QSYSOPR         Library:   QSYS
Default jobq:       QBATCH          Library:   QGPL
Hopper:              1



 CF3-Reader job display    CF5-Redisplay
```

The card reader display shows the current status of a reader and the parameters that were specified on the start command. The information displayed is:

A *Date/Time:* This is the date and time the reader was started.

B *Reader Name:* This is the fully qualified job name of the reader.

C *Started By:* This is the name of the user profile of the job that entered the start reader command.

D *Status*: This will contain one of the following:

- JOBQ: Reader is on a job queue.
- JOBQ/HLD: Reader is held on a job queue.
- ACT/READING: Reader is active and reading a job.
- ACT/HLD: Reader was executing and is held.
- ACT/WAITING FOR WORK: Reader is active and is waiting for additional input.

E *Current Job:* This is the name, user, and work number for the job being read.

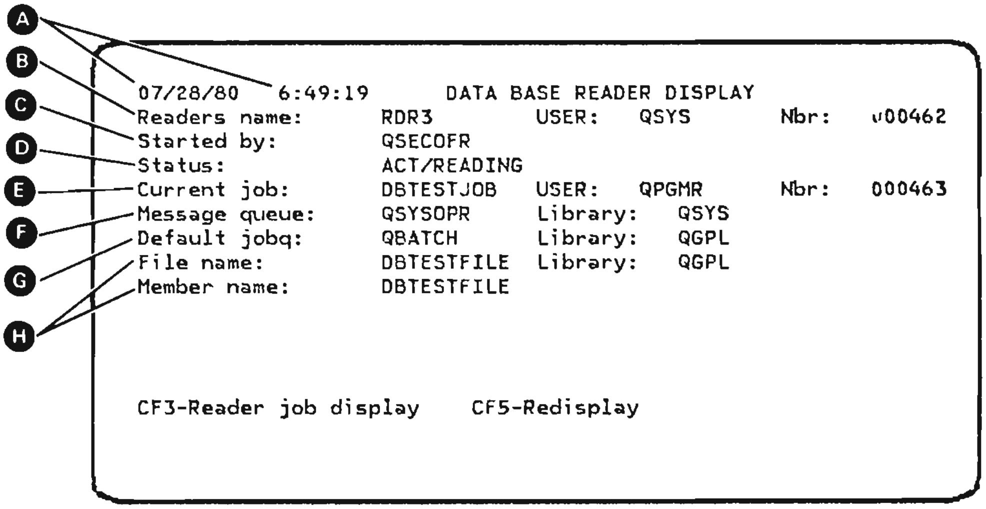
F *Message Queue:* This is the name of the message queue used by this reader for operational messages. This is the message queue name specified on the start reader command or the device if *DEVD or *REQUESTER was specified.

G *Default Jobq:* This is the name of the job queue on which jobs are enqueued if the job specifies JOBQ(*RDR). This is the job queue name specified on the start reader command.

H *Hopper:* This is the hopper number in which the input should be placed.

From the detailed reader display, the operator/user can use CF3 to obtain additional information about the reader job.

If you take the display reader option (option 1), and the reader is a data base reader, you get the following display:



```
07/28/80   6:49:19      DATA BASE READER DISPLAY
Readers name:       RDR3         USER:   QSYS        Nbr:     000462
Started by:         QSECOFR
Status:             ACT/READING
Current job:        DBTESTJOB    USER:   QPGMR       Nbr:     000463
Message queue:      QSYSOPR      Library:   QSYS
Default jobq:       QBATCH       Library:   QGPL
File name:          DBTESTFILE   Library:   QGPL
Member name:        DBTESTFILE




CF3-Reader job display    CF5-Redisplay
```

The data base reader display shows the current status of a reader and the parameters that were specified on the start command. The information displayed is:

A *Date/Time:* This is the date and time the reader was started.

B *Reader Name:* This is on the second line of the screen and is the fully qualified job name of the reader.

C *Started By:* This is the name of the user profile of the job that entered the start reader command.

D *Status*: This will contain one of the following:

- JOBQ: Reader is on a job queue.
- JOBQ/HLD: Reader is held on a job queue.
- ACT/READING: Reader is active and reading a job.
- ACT/HLD: Reader was executing and is held.
- ACT/WAITING FOR WORK: Reader is active and is waiting for additional input.

E *Current Job:* This is the name, user, and work number for the job being read.

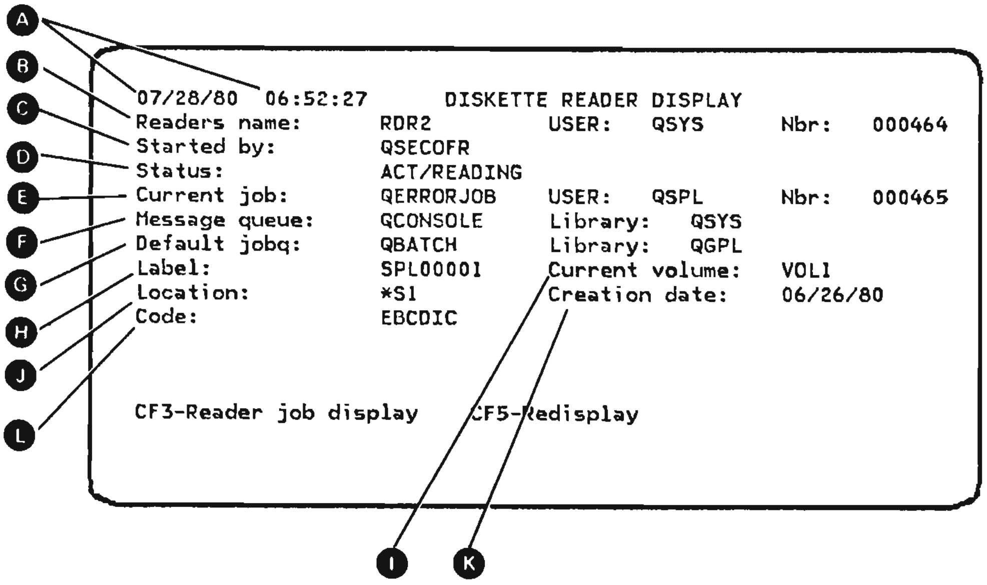
F *Message Queue:* This is the name of the message queue used by this reader for operational messages. This is the message queue name specified on the start reader command or the device if *DEVD or *REQUESTER was specified.

G *Default Jobq:* This is the name of the job queue on which jobs are enqueued if the job specifies JOBQ(*RDR). This is the job queue name specified on the start reader command.

H *File/Member Name:* This is the name of the data base file and member that contains the input stream.

From the detailed reader display, the operator/user can use CF3 to obtain additional information about the reader job.

If you take the display reader option (option 1), and the reader is a diskette reader, you get the following display.



```
07/28/80  06:52:27      DISKETTE READER DISPLAY
Readers name:      RDR2         USER:   QSYS      Nbr:    000464
Started by:        QSECOFR
Status:            ACT/READING
Current job:       QERRORJOB    USER:   QSPL      Nbr:    000465
Message queue:     QCONSOLE     Library:   QSYS
Default jobq:      QBATCH       Library:   QGPL
Label:             SPL00001     Current volume:   VOL1
Location:          *S1          Creation date:    06/26/80
Code:              EBCDIC



CF3-Reader job display    CF5-Redisplay
```

The detailed reader displays show the current status of a reader and the parameters that were specified on the start command. The information displayed is:

Ⓐ *Date/Time:* This is the date and time the reader was started.

Ⓑ *Reader Name:* This is the fully qualified job name of the reader.

Ⓒ *Started By:* This is the name of the user profile of the job that entered the start reader command.

Ⓓ *Status*: This will contain one of the following:

- JOBQ: Reader is on a job queue.
- JOBQ/HLD: Reader is held on a job queue.
- ACT/READING: Reader is active and reading a job.
- ACT/HLD: Reader was executing and is held.
- ACT/WAITING FOR WORK: Reader is active and is waiting for additional input.

Ⓔ *Current Job:* This is the name, user, and work number for the job being read.

Ⓕ *Message Queue:* This is the name of the message queue used by this reader for operational messages. This is the message queue name specified on the start reader command or the device if *DEVD or *REQUESTER was specified.

Ⓖ *Default Jobq:* This is the name of the job queue on which jobs are enqueued if the job specifies JOBQ(*RDR). This is the job queue name specified on the start reader command.

Ⓗ *Label:* This is the data identifier of the input stream on the diskette.

Ⓘ *Current Volume:* This is the volume identifier of the current volume being processed. (This is blank until the reader is active and processing a diskette.)

Ⓙ *Location:* This is the LOC parameter specified on the start command.

Ⓚ *Creation Date:* This is the creation date of the current diskette file. This is 00/00/00 for *NONE until diskette is open.

Ⓛ *Code:* This indicates if the diskette contains EBCDIC or ASCII data.

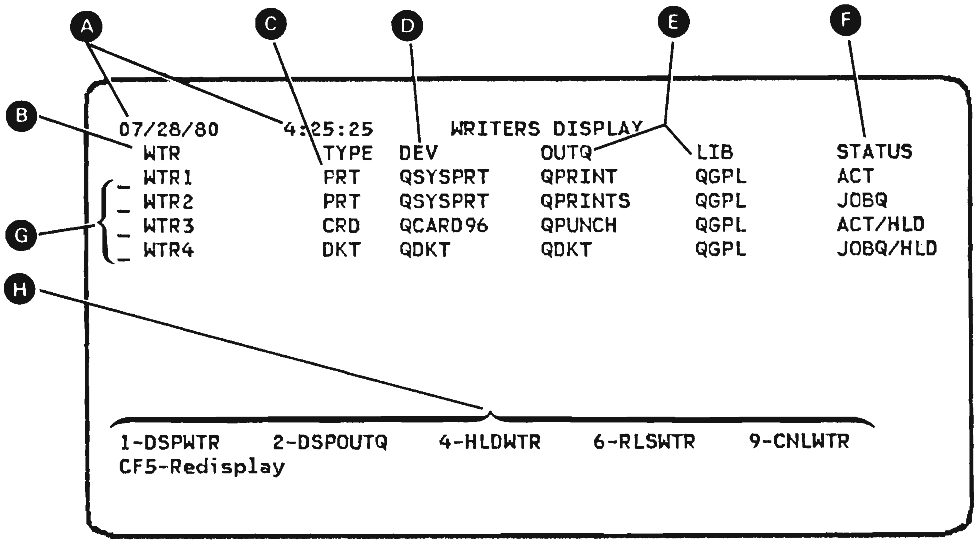
From the detailed reader display, the operator/user can use CF3 to obtain additional information about the reader job.

Writer Displays

The Display Writer (DSPWTR) command can be used to display the current status of all writers or to display detailed information about a particular writer. From this display you can display detailed information about a particular writer, hold a writer, release a writer, cancel a writer, or display the output queue assigned to the writer.

Interpreting Writer Displays

The command DSPWTR WTR(*ALL) displays all writers and produces the following display.



```
07/28/80         4:25:25          WRITERS DISPLAY
  WTR                  TYPE  DEV       OUTQ         LIB          STATUS
_ WTR1                 PRT   QSYSPRT   QPRINT       QGPL         ACT
_ WTR2                 PRT   QSYSPRT   QPRINTS      QGPL         JOBQ
_ WTR3                 CRD   QCARD96   QPUNCH       QGPL         ACT/HLD
_ WTR4                 DKT   QDKT      QDKT         QGPL         JOBQ/HLD




1-DSPWTR    2-DSPOUTQ    4-HLDWTR    6-RLSWTR    9-CNLWTR
CF5-Redisplay
```

Ⓐ *Date/Time:* This is the date and time the writer was started.

Ⓑ *Writer Name:* This is the name of the writer entered on the start command and is in the leftmost column under the heading WTR.

Ⓒ *Writer Type:* This is the type of writer PRT (print), CRD (card), or DKT (diskette) under the heading TYPE.

Ⓓ *Device:* This is the name of the device that the writer is using (or will use if the writer is still on the job queue) under the heading DEV.

Ⓔ *Output Queue:* This is the name of the output queue that the writer is processing. This is under the headings OUTQ and LIB.

Ⓕ *Status:* This is the same as for display all readers under the heading STATUS.

Ⓖ *Input Field:* This is a one-character input field where you can enter the options described under Ⓗ.

Ⓗ *Options:* These are the valid options that can be entered in the input field to perform the following:

1 DSPWTR: Displays a particular writer.

2 DSPOUTQ: Displays the output queue associated with a writer.

4 HLDWTR: Holds a particular writer.

6 RLSWTR: Releases a particular writer.

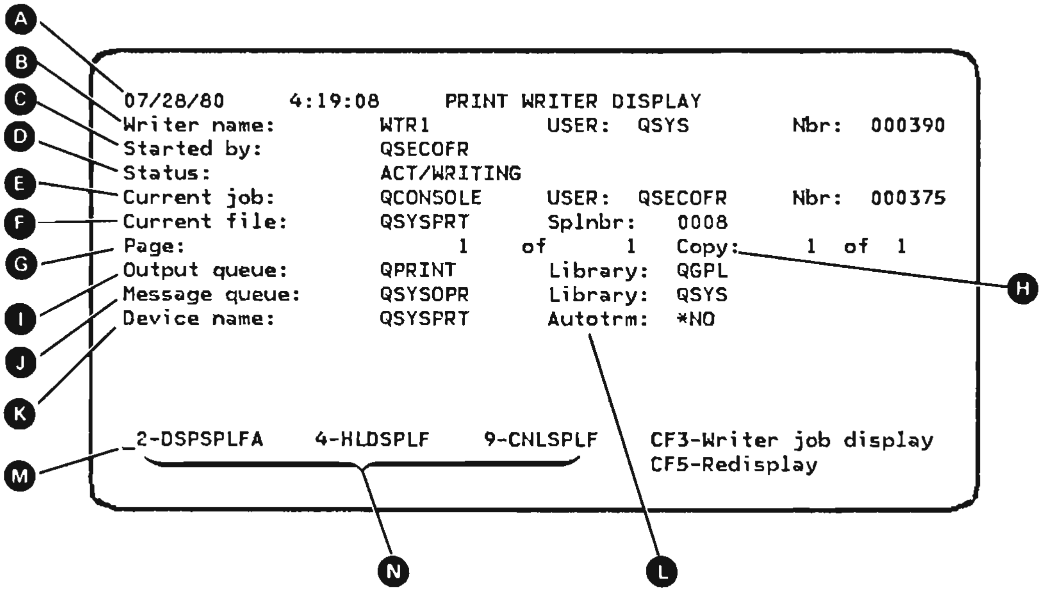
9 CNLWTR: Cancels a particular writer.

Note: The CF5 key can be used to ignore all input values and redisplay the writers with the status fields updated.

Interpreting a Specific Writer Display

You can obtain the display for a particular writer by using the writer name on the DSPWTR command or by taking option 1 from the Display Writers display.

If the writer is a print writer you will get the following display.



```
07/28/80     4:19:08     PRINT WRITER DISPLAY
Writer name:         WTR1        USER:  QSYS        Nbr:  000390
Started by:          QSECOFR
Status:              ACT/WRITING
Current job:         QCONSOLE    USER:  QSECOFR     Nbr:  000375
Current file:        QSYSPRT     Splnbr:   0008
Page:                      1    of      1   Copy:      1  of  1
Output queue:        QPRINT      Library:  QGPL
Message queue:       QSYSOPR     Library:  QSYS
Device name:         QSYSPRT     Autotrm:  *NO



_2-DSPSPLFA    4-HLDSPLF     9-CNLSPLF    CF3-Writer job display
                                          CF5-Redisplay
```

The detailed writer display shows the current status of a writer and the parameters that were specified on the start command. The following information is displayed:

Ⓐ *Date/Time:* This is the date and time the writer was started.

Ⓑ *Writer Name:* This is the fully qualified name of the writer.

Ⓒ *Started By:* This is the name of the user profile of the job that started the writer.

Ⓓ *Status:* This is the current status, which will be one of the following:

- JOBQ: Writer is on job queue and not executing.

- JOBQ/HLD: Writer is on job queue and held.

- ACT/WRITING: Writer is active and producing output.

- ACT/BETWEEN FILES: Writer is between spool files or there are no available spooled output file to process.

- ACT/BETWEEN COPIES: Writer is between copies of a spooled file.

- ACT/WAITING FOR DATA: Writer is waiting for work. Writer caught up to program producing immediate output.

Ⓔ *Current Job:* This is the qualified name of the job associated with the spool file that the writer is processing.

Ⓕ *Current File:* This is the file name and spool number that the writer is producing.

Ⓖ *Page Number:* This is the current page number and total number of pages in the file for printer files. If the file is redirected or *IMMED and incomplete the total number of pages will be blank.

🅗 *Copy Number:* This is the current copy number and the total number of copies requested.

🅘 *Output Queue:* This is the qualified name of the output queue and its current status (HELD or blank).

🅙 *Message Queue:* This is the qualified name of the message queue specified on the start writer command for operational messages or the device if *DEVD or *REQUESTER was specified.

🅚 *Device:* This is the name of the device specified on the start writer command.

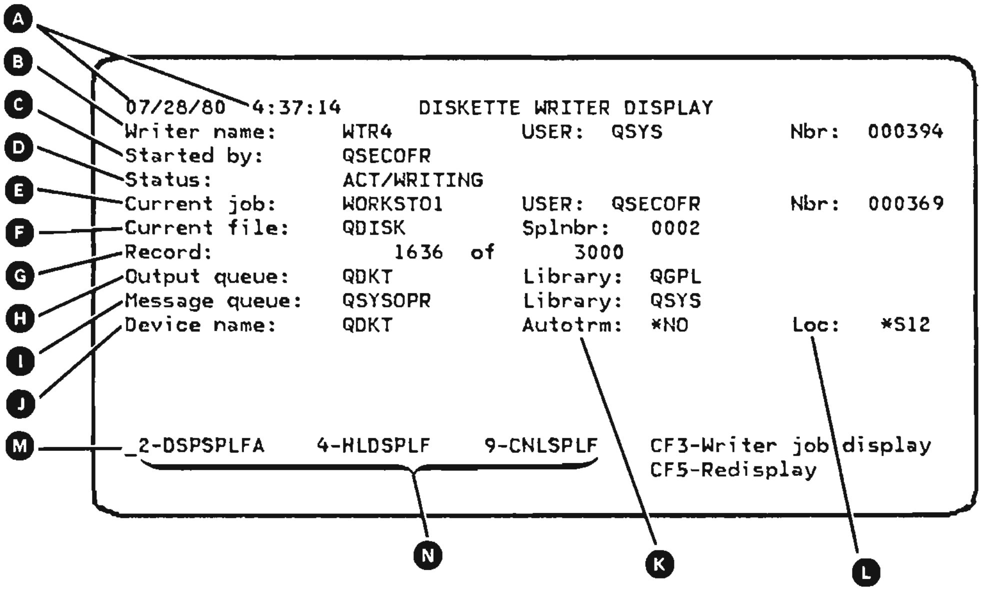
🅛 *AUTOTRM:* This is the writer termination option specified on the start command. If AUTOTRM was specified as (*YES XXXX) the second parameter (NORDYF or FILEEND) is shown (the *YES is not shown).

🅜 *Input field:* Enter the option you want to take on this field.

🅝 *Options:* From the detailed display for a print writer, the operator can display the attributes of the current spool file (DSPSPLFA), hold the current spool file (HLDSPLF) or cancel the current spool file (CNLSPLF). This is done by entering the proper option in the input field on the bottom of the screen.

Note: The CF3 function key can be used to obtain additional information for the writer job.

If the writer is a diskette writer, you will get the following display.



```
07/28/80  4:37:14      DISKETTE WRITER DISPLAY
Writer name:      WTR4           USER:  QSYS           Nbr:  000394
Started by:       QSECOFR
Status:           ACT/WRITING
Current job:      WORKSTO1       USER:  QSECOFR        Nbr:  000369
Current file:     QDISK          Splnbr:   0002
Record:               1636  of      3000
Output queue:     QDKT           Library:  QGPL
Message queue:    QSYSOPR        Library:  QSYS
Device name:      QDKT           Autotrm:  *NO         Loc:   *S12



_2-DSPSPLFA    4-HLDSPLF    9-CNLSPLF    CF3-Writer job display
                                         CF5-Redisplay
```

The detailed writer display shows the current status of a writer and the parameters that were specified on the start command. The following information is displayed:

A *Date/Time:* This is the date and time the writer was started.

B *Writer Name:* This is the fully qualified writer name.

C *Started By:* This is the name of the user that started the writer.

D *Status:* This is the current status, which will be one of the following:

- JOBQ: Writer is on job queue and not executing.
- JOBQ/HLD: Writer is on job queue and held.
- ACT/WRITING: Writer is active and producing output.
- ACT/BETWEEN FILES: Writer is between spool files or there are no available spooled output file to process.
- ACT/BETWEEN COPIES: Writer is between copies of a spooled file.
- ACT/WAITING FOR DATA: Writer is waiting for work. Writer caught up to program producing immediate output.

E *Current Job:* This is the qualified name of the job associated with the spool file that the writer is processing.

F *Current File:* This is the file name and spool number that the writer is producing.

G *Record Number:* This is the current record number and total number of records in the file for diskette files. When processing redirected spooled output files or *IMMED and incomplete, the total number of records will be blank.

H *Output Queue:* This is the qualified name of the output queue and its current status (HELD or blank).

I *Message Queue:* This is the qualified name of the message queue specified on the start writer command for operational messages or the device if *DEVD or *REQUESTER was specified.

J *Device:* This is the name of the device specified on the start writer command.

K *AUTOTRM:* This is the writer termination option specified on the start command. If AUTOTRM was specifed as (*YES XXXX) the second parameter (NORDYF or FILEEND) is shown (the *YES is not shown).

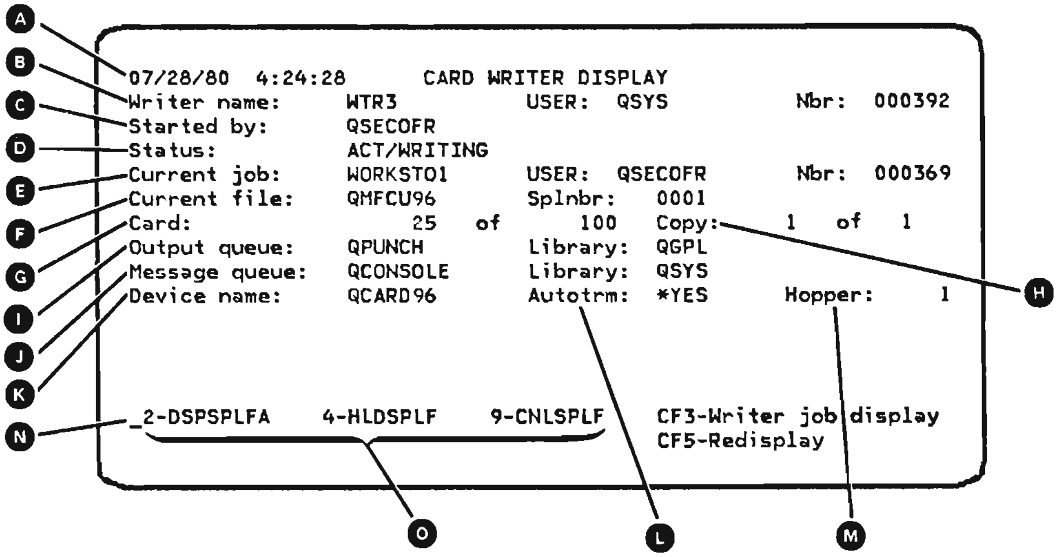
L *LOC:* This is the LOC location parameter specified on the start command.

M *Input Field:* Enter the option you want to take in this field.

N *Options:* From the detailed display for a diskette writer, the operator can display the attributes of the current spool file (DSPSPLFA), hold the current spool file (HLDSPLF) or cancel the current spool file (CNLSPLF). This is done by entering the proper option in the input field on the bottom of the screen.

Note: The CF3 function key can be used to obtain additional information for the writer job.

If the writer is a card writer, you will get the following display.



```
07/28/80  4:24:28      CARD WRITER DISPLAY
Writer name:     WTR3           USER:  QSYS          Nbr:  000392
Started by:      QSECOFR
Status:          ACT/WRITING
Current job:     WORKSTO1       USER:  QSECOFR       Nbr:  000369
Current file:    QMFCU96        Splnbr:   0001
Card:                  25   of       100   Copy:     1   of   1
Output queue:    QPUNCH         Library:  QGPL
Message queue:   QCONSOLE       Library:  QSYS
Device name:     QCARD96        Autotrm:  *YES      Hopper:     1



_2-DSPSPLFA     4-HLDSPLF     9-CNLSPLF     CF3-Writer job display
                                            CF5-Redisplay
```

The detailed writer display shows the current status of a writer and the parameters that were specified on the start command. The following information is displayed.

A *Date/Time:* This is the date and time the writer was started.

B *Writer Name:* This is the fully qualified writer name.

C *Started By:* This is the name of the user that started the writer.

D *Status:* This is the current status, which will be one of the following:

- JOBQ: Writer is on job queue and not executing.
- JOBQ/HLD: Writer is on job queue and held.
- ACT/WRITING: Writer is active and producing output.
- ACT/BETWEEN FILES: Writer is between spool files or there are no available spooled output file to process.
- ACT/BETWEEN COPIES: Writer is between copies of a spooled file.
- ACT/WAITING FOR DATA: Writer is waiting for work. Writer caught up to program producing immediate output.

E *Current Job:* This is the qualified name of the job associated with the spool file that the writer is processing.

F *Current File:* This is the file name and spool number that the writer is producing.

G *Card Number:* This is the current card number and total number of cards in the file for card files. When processing redirected spooled output files or *IMMED and incomplete, the total number of cards will be blank.

H *Copy Number:* This is the current copy number and the total number of copies requested.

I *Output Queue:* This is the qualified name of the output queue and its current status HELD or blank.

J *Message Queue:* This is the qualified name of the message queue specified on the start writer command for operational messages or the device if *DEVD or *REQUESTER was specified.

K *Device:* This is the name of the device specified on the start writer command.

L *AUTOTRM:* This is the writer termination option specified on the start command. If AUTOTRM was specifed as (*YES XXXX) the second parameter (NORDYF or FILEEND) is shown (the *YES is not shown).

M *Hopper Number:* This is, the hopper number to be used for the cards to be punched.

N *Input Field:* Enter the option you want to take in this field.

O *Options:* From the detailed display for a card writer, the operator can display the attributes of the current spool file (DSPSPLFA), hold the current spool file (HLDSPLF) or cancel the current spool file (CNLSPLF). This is done by entering the proper option in the input field on the bottom of the screen.

Note: The CF3 function key can be used with the display of a card writer to obtain additional information for the writer job.

DETERMINING AND CHANGING OUTPUT CHARACTERISTICS

Records produced as a result of processing a job are stored in one or more spooled output files that are associated with the output queue on which the output is placed. Each file has a set of *spool attributes* that define how the output is to be produced.

If you need to know the output characteristics of a particular file, you can request a display that shows its spool attributes. From the spool attributes display, you can change the attributes to obtain the desired output.

Obtaining the Display

To display the attributes of a spooled output file, do one of the following:

- Use option 2 from the output queue display
- Use option 2 from the display of the spooled files for a job by using either:
  - Option 6 of DSPJOB, or
  - Option 2 on DSPSBS
- Enter the Display Spooled File Attributes (DSPSPLFA) command from the system operator menu

If you use the DSPSPLFA command, you must enter at least the FILE parameter. Use the other three parameters as needed.

FILE Parameter: Enter this parameter and specify the name of the file, such as

```
FILE(RECORDS1)
```

JOB Parameter: This parameter defines the name of the job that created the file.

- If you submitted the job from your work station, you do not need to enter the parameter.
- If you did not submit the job, you should enter the parameter and specify the job name, along with the user name and job number if known, such as

  ```
  JOB(UPDATE.QPGMR)
  ```

  or

  ```
  JOB(UPDATE.QPGMR.000253)
  ```

SPLNBR Parameter: This parameter defines which spooled file is to be used if there is more than one with the same name in a job. The displays show the FILE, JOB, and SPLNBR filled in, so there should be no confusion as to which file is meant.

- If only one spooled file from the job has the file name you specified in the FILE parameter, you do not have to enter this parameter.

- If more than one file from the job has the specified name, you should enter the parameter and specify either SPLNBR(*LAST) if the attributes of the highest numbered file that is not already finished for the job are to be displayed, or the file number, such as SPLNBR(3) if the attributes of a file other than the one with the highest number are to be displayed.

OUTPUT Parameter: This parameter determines how the information is to be provided.

- If you do not enter the parameter, the attributes are displayed on your work station screen.

- If you specify OUTPUT(*LIST) the information is spooled for printing on the system printer and placed on the output queue associated with your job.

Examples:

1. To display at your work station the output characteristics of the file named LAMPS that is the only output file in the job CATALOG, enter

```
DSPSPLFA FILE(LAMPS) JOB(CATALOG)
```

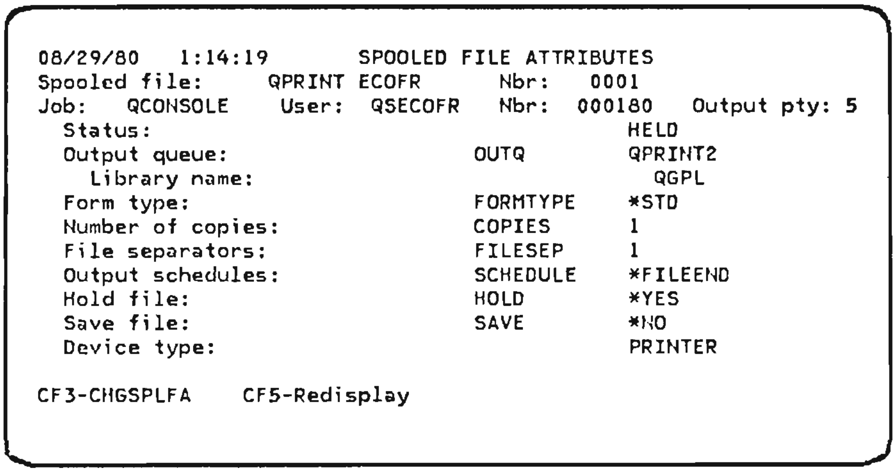
2. To print the output characteristics of the second of three files named CHECKS in the job PAY, which was submitted by the user DEPT25 and has a job number 000275, enter

```
DSPSPLFA FILE(CHECKS) JOB(PAY.DEPT25.000275)
 SPLNBR(5) OUTPUT(*LIST)
```

To display the spool file attributes of a file, use option 2 from the display of a particular output queue (DSPOUTQ), display of a particular writer (DSPWTR), and the display of job files (DSPJOB option 6).

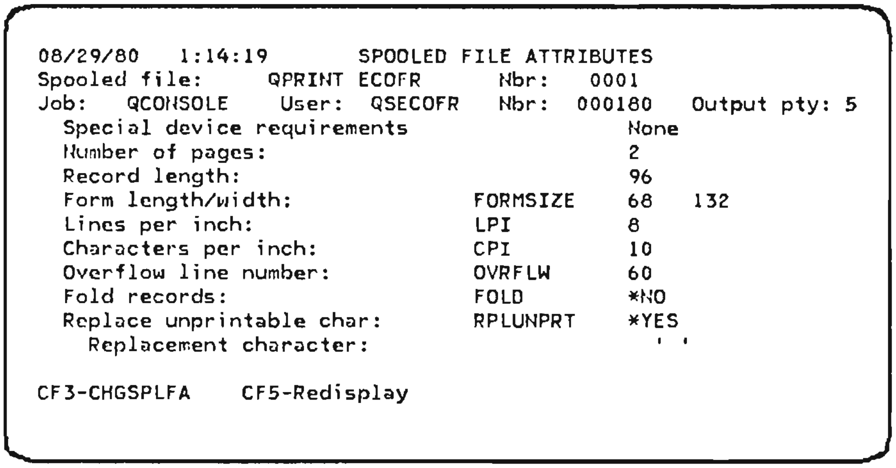
Interpreting the Display

If the spooled file you specified in the DSPSPLFA command is a print file, you receive three displays. The first display shows general output information, such as:



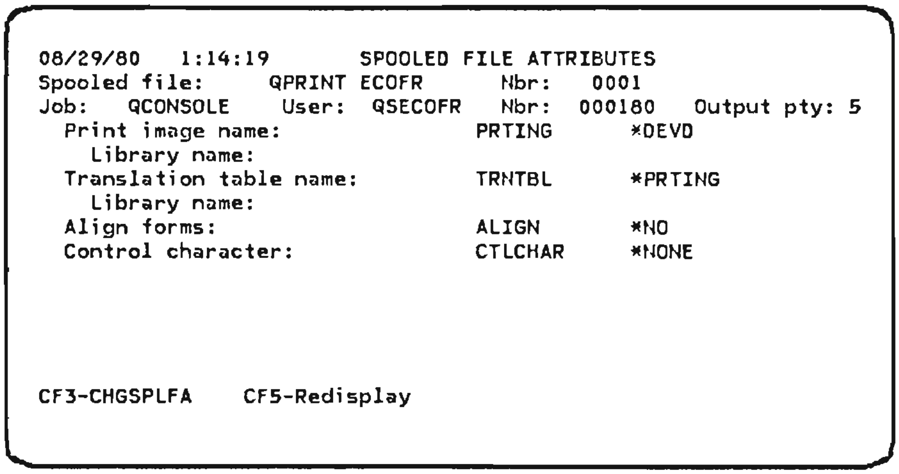
```
08/29/80   1:14:19       SPOOLED FILE ATTRIBUTES
Spooled file:     QPRINT ECOFR      Nbr:   0001
Job:   QCONSOLE    User:  QSECOFR   Nbr:  000180   Output pty: 5
  Status:                                     HELD
  Output queue:                   OUTQ        QPRINT2
    Library name:                               QGPL
  Form type:                      FORMTYPE    *STD
  Number of copies:               COPIES      1
  File separators:                FILESEP     1
  Output schedules:               SCHEDULE    *FILEEND
  Hold file:                      HOLD        *YES
  Save file:                      SAVE        *NO
  Device type:                                PRINTER

CF3-CHGSPLFA    CF5-Redisplay
```

The + symbol in the lower right corner of the first display indicates that there is another display to follow. When you are finished with the first display, press the Roll Up key. You then receive the second display, which provides information on the print image, such as:



```
08/29/80   1:14:19       SPOOLED FILE ATTRIBUTES
Spooled file:     QPRINT ECOFR      Nbr:   0001
Job:   QCONSOLE    User:  QSECOFR   Nbr:  000180   Output pty: 5
  Special device requirements                 None
  Number of pages:                            2
  Record length:                              96
  Form length/width:              FORMSIZE    68    132
  Lines per inch:                 LPI         8
  Characters per inch:            CPI         10
  Overflow line number:           OVRFLW      60
  Fold records:                   FOLD        *NO
  Replace unprintable char:       RPLUNPRT    *YES
    Replacement character:                      ' '

CF3-CHGSPLFA    CF5-Redisplay
```

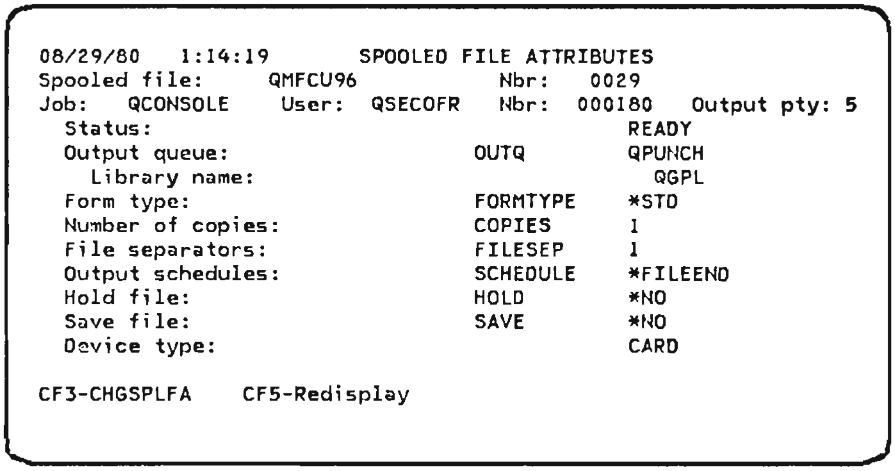
Press the Roll Up key to obtain the third display.



```
08/29/80   1:14:19       SPOOLED FILE ATTRIBUTES
Spooled file:     QPRINT ECOFR      Nbr:   0001
Job:   QCONSOLE    User:  QSECOFR   Nbr:  000180   Output pty: 5
  Print image name:               PRTING      *DEVD
    Library name:
  Translation table name:         TRNTBL      *PRTING
    Library name:
  Align forms:                    ALIGN       *NO
  Control character:              CTLCHAR     *NONE




CF3-CHGSPLFA    CF5-Redisplay
```

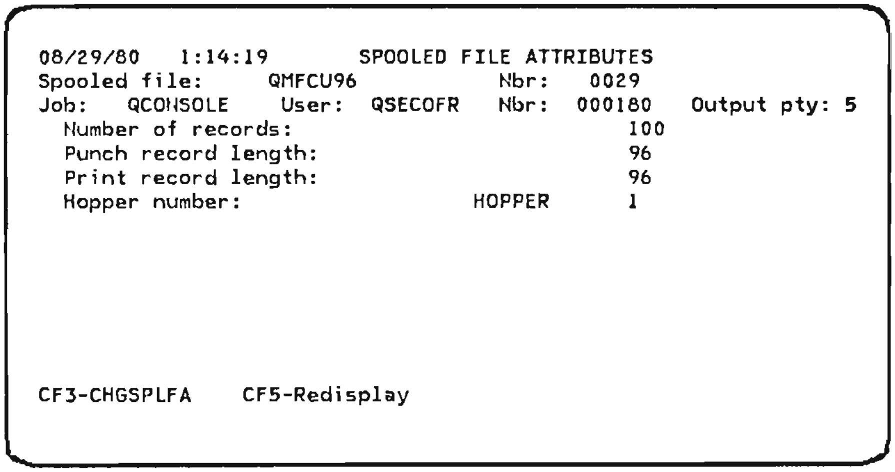
You can exit the print file displays by pressing the Enter key from the third screen. You can also exit the print file displays by pressing the CF1 key.

If the spooled file you specified is a card file, you receive two displays of the form:



```
08/29/80   1:14:19       SPOOLED FILE ATTRIBUTES
Spooled file:     QMFCU96           Nbr:   0029
Job:   QCONSOLE    User:  QSECOFR   Nbr:  000180   Output pty: 5
  Status:                                    READY
  Output queue:                     OUTQ       QPUNCH
    Library name:                              QGPL
  Form type:                        FORMTYPE   *STD
  Number of copies:                 COPIES     1
  File separators:                  FILESEP    1
  Output schedules:                 SCHEDULE   *FILEEND
  Hold file:                        HOLD       *NO
  Save file:                        SAVE       *NO
  Device type:                                 CARD

CF3-CHGSPLFA    CF5-Redisplay
```

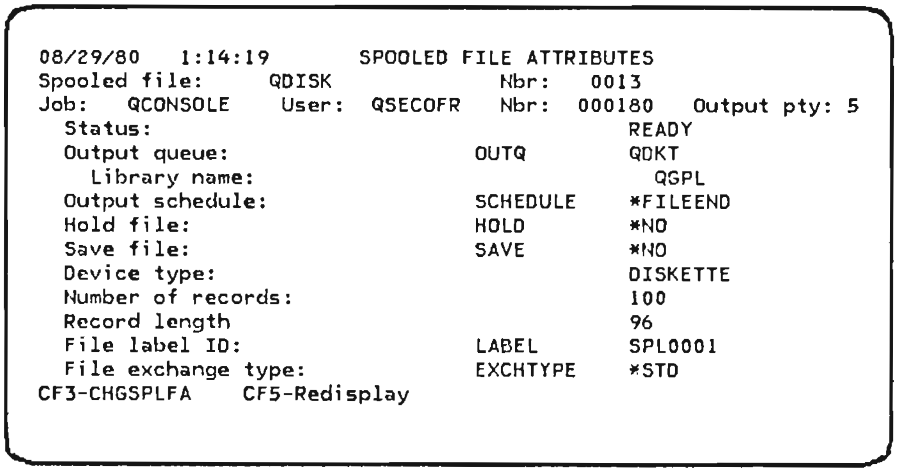
To obtain the second display, press the Roll Up key.



```
08/29/80   1:14:19       SPOOLED FILE ATTRIBUTES
Spooled file:     QMFCU96           Nbr:   0029
Job:   QCONSOLE    User:  QSECOFR   Nbr:  000180   Output pty: 5
  Number of records:                           100
  Punch record length:                         96
  Print record length:                         96
  Hopper number:                    HOPPER     1







CF3-CHGSPLFA    CF5-Redisplay
```

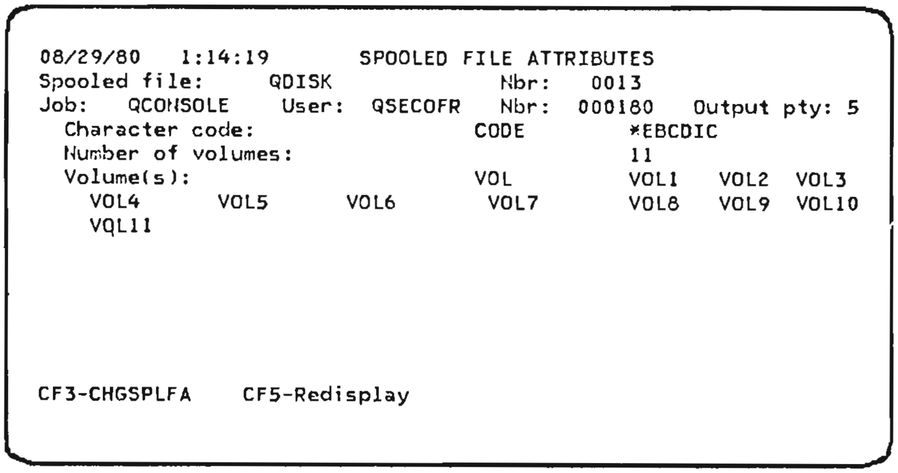
To exit the card file displays, press the Enter key after viewing the second display. You can also exit the card file displays by pressing the CF1 key.

If the spooled file you specified is a diskette file, you receive two displays of the form:



```
08/29/80   1:14:19       SPOOLED FILE ATTRIBUTES
Spooled file:     QDISK             Nbr:   0013
Job:   QCONSOLE    User:  QSECOFR   Nbr:  000180   Output pty: 5
  Status:                                     READY
  Output queue:                   OUTQ        QDKT
    Library name:                               QGPL
  Output schedule:                SCHEDULE    *FILEEND
  Hold file:                      HOLD        *NO
  Save file:                      SAVE        *NO
  Device type:                                DISKETTE
  Number of records:                          100
  Record length                               96
  File label ID:                  LABEL       SPL0001
  File exchange type:             EXCHTYPE    *STD
CF3-CHGSPLFA    CF5-Redisplay
```

You can obtain the second display by pressing the Roll Up key.



```
08/29/80   1:14:19       SPOOLED FILE ATTRIBUTES
Spooled file:     QDISK             Nbr:   0013
Job:   QCONSOLE    User:  QSECOFR   Nbr:  000180   Output pty: 5
  Character code:                 CODE        *EBCDIC
  Number of volumes:                          11
  Volume(s):                      VOL         VOL1   VOL2  VOL3
    VOL4      VOL5      VOL6       VOL7       VOL8   VOL9  VOL10
    VQL11






CF3-CHGSPLFA    CF5-Redisplay
```

To exit the diskette displays, press the Enter key after viewing the second display. You can also exit the diskette displays by pressing the CF1 key.

You will note that the first three lines of all the displays provide the same information regardless of the type of output device:

- The first line shows the date, time, and name of the display.
- The second line shows:
  - The name of the file (the narne you specified in the FILE parameter)
  - The spooled file number (NBR)
- The third line shows:
  - The name of the job
  - The name of the user profile from the job description
  - The job number
  - The output priority of the job
- The fourth line shows:
  - The output status of the file which could show HELD, READY, CLOSED, WRITING, or OPEN

Following these first four lines are the individual output attributes, which depend on the type of output device.

When you have finished viewing the display, press the Enter key to return to the original display from which you requested the output file information.

CHGSPLFA from the DSPSPLFA Display

If you want to change any of the spool file attributes, press the CF3 key. This will display the Change Spool File Attributes prompt and allow you to enter the changes. After you have entered the changes, press the Enter key to execute the change. Any error messages from the CHGSPLFA command as prompted will appear on the bottom of this screen.

If you press the CF1 key from the DSPSPLFA prompt, you will exit the prompt and also exit any other spool displays that called DSPSPLFA.

You should be aware that the CF1 key can take you back farther than you intended to go. For example, making the following entries will result in a return to command entry:

1. Command Entry
2. DSPOUTQ *ALL Option 1 for
3. DSPOUTQ QPRINT Option 2 for
4. DSPSPLFA FILE(QSYSPRT) JOB(JOB.USERX,000175) SPLNBR(5)
5. CF3 for DSPSPLFA prompt
6. CF1 will return you back to Command Entry

CONTROLLING READERS AND WRITERS

Once the spooling subsystem is started, you submit a job for processing by placing the input stream (cards or diskettes) on the appropriate device (or in a data base file) and starting the reader to the device (or data base file). You can also submit the same jobs for processing by using the Submit Jobs commands (SBMCRDJOB, SBMDBJOB, or SBMDKTJOB). The input is spooled and an entry for each job is placed on a job queue. Each job is processed when its turn comes. The spooled output records that result from the processing are automatically placed in a file that has an entry on an output queue. The output file is produced on a particular device when you start a writer to the appropriate output queue and device.

The setup instructions for each job should indicate when a particular reader, Submit Jobs command, or writer is needed. If the reader or writer does not automatically terminate when it is no longer needed, you can terminate it yourself by canceling it. You can use the //ENDINP command to terminate a reader or Submit Jobs command if the //ENDINP command is in the input stream being read. While a reader or writer is active, you can suspend its operation by holding it. If a reader or writer was held, you can release it to allow it to continue.

If you hold a reader or any job that is using a spooling function when it has a lock on an essential object that is used by many processes, exceptions may occur in other processes and cause jobs not to be completed.

The queue is locked if a spool exception message, CPF3330, is received where a job or output queue is displayed or if the display of all jobs or output queues takes a long time and one queue is blank for the number of entries.

If the problem continues, release held jobs and held readers to try to release the lock.

The reader and writer displays (DSPRDR, DSPWTR) provide an easy way to hold, release, and cancel readers and writers. However, the options provided from the displays give you the command defaults, which may not be what you want in every case.

Reader Operating Considerations

Commands to Start Readers

You use a different command to start a reader from each type of input source. The input sources and the associated start reader command are as follows:

Input Source	Command
Card	Start Card Reader (STRCRDRDR)
Data base	Start Data Base Reader (STRDBRDR)
Diskette	Start Diskette Reader (STRDKTRDR)

Each command is discussed separately in this chapter.

When you enter a start reader command, your request is handled as a job submitted to the spooling subsystem and is placed on the QSPL job queue. The reader is started when the spooling subsystem selects your request from the queue. If there are other requests of equal or higher priority on the QSPL job queue, your request to start a reader will not be processed until the other requests have been processed.

You can enter a start reader command while the spooling subsystem is inactive, but the reader will not be started until the spooling subsystem is started. No entries are taken from the QSPL job queue and processed while the spooling subsystem is inactive. The display of all readers (DSPRDR RDR(*ALL)) can show you whether your reader is on a job queue or active.

Reader Termination

A data base reader or diskette reader will terminate automatically at the end of the file being read. A card reader will remain active after the last card has been read, so that you can load additional cards (input stream) into the device and continue reading jobs and data into the system.

You can terminate a card reader, or any other active reader, by entering the Cancel Reader (CNLRDR) command (see *Canceling a Reader* in this chapter) or by using the End Input (//ENDINP) command in the input stream (see *Submit Jobs Operating Conditions* in this chapter).

Effect of Holding, Releasing, and Canceling a Reader

When a reader is held, it stops at the point in the input stream where it was currently reading (see *Holding a Reader* in this chapter). If it is released later, it resumes reading at the same point (see *Releasing a Reader* in this chapter).

If a reader is canceled using the *IMMED option of the Cancel Reader (CNLRDR) command, the job that is currently being read is not placed on the job queue, nor are any jobs that follow it in the input stream. For each job that a reader reads, a CPF3385 message is sent prior to the job being put on a job queue. If a reader was actively reading a job when it was canceled, message CPF3415 will be sent to the diagnostic message queue (MSGQ parameter) informing you that the reader ended before it finished the job.

A diskette or data base reader can only start at the beginning of an input stream. Therefore, if you start another reader to the input stream, all of the jobs in the input stream will be read in and placed on a job queue, even if some of them had been read in previously. Any jobs in the input stream that were on the job queue previously will have a second entry on the job queue, which means that they will be processed twice unless you prevent it. You can use the following procedure to prevent the jobs from being processed twice:

1. Hold the job queue (see *Holding Job and Output Queues* in this chapter). While the job queue is being held, none of the jobs currently on the queue or subsequently placed on the queue will be processed.

2. Start the reader.

3. When all of the jobs in the input stream have been placed on the job queue, cancel the jobs that were previously processed (see *Canceling a Job* in Chapter 12). The canceled jobs are removed from the queue but their job logs are written (put on an output queue) and you may wish to cancel these job logs also. They may not be sent to the same queue, so perhaps a search, using DSPSBS to show the jobs or an output queue, would be the quickest way to find all of them.

4. Release the job queue (see *Releasing Job and Output Queues* in this chapter). Processing will resume on the remaining jobs.

A card reader can start in the middle of an input stream, but it should start at the beginning of a job (the first card must begin with the characters // JOB). If you are starting another card reader to read in the remainder of an input stream, use the following procedure:

1. Remove the cards for all jobs in the input stream that the reader had completed reading. Those jobs have already been placed on a job queue for processing.

2. If the reader was stopped while it was in the process of reading a particular job in the input stream, that job was not placed on a job queue. Retrieve all cards to the beginning of the job (the last // JOB card read and any cards that follow it) and replace them in the hopper.

3. Ready the card device (for example, press the Start key on the MFCU).

4. Start the reader.

Reader Messages

If a problem occurs with the device or the input stream (cards, diskettes, data base files), an error message is sent to the message queue specified in the MSGQ parameter of the start reader command. Unless you specify a different message queue in the Start Reader (STRRDR) command, reader messages are sent to the system operator message queue, QSYSOPR (see *Message Queues* in Chapter 11). Examples of device error messages are given in the discussion of *Operator/Device Interaction* in Chapter 9.

Starting a Card Reader

To start a spooling reader that reads an input stream of one or more batch jobs from a card input device and places them on a job queue, enter the Start Card Reader (STRCRDRDR) command. You must enter the DEV parameter. All other parameters are optional.

DEV Parameter: Enter this parameter and specify the name of the device to be used.

RDR Parameter: This parameter allows you to enter a unique name for the reader to be used. The reader is identified by that name until it is complete.

- If you do not enter the parameter, the reader name is assumed to be the same as the device name you specified in the DEV parameter.

- The name you specify or let default must be different from any other reader name.

HOPPER Parameter: This parameter determines which hopper in the MFCU is used (see *5424 Multi-Function Card Unit* in Chapter 4).

- If you do not enter the parameter, the primary hopper (hopper 1) is used.

- If you specify HOPPER(2) the secondary hopper is used.

JOBQ Parameter: This parameter determines the job queue on which the reader places the input if JOBQ(*RDR) was specified in the JOB command used to submit the job.

- If you do not enter the parameter, the input is placed on the batch job queue (QBATCH).

- If you enter the parameter and specify a job queue name, the input is placed on that job queue.

MSGQ Parameter: This parameter determines where messages generated by the reader are sent (see message discussion in Chapter 11).

- If you do not enter the parameter, the messages are sent to the message queue of the device (*DEVD).

- If you enter the parameter, you can specify that messages be sent to your work station message queue (*REQUESTER) or to another message queue. The named message queue must not reside in the library QTEMP. This value becomes *DEVD for batch jobs.

Normally, you will not enter this parameter because you will want the messages to be sent to QSYSOPR. However, if QSYSOPR is currently assigned to the interactive job at another work station, you can enter this parameter and specify any message queue so that any messages relating to the read operation are sent to you.

Examples:

1. To start a reader to the card device QCARD96 so that the jobs on the cards in the primary hopper are read into the QBATCH job queue, enter

```
STRCRDRDR DEV(QCARD96)
```

2. To start the reader named ALT to the device CARD so that the jobs on cards in the secondary hopper are read and placed on the PAY job queue in the library PAYROLL and any related messages are sent to the WRKSTN3 message queue, enter

```
STRCRDRDR DEV(CARD) RDR(ALT) HOPPER(2)
 JOBQ(PAY.PAYROLL) MSGQ(WRKSTN3)
```

The reader will remain active after the last card is read (see *Reader Operating Considerations* in this chapter). You terminate the reader by entering the Cancel Reader (CNLRDR) command (see *Canceling a Reader* in this chapter) or by using the End Input (//ENDINP) command.

Starting a Data Base Reader

To start a spooling reader that reads a batch input stream from a data base file, enter the Start Data Base Reader (STRDBRDR) command. You must enter the FILE parameter. All other parameters are optional.

FILE Parameter: Enter this parameter and specify the name of the data base file that contains the input stream. The file named must not reside in the library QTEMP.

MBR Parameter: This parameter is used to specify a particular member of the file where the job or jobs are to be found.

- If you do not enter the parameter, the first member in the file is used.
- If you enter the parameter and specify a member name, the job or jobs are read from that member.

RDR Parameter: This parameter allows you to enter a unique name for the reader to be used. The reader is identified by that name until it is complete.

- If you do not enter the parameter, the reader name is assumed to be the same as the file name you specified in the FILE parameter.
- The name you specifiy or let default must be different from any other reader name.

The JOBQ parameter is used as described previously in *Starting a Card Reader*.

The MSGQ parameter defaults to QSYSOPR because there is no device associated with a data base file.

Examples:

1. To start a reader to read a job from member WKPAY35 in the data base file ACCSRC, enter

```
STRDBRDR FILE(ACCSRC) MBR(WKPAY35)
```

2. To start a reader having the name FRDR to read jobs from the data base file PARTS, located in the library INVTRY, into the job queue PROC, located in the library QGPL, enter

```
STRDBRDR FILE(PARTS.INVTRY) RDR(FRDR)
  JOBQ(PROC.QGPL)
```

3. To start the same reader from the same data base file to the same job queue, but send any messages related to the read operation to the DEPT27 message queue, enter

```
STRDBRDR FILE(PARTS.INVTRY) RDR(FRDR) JOBQ(PROC.QGPL)
  MSGQ(DEPT27)
```

A data base reader is automatically terminated at the end of the file or when an End Input (//ENDINP) command in the input stream is processed by the reader.

Starting a Diskette Reader

To start a spooling reader that reads a batch input stream from a diskette file, enter the Start Diskette Reader (STRDKTRDR) command. You must enter the DEV and LABEL parameters.

DEV Parameter: Use this parameter to specify the name of the diskette device to be used.

LABEL Parameter: Use this parameter to specify the data file identifier of the input stream to be read.

All of the other parameters are optional. Of these, the RDR, JOBQ, and MSGQ parameters are used as described in *Starting a Card Reader.* The remaining four parameters define the diskette or diskettes to be used and the coding format.

LOC Parameter: This parameter determines which diskettes are to be used in which magazine or I/O slot of the diskette magazine drive.

- If you do not enter the parameter, the operation begins with the first diskette in magazine 1 and continues through each successive diskette in both magazines to the last diskette in magazine 2 or until the end of the file.

- If you enter the parameter, you can specify one of the following options for the magazine or I/O slot:

*M12	magazine 1 followed by magazine 2
*M1	magazine 1 only
*M2	magazine 2 only
*S1	slot 1 only
*S2	slot 2 only
*S3	slot 3 only
*S12	slot 1 followed by slot 2
*S23	slot 2 followed by slot 3
*S123	slot 1 followed by slots 2 and 3

If you specified a magazine or more than one I/O slot, you can include two additional values that define the starting diskette and the ending diskette.

The second value specifies the starting diskette in the magazine or I/O slots.

- If you do not enter this value, the operation begins with the first, or leftmost, diskette as viewed from the front of the diskette magazine drive (the diskette in position 1).
- If you specify *CURRENT, the process begins with the diskette from which the diskette magazine drive last read a file.
- If you specified a magazine in the first value, you can specify a number from 1 through 10 for the second value, and the operation will begin with the diskette in that position in the magazine.

The third value specifies the ending diskette in the magazine or I/O slots.

- If you do not enter this value, the operation will end with the last, or rightmost, diskette (the diskette in position 10).
- If you specify *WRAP, the procedure used depends on whether you specified I/O slots, one magazine, or both magazines.

  I/O Slots: The operation begins at the leftmost specified slot (the slot with the lowest number) and proceeds from left to right to the rightmost specified slot (the slot with the highest number). You then receive a message to replace all the diskettes in the specified slots.

  One Magazine: The operation begins at the specified starting position in the magazine and continues to the last diskette in the magazine (the diskette in position 10). You then receive a message to mount a new magazine. The operation resumes with the diskette in position 1 of the new magazine.

  Both Magazines: The operation begins at the specified starting position in magazine 1 and continues through the last diskette in magazine 2. You then receive a message to mount two new magazines. The operation resumes with the diskette in position 1 of magazine 1.

- If you specify *ONLY, only the single starting diskette specified in the second value is used.
- If you specified a magazine in the first value, you can specify a number from 1 through 9 or 0 for the second value, and the operation will end with the diskette in that position in the magazine.

For additional information on the LOC parameter, see *Diskette Considerations* in Chapter 13.

VOL Parameter: This parameter allows you to specify that only a diskette or diskettes having a particular identifier is to be used.

- If you do not enter the parameter, VOL(*NONE) is assumed and any diskette in the position specified in the LOC parameter is used.

- If you enter the parameter and specify one or more volume identifiers (any combination of letters and numbers up to a maximum of six), such as VOL(ABCDE5) each diskette specified in the LOC parameter is used only if it has the specified volume identifier. If the diskette does not have the correct identifier, you will receive a message to mount the correct diskette.

For additional information on the VOL parameter, see *Diskette Considerations* in Chapter 13.

CRTDATE Parameter: This parameter allows you to specify that a diskette is to be used only if it contains a data file created on a certain date.

- If you do not enter the parameter, no check is made for a create date.
- If you enter the parameter and specify a create date, the diskette file is matched against that date. If it does not match, an error message is sent.

CODE Parameter: This parameter determines the type of character code to be used to read the diskette data onto the job queue.

- If you do not enter the parameter, EBCDIC code is used.
- If you specify CODE(*ASCII) ASCII code is used.

Examples:

1. To start a diskette reader and, using a device named QDKT, read a file labeled PAY from diskettes of magazines 1 and 2 and put the job in that file onto the batch job queue (QBATCH) in EBCDIC code, enter

   ```
   STRDKTRDR DEV(QDKT) LABEL(PAY)
   ```

2. To start a reader named INPUT1 and, using a device named DKTMD, read a file labeled UPDATE from the second through sixth diskettes of magazine 2 into the job queue INVENTORY of library RECORDS in ASCII code, enter

   ```
   STRDKTRDR DEV(DKTMD) LABEL(UPDATE) RDR(INPUT1)
     LOC(*M2 2 6) JOBQ(INVENTORY.RECORDS) CODE(*ASCII)
   ```

A diskette reader is automatically terminated at the end of the file or when an End Input (//ENDINP) command in the input stream is processed by the reader. You must start a reader for each separate file. For additional examples of using diskette commands, see Chapter 13 and *Starting a Diskette Writer* later in this chapter.

Holding a Reader

When you hold a reader, the process of reading input from a device or a data base file is stopped. The device that was being used stops operating but remains under control of the reader program; it cannot be used for another system operation. The device remains inactive, and no data is placed on the job queue as long as the reader is held. A reader that has been held can be either released or canceled.

To hold a reader, enter the Hold Reader (HLDRDR) command and specify the reader name in the RDR parameter. The name specified should be the same as the one specified in the RDR parameter of the command that started the reader. If no special name was specified when the reader was started (RDR parameter not entered), use the name specified in the DEV parameter of the STRCRDRDR or STRDKTRDR command or in the FILE parameter of STRDBRDR command.

Example:

To hold a reader named QCARD96, enter

```
HLDRDR RDR(QCARD96)
```

Holding a reader can also be done by selecting option 4 from the all readers display.

Releasing a Reader

When you release a reader that was previously held, the reading process resumes from the point where it was stopped. Data is not lost.

To release a reader, enter the Release Reader (RLSRDR) command and specify the reader name in the RDR parameter. This name should be the same as the one you specified in the HLDRDR command when you held the reader.

Example:

To release a reader named QCARD96, enter

```
RLSRDR RDR(QCARD96)
```

Releasing a reader can also be done by selecting option 6 from the all readers display.

Canceling a Reader

When you cancel a reader, the reading process is terminated, and the device being used is made available to the system for other uses. You can cancel a reader that is currently active or on the job queue.

To cancel a reader, enter the Cancel Reader (CNLRDR) command and specify the name of the reader in the RDR parameter. Use the OPTION parameter as needed to specify at what point the reading process is terminated. There are two options:

- If you do not enter the Option parameter, OPTION(*CNTRLD) is assumed and the reading process is not terminated until the job currently being read is completed and placed on the job queue.

- If you specify OPTION(*IMMED) the reading process is terminated immediately and the job currently being read is not placed on the job queue.

CAUTION
Do not use the CNLRDR command with OPTION(*IMMED) until you have tried a controlled cancel and the reader does not cancel. The *IMMED cancel may cause damage to system or user objects being used for the canceled reader.

Canceling a reader can also be done by selecting option 9 from the all readers display. OPTION(*CNTRLD) is the default when using option 9.

When you cancel a reader that was processing an input stream of one or more jobs from the MFCU, you must place the cards containing the JOB command and other records read by the MFCU back in the hopper with the rest of the input stream before you can start another reader to process any remaining jobs in the input stream (see *Reader Operating Considerations* in this chapter).

Examples:

1. To stop a reader named DATBASE after it has completed reading the current job, enter

```
CNLRDR RDR(DATBASE)
```

2. To stop a reader named CARD immediately such that the job currently being read is not placed on the job queue, enter

```
CNLRDR RDR(CARD) OPTION(*IMMED)
```

Writer Operating Considerations

Commands to Start Writers

You use a different command to start a writer to each type of output device. The types of output devices and the associated start writer command are as follows:

Output Device	**Command**
Card	Start Card Writer (STRCRDWTR)
Diskette	Start Diskette Writer (STRDKTWTR)
Printer	Start Printer Writer (STRPRTWTR)

Each command is discussed separately in this chapter.

When you enter a start writer command, your request is placed on the spooling subsystem job queue, QSPL, along with other spooling job requests. The writer is not started until your request is selected from the queue (for an additional explanation, see *Reader Operating Considerations*).

You can enter a command to start a writer when the spooling subsystem is inactive, but your request will not be taken from the queue, and the writer will not be started until the spooling subsystem is started.

The Start Print Writer (STRPRTWTR), Start Diskette Writer (STRDKTWTR), and Start Card Writer (STRCRDWTR) commands allow you to start a writer to a specific file on an output queue. Also, by using a parameter on the STRPRTWTR or STRCRDWTR command, you can select any page in the file as the point when you want to start printing, or select any card in a file where you want to start punching.

Writer Termination

When you enter a start writer command, you use the AUTOTRM parameter of the command to define whether the writer is to remain active or is to be terminated automatically after it has produced one file or when it has produced all of the spooled output files currently available (status (RDY) on display of output queue) on the output queue. Note that an output file is not considered available to be produced by a writer under any of the following conditions: (1) if it has been held on the queue (such as by the Hold Job or Hold Spooled File command) (2) if it is SCHEDULE(*JOBEND) and the job has not ended or (3) if it is still open and is not SCHEDULE(*IMMED).

Normally, you will want a printer writer to remain active so that it will produce any spooled output files that are placed on the queue later. It will remain active if you *do not* enter the AUTOTRM parameter when you enter the start writer command (the default value is *NO). You should specify AUTOTRM, (*YES) for a diskette or card writer, so that the device is made available for other uses (such as save/restore or another reader or writer).

You can terminate an active writer at any time by canceling it (see *Canceling a Writer* in this chapter).

Effect of Holding and Releasing a Writer

A writer sends a block of print, card, or diskette records at a time to the output device. Therefore, if you specify that the writer is to be held immediately (by entering the Hold Writer command with the *IMMED option), the device will not stop until the block of print, card, or diskette records has been produced (see *Holding a Writer* in this chapter). The device will stop anywhere within or at the end of a print line, or at the end of a card, or a diskette record. If you hold a writer with the *CNTLRD option, the device will not stop until the entire spooled output file currently being produced has been completed. If you hold a print writer with the *PAGEEND option, the device will stop at the end of a page.

When you release a diskette writer, the diskette magazine drive always restarts at the point where it stopped. When you release a card or printer writer, however, you have several options in specifying where to restart in the output file (see *Releasing a Writer* in this chapter). You can specify that the writer resume

- At the next print line or card
- At the beginning of the current spooled output file (the file that was being produced when the writer was held)
- At the beginning of the next spooled output file
- At a specific number of pages or cards beyond the page or card where the device stopped
- At a specific number of pages or cards previous to the page or card where the device stopped

Job/File Separators

A writer produces *job separators* at the beginning of the output of each new job to separate it from the previous job. The number of job separators, if any, is specified as an attribute of the output queue from which the writer is producing the output. You can change the number of job separators by using the JOBSEP parameter of the Change Output Queue (CHGOUTQ) command (see *Changing an Output Queue* in this chapter). If, instead of a number, you specify JOBSEP(*MSG), no job separators are produced and a message is issued that instructs you to remove the output produced for the previous job. Before the output is started for each new job, you should remove the output and reply to the message. When 0 is specified for job separators on an output queue, then no job separators are produced and no message to remove a job's output is issued.

A writer produces *file separators* at the beginning of the output of each new spooled output file to separate the file from the previous file. The number of file separators, if any, is based on the file separator attribute specified for the file. You can change the number of file separators by using the FILESEP parameter of the Change Spooled File Attributes (CHGSPFLA) command (see *Changing Spooled File Attributes* in this chapter).

If 0 is specified for file separators in output to be produced by a card writer, a message is issued before the output for each new file is started, so that you can remove the output from the previous file. The messages are sent to the message queue specified in the start writer command (see the following discussion of writer actions and messages).

Forms Type

When a writer is started, it is assumed that the standard forms have been mounted in the printer or that the standard cards have been placed in the hopper of the card device. Spooled output files for which FORMTYPE(*STD) is specified will be produced using these forms. You should check the device before starting the writer to ensure that the proper forms have been loaded.

If a particular forms type is specified for a file, a message is issued that instructs you to load the specified forms. The message is sent to the message queue specified in the start writer command (see the following discussion of writer actions and messages). You can change the type of forms for an output file by using the FORMTYPE parameter of the Change Spooled File Attributes (CHGSPFLA) command (see *Changing Spooled File Attributes* in this chapter).

Writer Actions and Messages

If a writer is unable to produce a particular spooled output file because of an error, it will issue a message that tells you it skipped the file and will go on to the next file on the output queue. The writer could, for example:

- Skip an output file for a card unit because the forms (cards) specified by the FORMTYPE parameter are not available.
- Skip an output file for a diskette because the correct volume is not available.
- Skip an output file for a printer because it requires a different print belt.

The file that was skipped remains on the output queue as held, and a message is issued that identifies the skipped file.

This type of message and any other messages that are related to the operation of a writer are sent to the message queue specified in the start writer command. If you do not specify a message queue when you enter the command to start a writer, the writer messages are sent to the message queue associated with the writer device, which may be the system operator message queue, QSYSOPR (see *Message Queues* in Chapter 11).

Messages regarding a problem with the device being used by a writer are also sent to the message queue specified in the start writer command (see *Operator/Device Interaction* in Chapter 9). Because these type of messages require you to take some action on a device, the specified message queue should be associated with a work station that is near the device. Most system devices are either contained in or connected to the 5381 System Unit; therefore, you should generally specify the system operator message queue (QSYSOPR), or the system console message queue (QCONSOLE) if QSYSOPR is allocated to a work station other than the system console. If the output is to be produced on a remote work station printer (5256 Printer), however, you should specify the message queue for a work station near the printer, so that any problems with the printer can be corrected by a person who is in the same area.

Notes:

1. A remote work station printer can be associated with the message queue of a work station near the printer device when the printer device description is created (see CRTDEVD command). This will allow the message queue to be used even when it is not specified on the STRPRTWTR command each time a writer is started.
2. Some messages regarding output errors may allow you to back up and rewrite records that were produced incorrectly; for example, reprinting the last few pages after a forms jam. If the output is being produced by a writer, the writer can recover records already produced and produce them again as indicated in the message. However, if the output is being produced by a program that does not use a writer, the program may not have the capability to recover records already produced, so that the backup option of the message does not work.

Starting a Card Writer

To start a writer that takes spooled output files from an output queue and produces the output on a card device, enter the Start Card Writer (STRCRDWTR) command. You must enter the DEV and OUTQ parameters.

DEV Parameter: Use this parameter to specify the name of the card device to be used to produce the spooled output.

OUTQ Parameter: Use this parameter to specify the name of the output queue from which the card writer is to produce the spooled output. If you know the library where the queue is located, include the library name also.

The remaining parameters are optional.

WTR Parameter: This parameter allows you to enter a unique name for the writer to be used. The writer is identified by that name until it terminates or is canceled.

- If you do not enter the parameter, the writer name is assumed to be the same as the device name you specified in the DEV parameter.

- The name you specify or let default must be different from any other writer name.

HOPPER Parameter: This parameter determines which hopper in the MFCU is used (see *5424 Multi-Function Card Unit* in Chapter 4).

- If you do not enter the parameter, the primary hopper (hopper 1) is used.

- If you specify HOPPER(2) the secondary hopper is used.

MSGQ Parameter: This parameter determines where messages generated by the writer are to be sent (see *Messages Queues* in Chapter 11).

- If you do not enter the parameter, the messages are sent to the message queue of the device (*DEVD).

- If you enter this parameter, you can specify that messages are to be sent to the message queue of the person who enters this command (*REQUESTER) or another message queue. For batch jobs, the value of *REQUESTER works the same as if it was *DEVD.

AUTOTRM Parameter: This parameter determines whether the writer terminates or remains active when the output queue is empty.

- If you do not enter the parameter, AUTOTRM(*NO) is assumed and the writer will remain active and produce any entries that appear later on the output queue.
- If you specify AUTOTRM(*YES) the default is no ready files *NORDYF and the writer is terminated when it can find no more available entries on the output queue. Spooled files on the queue that have been held are not considered available entries.
- If you specify AUTOTRM(*YES *FILEEND) the writer will terminate after it has produced the first file.

FILE Parameter: This parameter determines which spooled output file is processed first.

- If you do not enter this parameter, FILE(*NONE) is assumed and the writer will process the first available spooled output file on the output queue.
- If you specify FILE(QPRINT) the writer will process the file named QPRINT first.

JOB Parameter: This parameter specifies the job that created the spooled output file. This parameter applies only when FILE (spooled-file-name) is specified.

- If you do not specify the job parameter, the default says the same job that issued the start writer command is the job that created the spooled output file.
- If you specify JOB(JOB1.QSECOFR.000509) you identify the job, user, and job number that created the spooled output file.

SPLNBR Parameter: This parameter determines which spooled output file should be processed first if there is more than one spooled output file with the same name on an output queue. This parameter applies only when FILE (spool-file-name) is specified

- If you do not enter the SPLNBR parameter, SPLNBR(*ONLY) is assumed. If there is only one spooled output file with the specified spool file name on the output queue, the writer will process it first. If there is more than one spooled output file with the specified spool-file-name on the specified output queue, an error message is displayed.
- If you specify SPLNBR(*LAST) the writer will first process the highest numbered spooled output file with the specified spool-file-name on the specified output queue.
- If you specify SPLNBR(509) the writer will process spooled output file number 509 on the specified output queue.

CARD Parameter: This parameter determines which card the writer will process first. This parameter is valid only when the FILE parameter specifies a spooled output file.

- If you do not enter the CARD parameter, CARD(*BEGIN) is assumed and the writer will start punching the specified output file with the first card.

- If you specify CARD(4) the first card punched will be the fourth card in the output file.

Examples:

1. To write information from the QPUNCH output queue to the primary hopper of the device QCARD96 for punching, enter

```
STRCRDWTR DEV(QCARD96) OUTQ(QPUNCH)
```

2. To write information from the TERM output queue, located in the RECORDS library, to the secondary hopper of the device MFCU using the writer called PAYPUNCH, and terminate the writer when the queue is empty, enter

```
STRCRDWTR DEV(MFCU) OUTQ(TERM.RECORDS)
 WTR(PAYPUNCH) HOPPER(2) AUTOTRM(*YES)
```

Note: See the discussion of *Writer Operating Considerations* in this chapter.

Starting a Diskette Writer

To start a writer that takes spooled output files from an output queue and writes the output on one or more diskettes in the diskette drive, enter the Start Diskette Writer (STRDKTWTR) command. You must enter the DEV and OUTQ parameters.

DEV Parameter: Use this parameter to specify the name of the diskette device to be used for writing the spooled output on a diskette.

OUTQ Parameter: Use this parameter to specify the name of the output queue from which the diskette writer is to transfer the spooled output. If you know the library where the queue is located, include the library name also.

The remaining parameters are optional. The WTR, MSGQ, AUTOTRM, FILE, JOB, and SPLNBR parameters are used as described in *Starting a Card Writer*, earlier in this chapter. The LOC parameter is used as described in *Starting a Diskette Reader*.

A diskette writer will search for a particular volume identifier if a volume identifier was specified for an output file. If no volume identifier was specifed, the writer will produce the output on the diskette that is positioned for writing in the diskette drive. You will receive a message that identifies the diskette on which each output file was written.

Examples:

1. To write information out of the QDKT output queue onto diskettes in magazines 1 and 2 of the device called SAVRST, enter

```
STRDKTWTR DEV(SAVRST) OUTQ(QDKT)
```

2. To write information, using the writer WRDKT, from the INVREC output queue, located in the RECORDS library, onto the fifth through the last diskettes in magazine 2 of the device called STORE and terminate the writer when the output queue is empty, enter

```
STRDKTWTR DEV(STORE) OUTQ(INVREC.RECORDS) WTR(WRDKT)
  LOC(*M2 5) AUTOTRM(*YES)
```

Note: See the discussion of *Writer Operating Considerations* in this chapter.

Starting a Print Writer

To start a writer that takes spooled output files from an output queue and prints the output on a printer, enter the Start Print Writer (STRPRTWTR) command. You must enter the DEV and OUTQ parameters.

DEV Parameter: Use this parameter to specify the name of the printer to be used for printing the spooled output.

OUTQ Parameter: Use this parameter to specify the name of the output queue from which the writer is to print the spooled output. If you know the library where the queue is located, include the library name also.

The WTR, MSGQ, AUTOTRM, FILE, JOB, and SPLNBR parameters are optional. They are used as described in *Starting a Card Writer* earlier in this chapter. Using option 7 on the system operator menu is the same as entering this command and specifying only the DEV and OUTQ parameters.

PAGE Parameter: This parameter determines which page the writer will process first. This parameter applies only when the FILE parameter specifies a spooled output file.

- If you do not enter the PAGE parameter, PAGE(*BEGIN) is assumed and the writer will start printing the specified output file with the first page.

- If you specify PAGE(4) the first page printed will be the fourth page in the output file.

If the printer to be used is a remote work station printer, you should include the MSGQ parameter and specify the name of a message queue for a work station in the same area or specify that message queue when the device description for the printer is created (see CRTDEVD command). This will ensure that messages regarding any problem with the printer, such as a forms alignment, will go to someone near the printer who can correct the problem.

Examples:

1. To write information out of the QPRINT output queue (for output using one-part paper) so that it is printed on the system printer named QSYSPRT, enter

   ```
   STRPRTWTR DEV(QSYSPRT) OUTQ(QPRINT)
   ```

2. To print information on work station printer 5 (WSPRT5) from the QPRINT2 output queue (for output on two-part paper) using the writer called SPECIAL and have any related messages sent to work station 5 (WRKSTN5) located next to the printer, enter

   ```
   STRPRTWTR DEV(WSPRT5) OUTQ(QPRINT2) WTR(SPECIAL)
     MSGQ(WRKSTN5)
   ```

3. To write information out of the QPRINTS output queue (for output on special forms) so that it is printed on the second system printer named QSYSPRT2 and have the writer terminate when the output queue is empty, enter

   ```
   STRPRTWTR DEV(QSYSPRT2) OUTQ(QPRINTS) AUTOTRM(*YES)
   ```

4. To write a file to a printer device QSYSPRT, from a file named QPRTSPLO, so that the first page printed is page 10, you enter

   ```
   STRPRTWTR DEV(QSYSPRT) OUTQ(QPRINT)
     FILE(QPRTSPLO) JOB(QCONSOLE.QSECOFR.000625)
     SPLNBR(6) PAGE(10)
   ```

Holding a Writer

When you hold a writer, the process of producing output from an output queue is stopped. The device that was being used stops operating but remains under control of the writer program; it cannot be used for another system operation. The device remains inactive, and no output is produced from the output queue as long as the writer is held. A writer that has been held can be either released or canceled.

To hold a writer, enter the Hold Writer (HLDWTR) command and specify the writer name in the WTR parameter. The name specified should be as specified in the WTR parameter (or the DEV parameter if the WTR parameter was not used) of the command that started the writer.

Use the OPTION parameter as needed to specify at what point the writing process is to terminate. There are three options:

- If you do not enter the parameter, OPTION(*IMMED) is assumed and the writer stops as soon as it has completed the current block of output records, which is
  - Anywhere within or at the end of a print line for a printer
  - At the end of a card being punched in a card device
  - At the end of a record in a diskette device

  For an additional explanation, see *Writer Operating Considerations* in this chapter.

- If you specify OPTION(*PAGEEND) the print writer stops at the end of a page. This option is valid only for print writers.

- If you specify OPTION(*CNTRLD) the writer stops at the end of the current file; that is, at the end of the copy of the file currently being produced if multiple copies were to be made.

Examples:

1. To stop further processing of a writer named QSYSPRT as soon as it has completed the current block of print records, enter

```
HLDWTR WTR(QSYSPRT)
```

2. To stop further processing of a writer named QDKT when it reaches the end of the current file, enter

```
HLDWTR WTR(QDKT) OPTION(*CNTRLD)
```

To hold a writer use option 4 on the all writers display. Option (*IMMED) is defaulted.

Releasing a Writer

When you release a writer, the writing process resumes.

To release a writer, enter the Release Writer (RLSWTR) command and specify the writer name in the WTR parameter. This name should be the one you specified in the HLDWTR command when you held the writer.

The point where writing resumes is determined by the OPTION parameter:

- If you do not enter this parameter, OPTION(*CURRENT) is assumed and the writer resumes at the point it was held. This is the only valid option when you are releasing a diskette writer.

- If you specify OPTION(*BEGIN) the writer resumes at the beginning of the current file.

- If you specify OPTION(*BYPASS) the writer resumes at the beginning of the next file, but the current file (the file it was writing when held) is held on the output queue.

- If you specify + and a number, such as OPTION(+2) the writer resumes that number of pages (for a printer) or cards (for a card unit) *beyond* the page or card where the device stopped.

- If you specify - and a number, such as OPTION(-2) the writer resumes that number of pages or cards *previous* to the page or card where the device stopped.

- Releasing a writer can also be done using option 6 on the all writers display. Option (*CURRENT) is defaulted.

Examples:

1. To release the writer named QDKT at the point where it was stopped, enter

   ```
   RLSWTR WTR(QDKT)
   ```

2. To release the writer named QSYSPRT so that it resumes at the beginning of the file, enter

   ```
   RLSWTR WTR(QSYSPRT) OPTION(*BEGIN)
   ```

3. To release the writer named QCARD96 so that it resumes with the next file on the output queue, enter

   ```
   RLSWTR WTR(QCARD96) OPTION(*BYPASS)
   ```

4. To release the writer named WSPRT so that it backs up two pages and resumes at that point, enter

   ```
   RLSWTR WTR(WSPRT) OPTION(-2)
   ```

Canceling a Writer

When you cancel a writer, the writing process is terminated and the device being used is made available to the system for other uses. You can cancel a writer that is currently active or one that is being held or one that is still on the job queue.

To cancel a writer, enter the Cancel Writer (CNLWTR) command and specify the name of the writer in the WTR parameter. Use the OPTION parameter as needed to specify at what point the writer is to be terminated. There are three options:

- If you do not enter the OPTION parameter, OPTION(*CNTRLD) is assumed and output stops at the end of the file (or current copy of the file) being written to the output device.
- If you specify OPTION(*PAGEEND) the output stops at the end of a page, and the file being produced remains on the output queue. If a file was being produced at the time the writer was canceled, and another writer is started to the output queue at some later time, the file is produced in its entirety even if part of it had been produced before. This option is valid only on print writers.
- If you specify OPTION(*IMMED) output stops immediately, but the file being produced remains on the output queue. If a file was being produced at the time the writer was canceled and another writer is started to the output queue at some later time, the file is produced in its entirety even if part of it had been produced before.

CAUTION
Before using the CNLWTR command with the OPTION(*IMMED) parameter you should try:

- CNLWTR command with OPTION(*CNTRLD) for card, diskette or print writers.
- CNLWTR command with OPTION(*PAGEEND) for print writers.
- HLDWTR

The OPTION(*IMMED) parameter should not be used unless any of the above commands fail to stop the writer. Using the OPTION(*IMMED) parameter value may cause damage to system and user objects used by the writer.

Using option 8 on the system operator menu is the same as entering this command and specifying only the WTR parameter.

Canceling a writer can be done using option 9 on the all writers display. Option (*CNTRLD) is defaulted.

Examples:

1. To terminate a writer named QSYSPRT after it has completed writing the current file, enter

   ```
   CNLWTR WTR(QSYSPRT)
   ```

2. To terminate a writer named QDKT immediately such that the entire file it is currently writing remains on the output queue, enter

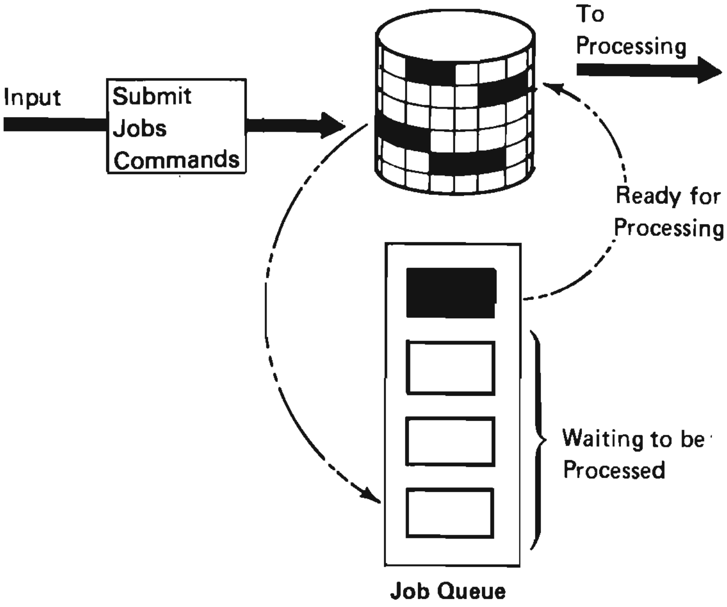
   ```
   CNLWTR WTR(QDKT) OPTION(*IMMED)
   ```

SUBMIT JOBS COMMANDS

The Submit Jobs commands read an input stream of job information into the system from an input source. The input stream consists of one or more batch jobs and any related inline data files to be processed. The input stream may be

- A diskette file on one or more diskettes that have been placed in the diskette magazine drive (see Chapter 4).
- A data base file in internal storage.
- A card file (stack of cards) that has been placed in a card device such as the MFCU (see Chapter 4).

When a batch job is read from an input source by the Submit Jobs commands, the input stream is stored in the system and an entry for the job is placed on a job queue. The job information remains stored in the system where it was placed by the reader until the job entry is selected from the job queue for processing by a subsystem (see *Subsystems* in Chapter 8). The spooling subsystem does not have to be active to use the Submit Jobs commands.



Submit Jobs Operating Considerations

You use a different command for each type of input source. The input sources and the associated commands are as follows:

Input Source	Command
Diskette	Submit Diskette Jobs (SBMDKTJOB)
Data base	Submit Data Base Jobs (SBMDBJOB)
Card	Submit Card Jobs (SBMCRDJOB)

Each command is discussed separately in this chapter.

The spooling subsystem is not used by the Submit Jobs commands. These commands execute within the job and subsystem where they are entered.

Submit Jobs Commands Termination

The SBMDBJOB command and the SBMDKTJOB command will terminate automatically at the end of the file being read. A Submit Card Jobs (SBMCRDJOB) command will remain active after the last card has been read, so that you can load additional cards (input stream) into the device and continue reading jobs and data into the system. The End Input (//ENDINP) command must be used to terminate a Submit Card Jobs (SBMCRDJOB) command. The End Input command in the input stream will also terminate a Submit Data Base Jobs (SBMDBJOB) command and a Submit Diskette Jobs (SBMDKTJOB) command.

A Submit Diskette Jobs (SBMDKTJOB) or Submit Data Base Jobs (SBMDBJOB) command can start only at the beginning of an input stream. Therefore, if you enter another Submit Jobs command for the input stream, all of the jobs in the input stream will be read in and placed on a job queue, even if some of them had been read in previously. Any jobs in the input stream that were on the job queue previously will have a second entry on the job queue, which means that they will be processed twice unless you prevent it. You can use the following procedure to prevent the jobs from being processed twice:

1. Hold the job queue (see *Holding Job and Output Queues* in this chapter). While the job queue is being held, none of the jobs currently on the queue or subsequently placed on the queue will be processed.

2. Enter the Submit Jobs command.

3. When all of the jobs in the input stream have been placed on the job queue, cancel the jobs that were previously processed (see *Canceling a Job* in Chapter 12). The canceled jobs are removed from the queue but their job logs are written (put on an output queue) and you may wish to cancel these job logs also. They may not be sent to the same queue, so perhaps a search, using DSPSBS to show the jobs or an output queue, would be the quickest way to find all of them.

4. Release the job queue (see *Releasing Job and Output Queues* later in this chapter). Processing will resume on the remaining jobs.

A Submit Card Jobs (SBMCRDJOB) command can start in the middle of an input stream, but it should start at the beginning of a job (the first card must begin with the characters // JOB). If you are entering another Submit Card Jobs (SBMCRDJOB) command to read in the remainder of an input stream, use the following procedure:

1. Remove the cards for all jobs in the input stream that the Submit Jobs command had completed reading. Those jobs have already been placed on a job queue for processing.

2. If the Submit Jobs command was stopped while it was in the process of reading a particular job in the input stream, that job was not placed on a job queue. Retrieve all cards to the beginning of the job (the last // JOB card read and any cards that follow it) and replace them in the hopper.

3. Ready the card device (for example, press the Start key on the MFCU).

4. Enter the Submit Card Jobs (SBMCRDJOB) command.

Submit Jobs Messages

If a problem occurs with the device (card or diskette), an error message is sent to the message queue specified in the MSGQ parameter of the Submit Jobs command. Unless you specify a different message queue in the Submit Jobs command, the messages are sent to the message queue defined for the device being used. Examples of device error messages are given in the discussion of *Operator/Device Interaction* in Chapter 9.

Submitting Diskette Jobs

To read a batch input stream from a diskette file, enter the Submit Diskette Jobs (SBMDKTJOB) command. You must enter the DEV and LABEL parameters.

DEV Parameter: Use this parameter to specify the name of the diskette device to be used.

LABEL Parameter: Use this parameter to specify the data file identifier of the input stream to be read.

Using option 9 on the system operator menu is the same as entering this command with only the DEV and LABEL parameters.

The remaining parameters define the diskette or diskettes to be used and the coding format.

LOC Parameter: This parameter determines which diskettes are to be used in which magazine or I/O slot of the diskette magazine drive.

- If you do not enter this parameter, the operation begins with the first diskette in magazine 1 and continues through each successive diskette in both magazines to the last diskette in magazine 2 or until the end of the file.

- If you enter this parameter, you can specify one of the following options for the magazine or I/O slot:

*M12	magazine 1 followed by magazine 2
*M1	magazine 1 only
*M2	magazine 2 only
*S1	slot 1 only
*S2	slot 2 only
*S3	slot 3 only
*S12	slot 1 followed by slot 2
*S23	slot 2 followed by slot 3
*S123	slot 1 followed by slots 2 and 3

- If you specified a magazine or more than one I/O slot, you can include two additional values that define the starting diskette and the ending diskette.

- The second value specifies the starting diskette in the magazine or I/O slots.
  - If you do not enter this value, the operation begins with the first, or leftmost, diskette as viewed from the front of the diskette magazine drive (the diskette in position 1).
  - If you specify *CURRENT, the process begins with the diskette from which the diskette magazine drive last read a file.
  - If you specified a magazine in the first value, you can specify a number from 1 through 10 for the second value, and the operation will begin with the diskette in that position in the magazine.

- The third value specifies the ending diskette in the magazine or I/O slots.
  - If you do not enter this value, the operation will end with the last, or rightmost, diskette (the diskette in position 10).
  - If you specify *WRAP, the procedure used depends on whether you specified I/O slots, one magazine, or both magazines.

    I/O Slots: The operation begins at the leftmost specified slot (the slot with the lowest number) and proceeds from left to right to the rightmost specified slot (the slot with the highest number). You then receive a message to replace all the diskettes in the specified slots.

    One Magazine: The operation begins at the specified starting position in the magazine and continues to the last diskette in the magazine (the diskette in position 10). You then receive a message to mount a new magazine. The operation resumes with the diskette in position 1 of the new magazine.

    Both Magazines: The operation begins at the specified starting position in magazine 1 and continues through the last diskette in magazine 2. You then receive a message to mount two new magazines. The operation resumes with the diskette in position 1 of magazine 1.

  - If you specify *ONLY, only the single starting diskette specified in the second value is used.
  - If you specified a magazine in the first value, you can specify a number from 1 through 9 or 0 for the second value, and the operation will end with the diskette in that position in the magazine.

For additional information on the LOC parameter, see *Diskette Considerations* in Chapter 13.

VOL Parameter: This parameter allows you to specify that only a diskette or diskettes having a particular identifier is to be used.

- If you do not enter this parameter, any diskette in the position specified in the LOC parameter is used.

- If you enter this parameter and specify one or more volume identifiers (any combination of letters and numbers up to a maximum of six), such as VOL(ABCDE5) each diskette specified in the LOC parameter is used only if it has the specified volume identifier. If the diskette does not have the correct identifier, you will receive a message to mount the correct diskette.

For additional information on the VOL parameter, see *Diskette Considerations* in Chapter 13.

CRTDATE Parameter: This parameter allows you to specify that a diskette is to be used only if it contains a data file created on a certain date.

- If you do not enter this parameter, no check is made for a create date.
- If you enter this parameter and specify a create date, the diskette file is matched against that date. If it does not match, an error message is sent.

CODE Parameter: This parameter determines the type of character code to be used to read the diskette data onto the job queue.

- If you do not enter this parameter, EBCDIC code is used.
- If you specify CODE(*ASCII) ASCII code is used.

JOB Parameter: This parameter determines the job queue on which the job is placed if JOBQ(*RDR) was specified in the JOB command used to define the job.

- If you do not enter this parameter, the input is placed on the batch job queue (QBATCH).
- If you enter this parameter and specify a job queue name, the input is placed on that job queue.

MSGQ Parameter: This parameter determines where device related messages are sent (see message discussion in Chapter 11).

- If you do not enter this parameter, the messages are sent to the message queue of the device (*DEVD).
- If you enter this parameter, you can specify that messages be sent to your work station message queue (*REQUESTER) or to another message queue.

Normally, you will not enter this parameter because you will want the messages to be sent to QSYSOPR. However, if QSYSOPR is currently assigned to the interactive job at another work station, you can enter this parameter and specify any message queue so that any device related messages are sent to you.

DSPSBMJOB Parameter: This parameter specifies whether the jobs being submitted are allowed to be displayed on the submitted jobs display. Any submitted job of the type specified by the SBMFROM parameter of the DSPSBMJOB command can be displayed.

- If you do not enter this parameter, the job will be displayed by the DSPSBMJOB command.
- If you enter *NO, the job will not be displayed by the DSPSBMJOB command.

Examples:

1. To submit jobs from diskette, using a device named QDKT, read a file labeled PAY from diskettes of magazines 1 and 2 and put the job in that file onto the batch job queue (QBATCH) in EBCDIC code, enter

```
SBMDKTJOB  DEV(QDKT)  LABEL(PAY)
```

2. To submit jobs from diskette, using a device named DKTMD, read a file labeled UPDATE from the second through sixth diskettes of magazine 2 into the job queue INVENTORY of library RECORDS in ASCII code,·enter

```
SBMDKTJOB  DEV(DKTMD)  LABEL(UPDATE)
  LOC(*M2  2  6) JOBQ(INVENTORY.RECORDS)  CODE(*ASCII)
```

A Submit Diskette Jobs (SBMDKTJOB) command is automatically terminated at the end of the file or when an End Input (//ENDINP) command is processed in the input stream. Use a separate Submit Diskette Jobs (SBMDKTJOB) command for each separate file. For additional examples of using diskette commands, see Chapter 13 and *Starting a Diskette Writer* later in this chapter.

Submitting Data Base Jobs

To read a batch input stream from a data base file, enter the Submit Data Base Jobs (SBMDBJOB) command. You must enter the FILE parameter. All other parameters are optional.

FILE Parameter: Enter this parameter and specify the name of the data base file that contains the input stream.

MBR Parameter: This parameter is used to specify a particular member of the file where the job or jobs are to be found.

- If you do not enter this parameter, the first member in the file is used.

- If you enter this parameter and specify a member name, the job or jobs are read from that member.

The JOBQ and DSPSBMJOB parameters are used as described previously in *Submitting Diskette Jobs*.

Using Option 10 on the system operator menu is the same as entering this command with only the FILE and MBR parameter.

Examples:

1. To read a job from member WKPAY35 in the data base file ACCSRC, enter

```
SBMDBJOB FILE(ACCSRC) MBR(WKPAY35)
```

2. To read jobs from the data base file PARTS, located in the library INVTRY, into the job queue PROC, located in the library QGPL, enter

```
SBMDBJOB FILE(PARTS.INVTRY)
 JOBQ(PROC.QGPL)
```

A Submit Data Base Jobs (SBMDBJOB) command is automatically terminated at the end of the file or when an End Input (//ENDINP) command is processed within the job.

Submitting Card Jobs

To read an input stream of one or more batch jobs from a card input device and place them on a job queue, enter the Submit Card Jobs (SBMCRDJOB) command. You must enter the DEV parameter. All other parameters are optional.

DEV Parameter: Enter this parameter and specify the name of the device to be used.

HOPPER Parameter: This parameter determines which hopper in the MFCU is used (see *5424 Multi-Function Card Unit* in Chapter 4).

- If you do not enter this parameter, the primary hopper (hopper 1) is used.
- If you specify HOPPER(2) the secondary hopper is used.

The JOBQ, MSGQ, and DSPSBMJOB parameters are used as described previously in *Submitting Diskette Jobs*.

Examples:

1. To submit card jobs with the card device QCARD96 so that the jobs on the cards in the primary hopper are read into the QBATCH job queue, enter

```
SBMCRDJOB  DEV(QCARD96)
```

2. To submit card jobs with the device CARD so that the jobs on cards in the secondary hopper are read and placed on the PAY job queue in the library PAYROLL and any device related messages are sent to the WRKSTN3 message queue, enter

```
SBMCRDJOB  DEV(CARD)  HOPPER(2)
  JOBQ(PAY.PAYROLL)  MSGQ(WRKSTN3)
```

The Submit Card Jobs (SBMCRDJOB) command will remain active after the last card is read unless an End Input (//ENDINP) command followed by a /* card is read.

Comparison of Start Reader Commands and Submit Jobs Commands

Submit Jobs commands can be used to accomplish most of the functions performed by spool readers, that is, they both interpret input streams and submit jobs to job queues. The differences between the submit jobs function and a reader function are that the Submit Jobs commands run in the same job as the requester of the function and do not perform syntax checking while the reader runs in its own job and will do snytax checking if specified on the JOB command.

The choice between using Submit Jobs commands and using Start Reader commands is one of operation. Because the submit jobs function runs synchronously within the job that invokes it, the user cannot continue other work on the same work station or batch job until the command is completed. Also, syntax checking of the input stream is not available on the Submit Jobs commands. On the other hand, the overhead to the system is less with a submit jobs function than a reader because a separate job does not have to be created to do the work. Because of this, the submit jobs functions use fewer system resources, have fewer effects on the system in general, and start reading somewhat sooner than a reader. The QSPL subsystem is not involved in the Submit Jobs commands and does not have to be started to use the Submit Jobs commands. In contrast, a reader does its work asynchronously with respect to the user job, and thus may be started and allowed to do its work while other work is being done by the user. A reader will also do syntax checking of jobs.

The Submit Jobs commands cannot be controlled with any hold, release, cancel, or display commands, but like the readers, they will terminate after the End Input (//ENDINP) command is processed within the input stream.

Comparison of the Submit Job (SBMJOB) Command and the Submit Jobs (SBMCRDJOB), (SBMDKTJOB), (SBMDBJOB) Commands

All of these commands can be used to submit batch jobs to a batch job queue. The difference in usage of these commands is where the job comes from. The following summary will help you choose which command to use.

- Submit Job (SBMJOB) Command: This command is used to submit a job to a batch job queue by using the job description name that is in the system. You can also submit a single command to a batch job queue from a work station.

- Submit Jobs (SBMCRDJOB), (SBMDKTJOB), (SBMDBJOB) Commands: These commands are used to submit a job to a batch job queue from a card device, a diskette device, or a data base file. The job description is taken from the //JOB statement in the input stream.

For details on these commands, see the *System/38 CL Reference Manual,* SC21-7731.

CONTROLLING SPOOLING QUEUES AND FILES

There may be occasions where you need to intervene in the way the spooling queues or the spooled files are handled on the system. For example, you may want to hold batch jobs at times when the interactive work becomes extremely heavy, or you may want to hold certain printing output until other output has been completed on a printer. You can hold, release, and clear a job or output queue, which affects all jobs or files on the queue. You can also hold, release, cancel and change the attributes of an individual spooled file.

Holding Job and Output Queues

When you hold a job queue, no jobs can be taken from the queue to be processed. Jobs can still be added to the queue, but they are also held. Similarly, when you hold an output queue, none of the files on the queue can be produced, including any files added after the queue was held; however, any file that was being produced when the queue was held will be completed. Note that when you hold a job queue, any job on the queue is held in its entirety. When you hold an output queue, only the files from a job that are on the queue will be held; files from a job that are on another output queue will not be held.

While a job queue is being held, you can remove a job from it by using the Cancel Job (CNLJOB) command (see *Canceling a Job* in Chapter 12). Similarly, you can use the Cancel Spooled File (CNLSPLF) command to remove spooled files from an output queue that is being held or use CHGSPLFA to move a file to a queue that is not held. (See *Canceling a Spooled File* in this chapter.)

To hold a job queue, enter the Hold Job Queue (HLDJOBQ) command and specify the name of the job queue in the JOBQ parameter. To hold an output queue, enter the Hold Output Queue (HLDOUTQ) command and specify the name of the output queue in the OUTQ parameter. If you know the library where the job queue or output queue is located, include the library name with the queue name.

Holding job queues and output queues can also be done from the display of all job queues and the display of all output queues by selecting option 4. (This uses the fully qualified queue name.)

Examples:

1. To suspend processing of any jobs in the batch job queue, enter

```
HLDJOBQ JOBQ(QBATCH)
```

2. To prevent output from being written from the QDKT output queue located in the library QGPL, enter

```
HLDOUTQ OUTQ(QDKT.QGPL)
```

Releasing Job and Output Queues

When you release a job queue, the jobs on the queue become available for processing. When you release an output queue, the spooled files on the queue can be produced. Note, however, that if specific jobs on a job queue or specific files on an output queue were held by some means other than the HLDJOBQ or HLDOUTQ command, those jobs or files are not released (see *Holding Jobs* in Chapter 12 and *Holding Spooled Files* later in this chapter).

To release a job queue, enter the Release Job Queue (RLSJOBQ) command and specify the name of the job queue in the JOBQ parameter. To release an output queue, enter the Release Output Queue (RLSOUTQ) command and specify the name of the output queue in the OUTQ parameter. If you know the library where the job queue or output queue is located, include the library name with the queue name.

Releasing job queues and output queues can also be done from the display of all job queues and the display of all output queues by selecting option 6. (This uses the fully qualified name.)

Examples:

1. To resume processing of jobs in the batch job queue, enter

```
RLSJOBQ JOBQ(QBATCH)
```

2. To release all files in the QPUNCH output queue, located in the QGPL library, for writing to an output device, enter

```
RLSOUTQ OUTQ(QPUNCH.QGPL)
```

Clearing Job and Output Queues

When you clear a job queue, all job entries currently waiting on the queue to be processed, or being held, are removed. Similarly, when you clear an output queue, all file entries waiting on the queue to be produced, or being held, are removed. However, any jobs currently being read in by a reader or Submit Jobs commands are not removed from a job queue, nor are any output files with a status of open (OPN), or writing (WTR) removed from an output queue. Also, entries may be left on the job or output queue if all necessary resources are not available to completely clear the queue, and interactive jobs that are on a job queue as the result of a TFRJOB command are not removed.

To clear a job queue, enter the Clear Job Queue (CLRJOBQ) command and specify the name of the job queue in the JOBQ parameter. To clear an output queue, enter the Clear Output Queue (CLROUTQ) command and specify the name of the output queue in the OUTQ parameter. If you know the library where the job or output queue is located, include the library name with the queue name. These commands return a message indicating how many entries are removed and how many are kept. The queue name is also given so that CL program job logs will be meaningful.

Examples:

1. To remove all job entries currently on the batch job queue, enter

   ```
   CLRJOBQ JOBQ(QBATCH)
   ```

2. To remove all output files currently on the QDKT output queue in the QGPL library, enter

   ```
   CLROUTQ OUTQ(QDKT.QGPL)
   ```

Holding a Spooled File

When you hold a spooled output file, no output can be written from that file, but other files on the output queue are still available to be produced. If the file that is held was in the process of being produced, the writer stops writing the file and goes to the next available file on the queue. The file remains held until:

- The file is released.
- The file is canceled.
- The job that created the file is canceled and SPLFILE(*YES) was specified in the CNLJOB command.
- The output queue containing the file is cleared.

To hold a spooled file, enter the Hold Spooled File (HLDSPLF) command and specify the name of the file in the FILE parameter. There are two optional parameters:

JOB Parameter: This parameter supplies the name of the job that created the file.

- If you submitted the job from your work station, you do not need to enter this parameter.
- If you did not submit the job, you should enter the parameter and specify the job name, along with the user name and job number if known.

SPLNBR Parameter: This parameter supplies the number of the spooled file to be held.

- If only one spooled file from the job has the file name you specified in the FILE parameter, you do not need to enter this parameter.

- If more than one file from the job has the specified file name, you should enter the parameter and specify either SPLNBR(*LAST) if the highest numbered file for the job is to be held, or the file number such as SPLNBR(3) if a file other than the one with the highest number is to be held.

Holding a spooled file can also be done using:

- Option 4 from a specific output queue display.

- Option 4 from a job's files display. The job's files display can be obtained from option 6 of DSPJOB, option 2 of DSPSBMJOB, and option 2 of DSPSBS.

- Option 4 from a specific writer display.

The displays use all three parameters. SPLNBR is the file number so the file is completely specified.

Examples:

1. To prevent the file named LAMPS, which is the only output file in the job CATALOG, from being punched on cards, enter

```
HLDSPLF FILE(LAMPS) JOB(CATALOG)
```

2. To prevent the second of three files named CHECKS in the job PAY, which was submitted by the user DEPT25 and has a job number 000275, from being printed, enter

```
HLDSPLF FILE(CHECKS) JOB(PAY.DEPT25.000275) SPLNBR(2)
```

Releasing a Spooled File

When you release a spooled file that was previously held, it again becomes available to be produced from the beginning of the file whether it was held before it could be written out or while it was being written out.

To release a spooled file, enter the Release Spooled File (RLSSPLF) command and specify the name of the file in the FILE parameter. You should also use the JOB parameter if you did not submit the job that created the file, and the SPLNBR parameter if there is more than one file in the job with the file name you specified. All of the parameters are used as described in *Holding a Spooled File.*

Releasing a spooled file can also be done using option 6 from the job's files display or the display of a specific output queue. The job's files display can be obtained from option 6 of DSPJOB, option 2 of DSPSBMJOB, and option 2 of DSPSBS. The displays use all three parameters. SPLNBR is the file number so the file is completely specified.

Examples:

1. To release the file named LAMPS, which is the only file in the job CATALOG, so that it can be punched on cards, enter

   ```
   RLSSPLF FILE(LAMPS) JOB(CATALOG)
   ```

2. To release the highest numbered file with the name NEW in the printing job you submitted, enter

   ```
   RLSSPLF FILE(NEW) SPLNBR(*LAST)
   ```

Canceling a Spooled File

When you cancel a spooled file, that file is removed from the output queue and cannot be produced on an output device. If the file is being produced when it is canceled, the writer stops producing the file and goes on to the next file in the queue. Any output that has not been produced from the file is lost.

To cancel a spooled file, enter the Cancel Spooled File (CNLSPLF) command and specify the name of the file in the FILE parameter. You can cancel one file at a time with the CNLSPLF command. You should also use the JOB parameter if you did not submit the job that created the file, and the SPLNBR parameter if there is more than one file in the job with the file name you specified. All of the parameters are used as described in *Holding a Spooled File*.

Canceling a spooled file can also be done using option 9 from:

- Job's files display. This job's files display can be obtained from option 6 of DSPJOB, option 2 of DSPSBMJOB, and option 2 of DSPSBS.

- Specific output queue display.

- Specific writer's display.

The displays use all three parameters. SPLNBR is the file number so the file is completely specified.

Examples:

1. To remove from the output queue the highest numbered file named SHOES in the job CATALOG, enter

   ```
   CNLSPLF FILE(SHOES) JOB(CATALOG) SPLNBR(*LAST)
   ```

2. To remove from the output queue the second of three files named CHECKS in the job PAY, which was submitted by the user DEPT25 and has a job number 000275, enter

   ```
   CNLSPLF  FILE(CHECKS) JOB(PAY.DEPT25.000275)
    SPLNBR(2)
   ```

CHANGING SPOOLING QUEUES AND FILES

If the output requirements of a spooled file change, you can change the attributes of the file; that is, the characteristics that define how the file output is to be handled. You can also change an output queue to modify the number of separators between jobs. If a file is being produced by a writer, only COPIES and SAVE can be changed.

Changing Spooled File Attributes

To change the attributes of a spooled output file, enter the Change Spooled File Attributes (CHGSPLFA) command and specify the name of the file in the FILE parameter. Use the JOB and SPLNBR parameters when needed to further define which file is to be changed, as described in *Holding a Spooled File.* You can also receive a prompt to enter this command from the DSPSPLFA display; the FILE, JOB, and SPLNBR are filled in on the prompt.

The remaining parameters are used as follows.

OUTQ Parameter: This parameter determines on which output queue the file is placed.

- If you do not enter the parameter, OUTQ(*SAME) is assumed and the file remains on the same output queue.

- If you specify another output queue, such as OUTQ(QPRINTS) the file is moved to that queue.

You might use this parameter to put a spooled output file

- On a special output queue for low- or high-priority output

- On an output queue from which a writer is currently producing output

- On an output queue for a different type of device so that the output is redirected to that device

Note: Redirecting output to the MFCU will result in punch checks if the data being redirected contains characters other than the 64 characters supported by the MFCU.

FORMTYPE Parameter: This parameter, which is used only for printed and card output, determines the type of forms on which the file is produced.

- If you do not enter the parameter, FORMTYPE(*SAME) is assumed and the type of forms is not changed.

- If you specify a forms identifier, such as FORMTYPE(M02-8416-1) you are instructing the system that the output is to be printed on the specified form.

COPIES Parameter: This parameter, which is used only for card and printed output, determines the number of copies that are made of the file.

- If you do not enter the parameter, COPIES(*SAME) is assumed and the number of copies remains the same.

- If you specify a number from 1 through 99, such as COPIES(12) that number of copies is made of the file when it is written out.

FILESEP Parameter: This parameter, which is used only for punched card or printed output, determines how many pages or cards are to be produced at the beginning of the file output to separate it from previous file output (see the discussion of file separators under *Writer Operating Considerations* in this chapter).

- If you do not enter the parameter, FILESEP(*SAME) is assumed and the number of separator pages or cards remains the same.

- If you specify a number (from 0 through 9), such as FILESEP(5) that number of separator pages or cards are produced at the beginning of each copy of the file output.

SCHEDULE Parameter: This parameter determines when the spooling writer can begin producing output from the file on the output device.

- If you do not enter the parameter, SCHEDULE(*SAME) is assumed and the schedule remains the same.

- If you specify SCHEDULE(*JOBEND) output is not produced on the device until the entire job is completed.

- If you specify SCHEDULE(*FILEEND) output is produced on the device after the last output record has been processed by the program and the file has been closed.

- If you specify SCHEDULE(*IMMED) output is produced on the device as soon as the first output records are made available to the file; that is, the output writer can begin writing output even while the file is being produced.

SAVE Parameter: This parameter determines whether the file is to be retained on the output queue (saved) so that it can be used to produce other copies of the output.

- If you do not enter this parameter, SAVE(*SAME) is assumed and the file is saved or not saved as previously specified.

- If you specify SAVE(*NO) the file is not saved.

- If you specify SAVE(*YES) the file is saved.

PRTIMG Parameter: This parameter, which is used for print files only, specifies the print image that is to be used to produce the file on a printer.

- If you do not specify this parameter, PRTIMG(*SAME) is assumed and the print image associated with the output file at the time the printer device file was opened is used.

- If you specify PRTIMG(*DEVD) the standard print image specified in the printer device description is used.

- If you specify PRTIMG(CHAR48PI) the print image named CHAR48PI is used.

TRNTBL Parameter: This parameter, which is used for print files only, specifies the translate table to be used when the output data is to be translated before it is printed.

- If you do not specify this parameter, TRNTBL(*SAME) is assumed and the translate table associated with the output file at the time the printer device file was opened is used.

- If you specify TRNTBL(*PRTIMG) the translate table with the same qualified name as the print image is used.

- If you specify TRNTBL(*NONE) no translation is done when the output file is produced.

- If you specify TRNTBL(SCRAMTBL) the translate table named SCRAMTBL is used to convert unprinted characters before the output file is produced.

VOL Parameter: This parameter, which is for diskette output files only, specifies one or more volume identifiers of the diskettes on which the output file is to be written.

- If you do not specify this parameter, VOL(*SAME) is assumed and the volume identifiers associated with the output file at the time the diskette device file was opened is used.

- If you specify VOL(*NONE) the output file is written to the first available diskette.

- If you specify VOL(PAYROL) the diskette volume named PAYROL is to be used for this output file.

LABEL Parameter: This parameter, which is for diskette output files only, specifies the data file identifier to be written on diskette from this output file.

- If you do not specify this parameter, LABEL(*SAME) is assumed and the data file identifier associated with the output file at the time the diskette device file was opened is used.

- If you specify LABEL(PAYROLL1) the data file identifier PAYROLL1 will be written on diskette for the output file.

CODE Parameter: This parameter, which is for diskette files only, specifies the type of character code to be used when the output file is written to diskette.

- If you do not enter this parameter, CODE(*SAME) is assumed and the character code associated with the output file at the time the diskette device file was opened is used.

- If you specify CODE(*EBCDIC) the EBCDIC character code is used.

- If you specify CODE(*ASCII) the ASCII character code is used.

Examples:

1. To change the file named DEPT511 so that two copies are punched with five separator cards at the beginning of each set of cards, enter

   ```
   CHGSPLFA FILE(DEPT511) COPIES(02) FILESEP(5)
   ```

2. To change the file named SALES of the job BILLING, which was submitted by the user JONES and has a job number 000147, so that it is moved to the QPRINT2 output queue and printed on form 1140-6, enter

   ```
   CHGSPLFA FILE(SALES) JOB(BILLING.JONES.000147)
    OUTQ(QPRINT2) FORMTYPE(1140-6)
   ```

Changing spooled file attributes is also available as a prompt when CF3 is pressed from the spooled file attributes display.

Changing an Output Queue

To change an output queue, enter the Change Output Queue (CHGOUTQ) command and specify the name of the output queue (and the name of the library where it is located if known) in the OUTQ parameter. You normally will not change the DSPDTA and OPRCTL parameters; they define who has authority to control the queue or to display data in the queue. If you find that you do not have authority to change an output queue, ask an authorized person to change it for you. You are most likely to change the JOBSEP parameter, but you may also change the TEXT parameter.

JOBSEP Parameter: This parameter, which is used only for punched card or printed output, determines how many separator pages or cards are to be produced between jobs on the queue (see the discussion of job separators under *Writer Operating Considerations* in this chapter). Note the similarity to the FILESEP parameter of the CHGSPLFA command. The number of job separators cannot be changed if a writer is active.

- If you do not enter the JOBSEP parameter, JOBSEP(*SAME) is assumed and the number of separator pages or cards remains the same.

- If you specify a number (from 0 through 9), such as JOBSEP(5) that number of separator pages or cards is produced at the beginning of each job on the queue when each is written out.

- If you specify JOBSEP(*MSG) no job separator pages or cards will be produced at the beginning of each job on the queue, but a message will be sent at the beginning of each job's output so that previous output can be removed.

TEXT Parameter: This parameter can be used to include text that describes the output queue. The text will appear when the output queue is displayed by the DSPOBJD command.

- If you do not enter the parameter, TEXT(*SAME) is assumed and the text, if any, will remain the same.

- If you enter the parameter, you can specify that there is to be no text (*BLANK) or up to 50 characters of text that will replace any existing text.

Examples:

1. To change the output queue named QPRINTS so that two separator pages are placed before each job printed from the queue, enter

```
CHGOUTQ OUTQ(QPRINTS) JOBSEP(2)
```

2. To change the output queue named QPUNCH in the library QGPL to include the statement that it is the 'default queue for all 96-col cards', enter

```
CHGOUTQ OUTQ(QPUNCH.QGPL)
 TEXT('Default queue for all 96-col cards')
```

Chapter 11. Message Handling

Messages provide the means of communication between the system and users of the system, as well as within the system itself. When you enter a command, the system may respond with a message indicating the status of your request. Programs processing work on the system also communicate through messages. One program may send a message to another program specifying that an action was taken. The second program may respond with a message and send a message to a third program. Messages regarding significant system events or actions taken on specific jobs are recorded in system logs or job logs, or in both.

As the system operator, you receive messages from system users and system programs that communicate conditions to you and indicate actions you need to take. As a system user, you receive messages in response to your entries at a work station.

MESSAGE CHARACTERISTICS

A message is a communication sent to one system user or program by another. An extensive set of *predefined* messages is supplied with the system. These messages are used for communication between programs within the system or between the system and users of the system such as you. For example, the system response to your command to start the batch subsystem is in the form of a predefined message:

> Start of subsystem QBATCH in library QGPL in progress

In addition, you can communicate with other users of the system through *impromptu* messages that are created at the same time they are sent. An example of an impromptu message you might send is

> SYSTEM GOING DOWN AT 9:00

Message Types

How a message is handled depends on its type as defined to the system. In certain message commands you will use, you must specify the message type. The message types you are likely to use are:

*Informational (*INFO):* Messages that convey information about the condition of the system and help in the use of the system or an application. You receive informational messages from the system or users and send informational messages to users.

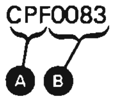
*Inquiry (*INQ):* Messages that convey information but also require a reply. You receive inquiry messages from the system or system users and may occasionally send an inquiry message to a user.

*Reply (*RPY):* Messages that respond to previously received inquiry messages. You send reply messages to senders of inquiry messages.

*Sender's Copy (*COPY):* Messages that are copies of inquiry messages. You may want to see a copy of an inquiry message that you had sent previously.

Message Identifier

In some message displays, you will note that the message is accompanied by a letter and number code such as



This is the message identifier that is used to distinguish the message from others in a message file. The first three letters 🅐 indicate the message category. The following list shows some typical message identifiers:

CPF	Control Program Facility
CPA	Messages that require operator action
CPC	Completion Messages
CPD	Diagnostic Messages
MCH	Machine Interface
RPG	RPG Language
IDU	Interactive Data Base Utilities
CBE	COBOL Language (execution messages)
CBL	COBOL Language (compiler messages)
CSC	COBOL Language (syntax checker)

The remaining four digits 🅑 indicate the sequence number of the message. The example message identifier shown indicates this is CPF message number 83.

Message information is listed in numerical order according to message identifier in the *System/38 Messages Guide* and *System/38 Messages Guide: COBOL*.

Severity Codes

The severity code for a message indicates how serious the message is in terms of system operations. The severity code can be any number from 00 through 99. The higher the number, the greater the effect on system operations.

Not all messages you receive will have a severity code. There will, however, be a severity code for all messages regarding errors related to your input at a work station and for all messages regarding system error conditions.

The following is a list of severity codes and their meanings.

00 (Information): A message of this severity is for information purposes only; no error was detected and no reply is needed. The message could indicate that a function is in progress or it has reached a successful completion.

10 (Warning): A message of this severity indicates a potential error condition. The program may have taken a default, such as supplying missing input. The results of the operation are assumed to be what was intended.

20 (Error—Default Used): This message indicates an error has been detected, but it is one for which automatic recovery procedures probably were applied; processing has continued. A default may have been taken to replace erroneous input. The results of the operation may not be valid. The function may have been only partially completed; for example, some items in a list may have been processed correctly while others may have failed.

30 (Input Value Error): This message indicates the error detected is too severe for automatic recovery, and no defaults are possible. If the error was in source data, the entire input record was skipped. If the error occurred during program execution, it will lead to an abnormal termination of the program (severity 40). The results of the operation are not valid.

40 (Error—Requested Function Terminated): This message indicates the operation has terminated, possibly because it was unable to handle invalid data, or possibly because the user has canceled it.

50 (Abnormal Termination of Job): This message indicates the job was terminated or was not started. A routing step may have terminated abnormally or failed to start, a job-level function may not have been performed as required, or the job may have been canceled.

60 (System Status): This message gives either the status of or a warning about a device, a subsystem, or the system.

70 (Device Integrity): This message indicates that a device is malfunctioning or in some way is no longer operational. The user may be able to recover from the failure or may require the assistance of a service representative.

80 (System Alert): A message of this severity is issued for impromptu informational messages to the system operator. It warns of a condition that, although not severe enough to stop the system now, could become more severe unless preventive measures are taken.

90 (System Integrity): A message of this severity is issued only to the system operator. It describes a condition that renders either a subsystem or the system inoperative.

99 (Action): A message of this severity indicates that some manual action is required, such as entering a reply, changing printer forms, or replacing diskettes.

By specifying a severity code in a command, you can define which types of messages you receive (see *Changing a Message Queue* and *Displaying Messages from a Queue* in this chapter).

Message Text Levels

If an error condition occurs, you receive the *first-level text* of an error message that briefly identifies the error. By positioning the cursor anywhere on the line under the message, and pressing the Help key, you obtain a separate second-level message display that contains the message identifier, the name of the program issuing the message, and the *second-level text*. The second-level text contains additional information on the error or describes what corrective action to take.

If there is no second-level text, refer to the *Problem Determination Guide*.

MESSAGE QUEUES

Messages in response to a direct request, such as a command, are displayed automatically on the display from which you made the request. However, all other messages sent from one location in the system to another are transmitted to message queues. The system user or program must get the message from the queue.

Types of Message Queues

There are several types of message queues in the system. All types provide essentially the same message handling capabilities through a common set of commands.

Work Station Message Queues: Each work station, including the system console, has a system message queue permanently assigned to it. A message sent to a work station message queue is available to anyone signed on at the work station. The message queue has the same name as the work station it is associated with.

The system console message queue is identified by the name QCONSOLE.

User Message Queues: A message queue can be created for individual users of the system. This message queue could be assigned to a single person or a group of people having a common job function.

Each user message queue exists independently of the work station message queue. Messages on a user message queue can be obtained by anyone authorized to use the queue, regardless of which work station that person is using. However, the queue must be specifically allocated (assigned) to the user's interactive job at a work station before it will automatically deliver messages to that work station. You can allocate a user message queue to your job at a work station by changing its delivery mode (see *Changing a Message Queue* in this chapter).

System Log Message Queues: Information to be recorded in the system history log (QHST), service log (QSRV), or programming change log (QCHG) is sent to a special message queue for each of those logs (see *System Logs* in Chapter 8).

Job Message Queues: A job message queue is created for each job on the system to handle all the message requirements of the job.

System Operator Message Queue

The system operator message queue is a special user message queue supplied by IBM. It is identified by the name QSYSOPR and stored in the system library, QSYS. All messages from system programs and application programs in jobs running on the system regarding operator intervention in system operations are sent to this queue. Individual users at work stations who need assistance in, or information about, system operations also send messages to the queue.

When you start the system, QSYSOPR is allocated to your interactive job, that is, the interactive job you created when you signed on the system console (however, note the exeptions discussed later in *Delivery Status at Sign-On and Sign-Off*). QSYSOPR is not tied to the system console. It can be reallocated (reassigned) to another work station. Messages on system operations are still sent to the queue, but they are displayed at the other work station. You can also arrange to share QSYSOPR with one or more others who are also performing system operator tasks. Those procedures are described in *Reallocating and Sharing the System Operator Message Queue*, in this chapter.

Remember that the work station you are using also has a work station message queue associated with it. Therefore, you could receive messages at that message queue as well. For example, if you are using the system console, you could receive some messages at the system operator message queue, QSYSOPR, and other messages at the system console message queue, QCONSOLE. However, all messages regarding system operations will be sent to QSYSOPR unless specified otherwise in a spooling command (see *Reader Operating Considerations* and *Writer Operating Considerations* in Chapter 10).

When a message is sent to QSYSOPR, it is also sent to the message queue for the system history log, QHST, so that it is stored in the log in the same sequence that it was sent. The system history log, therefore, provides a cumulative record of system events in the same sequence that they happened. You can display or print the messages in the system history log by using the Display Log (DSPLOG) command (see *System Logs* in Chapter 8).

The relationship of QSYSOPR to the system and other system users is summarized in Figure 11-1.

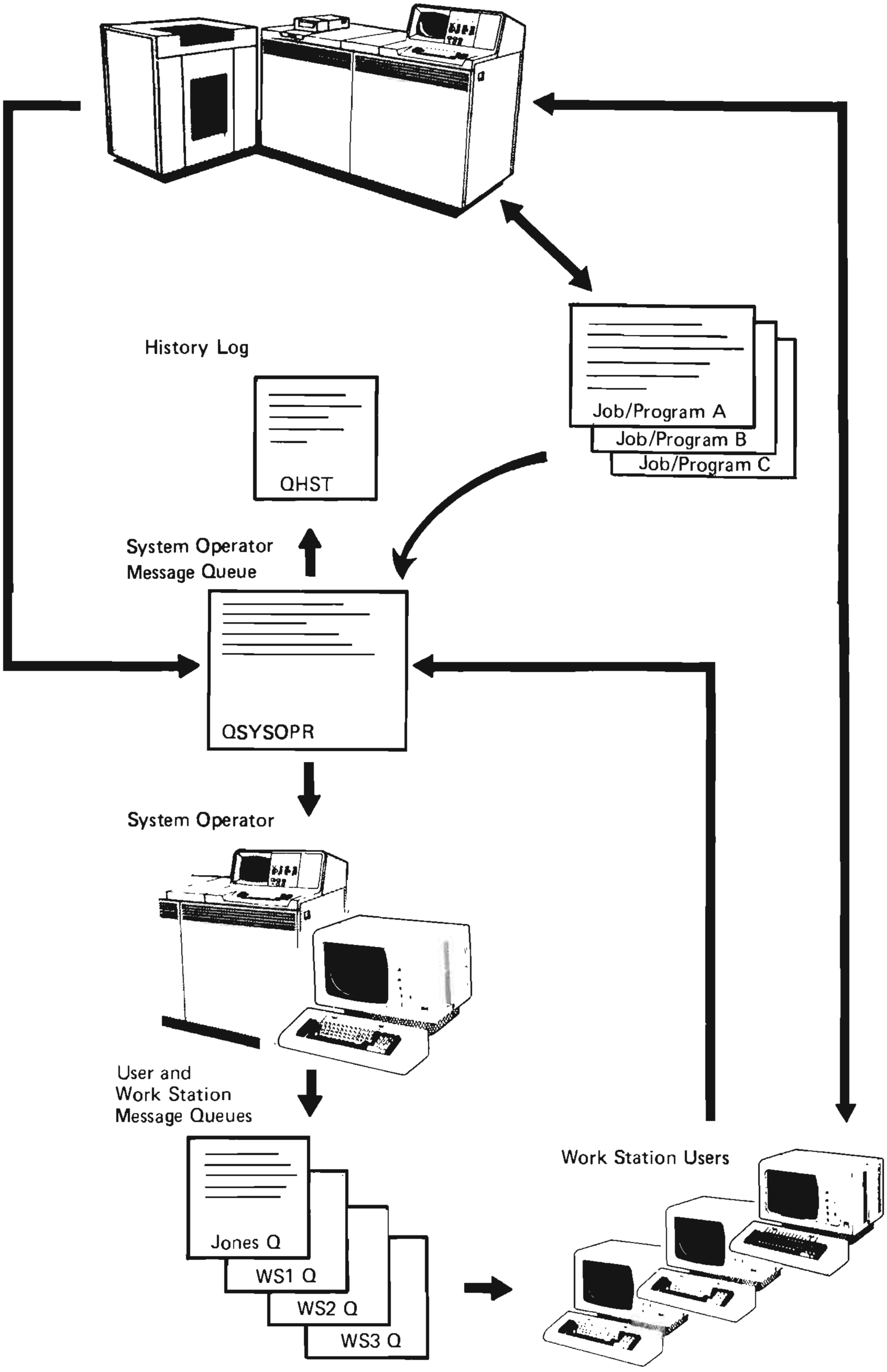


Figure 11-1. The System Operator Message Queue and the System

Message Queue Delivery Modes

A work station or user message queue, including QSYSOPR, can deliver messages to your work station in four possible modes. The modes are:

- *Break Mode*: Your work is interrupted, and a separate message display containing the message appears on your work station screen (unless you have specified a program that delivers the message differently).

  If you are using the system console and you have the alarm feature, the alarm sounds until you reply to the message or otherwise indicate a response (such as by pressing a Roll key).

  If you are using a 5251/5252 work station, the alarm only buzzes briefly.

  When you press the Enter key after viewing the message, the message display is replaced by your previous display. However, if you received an inquiry message, you should reply to it before you return to your previous display. If necessary, you can exit from an inquiry message display by pressing the CF1 key. If another message arrives in the queue before you return to your previous display, the message display will be redisplayed with the new message when you press the Enter key.

- *Notify Mode*: Your work is not interrupted, but you are notified that a message has arrived.

  If you are using the system console, the Attention indicator on the screen comes on (■) to notify you of the message. If you have the alarm feature, the Attention light on the operator/service panel also comes on.

  If you are using a 5251/5252 work station, the Message Waiting indicator on the screen comes on (■) and the alarm buzzes briefly to notify you of the message.

  To obtain the message, you use the Display Messages command (see *Displaying Messages from a Queue* in this chapter). The screen indicator and Attention light, if any, are turned off when you display the messages. When you press the Enter key after viewing the message display, the display from which you requested the messages is returned to your screen.

- *Hold Mode*: The message queue holds the messages until you ask for them by using the Display Messages command. You are not notified when a new message arrives.

- *Default Delivery*: All messages are ignored, and any messages requiring a reply are sent the default reply for the message. For QSYSOPR, the messages are retained in the queue and logged in the system history log, QHST. For any other user or work station message queue, none of the messages are retained in the queue.

For the break and notify modes, you can filter messages by specifying which kinds are to be delivered. You do this by specifying a minimum severity code. Only messages with a severity equal to or greater than the minimum severity code are delivered. Thus, you can set your message queue so that you are notified of messages only above a particular severity level.

By changing the delivery mode of a user message queue, such as the system operator message queue, you can allocate it to or deallocate it from an interactive job at a work station. When a user message queue is set to the break or notify mode, it is allocated exclusively to the interactive job that set it. That is, if you set a message queue to the break or notify mode, it will deliver messages only to your work station until you sign off. When a user message queue is set to the hold mode, it is effectively deallocated. This means that anyone with authority to use the queue can display messages from it.

By selecting the default delivery mode, you can set the system for unattended operation.

Note that setting the delivery mode of your message queue does not affect messages in response to commands you enter. Those messages are displayed on your screen automatically, regardless of how your message queue is set.

The current delivery mode and message queue severity are displayed on the heading line of the message queue display.

Delivery Status at Sign-On and Sign-Off

The message delivery mode of work station message queues, the system operator message queue, and other user message queues is generally reset when you or another user signs on or off a work station. All work stations are automatically signed off when the system is powered down.

System Operator Message Queue: When the system is being prepared for operation, the system operator message queue, QSYSOPR, is reset to the break mode, and any messages in the queue with severity greater than or equal to the severity code which was most recently specified for the queue are displayed automatically on the system console screen. The queue is allocated exclusively to the interactive job of the person who signed on at the system console.

There are situations, however, where the queue is not reset when the system is started. These involve either a restart by the auto-IMPL process or a startup that bypasses the system console (see *Variations of the Startup Procedure* in Chapter 8). Because defaults are used for the start CPF process and you do not sign on until the process is completed, QSYSOPR is handled as any other user message queue. This means that the queue will not be allocated to your work station; it will be set to the default mode. Therefore, if your system is restarted by an auto-IMPL or the system console is bypassed during startup, you should change QSYSOPR to the break mode so that you receive messages regarding system conditions (see *Changing a Message Queue* in this chapter).

When you sign off or power down the system, the queue reverts to the hold mode unless you had set it to the default mode, in which case it remains in default mode.

Note that if you sign off and then sign on again while the system is active, the queue will have been reset, and you will have to set it to the desired delivery mode after you sign on.

User Message Queues: User message queues are not affected by sign-on because they are not automatically assigned to a specific work station. They become assigned to a work station when they are specifically allocated to a user's interactive job at the work station.

When the user to which the message queue was allocated signs off, the user message queue reverts to the hold mode unless it was set to the default mode, in which case it remains in that mode.

Work Station Message Queues: When a user signs on a work station, including the system console, the work station message queue is reset to the notify mode, and the Attention or Message Waiting screen indicator comes on as soon as there are any messages to be displayed.

At sign-off, the work station message queue reverts to the hold mode.

Notification of Message Sent to QSYSOPR

When an operator action is required for a particular system function or user job, a message is sent to the system operator message queue, QSYSOPR. The affected job or system function will wait until you take the necessary action and respond to the message (see, for example, *Operator/Device Interaction* in Chapter 9).

You will be notified when a message is sent to QSYSOPR if the queue is set to the break or notify mode. If QSYSOPR is in the break mode, a message sent to the queue with a severity greater than or equal to the severity code specified for the queue will be displayed automatically. If QSYSOPR is set to the notify mode, the arrival of a message of great enough severity will cause the work station to signal you (for details, see the previous discussion of *Message Queue Delivery Modes*). You display the messages by entering the Display Messages command (or by pressing the CF6 key if you are using the system operator menu).

When QSYSOPR is set to the hold or default mode, you are not notified when a message is sent to the queue. If you set QSYSOPR to the hold mode, you should display the messages in the queue periodically and take any action necessary, so that system operations continue smoothly. If you set QSYSOPR to the default mode, the system supplies default replies for all inquiry messages sent to the queue. The default reply for some messages is an instruction to cancel the job. Because you do not receive message that require an operator response, the affected operation is suspended while the system waits for a reply. The system resources being used for the operation are unavailable for other uses. QSYSOPR should be set to the default mode, therefore, only when system functions that normally send messages to the queue (such as batch jobs and spooling readers and writers) are not active. QSYSOPR can be set to the default mode without adverse effect when only interactive jobs that do not normally send messages to the queue are being processed.

Situations could occur that cause QSYSOPR to be set to the hold mode implicitly; that is, without you changing it intentionally. Examples are:

- QSYSOPR is no longer allocated to your interactive job because
  - You signed off and did not change the queue back to the break or notify mode when you signed on again.
  - Your interactive job was abnormally terminated.

- The interactive job to which QSYSOPR was allocated is currently suspended though the system request function (see *Using the System Request Menu* in Chapter 6). While a job is suspended, a message queue allocated to the job can neither signal nor deliver messages to the work station at which the job was active.

Because system operations could be adversely affected when you do not perform some required operator action, the system checks for conditions where QSYSOPR is not set to the hold or default mode intentionally. If such a condition is detected, the system sends another message to a *work station message queue* to notify you that a message requiring operator action was sent to QSYSOPR. The system value QSYSOPRDEV defines the work station message queue that is to receive the second message. When the system is shipped from IBM, the work station message queue specified in QSYSOPRDEV is QCONSOLE. If you normally use a 5251 or 5252 work station to control system operations, you can specify the work station message queue for that work station in QSYSOPRDEV, instead of QCONSOLE (see *Changing System Values* in Chapter 8). The message queue specified in QSYSOPRDEV should not be set to the hold or default mode because you would not be notified when a message arrived.

When you sign on a work station, the associated work station message queue is set to the notify mode. If you do not change the delivery mode of the message queue, therefore, the work station will signal you each time a message is sent to the message queue. For the system console, this means that the Attention indicator on the side of the screen will come on; the Attention light on the operator/service panel will also come on if you have the feature. For a 5251 or 5252 work station, the Message Waiting indicator will come on and the alarm will buzz briefly.

The message that is sent to the work station message queue specified in QSYSOPRDEV will inform you that a message requiring an operator action was sent to QSYSOPR. The procedure you use to display the message in QSYSOPR depends on the reason the queue was implicitly set to the hold mode. If QSYSOPR was deallocated because your interactive job terminated, you can display the messages by using the Display Messages (DSPMSG) command, provided your current job has the necessary authority. If QSYSOPR is still allocated to a job that is suspended, you must resume the suspended job to display the message. Usually, the job will be suspended because you used option 1 of the system request menu to transfer to another job. The suspended job will be resumed when you use the system request menu to transfer back to the job. As soon as the job is resumed, the message in QSYSOPR is delivered to your work station in the manner that is appropriate for the delivery mode of the queue. If QSYSOPR is in the break mode, the message will be displayed automatically as soon as your interactive job is resumed. If the queue is in the notify mode, you will be signaled and you display the message by entering the Display Messages (DSPMSG) command.

Changing a Message Queue

To change the message delivery mode or another characteristic of a message queue, you use the Change Message Queue (CHGMSGQ) command. When you enter the command, you must include the MSGQ parameter to identify the message queue. You then include any of the other parameters as needed to change one or more particular characteristics. If you do not enter a parameter, the characteristic determined by that parameter remains the same.

MSGQ Parameter: You must enter this parameter to specify the name of the message queue to be changed, such as

```
MSGQ(QSYSOPR)
```

DLVRY Parameter: Use this parameter to specify the delivery mode. Note that this change is effective only for your current interactive job. When you end your interactive job, by signing off or powering down the system, your message queue reverts to the delivery mode described previously in *Delivery Status at Sign-On and Sign-Off.*

- If you do not enter the parameter, DLVRY(*SAME) is assumed and the delivery mode remains the same as before.

- If you enter the parameter, you can specify *HOLD, *BREAK, *NOTIFY, or *DFT (default); for example, DLVRY(*NOTIFY).

PGM Parameter: If you specified the break delivery mode (*BREAK) in the DLVRY parameter, you can use this parameter to determine the way in which you obtain the message.

- If you do not enter the PGM parameter, PGM(*SAME) is assumed and there is no change; however, if a means of delivery was not specified previously, *DSPMSG is assumed.

- If you specify PGM(*DSPMSG) each message is displayed automatically when it arrives in the queue.

- If you specify a program, such as PGM(BRKDLVY) that program is invoked to deliver break messages. The program determines what you do to obtain the messages.

SEV Parameter: This parameter defines a severity code for delivery of messages provided the message queue is set to the break or notify delivery mode.

- If you do not enter the parameter, SEV(*SAME) is assumed and the severity code, if any, remains the same.

- If you enter the parameter and specify a severity code, such as SEV(30) messages will be delivered in the break or notify mode only when a message appears in the queue with a severity equal to or greater than that specified.

The following chart shows the kinds of messages that will be displayed at the various severity levels. An X in the column indicates the message will be displayed in break delivery mode.

	Severity							
	00	10	40	50	60	70	80	90
Messages that Require a Reply	X	X	X	X	X	X	X	X
System Alert and Impromptu Messages	X	X	X	X	X	X	X	
Device Problems	X	X	X	X	X	X		
Subsystem Problems	X	X	X	X	X			
Job Abnormal Termination	X	X	X	X				
Power Off Device	X	X	X					
Warnings	X	X						
Informational	X							

To cut down the number of break messages, a severity level of 50 or 60 can be used.

TEXT Parameter: This parameter determines whether any text is to be included in the message queue description.

- If you do not enter the parameter, TEXT(*SAME) is assumed and the text, if any, remains the same.

- If you specify TEXT(*BLANK) any existing text is deleted.

- If you include specific text, such as TEXT('System Operator Message Queue') that text replaces any existing text.

Do not enter the RESET or FORCE parameters. They are used only for programs that process messages.

Examples:

1. To change the work station message queue for the system console so that messages are held in the queue but you are not notified when they arrive, enter

```
CHGMSGQ MSGQ(QCONSOLE) DLVRY(*HOLD)
```

2. To change the system operator message queue so that you are signaled at the arrival of any message having a severity of 10 or greater, enter

```
CHGMSGQ MSGQ(QSYSOPR) DLVRY(*NOTIFY) SEV(10)
```

3. To change the system operator message queue so that any message with a severity of 80 or greater is displayed automatically, enter

```
CHGMSGQ MSGQ(QSYSOPR) DLVRY(*BREAK) PGM(*DSPMSG)
  SEV(80)
```

Setting QSYSOPR for Unattended Operation

Using the CHGMSGQ command just discussed, you can set the system for unattended operation. You do this by specifying the default message delivery mode for the system operator message queue; for example, by entering

```
CHGMSGQ MSGQ(QSYSOPR) DLVRY(*DFT)
```

When the QSYSOPR is set in the default mode, the CPF programming supplies default replies for all messages sent to the message queue. Should an error condition occur, therefore, the default reply could result in a job being terminated (see the previous discussion, *Notification of Message Sent to QSYSOPR*, in this chapter). To minimize this possibility, you should take certain precautions before putting the QSYSOPR in the default mode:

- Display all messages to make sure the system is operating properly.
- Make sure that all input/output devices are supplied with necessary materials (paper, cards, diskettes).
- Make sure that all subsystems likely to be needed are started.

Even with these precautions, unattended operation is practical only for a short period if input/output devices are involved. A printer will eventually run out of forms. The MFCU will eventually run out of cards. Diskettes will eventually need to be loaded or replaced in the diskette magazine drive.

If you suspect that an error condition has occurred during unattended operation, you should check the system operator message queue for error messages.

Reallocating and Sharing QSYSOPR

If you move from the system console to another work station, messages for the system operator message queue, QSYSOPR, will not be delivered to your new work station unless you take some action to reallocate (reassign) the queue. There may also be situations where you will want to share the queue with other persons who are also performing job control functions. You can reassign or share QSYSOPR by using the CHGMSGQ command.

Reassigning the Queue

There are various ways you can reassign the system operator message queue. The following procedure is one of the simplest:

1. Sign off the work station you are currently using.

2. Sign on the work station you are going to use.

3. At the new work station, enter the CHGMSGQ command, using the DLVRY parameter to specify the break or notify message delivery mode for the system operator message queue (QSYSOPR).

When you sign off your current work station, QSYSOPR is deallocated. You could also deallocate the queue by changing its delivery mode to the hold mode without signing off. When you change QSYSOPR to the break or notify mode, it is allocated exclusively to your interactive job at the new work station and will deliver messages only to that work station.

Example:

You have been operating at the system console with the system operator message queue in the break mode. Now you want to transfer operations to a 5251 work station. Therefore, you do the following:

1. Sign off the system console.

2. Move to the 5251 and sign on.

3. Reallocate the system operator message queue by entering

```
CHGMSGQ MSGQ(QSYSOPR) DLVRY(*BREAK)
```

Once the CHGMSGQ command is executed, messages will be delivered automatically from QSYSOPR to the 5251 work station instead of the system console.

Sharing the Queue

You can make QSYSOPR available to other users performing system operator tasks by changing the queue to the hold delivery mode. When you change it to the hold mode by entering

```
CHGMSGQ MSGQ(QSYSOPR) DLVRY(*HOLD)
```

it is deallocated from your interactive job. This means that other authorized users can display the message in the queue by entering the Display Messages command (see *Displaying Messages from a Queue* in this chapter).

Note, however, that neither you nor any other user will be signaled when a message arrives regarding a condition that requires an operator response (see the previous discussion, *Notification of Message Sent to QSYSOPR*, in this chapter).

Managing Messages in a Queue

Messages sent to the system operator message queue will remain in the queue unless you remove them. The same is true of other message queues unless they are set to the default mode. As messages accumulate in a message queue, they occupy increasing amounts of system storage that could be used for other purposes. Therefore, you should remove a message from a message queue as soon as you have displayed it and have taken any necessary actions. While you are displaying the messages in a message queue, you can use the CF6 key to remove an individual message and the CF8 key to remove all messages from the queue (see *Displaying Messages from a Queue* in this chapter).

If you take over system operations from another operator, you should immediately display the messages in QSYSOPR to verify that there are no messages that require an action by you. After you have taken any actions needed, you should remove all the messages from the queue.

Messages cannot be displayed from a message queue after they have been removed. Note, however, that all messages sent to QSYSOPR are also recorded in the system history log. You can review messages that were previously in QSYSOPR by displaying the history log (see *System Logs* in Chapter 8).

Handling a Damaged Message Queue

When a message queue becomes damaged, messages cannot be sent to it or displayed from it. If you attempt to display the messages in a damaged queue, or send a message to it, you will receive an error message that indicates the queue is damaged.

A damaged message queue must be deleted and recreated. The procedure you use depends on the type of message queue.

The system operator message queue, QSYSOPR, can only be deleted and recreated by powering down the system and restarting it. You can use the following procedure to recover from damage to QSYSOPR:

1. If possible, warn all work station users that the system must be powered down temporarily and that they should sign off. You can do this by sending a break message to all work station users (see *Sending Break Messages* in this chapter).

2. Power down the system by entering the PWRDWNSYS command (see *Powering Down the System* in Chapter 8).

3. Set the rotary switches on the operator/service panel for an IMPL Abbr and press the Power On switch to restart the system.

4. During the start CPF process, the damaged message queue is automatically deleted and recreated. You can then resume normal operations.

If the message queue for any of the system logs (QHST, QSRV, and QCHG) is damaged, that message queue is also automatically deleted and recreated during the start CPF process. However, you do not have to power down and restart the system if the QHST, QSRV, or QCHG message queue becomes damaged during system operations. Contact the system security officer and ask him to enter the commands that delete and recreate the damaged system log message queue.

If a user message queue (other then QSYSOPR) is damaged, it must be deleted and recreated by commands. Contact the owner of the queue, or another appropriate authority (such as the system security officer), and ask him to enter the commands that delete and recreate the queue.

If a work station message queue is damaged, you should vary the work station offline and then online again (see *Varying Devices, Control Units, and Lines Online and Offline* in Chapter 9). When the work station is varied offline and online, the damaged work station message queue is automatically deleted and recreated.

The messages in a damaged message queue are lost. If the damaged queue was QSYSOPR, however, all messsages that were sent to the queue prior to the time of the damage were also recorded in the system history log, QHST. You can obtain those messages by displaying the history log. If the message queue for a system log, such as QHST, is damaged, only the messages in the current version of the log are lost.

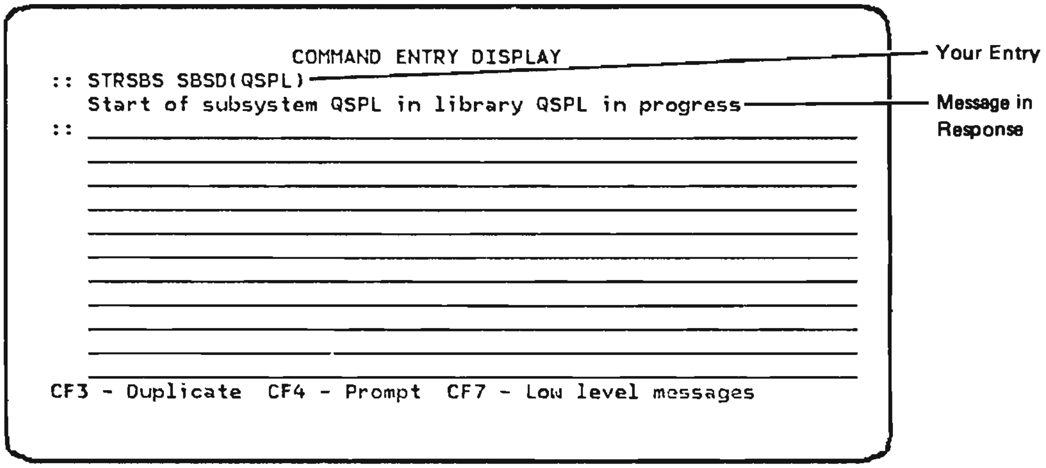
RECEIVING AND DISPLAYING MESSAGES

As a user of the system, you receive system completion or diagnostic messages in response to your requests. As the system operator, you receive messages, through the system operator message queue, from the system, jobs being processed by the system, and individual users at work stations.

The way in which messages are displayed depends on which display you are currently using and, for messages sent to your message queue, the delivery mode of your queue.

Response Messages

Messages in response to your entered requests are automatically displayed on the current display you are using. If you are using the command entry display, the message appears on the next available line.

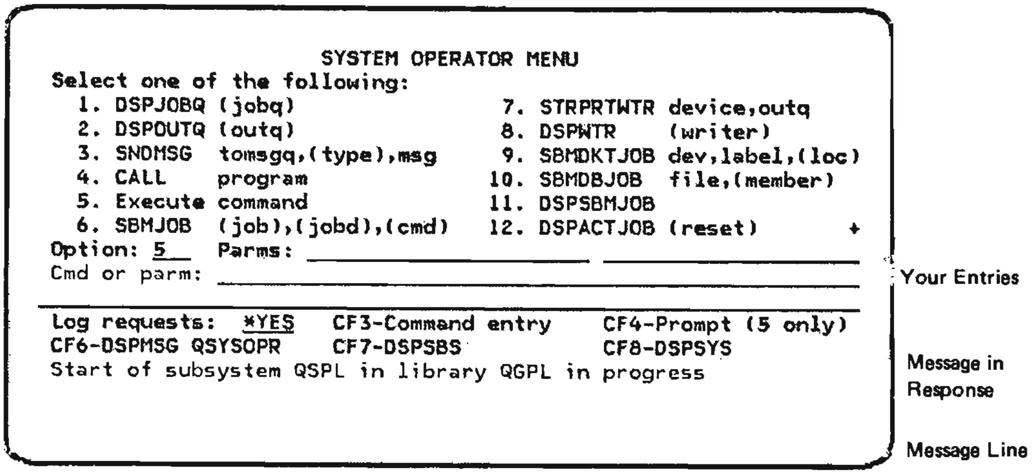


```
                    COMMAND ENTRY DISPLAY
:: STRSBS SBSD(QSPL)
   Start of subsystem QSPL in library QSPL in progress
:: 











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

Your Entry

Message in Response

For any other kind of display, the message appears at the bottom of the screen.



```
                       SYSTEM OPERATOR MENU
Select one of the following:
  1. DSPJOBQ (jobq)               7. STRPRTWTR device,outq
  2. DSPOUTQ (outq)               8. DSPWTR    (writer)
  3. SNDMSG  tomsgq,(type),msg    9. SBMDKTJOB dev,label,(loc)
  4. CALL    program             10. SBMDBJOB  file,(member)
  5. Execute command             11. DSPSBMJOB
  6. SBMJOB  (job),(jobd),(cmd)  12. DSPACTJOB (reset)       +
Option: 5__  Parms: ______________________ ____________________
Cmd or parm: _________________________________________________
________________________________________________________________
Log requests: *YES   CF3-Command entry     CF4-Prompt (5 only)
CF6-DSPMSG QSYSOPR   CF7-DSPSBS            CF8-DSPSYS
Start of subsystem QSPL in library QGPL in progress
```

If you need additional information about a message, you can request a second-level message display. This display is particularly useful when you need to respond to an error message (see *Handling Error Messages* in this chapter). To request the second-level message display, position the cursor on the same line as the message and press the Help key.

Specific types of messages will appear on the bottom line of the screen (message line). If you press the Help key while there is a message on the message line, and input is inhibited, you will receive the second-level text for the message on the message line regardless of where the cursor is positioned.

Sometimes when a command fails, the information on the message line may not be sufficient to correct the problem. It may be necessary to access the lower level messages that have been sent to the log by the command-processing programs. The following procedure can be used:

1. From the system operator menu press the CF3 key to obtain the command entry display.

2. Press the CF7 key. This will display the lower level messages for the command that failed.

3. Press the Roll Down key. The messages will appear after the commands in chronological order. Both high level and low-level messages will be displayed.

Displaying Messages from a Queue

Unless a message queue is in the break mode, a message sent to the queue is not automatically displayed. You must take some action to obtain the message. Your current display is then replaced by a separate message display.

Obtaining the Message Display

If your message queue is in the *break* mode:

- You receive the message automatically if you specified PGM(*DSPMSG) in the CHGMSGQ command. Note that this is the way you obtain messages from the system operator message queue if you have not changed its delivery mode since starting the system.

- You receive the message when you respond as indicated to an invoked program if you changed your message queue to specify such a program to handle break delivery. The program might, for example, show a statement on your current display that indicates a message has arrived and instructs you what key to press to obtain the message.

If your message queue is in the *notify* or *hold* mode, you obtain the display by entering the Display Messages (DSPMSG) command. Pressing the CF6 key from the system operator menu is the same as entering the command in the form

```
DSPMSG MSGQ(QSYSOPR)
```

MSGQ Parameter: This parameter determines the message queue from which the messages are to be displayed.

- If you do not enter this parameter, MSGQ(*WRKSTN) is assumed and the messages displayed are from your work station message queue. For example, if you were using the system console, the messages displayed would be from the system console message queue, *not* the system operator message queue.

- To display messages from the system operator message queue, you must enter the parameter and specify either MSGQ(QSYSOPR.QSYS) or simply MSGQ(QSYSOPR).

  You can also use this parameter to specify the name of another user message queue, but you can display messages only from your own *work station* message queue.

MSGTYPE Parameter: This parameter determines whether all types of messages or only a specific type of message is displayed.

- If you do not enter the parameter, MSGTYPE(*ALL) is assumed and all types of messages in the queue are displayed.

- If you specify MSGTYPE(*INFO) only informational messages are displayed.

- If you specify MSGTYPE(*INQ) only inquiry messages and the replies to them are displayed.

- If you specify MSGTYPE(*COPY) only your copies of inquiry messages you sent to other work stations and any replies to the message are displayed. By displaying these copies, you can determine whether you have received replies to all of the inquiry messages you sent.

START Parameter: This parameter determines the order in which the messages are to be shown.

- If you do not enter the parameter, START(*LAST) is assumed and the last message to arrive is shown on the bottom display line, the next to the last message is on the second line from the bottom, and so on, thus displaying the messages in the sequence from newest to oldest. To view older messages not shown on the screen, press the Roll Down key.

- If you specify START(*FIRST) the first message that arrived on the queue is shown on the top display line, the next message to arrive is shown on the second line, and so on, thus displaying the messages in the sequence from oldest to newest. To view newer messages not currently shown on the screen, press the Roll Up key.

SEV Parameter: This parameter determines whether only messages greater than a certain severity level are displayed (see *Severity Codes*).

- If you do not enter the parameter, SEV(00) is assumed and messages of all severity levels are displayed.

- If you specify SEV(*MSGQ) only messages having a severity equal to or greater than the level specified for the queue are displayed (see *Changing a Message Queue*).

- If you specify a severity code, such as SEV(30) only messages having a severity equal to or greater than that level are displayed. This overrides any severity level previously established from the queue.

OUTPUT Parameter: This parameter determines how the messages are provided.

- If you do not enter the parameter, the messages are displayed on your work station screen.

- If you specify OUTPUT(*LIST) the messages are printed in the sequence from oldest to newest.

Examples:

1. To display on the system console screen all types of messages that are on the system console message queue, enter on the system console

```
DSPMSG
```

2. To print out all types of messages that are on the system operator message queue, enter

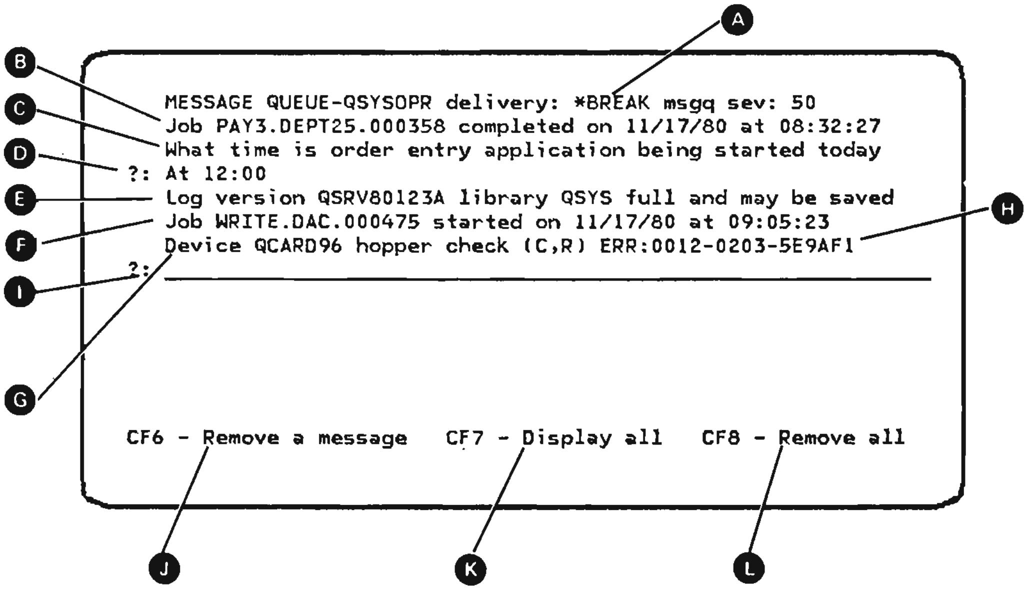
```
DSPMSG MSGQ(QSYSOPR) OUTPUT(*LIST)
```

3. To display at your work station only inquiry messages (those requiring a reply) on the system operator message queue that have a severity code equal to or greater than 20, enter

```
DSPMSG MSGQ(QSYSOPR) MSGTYPE(*INQ) SEV(20)
```

Interpreting the Display

The messages are displayed in a form such as:



```
        MESSAGE QUEUE-QSYSOPR delivery: *BREAK msgq sev: 50
        Job PAY3.DEPT25.000358 completed on 11/17/80 at 08:32:27
        What time is order entry application being started today
     ?: At 12:00
        Log version QSRV80123A library QSYS full and may be saved
        Job WRITE.DAC.000475 started on 11/17/80 at 09:05:23
        Device QCARD96 hopper check (C,R) ERR:0012-0203-5E9AF1
     ?: ____________________________________________________________



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

A This is the name, delivery mode, and severity level of the message queue from which the messages are displayed.

B This is an informational message from the system regarding the status of a job.

C This is an inquiry message from a system user at another work station.

D This is your reply to the inquiry message **C**. If you had not already replied to the message, there would be an underlined input field here in which you enter your reply.

E This message informs you that the current version of the system service log is full. If you are to retain the information contained in the log version, you should save it on a diskette or tape (see *Saving Information* in Chapter 13).

F This is another message on the status of a job.

G This is an inquiry message about an error condition on a system device that you need to correct.

H Each device error message is accompanied by a string of characters such as shown here. These characters are actually part of the error message and provide information for service personnel.

Ⓘ You reply to the error message Ⓖ by entering R or C in this input field. Entering R means that you have corrected the condition and the operation should be retried. Entering C means that the operation should be canceled. You can exit an inquiry display without answering by pressing the CF1 key.

Ⓙ If you position the cursor on the line containing a particular message and press the CF6 key, that message is removed from the queue.

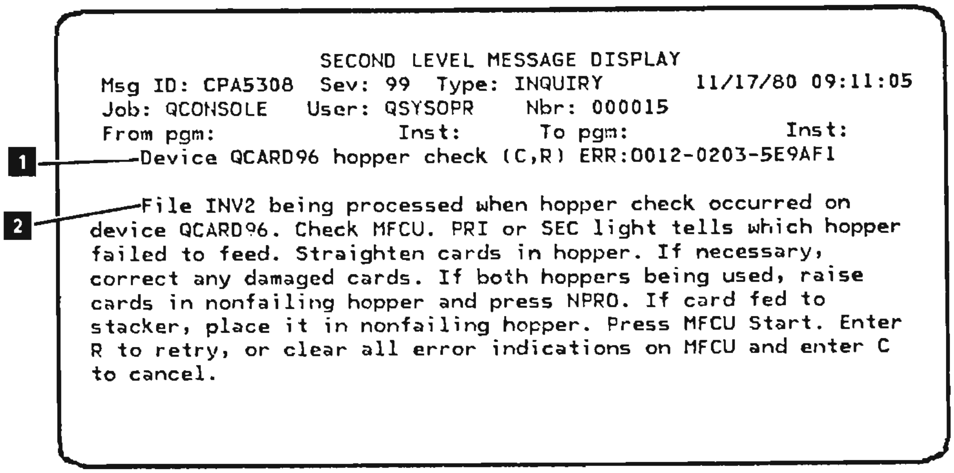
Ⓚ If you did not request to see all messages on the queue, you can display all messages by pressing the CF7 key.

Ⓛ You can remove all the messages from the queue by pressing the CF8 key.

If there are more messages in the queue than can be displayed at one time, a + appears on the lower right side of the screen. You obtain the additional message by pressing the Roll Up key.

Note that if the message queue is in the break mode, the display will be adjusted so that the message causing the current break in your interactive job is shown as the first message on the display. Any messages that arrive in the queue after that first message, will be displayed below it. Usually, you will see only one message at a time. You can display previous messages by pressing the Roll Down key. If the message that caused break delivery is an inquiry message such as Ⓒ or Ⓖ, you cannot return to your previous display until you reply to the message. This is to ensure that an operation is not held up for an extended period because a required action was not taken.

For some types of messages, only one line of the first-level text is shown on the initial message display. If you position the cursor on the line containing a particular message on the initial display and press the Help key, you receive a separate second-level display for that message, such as



```
                         SECOND LEVEL MESSAGE DISPLAY
 Msg ID: CPA5308  Sev: 99  Type: INQUIRY        11/17/80 09:11:05
 Job: QCONSOLE   User: QSYSOPR    Nbr: 000015
 From pgm:               Inst:      To pgm:             Inst:
    Device QCARD96 hopper check (C,R) ERR:0012-0203-5E9AF1

    File INV2 being processed when hopper check occurred on
device QCARD96. Check MFCU. PRI or SEC light tells which hopper
failed to feed. Straighten cards in hopper. If necessary,
correct any damaged cards. If both hoppers being used, raise
cards in nonfailing hopper and press NPRO. If card fed to
stacker, place it in nonfailing hopper. Press MFCU Start. Enter
R to retry, or clear all error indications on MFCU and enter C
to cancel.
```

This second-level display provides all available information about the message, including the first-level text 1 and any second-level text 2. Not all messages will have second-level text. However, the complete first-level text will always be shown, including any part that was too long to fit on a single line of the original message display. The second-level text of an error message, as in the example display shown here, describes how to correct the error (see *Handling Error Messages* in this chapter).

This second-level message display also indicates where the message originated. If the message was sent from the system, as in the example shown here, the *Job*, *User*, and *Nbr* fields on the display will indicate your own interactive job as follows:

- *Job*: the name of the work station you are using.
- *User*: the user profile associated with the keyword you entered when you signed on.
- *Nbr*: the job number assigned to your interactive job.

However, if the message was sent by a person at another work station, the same three fields will indicate the following:

- *Job*: the name of the work station from which the message was sent.
- *User*: the user profile of the person using the work station.
- *Nbr*: the job number assigned to the interactive job at the work station.

Pressing the Enter key from this display returns you to the original message display. Pressing the Enter or CF1 key from the original message display returns you to the display from which you requested the messages (such as the system operator menu or the command entry display).

SENDING MESSAGES

On occasion you will want to send impromptu messages to users at work stations to inform them of system conditions or provide general system information. There are two commands you can use to send messages. The primary difference between the two is how the message is delivered.

Sending Nonurgent Messages

Using the Send Message (SNDMSG) command, you can send an impromptu message from your work station to one or more message queues, including both work station and individual user message queues. You can also request a reply from the receiver of your message. However, the delivery of a message sent by this command depends on what delivery mode the receiver's message queue is set to. You should use this command to send only general information type messages not requiring timely delivery. If you need to ensure that your message will be delivered immediately, you should use the Send Break Message (SNDBRKMSG) command described later. When you select option 6 on the system operator menu, you are using the SNDMSG command.

When entering the SNDMSG command, you must enter the MSG and TOMSGQ parameters. There are two additional parameters you can use as needed.

MSG Parameter: Use this parameter to specify the message to be sent. The actual message must be enclosed in apostrophes (also called single quotes); for example,

```
MSG('New forms 1010 available')
```

TOMSGQ Parameter: Use this parameter to specify where the message is to go. It can be sent to a work station message queue such as

```
TOMSGQ(WS3)
```

or to a user message queue, such as

```
TOMSGQ(JONESQ)
```

You can specify up to 50 message queues in this parameter if you are sending an information message; but you can specify only one message queue if you are sending an inquiry message.

MSGTYPE Parameter: Use this parameter when you are sending an inquiry message that requires a reply.

- If you do not enter the parameter, MSGTYPE(*INFO) is assumed and your message is assumed to be informational and the receiver's message display is not formatted for a reply.

- If you specify MSGTYPE(*INQ) the receiver's message display is formatted with an input field for a reply on the line below the message, such as

  Are you using the INVENTORY subsystem?

  ?: ____

  Note that just putting a question mark at the end of the message does not tell the system it is an inquiry message. You must use this parameter. An inquiry message can be sent to only one message queue at a time.

You can exit from an inquiry message screen by answering the inquiry or pressing the CF1 key. The message remains on the message queue and no defaults are taken.

RPYMSGQ Parameter: If you specified that the message is an inquiry, that is,

```
MSGTYPE(*INQ)
```

use this parameter to specify where the reply is to go.

- If you do not enter this parameter, RPYMSGQ(*WRKSTN) is assumed and any reply is sent to the message queue of the work station you are using, such as QCONSOLE if you are using the system console.
- Generally, you will want to have the reply sent to the system operator message queue, so you should specify RPYMSGQ(QSYSOPR). You could also specify any other work station or user message queue.

Examples:

1. To send the message

   Sales history file now on line

   to work station WS1, enter

   ```
   SNDMSG MSG('Sales history file now on
    line') TOMSGQ(WS1)
   ```

2. To send the same message to work stations WS2 and WS3, enter

   ```
   SNDMSG MSG('Sales history file now on
    line') TOMSGQ(WS2 WS3)
   ```

3. To send the message

   Batch job ready. Hold for pickup?

   to the message queue identified as SMITH and require a reply to the system operator message queue, enter

   ```
   SNDMSG MSG('Batch Job Ready. Hold for pickup?')
    TOMSGQ(SMITH) MSGTYPE(*INQ) RPYMSGQ(QSYSOPR)
   ```

Sending Break Messages

Using the Send Break Message (SNDBRKMSG) command, you can send an impromptu message from your work station to one or more work stations and force it to be delivered in the break mode regardless of what delivery mode the receiver's message queue is set to. Unlike the SNDMSG command, this command can be used to send a message only to work station message queues, not individual user message queues. Like the SNDMSG command, this command can be used to send a message that requires a reply.

You should use the SNDBRKMSG command when sending any message that requires the immediate attention of a work station user, such as a message about a current or impending system condition.

When entering the SNDBRKMSG command, you must enter the MSG parameter. The remaining three parameters are optional. Note the similarity to the parameters of the SNDMSG command.

MSG Parameter: In this parameter you specify the message to be sent. The message must be enclosed in apostrophes.

TOMSGQ Parameter: Use this parameter when you need to specify what message queue or queues the message is to be sent to.

- If you do not enter the parameter, TOMSGQ(*ALLWS) is assumed and the message is sent to all work station message queues. This is known as a *broadcast* message.

- If you specify one or more message queue names (up to a maximum of 50), the message is sent to the specified queue or queues. Note that you can specify only work station message queues.

MSGTYPE Parameter: Use this parameter as needed to specify an inquiry message that requires a reply.

- If you do not enter the parameter, MSGTYPE(*INFO) is assumed and the message is assumed to be informational only.

- If you specify MSGTYPE(*INQ) the receiver's message display is formatted for a reply. Note that a message requiring a reply can be sent to only one message queue at a time.

RPYMSGQ Parameter: If you specified MSGTYPE(*INQ), the RPYMSGQ parameter determines where the reply is sent.

- If you *do not* enter this parameter, RPYMSGQ(QSYSOPR.*LIBL) is assumed and the reply is sent to the system operator message queue.

- If you want the reply to go to a queue other than the system operator message queue, enter this parameter and specify the name of the queue, such as RPYMSGQ(ALTSYSOPR).

Examples:

1. To send the message

   System printer not available until 2:00.
   Special forms in use

   in the break mode to all work stations, enter

   ```
   SNDBRKMSG  MSG('System printer not available until 2:00.
    Special forms in use')
   ```

2. To send the message

   Are you finished with DATA2 subsystem for today?

   in the break mode to work station WS7 and require a reply, enter

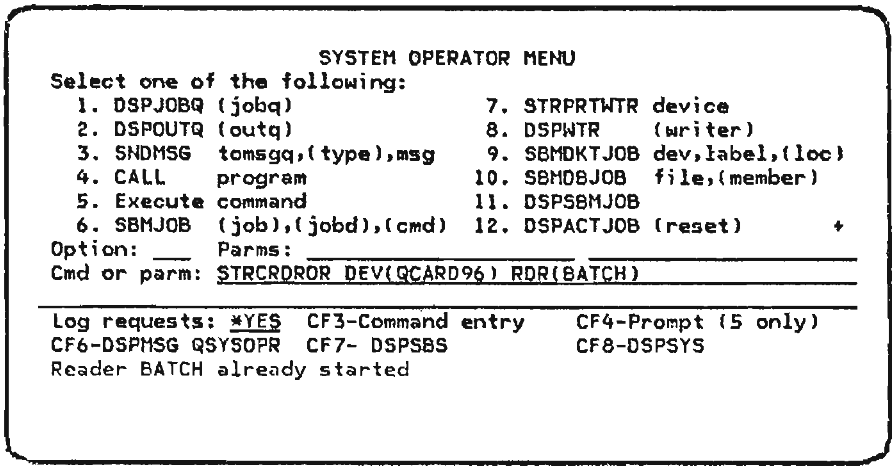
   ```
   SNDBRKMSG  MSG('Are you finished with DATA2
    subsystem for today?') TOMSGQ(WS7)  MSGTYPE(*INQ)
   ```

HANDLING ERROR MESSAGES

You will receive a variety of messages that indicate conditions ranging from simple input errors to problems with system devices or programs. The message could be a response message on your current display or a message regarding a system problem that is sent to the system operator message queue and displayed on the separate message display.

Response Messages

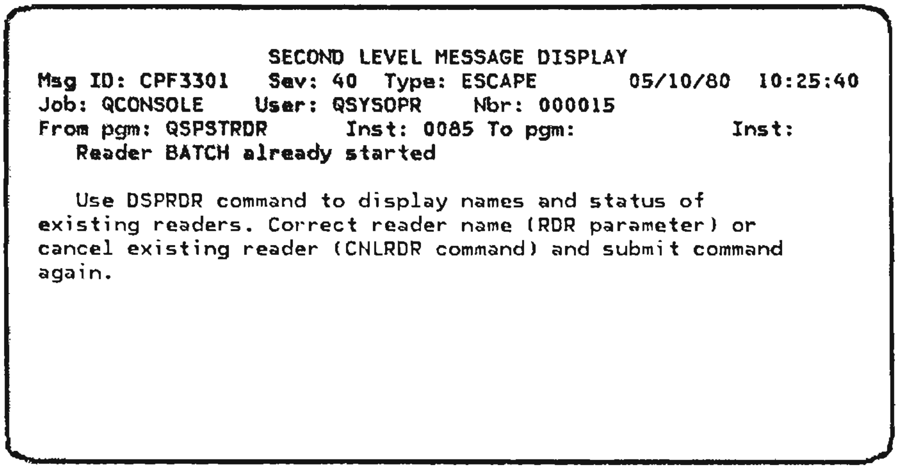
If you make a request that the system cannot process because of an error, you will receive an error message on your current display, such as



```
                       SYSTEM OPERATOR MENU
Select one of the following:
  1. DSPJOBQ (jobq)                 7. STRPRTWTR device
  2. DSPOUTQ (outq)                 8. DSPWTR    (writer)
  3. SNDMSG  tomsgq,(type),msg      9. SBMDKTJOB dev,label,(loc)
  4. CALL    program               10. SBMDBJOB  file,(member)
  5. Execute command               11. DSPSBMJOB
  6. SBMJOB  (job),(jobd),(cmd)    12. DSPACTJOB (reset)        +
Option: ___  Parms: ____________________ ____________________
Cmd or parm: STRCRDROR DEV(QCARD96) RDR(BATCH)
_____________________________________________________________
Log requests: *YES  CF3-Command entry     CF4-Prompt (5 only)
CF6-DSPMSG QSYSOPR  CF7- DSPSBS           CF8-DSPSYS
Reader BATCH already started
```

Depending on the display, the keyboard may also lock.

To obtain additional information about the error, position the cursor on the same line as the message and press the Help key. You then receive a separate display that contains the second-level text for the message:



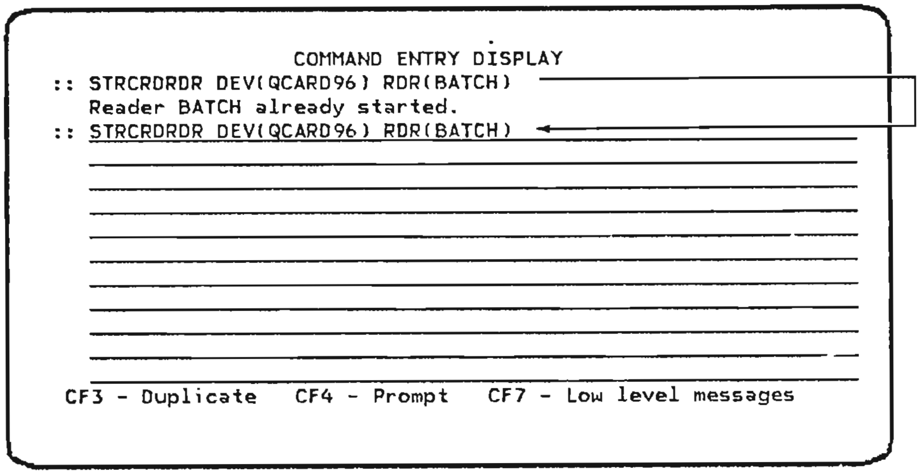
```
                    SECOND LEVEL MESSAGE DISPLAY
Msg ID: CPF3301   Sev: 40  Type: ESCAPE       05/10/80  10:25:40
Job: QCONSOLE    User: QSYSOPR    Nbr: 000015
From pgm: QSPSTRDR      Inst: 0085 To pgm:            Inst:
   Reader BATCH already started

   Use DSPRDR command to display names and status of
existing readers. Correct reader name (RDR parameter) or
cancel existing reader (CNLRDR command) and submit command
again.
```

To view any other messages associated with the command in error do the following:

1. Press Enter or CF1 to return you to the system operator menu.

2. Press CF3 from the system operator menu to obtain the command entry display.

The command entry display shows the command in error and its associated messages. To view other low-level messages associated with the command, press the CF7 key. If you recognize the error, you can press the CF3 key to duplicate the erroneous command in the current input area so that you can correct it.



```
                    COMMAND ENTRY DISPLAY
:: STRCRDRDR DEV(QCARD96) RDR(BATCH)
   Reader BATCH already started.
:: STRCRDRDR DEV(QCARD96) RDR(BATCH)

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

Press the CF3 key.

For some types of displays, such as prompts, you may have to press the Reset key to unlock the keyboard before you can correct the error. You correct the error by keying in the correct characters on top of the incorrect characters.

If you are still not sure how to correct the error after examining the second-level text of the error message, refer to the problem determination procedures in the *Problem Determination Guide*.

Messages Sent to QSYSOPR

Messages regarding job, system, or input/output device errors are sent to the system operator message queue and appear on the separate message display (see *Displaying Messages from a Queue*).

If you are uncertain about what to do, your next step is to request the second-level text for the message by positioning the cursor on the same line as the message and pressing the Help key. This results in a second-level message display that generally provides enough information for you to correct the error.

If you still are uncertain about what to do after examining the second-level text for an error message, refer to the problem determination procedures in the *Problem Determination Guide.* If there is no second-level text, refer to the *Messages Guide.*

Using the Messages Guide and Problem Determination Guide

The *Messages Guide* and *Problem Determination Guide* (see *About This Manual*) contain problem determination procedures for various types of error conditions. These procedures may help resolve an error that you could not resolve from the first- or second-level text of an error message.

The entries in the *Messages Guide* are organized according to the message identifier. To obtain the message identifier for a particular message, use the Help key to request the second-level message display (the display containing the second-level text). The message identifier is shown in the *Msg ID* field of this display.

When you have found the entry in the *Messages Guide* that has the same message identifier as the message you received, look for the *PDP code.* Then refer to the problem determination procedures associated with that PDP code in the *Messages Guide* or the *Problem Determination Guide.*

Keyboard Errors on 5251/5252

If you are using a 5251 or 5252 work station and make a keying error that is detected by the work station, only a four-digit error code, such as 0007 is shown on your display initially. When that occurs, press the Help key to obtain the explanatory message, such as

    Mandatory enter field, must enter data

You can obtain the second-level message text associated with that error by pressing the Help key a second time. To correct the error, you must press the Reset (or Error Reset) key to unlock the keyboard.

Chapter 12. Job Operations

Work is processed by the system in terms of jobs. Each job is a separate unit of work. Using commands, you can determine the status of jobs, control the processing of jobs, call programs that perform specific functions as parts of jobs, or start user batch jobs. Your duties may also include moving information in files into and out of the system as well as reorganizing the information in the system.

JOB CHARACTERISTICS

There are two basic types of jobs on the System/38.

- *Interactive jobs* of users at work stations
- *Batch jobs* that are processed by the system independently of users at work stations

From your point of view, you and the system are engaged in a continuing dialog during an interactive job; you enter requests from a work station and receive a response at that work station for each request. The interactive job consists of all functions requested through the work station from sign-on to sign-off.

For a batch job, on the other hand, you submit a predefined collection of commands and related data, such as a set of punched cards, at one time and receive a complete file of results, such as a printout, at some time later. You receive messages from the system only when circumstances arise that require some action by you. The job consists of all the processing actions that result from input contained within the job. Basically, a batch job is all the commands between a Job command and an End Job command or the next Job · command. For example, the following shows two jobs:

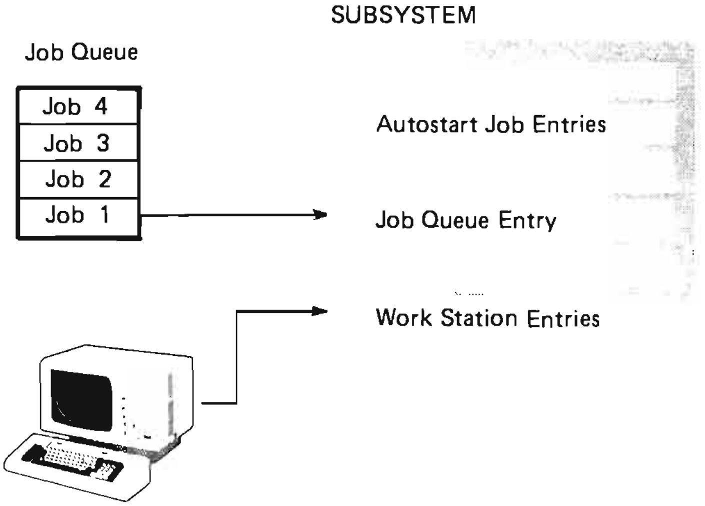
```
// JOB OEWEEKLY    }
CALL PGMA          }   Job: Commands included in the job.
CALL PGMB          }
CALL PGMC          }

// JOB OEMONTHLY   }
CALL PGME          }
// DATA            }
•                  }   Job: Inline data file used in the job.
•                  }
•                  }
// ENDJOB          }
```

Other kinds of jobs that do not have a Job command associated with them are also considered to be batch jobs. These are:

- Autostart jobs that start automatically when a subsystem is started.
- Spooling readers and writers that transfer information into the system from an input device and out of the system to an output device.
- Jobs submitted for independent processing directly from a work station.

Work Entries

All jobs are processed within the environment of a specific subsystem (see *Subsystems* in Chapter 8). Work entries in the subsystem description define the sources from which work can be accepted. For each subsystem, such as the controlling and batch subsystems, there can be three types of work entries:



Work Station: A work station entry specifies the work stations from which interactive jobs (sign-ons) will be accepted. An interactive job becomes active in a subsystem when a user signs on one of the specified work stations, or when another interactive job is transferred into the subsystem.

Autostart Job: An autostart job entry specifies a job that is to become active when a specific subsystem becomes active. When the subsystem is started, the job starts automatically.

Job Queue: A job queue entry specifies the job queue from which a subsystem can accept jobs. When you submit a batch job for processing by a subsystem, it is placed on the job queue associated with the subsystem. The job becomes active in the subsystem when it is taken off the job queue and initiated.

For example, when you submit a job to the batch subsystem, QBATCH, the job is placed on the QBATCH job queue. Processing is started when the job is taken from the queue. Similarly, when you enter a command to start a spooling reader or writer, your request is handled as a job and placed on the QSPL job queue. The reader or writer is started when the job is taken from the queue.

Job Descriptions

All jobs on the system must have a *job description* that defines how the job is to be handled by the system. Standard job descriptions are supplied with the system for interactive jobs, batch jobs, and the various spooling readers and writers. Additional job descriptions may be created for your system.

When a job is submitted to the system, a particular job description must be specified.

For an autostart job, the job description is referenced by the autostart job entry in the subsystem description of the subsystem that is to process the job.

For interactive jobs, the job description to be used is referenced in the work station entry.

When a batch job (other than an autostart job) is submitted to the system, the standard job description QBATCH is used unless you specify another job description. For example, if a job on punched cards requires a special job description, the job description name is specified in the Job (//JOB) command that appears on the first card, such as

$$\text{// JOB}\ \underbrace{\text{WKPAY}}_{\substack{\textbf{Name}\\\textbf{of Job}}}\ \underbrace{\text{CHECKS}}_{\substack{\textbf{Name of Job}\\\textbf{Description}}}$$

The Job (//JOB) command is provided as part of the input for the job.

When a batch job is submitted from a work station, the QBATCH job description is used unless another job description is specified in the Submit Job (SBMJOB) command that is entered at the work station.

Job Numbers

Every job in the system has a unique *job number* by which it is distinguished from all other jobs. This includes not only user jobs but also system jobs such as the system arbiter (QSYSARB) that control system functions. The system assigns a job number to each job when the job enters the system; the job retains that job number until it leaves the system. An interactive job enters the system when a user signs on at a work station; the job leaves the system when the user signs off (or the job is canceled) and all output has been produced. A batch job enters the system when it is placed on a job queue; the job leaves the system when its processing has been completed (or the job is canceled) and all output has been produced.

Job numbers consist of six digits and can have any value from 000001 through 999999. The numbers are assigned to jobs in the sequence from lowest to highest in the same order that the jobs enter the system. For example, the tenth job to enter the system is assigned the job number 000010. When all numbers have been assigned through 999999, the assignment process wraps around (begins again at 000001). For a short time after the assignment process has reached the limit of 999999 and has wrapped around, the job number of new jobs will be less than that of jobs already in the system.

Job Logs

A *job log* consists of messages (including commands) generated during the processing of a job. There is a job log for each interactive job at a work station as well as each autostart or submitted batch job.

Whether all, some, or none of those messages is logged is defined by the LOG parameter in the job description or in the command used when the job was submitted. You can change the logging level as needed, or requested, by using the LOG parameter of the Change Job (CHGJOB) command (see *Changing a Job* later in this chapter). If the maximum logging level was specified for your interactive job, your job log would show:

- All commands you entered (but not the individual commands in any CL programs you invoked).
- All response messages to your requests.
- Messages relating to lower level programs that were invoked as a result of your requests. These messages normally would not be displayed at your work station.

Job logs are printed on the system printer after their jobs end (Figure 12-1). However, some job logs may not be printed. The job log for an interactive job is not printed unless LOG(*LIST) is specified in the SIGNOFF command when a user signs off a work station (see *Work Station Sign-Off* in Chapter 6). If you cancel a job, you can limit the number of job log entries (messages) that are printed; if the limit is zero, the job log is not printed (see *Canceling a Job* in this chapter). When the system is started after an abnormal termination, you can specify on the start CPF prompt whether the job logs are to be printed for the jobs that were active at the time of the termination (see *Starting CPF* in Chapter 8).

An example with an explanation of a job log is shown in the *Problem Determination Guide.*

As shipped from IBM, the system is set up such that all job logs to be printed are written to the spooled output file QPJOBLOG. This file is placed on the default output queue for the job. The job logs can be printed if the spooling subsystem is active and a print writer has been started to produce output from the output queue. If the spooling subsystem is not active, you can start it by entering

```
STRSBS SBSD(QSPL.QGPL)
```

You can start a print writer to the QPRINT output queue to the system printer by entering

```
STRPRTWTR DEV(QSYSPRT) OUTQ(QPRINT)
```

Job logs for system jobs such as the system arbiter (QSYSARB), which are not completed until the final stages of terminating the system, will not be printed until the next time the system is started.

It will be your responsibility to collect the job logs and provide them to system users according to the procedures established for your system.

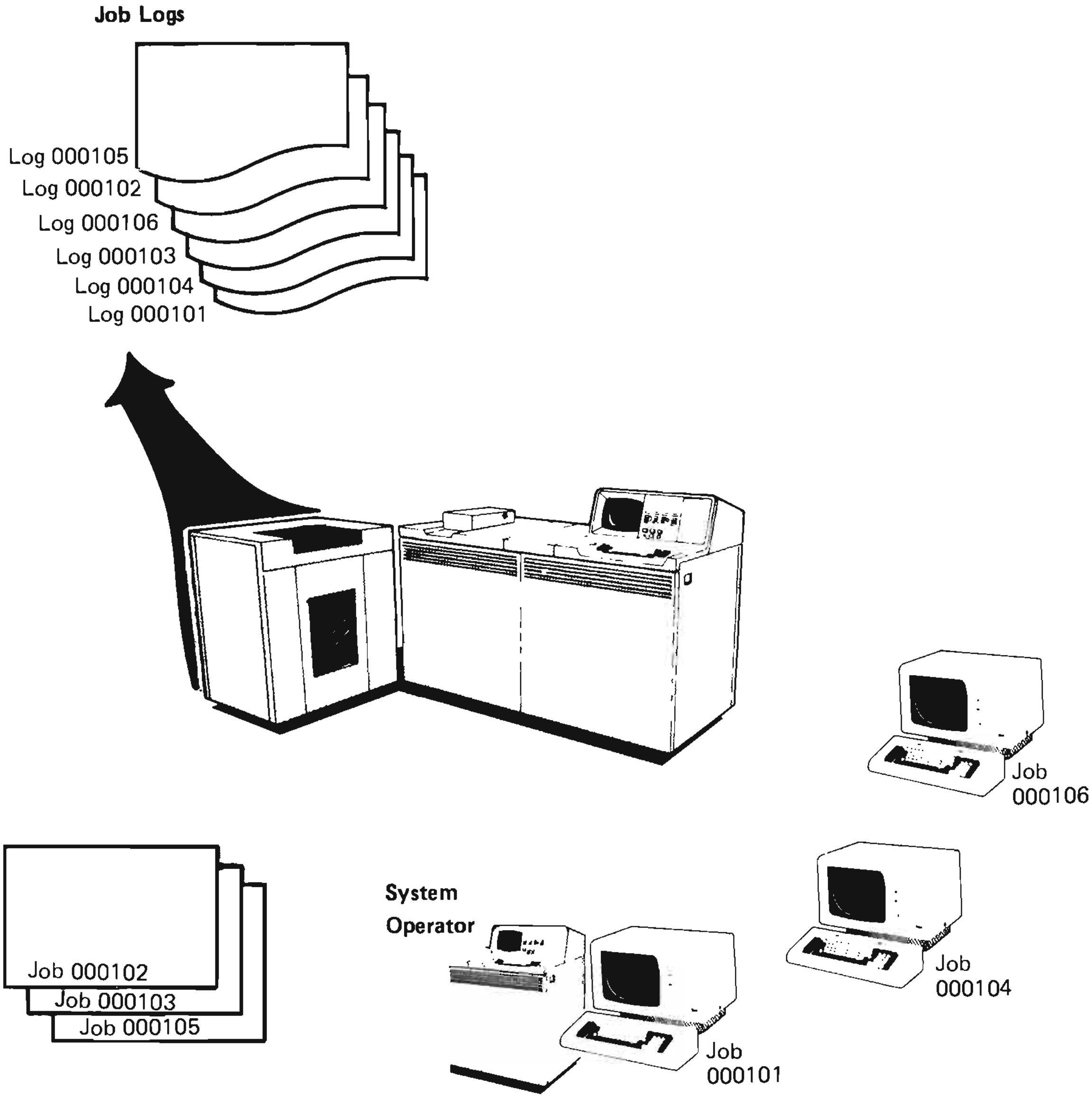


Figure 12-1. Job Logs (Printed After Job Ends)

YOUR INTERACTIVE JOB

As described in Chapter 6, when you sign on a work station, you create an interactive job in the same way as any other user. That single job continues until you sign off or until it is otherwise terminated such as when the system is powered down or the subsystem in which it is being processed is terminated or is canceled by another user who has the proper authority.

During your interactive job, you use the system operator menu and, as needed, the command entry display, menus, and prompts to control system operations. Between sign-on and power down at the system console, for example, you may enter commands to:

- Start the batch and spooling subsystems
- Start a card reader to the batch job queue
- Start a printer writer from an output queue
- Hold an individual job or a queue containing several jobs
- Save a version of a system log
- Display messages sent to the system operator message queue
- Initiate power down

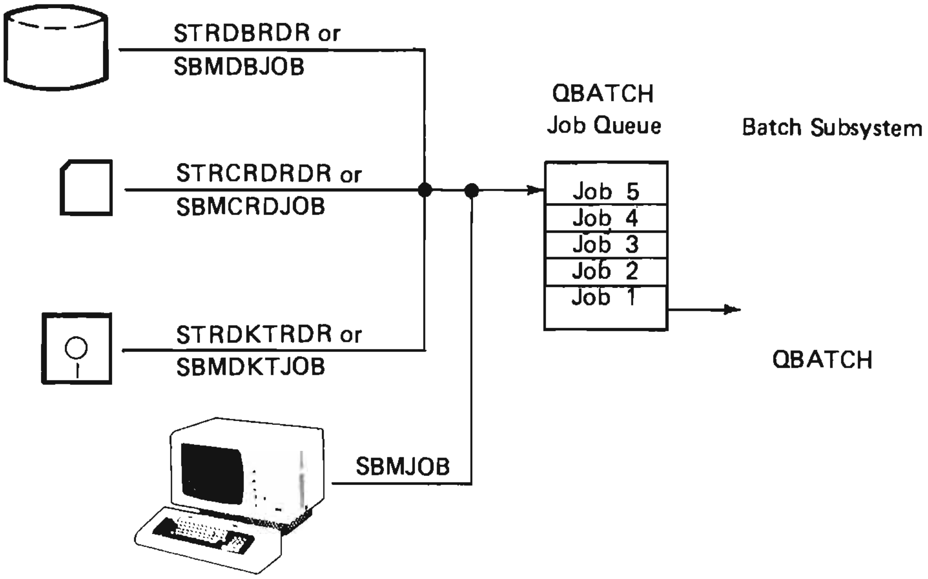
Each function is performed by a separate command. Together, these commands constitute your interactive job.

SUBMITTING BATCH JOBS

You can submit a batch job in three ways:

- Place the job directly on a job queue using the Submit Job (SBMJOB) command at a work station.
- Use the Submit Diskette Jobs (SBMDKTJOB) command, Submit Data Base Jobs (SBMDBJOB) command, or Submit Card Jobs (SBMCRDJOB) command that takes the job from the data base or an input device and places the job on a job queue.
- Start a spooling reader that transfers the job from the data base or an input device to a job queue (the spooling subsystem must be active).

All three of these procedures involve entering one or more commands (below). Your operating instructions or the individual job setup instructions should indicate how each command is to be entered; that is, which parameters to enter and which values to specify for those parameters.



Note that a job can be placed on any batch job queue while the batch subsystem is inactive, but it will not be processed until the subsystem is started and its priority and time stamp make it available to be taken from the queue.

Using Spooling Readers

Using a spooling reader, you can submit a batch job from the data base, from cards placed in a card device, or from diskettes placed in the diskette magazine drive.

Jobs from Data Base

Because the information for a job being submitted from the data base does not have to be supplied for the system, your only task is to enter the STRDBRDR command that starts the reader, such as

```
STRDBRDR FILE(RECPRT.RECORDS)
```

For details on the command, see *Starting a Data Base Reader* in Chapter 10.

Jobs on Punched Cards

If your system has the Multi-Function Card Unit (see Chapter 4), batch jobs can be submitted on 96-column cards. The stack of cards supplied to you may contain a single job or several jobs in succession (a job stream). Either way, your procedures are the same.

Before loading the cards in the MFCU, you can help ensure that the job input stream will be processed correctly if you:

- Check the first card to verify that the first few characters are the job delimiter

  ```
  // JOB
  ```

- Check the last card to verify that it contains the end-of-file delimiter

  ```
  /*
  ```

Use the following procedure to submit the job:

1. Enter the DSPDEVCFG command and verify that no reader or job is using the device and that it is varied on.
2. Press the NPRO key to clear any cards left in the MFCU.
3. Load the cards in the hopper indicated in the setup instructions for the job. The cards should be placed in the hopper face down with their top edge toward the left edge of the hopper.
4. Press the Start key on the MFCU operator panel.
5. Enter the STRCRDRDR command on your work station, exactly as indicated in the job setup instructions. For example, the command might be:

   ```
   STRCRDRDR DEV(QCARD96)
   ```

For details on the command, see *Starting a Card Reader* in Chapter 10.

Jobs on Diskettes

If you are to submit a job from information on a diskette, you should either be supplied with the diskettes or given instructions on which diskettes to use. You may have to insert diskettes in a magazine (see Chapter 3 for details).

Use the following procedure to submit the job:

1. Load the diskettes or magazine(s) in the position of the diskette drive that is indicated in the job setup instructions.

2. Enter the STRDKTRDR command exactly as indicated in the job setup instructions. Note that the value or values specified in the LOC parameter of the command must match the location in which you placed the diskettes or magazines. If you are instructed to specify one or more values for the VOL parameter, those values should match the volume identifiers indicated on the diskettes. For example, if you are instructed to enter the command

   ```
   STRDKTRDR DEV(QDKT) LABEL(RECORDS)
    LOC(*M2 3 4) VOL(RECA RECB)
   ```

   the diskette identified as RECA must be in the third position and diskette RECB must be in the fourth position of a magazine, and the magazine must be placed in the right magazine position (*M2) of the diskette drive (for reference, see Figure 13-1 in Chapter 13).

Details on the STRDKTRDR command are given in *Starting a Diskette Reader* in Chapter 10.

Using the Submit Jobs (SBMCRDJOB, SBMDBJOB, SBMDKTJOB) Commands

You can use the Submit Data Base Jobs (SBMDBJOB) command to submit a batch job from the data base, the Submit Card Jobs (SBMCRDJOB) command to submit a batch job from cards, or the Submit Diskette Jobs (SBMDKTJOB) command to submit a batch job from diskette.

Note that these commands place a job directly on a jobqueue, bypassing the spooling subsystem. However, if output from th job is to be produced on an output device (printer, diskette drive, MFCU), that output will be handled by a spooling writer in the same way as output for jobs that were placed on a job queue by a spooling reader.

Jobs from Data Base

Because the information for a job being submitted from the data base does not have to be supplied for the system, your only task is to enter the SBMDBJOB command that reads the job, such as:

```
SBMDBJOB  FILE(RECPRT.RECORDS)
```

For details on the command, see *Submitting Data Base Jobs* in Chapter 10.

Jobs on Punched Cards

If your system has the Multi-Function Card Unit (see Chapter 4), batch jobs can be submitted on 96-column cards. The stack of cards supplied to you may contain a single job or several jobs in succession (a job stream). Either way, your procedures are the same.

Before loading the cards in the MFCU, you can help ensure that the job input stream will be processed correctly if you:

- Check the first card to verify that the first few characters are the job delimiter

  ```
  // JOB
  ```

- Check the last two cards to verify that they contain the End Input (ENDINP) command followed by the end of file delimiter (/*)

  ```
  //ENDINP
  /*
  ```

Use the following procedure to submit the job:

1. Enter the DSPDEVCFG command and verify that no reader or job is using the device and that it is varied on.
2. Press the NPRO key to clear any cards left in the MFCU.
3. Load the cards in the hopper indicated in the setup instructions for the job. The cards should be placed in the hopper face down with their top edge toward the left edge of the hopper.
4. Press the Start key on the MFCU operator panel.
5. Enter the SBMCRDJOB command on your work station, exactly as indicated in the job setup instructions. For example, the command might be:

```
SBMCRDJOB DEV(QCARD96)
```

For details on the command, see *Submitting Card Jobs* in Chapter 10.

Jobs on Diskettes

If you are to submit a job from information on a diskette, you should either be supplied with the diskettes or given instructions on which diskettes to use. You may have to insert diskettes in a magazine (see Chapter 3 for details).

Use the following procedure to submit the job:

1. Load the diskettes or magazine(s) in the position of the diskette drive that is indicated in the job setup instructions.
2. Enter the SBMDKTJOB command exactly as indicated in the job setup instructions. Note that the value or values specified in the LOC parameter of the command must match the location in which you placed the diskettes or magazines. If you are instructed to specify one or more values for the VOL parameter, those values should match the volume identifiers indicated on the diskettes. For example, if you are instructed to enter the command

```
SBMDKTJOB DEV(QDKT) LABEL(RECORDS)
  LOC(*M2 3 4) VOL(RECA RECB)
```

   the diskette identified as RECA must be in the third position and diskette RECB must be in the fourth position of a magazine, and the magazine must be placed in the right magazine position (*M2) of the diskette drive (for reference, see Figure 13-1 in Chapter 13).

Details on the SBMDKTJOB command are given in *Submitting Diskette Jobs* in Chapter 10.

Using the Submit Job (SBMJOB) Command

The SBMJOB command can be used to submit a batch job if the job description being used is already stored in the system. The job description QBATCH is used if you do not specify another job description name. If you do not specify a job name, the job description name is used as the job name.

Note that this command places a job directly on a job queue, bypassing the spooling subsystem. However, if output from the job is to be produced on an output device (printer, diskette drive, MFCU), that output will be handled by a spooling writer in the same way as output for jobs that were placed on a job queue by a spooling reader.

By using a simple form of the SBMJOB command that includes the RQSDTA parameter, you can submit an individual command or call a program that is to be run on the system independently of your interactive job. The command to be submitted is specified as the value of the RQSDTA parameter. Similarly, you can specify the CALL command in the RQSDTA parameter to call a program that performs a particular kind of batch processing. After you have entered the SBMJOB command, you can go on to other tasks and enter other commands at your work station without waiting for the execution of the submitted command or program to be completed.

When you select option 6 on the system operator menu and enter a command in the *Msg or cmd* field, you are using the SBMJOB command with only the RQSDTA parameter. If you enter the SBMJOB command without using option 6, the value you specify in the RQSDTA parameter must be enclosed in apostrophes. For example, if you select option 5 on the system operator menu and enter the complete SBMJOB command in the *Msg or cmd* field, or if you enter the SMBJOB command on the command entry display, the command might look like this:

```
SBMJOB RQSDTA('CALL PAYPGM')
```

Not all commands can be entered as batch jobs. The box in the lower right corner of the syntax diagram indicates whether batch entry is valid. If a B is shown following the word *Job*, such as

```
Job:B,I or Job:B
```

the command can be submitted as a batch job.

Messages regarding the command or program that you submitted as a batch job are recorded in the job log, which is printed after the job is completed. Messages that indicate when the job was started and completed are also recorded in the system history log (see *System Logs* in Chapter 8). You can also specify a message queue for the completion message by using the MSGQ parameter. For example, an entry in the history log that indicates the completion of a batch job you submitted might look like this:

```
Job QBATCH.QSYSOPR.715475 completed on 11/17/80 at 08:23:47
```

The job description name, QBATCH, is used here as the job name because you did not specify a job name when you submitted the job.

Examples:

1. To submit as a batch job the command to save the large library ORDERS, enter

   ```
   SBMJOB RQSDTA('SAVLIB LIB(ORDERS)')
   ```

2. To submit the job WKPAY, which has the job description CHECKS that is located in the library PAYROLL, and call the program PAYPGM to process the job, enter

   ```
   SBMJOB JOB(WKPAY) JOBD(CHECKS.PAYROLL)
    RQSDTA('CALL PAYPGM')
   ```

Starting Writers

When you submit a batch job, you should also start the writer that will be producing output from the job if output is to be produced. The job setup instructions should specify which writer and which device is to be used and what to specify in the writer command.

Details on the commands that start spooling writers are given in Chapter 10.

CALLING PROGRAMS

Jobs submitted to the system may supply data and call programs stored in the system to process the data. As part of your job, you may need to call programs especially designed for your system. These programs may perform standard operations for users of your system, such as printing out a weekly inventory of records; or they may simplify a complicated command or combine several commands, such as a program to start all spooling functions.

To call a program, you use the CALL command. When you select option 3 on the system operator menu, you are using this command. 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 (see *Command Syntax* in Chapter 6). The keyword-with-value form of the same example would be

```
CALL PGM(INVENTORY)
```

You can also add a library qualifier to the program name, such as

```
CALL PGM(INVENTORY.RECORDS)
```

The only other parameter of the command is PARM, which is used to pass parameters to the program; that is, to specify information to be used by the program.

Examples:

1. To call a program SPLSTR that starts the spooling subsystem and specific readers and writers, enter

   ```
   CALL PGM(SPLSTR)
   ```

   or simply

   ```
   CALL SPLSTR
   ```

2. To call a program INVPRT, located in the library RECORDS, that prints a weekly inventory, and pass two parameters, CURRENT and UPDATE, to it, enter

   ```
   CALL PGM(INVPRT.RECORDS) PARM(CURRENT UPDATE)
   ```

   or simply

   ```
   CALL INVPRT.RECORDS (CURRENT UPDATE)
   ```

INTERACTIVE VERSUS BATCH JOBS

Although your interactive job allows you to make a request and receive an almost immediate response in most situations, there may be situations where entering requests interactively hampers your job control duties and makes inefficient use of the system.

For example, if you enter a command to call a long-running program, such as

```
CALL FILECHECK
```

your work station is under control of the program until it is completed. During that time you cannot enter another command in your interactive job. If you submitted a batch job that called the same program, such as

```
SBMJOB RQSDTA('CALL FILECHECK')
```

the job would be placed on the batch job queue to be processed independently of your work station job, and control of your work station would return immediately to you. Similarly, if you are routinely entering the same sets of commands, those commands could be compiled into a CL program that could be called from a single batch job you submitted from your work station.

DETERMINING JOB STATUS

There are several display commands that you can use to obtain information about jobs. For general information, you can use one of the system or subsystem display commands (Chapter 8), or one of the spooling display commands (Chapter 10). For detailed information, you use the Display Job (DSPJOB) command.

General Job Information

The following summarizes what types of general job information you can obtain from various display commands.

Display System (DSPSYS) Command: This command requests a display that indicates the number of jobs in each subsystem (see Chapter 8).

Display Subsystem (DSPSBS) Command: This command requests a display that lists the names of all jobs in one or all subsystems, and indicates the status of each (see Chapter 8).

Display Job Queue (DSPJOBQ) Command: This command requests a display that lists either the number of jobs on each job queue or specific information about each job on a particular job queue, such as the name of the job, who submitted it, its job number, its priority, and its status (see Chapter 10).

Display Output Queue (DSPOUTQ) Command: This command requests a display that lists either the number of files on each output queue or specific information about each file on a particular output queue, such as name of file, file number, name of job, name of user who submitted the job, job number, priority, and status.

Display Reader (DSPRDR) Command: This command requests a display that lists either all the readers and their status or detailed information about a specific reader.

Display Writer (DSPWTR) Command: This command requests a display that lists either all the writers and their status or detailed information about a specific writer.

Display Active Jobs (DSPACTJOB) Command: This command requests a display that lists information on all active jobs in the system. The display shows performance statistics and the status of the job.

Obtaining the Active Jobs Display

To request a display of all active jobs on the system, enter the Display Active Jobs (DSPACTJOB) command or request option 12 from the system operator menu, or press the CF8 key from the System Status display. This command has two parameters.

OUTPUT Parameter: This parameter determines if the display is to be displayed at a work station or listed on a printer.

- If you do not enter this parameter, the display will be shown on your work station screen.
- If you enter *LIST, the display will be listed with your job's spooled output on a printer.

RESET Parameter: This parameter determines whether the job statistics are reset to start a new measurement period.

- If you do not enter this parameter, the job statistics are not reset and the information is the accumulated job status since the last reset.
- If you enter *YES, the job statistics are reset and you begin accumulating job statistics for a new time period.

Examples:

1. To display the status of all active jobs on the system, enter

```
DSPACTJOB
```

2. To print the active job information, enter

```
DSPACTJOB OUTPUT(*LIST)
```

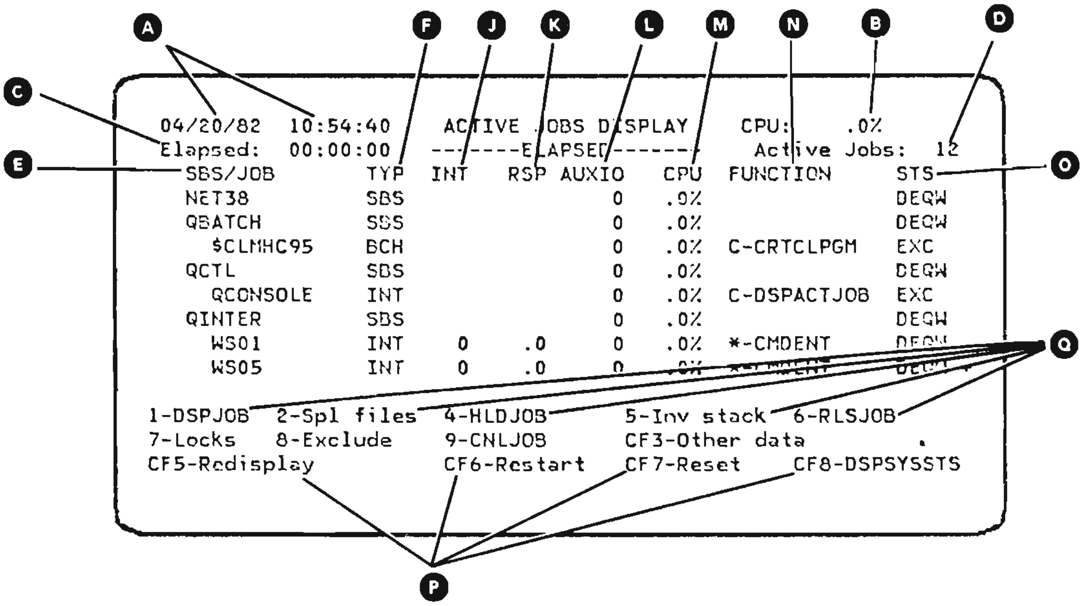
3. To reset the active job statistics and start a new measurement period, enter

```
DSPACTJOB RESET(*YES)
```

Interpreting the Active Jobs Display

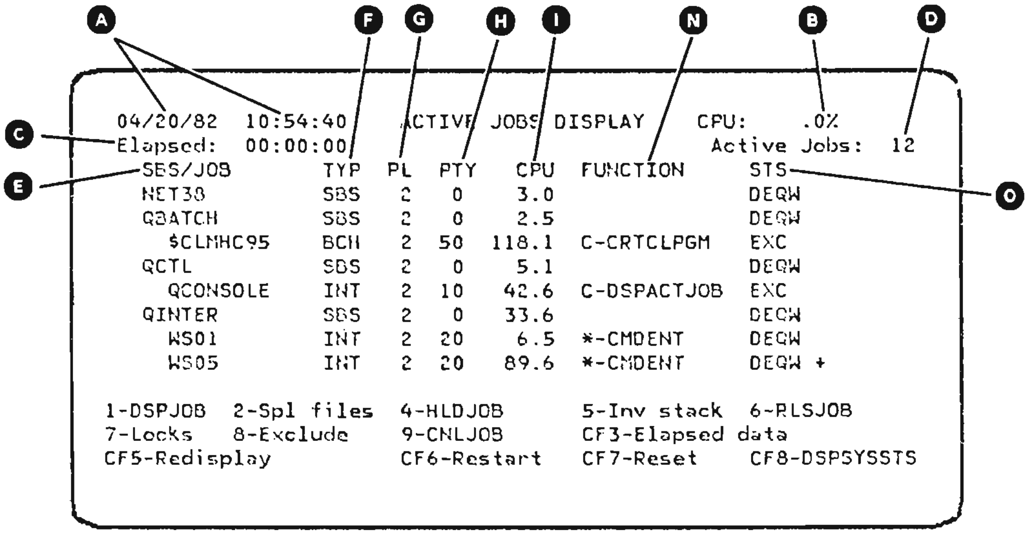
When you are using the system console and you enter the DSPACTJOB command, enter option 12 from the system operator menu, or press the CF8 key from the System Status display, you receive the first of two displays. To obtain the second display, press the CF3 key.

First Display



```
04/20/82  10:54:40    ACTIVE JOBS DISPLAY    CPU:     .0%
Elapsed:  00:00:00    ---------ELAPSED--------   Active Jobs:  12
  SBS/JOB         TYP  INT    RSP AUXIO   CPU  FUNCTION        STS
  NET38           SBS                 0   .0%                  DEQW
  QBATCH          SBS                 0   .0%                  DEQW
    $CLMHC95      BCH                 0   .0%  C-CRTCLPGM      EXC
  QCTL            SBS                 0   .0%                  DEQW
    QCONSOLE      INT                 0   .0%  C-DSPACTJOB     EXC
  QINTER          SBS                 0   .0%                  DEQW
    WS01          INT    0     .0     0   .0%  *-CMDENT        DEQW
    WS05          INT    0     .0     0   .0%  *-CMDENT        DEQW +

1-DSPJOB  2-Spl files  4-HLDJOB     5-Inv stack  6-RLSJOB
7-Locks   8-Exclude    9-CNLJOB     CF3-Other data
CF5-Redisplay          CF6-Restart  CF7-Reset    CF8-DSPSYSSTS
```

Second Display



```
04/20/82  10:54:40    ACTIVE JOBS DISPLAY    CPU:     .0%
Elapsed:  00:00:00                           Active Jobs:   12
  SBS/JOB         TYP  PL  PTY   CPU   FUNCTION       STS
  NET38           SBS  2    0    3.0                  DEQW
  QBATCH          SBS  2    0    2.5                  DEQW
    $CLMHC95      BCH  2   50  118.1   C-CRTCLPGM     EXC
  QCTL            SBS  2    0    5.1                  DEQW
    QCONSOLE      INT  2   10   42.6   C-DSPACTJOB    EXC
  QINTER          SBS  2    0   33.6                  DEQW
    WS01          INT  2   20    6.5   *-CMDENT       DEQW
    WS05          INT  2   20   89.6   *-CMDENT       DEQW +

1-DSPJOB  2-Spl files  4-HLDJOB     5-Inv stack   6-RLSJOB
7-Locks   8-Exclude    9-CNLJOB     CF3-Elapsed data
CF5-Redisplay          CF6-Restart  CF7-Reset     CF8-DSPSYSSTS
```

To return to the first display, press the CF3 key.

Ⓐ *Date/Time:* This is the job date and the system time when you requested the display.

Ⓑ *CPU:* This is the percent of time used by the CPU during the Elapsed Ⓒ time period. The CPU time percentage is normally higher than the CPU time used by the displayed jobs because it includes CPU time used for system overhead and jobs that have ended during the elapsed time period.

Ⓒ *Elapsed:* This is the amount of time that has elapsed between the measurement start time and the current system time. The measurement time starts when the display is initially requested or when you press the CF7 key.

Ⓓ *Active Jobs:* This is the number of active jobs in the system. This number includes system jobs such as start CPF and system arbiter.

Ⓔ *SBS/JOB:* This is the job name of the active job. Jobs that execute in a subsystem are indented two positions under the subsystem that they are associated with.

Ⓕ *TYP:* This is the job type. The types of jobs are:

- ASJ (auto start job)
- BCH (batch)
- INT (interactive)
- RDR (spool reader)
- SBS (subsystem monitors)
- SYS (system)
- WTR (spool writer)

Ⓖ *PL:* This is the system pool that the job is running in. These identifiers are not the same as those specified in the subsystem description. The identifiers shown in this column are the same as the system pool identifiers shown on the system status display.

Ⓗ *PTY:* This is the execution priority of the job. System jobs, (subsystem monitor, start CPF, system aribter) that have a higher priority than user jobs, will display a priority of 0.

Ⓘ *CPU:* This is the total CPU time (seconds) used by the job.

Ⓙ *INT:* This is the number of operator interactions during the measurement time interval. This number includes actions such as the operator pressing the Enter key or CF keys. This field is for interactive work station jobs only and does not include the console.

K *RSP:* This is the average system response time (seconds) over the measurement time interval. This is for interactive work station jobs only and does not include the console.

L *AUXIO:* This is the number of auxiliary storage reads and writes the job has made during the time interval. This includes data base and non-data base paging.

M *CPU:* This is the percent of CPU time used by the job over the measurement interval.

N *FUNCTION:* This is the high-level function being performed by the job. This field will be blank when a logged function has not been performed. The first character in this field indicates the type of function that is in process. If the first character is:

C: A command is in process. The name following the C is the name of the command. Commands shown are commands entered interactively, in a batch job stream, or requested from a system menu. Commands in CL programs do not appear.

P: A program is in process. The name following the P is the name of the high-level program that is called interactively, called in a batch job stream, is the name of the initial program specified in the user profile, or the name of a system request processor such as QOPRMENU, QPGMMENU, or QCALLMENU.

I: An index rebuild is in process for the file identified by the name following the I.

L: A logging function is in process. The name following the L identifies the log to which the logging is being done:

- QHST (is being logged to a data base file)

- QSRV (is being logged to a data base file)

- QCHG (is being logged to a data base file)

This is one of the following special values:

- ADLACTJOB: Auxiliary storage is being allocated for the number of jobs specified in the system value QADLACTJ.

- ADLTOTJOB: Auxiliary storage is being allocated for the number of jobs specified in the system value QADLTOTJ.

- CMDENT: The Command Entry display is being displayed.

- DUMP: A dump is in progress.

- JOBLOG: A job log is being produced.

O *STS:* This is the status of the job. Only one status is displayed per job. If the hold job, release job, or cancel job functions are executed against a job through the Active Jobs display, the status will show *HLD or *RLS. Other possibilities are:

- CNL: The job has been canceled with the *IMMED option or delay time has expired with the *CNTRLD option.
- SRQ: The job is inactive as the result of a system request.
- DEQA: The job is waiting on a dequeue and is in the pool activity level.
- DEQW: The job is waiting on a dequeue and is not in the pool activity level.
- LCKW: The job is waiting for a lock and is not in the pool activity level.
- EVTW: The job is waiting on an event and is not in the pool activity level.
- INEL: The job is ineligible and not in the pool activity level.

P *CF Keys:*

- CF3: Allows you to switch, back and forth, between the first display and the second display.
- CF5: Updates the statistics on the display. The previous start time continues to be used as the start time for the measurement interval.
- CF6: Resets the start time to start a new measurement interval. This causes the new measurement interval start time to be set to the previous display time shown on line 1.
- CF7: Resets the display. This causes the start time to be set to the current time. This has the same effect as entering the command with RESET(*YES).

Q *Options:* An input field to the left of each job name can be used to enter any one of the options shown at the bottom of the display. If you enter options for more than one job, the options are performed on the jobs in the order in which the jobs are shown on the display. The system executes these options using default values for all the parameters.

1 DSPJOB: The display job menu is displayed. When this option is selected for a reader or writer job, the DSPRDR or DSPWTR display is displayed. This option is not valid for system jobs or subsystem monitor jobs.

2 Spl files: The job's spooled output files are displayed. This option is valid for all jobs.

4 HLDJOB: The job is held, but its spooled files are not held. The HLDRDR or HLDWTR command is executed if this option is entered for a reader or writer. This option is not valid for system jobs or subsystem jobs.

5 Inv stack: The job's program invocation stack is displayed. This option is valid for all jobs.

6 RLSJOB: The job, which must be in the held state, is released. The RLSRDR or RLSWTR command is executed if this option is entered for a reader or writer. This option is not valid for subsystem jobs or system jobs. If the command is successfully executed, *RLS will appear in the status field.

7 Locks: The job's locks are displayed. This option is valid for all jobs. The locks displayed do not include data base record locks.

8 Exclude: The job is excluded from the display. This option does not effect the job, only the display.

9 CNLJOB: The job is canceled but the spooled output files produced by the job are not canceled. A controlled cancel is performed as if the CNLJOB command was entered using all defaults for the parameters. The CNLRDR or CNLWTR command is executed if the job is a reader or writer.

When all of the commands have been executed, the Active Jobs display is reshown with no updated information. The same set of jobs will be shown unless an error occured while processing a command. If an error occured, the first job with an error is shown. Error or completion messages are shown at the bottom of the display. An indication of successful nondisplay commands will be shown in the status field of the job.

If there are more jobs than can be shown on a single display, the roll keys can be used and options entered on more than one display before the Enter key is pressed.

The CF1 key can be used to exit from the Active Jobs display or to exit from a display presented as the result of executing commands. Options entered for jobs after the CF1 key is pressed are not performed.

Specific Job Information

You can obtain information about a particular job by requesting a specific job display. Each job display gives the basic characteristics and the current status of the job. From the job display menu, you can obtain one or all of the following additional displays:

- A display of job status attributes
- A display of job definition attributes
- A display of job execution attributes
- A display of the program stack
- A display of the job's spooled output files

If you obtain the display of spooled output files from the job display, you can execute commands to display, hold, release, or cancel particular spooled output files directly from that output files display.

Obtaining the Job Display

To request a specific job display, enter the Display Job (DSPJOB) command. The command has two parameters.

JOB Parameter: This parameter determines which job is to be displayed.

- If you do not enter the parameter, information is provided about your own interactive job.
- If you specify the name of a job, information is provided on that job. You should include the fully qualified job name if you know it; that is, the job name followed by the user name and job number, such as

```
JOB(ORDER3.JONES.000355)
```

If you do not enter the fully qualified job name and more than one job in the system is identified by the same job name, or job name and user name, you receive messages that list the job name, user name, and job number for all jobs that have the name you specified. You can then correct the command to include the fully qualified name of the job you want displayed.

OUTPUT Parameter: This parameter determines how the requested information is provided.

- If you do not enter the parameter, the information is displayed at your work station.
- If you specify OUTPUT(*LIST) the information is spooled on an output file and placed on an output queue for printing on the system printer.

Note: You cannot display job information on certain system jobs, such as the system arbiter (QSYSARB) or subsystem monitors.

Examples:

1. To display information about your interactive job, enter

```
DSPJOB
```

2. To print system information about the job PAY3 with the job number 000358 that was submitted by the user named DEPT25, enter

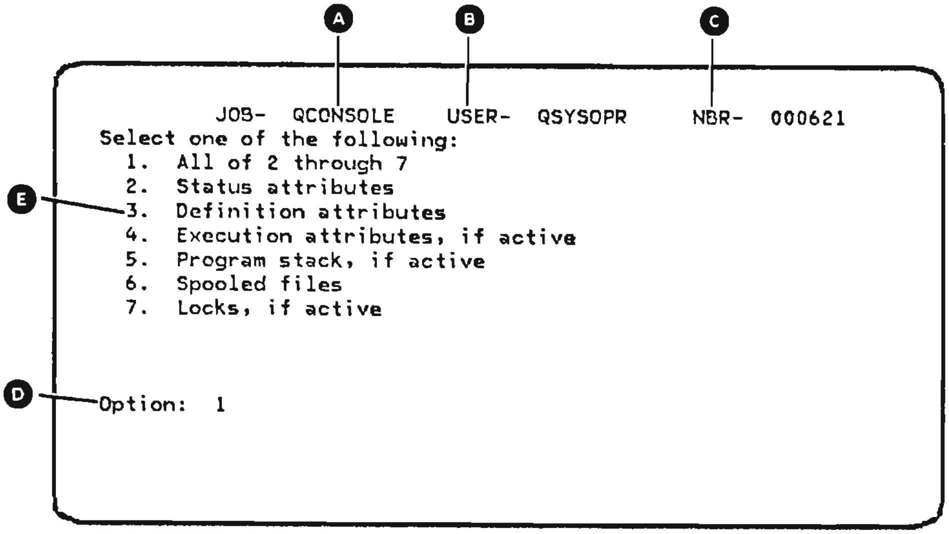
```
DSPJOB JOB(PAY3.DEPT25.000358) OUTPUT(*LIST)
```

A job display is also available by using CF3 from DSPRDR or DSPWTR display for a specific reader or writer. The display that follows CF3 is the same as you would get by entering Display Job (DSPJOB) command and entering the reader or writer as the job name and QSYS as the user.

A job display is also available from DSPJOBQ of a particular job queue, DSPSBMJOB, DSPSBS, DSPLINSTS, DSPCUDSTS, or DSPDEVSTS by taking Option 1.

Interpreting the Job Display

When you enter the DSPJOB command, you receive a menu display of the form:



```
              JOB-  QCONSOLE    USER-  QSYSOPR     NBR-  000621
 Select one of the following:
   1.  All of 2 through 7
   2.  Status attributes
   3.  Definition attributes
   4.  Execution attributes, if active
   5.  Program stack, if active
   6.  Spooled files
   7.  Locks, if active



 Option:  1
```

Ⓐ *Job Name:* This is the name of the job.

- For autostart jobs, it is the name provided through the autostart job entry of the subsystem description.
- For batch jobs, it is the name specified when the job was submitted.
- For interactive jobs, it is the name of the work station being used for the interactive session.

Ⓑ *User:* This is the name of the user who submitted the job.

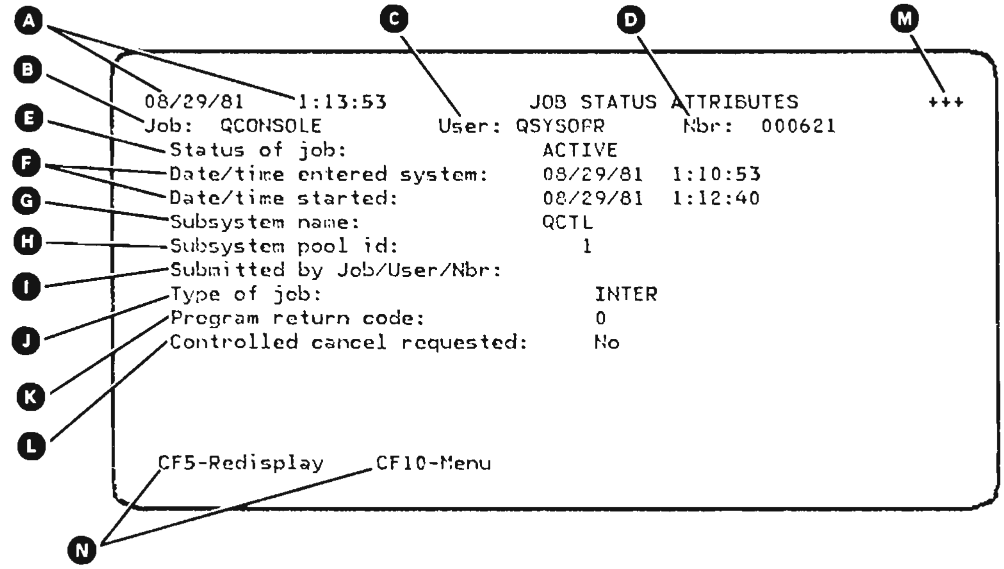
- For an autostart job, it is the name defined in the job description referenced by the autostart job entry.
- For a batch job, this is the name given in the job description. The job description is specified in the JOB command, such as on the first card of card input, or in the SBMJOB command that you or another user enter when you submit a job from a work station. For readers and writers the user is QSYS.
- For an interactive job, it is usually the name of the user profile associated with the user password entered at sign-on. If the work station was signed on without a password, the name shown is the name specified in the job description referenced by the work station entry of the subsystem description.

Ⓒ *NBR:* This is the job number assigned to the job by the system when it entered the system. No other job in the system has the same job number (see *Job Numbers* in this chapter).

Ⓓ *Input Field:* Enter the one digit option code Ⓔ in the input field.

Ⓔ *Option Code:* Enter option code in the input field. If you do not enter an option, the option defaults to 1.

The job display option 2 produces the following job status display:



```
08/29/81    1:13:53              JOB STATUS ATTRIBUTES            +++
Job:  QCONSOLE          User: QSYSOPR       Nbr:  000621
  Status of job:                  ACTIVE
  Date/time entered system:       08/29/81  1:10:53
  Date/time started:              08/29/81  1:12:40
  Subsystem name:                 QCTL
  Subsystem pool id:                1
  Submitted by Job/User/Nbr:
  Type of job:                        INTER
  Program return code:                0
  Controlled cancel requested:        No



 CF5-Redisplay     CF10-Menu
```

A *Date/Time:* The date and the time shown are the job date and the system time when you requested the display.

B *Job Name:* This is the name of the job.

- For autostart jobs, it is the name provided through the autostart job entry of the subsystem description.
- For batch jobs, it is the name specified when the job was submitted.
- For interactive jobs, it is the name of the work station being used for the interactive session.

C *User:* This is the name of the user who submitted the job.

- For an autostart job, it is the name defined in the job description referenced by the autostart job entry.
- For a batch job, this is the name given in the job description. The job description is specified in the JOB command, such as on the first card of card input, or in the SBMJOB command that you or another user enter when you submit a job from a work station.
- For an interactive job, it is usually the name of the user profile associated with the user password entered at sign-on. If the work station was signed on without a password, the name shown is the name specified in the job description referenced by the work station entry of the subsystem description.

D *NBR:* This is the job number assigned to the job by the system when it entered the system. No other job in the system has the same job number (see *Job Numbers* in this chapter).

E *Status of Job:* The status of the job is shown in two columns. The first column indicates:

- ACTIVE: The job is active.
- JOBQ: The job is on a job queue waiting to be processed.
- OUTQ: The output from a job is on an output queue waiting to be produced.
- TFRJOB: The job has been initiated but is currently residing on a job queue as a result of a TFRJOB command.
- SYSREQ: The interactive job was suspended through the use of the System Request key.
- CANCEL: The job is being canceled with the *IMMED option, or the subsystem the job is in is being terminated with the *IMMED option.

The second column indicates:

- HELD: The job has been held. The HELD for an OUTQ job means that output was held by HLDJOB SPLFILE(*YES).
- Blank: The job has not been held.

Both columns can indicate:

- JOBLOG PENDING: If the job log has not yet been written, this indication will appear until the job log can be written.
- FIN: The job has not properly completed its termination. It will be removed the next time the system is started or may be removed by the CNLJOB command.

F *Date/Time Entered System:* The system date and time the job entered the system.

Date/Time Started: The system data and time when processing began on the job.

G *Subsystem Name:* If the job is being processed, the name of the subsystem in which it is being processed is shown. If nothing is shown, the job is not being processed.

Ⓗ *Subsystem Pool ID:* The subsystem pool in which the job's current routing step executes.

Ⓘ *Submitted by Job/User/Nbr:* If the job was submitted by the Submit Job command.

Ⓙ *Type of Job:* The type of job being displayed (INTER, BATCH, AUTO, RDR, or WTR).

Ⓚ *Program Return Code:* The RPG return code.

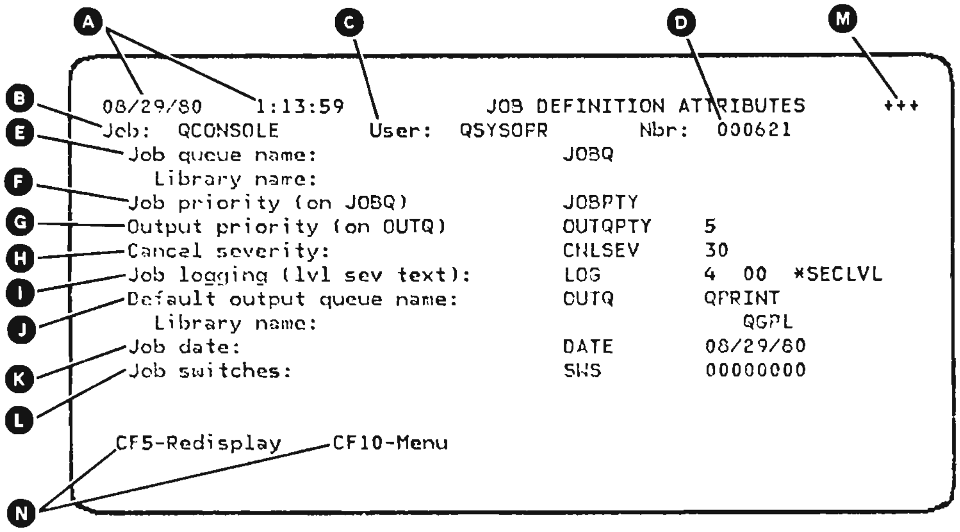
Ⓛ *Controlled Cancel Requested:* Indicates if a controlled cancel has been requested.

Ⓜ If you selected option 1 or took the default, three pluses (+++) will appear in this field and pressing the Enter key will take you to the next display.

Ⓝ *CF Keys:*

- CF5: Redisplay with updated values
- CF10: Return to Display Job Menu

Selection of option 3 from the job display menu produces the following Job Definition Attributes display:



```
 08/29/80     1:13:59               JOB DEFINITION ATTRIBUTES       +++
 Job:  QCONSOLE        User:  QSYSOPR        Nbr:  000621
   Job queue name:                      JOBQ
     Library name:
   Job priority (on JOBQ)               JOBPTY
   Output priority (on OUTQ)            OUTQPTY     5
   Cancel severity:                     CNLSEV      30
   Job logging (lvl sev text):          LOG         4  00  *SECLVL
   Default output queue name:           OUTQ        QPRINT
     Library name:                                     QGPL
   Job date:                            DATE        08/29/80
   Job switches:                        SWS         00000000


  CF5-Redisplay     CF10-Menu
```

A *Date/Time:* The date and the time shown are the job date and the system time when you requested the display.

B *Job Name:* This is the name of the job.

- For autostart jobs, it is the name provided through the autostart job entry of the subsystem description.

- For batch jobs, it is the name specified when the job was submitted.

- For interactive jobs, it is the name of the work station being used for the interactive session.

C *User:* This is the name of the user who submitted the job.

- For an autostart job, it is the name defined in the job description referenced by the autostart job entry.

- For a batch job, this is the name given in the job description. The job description is specified in the JOB command, such as on the first card of card input, or in the SBMJOB command that you or another user enter when you submit a job from a work station.

- For an interactive job, it is usually the name of the user profile associated with the user password entered at sign-on. If the work station was signed on without a password, the name shown is the name specified in the job description referenced by the work station entry of the subsystem description.

D *NBR:* This is the job number assigned to the job by the system when it entered the system. No other job in the system has the same job number (see *Job Numbers* in this chapter).

E *Job Queue Name:* The name shown is the job queue on which the job was placed when submitted. The name of the library where the queue is located is shown on the next line. If the job is not on the job queue, no job queue name is shown.

F *Job Priority:* The number shown indicates the priority of the job on the job queue. The lower the number, the higher the priority and the sooner the job will be processed. The highest priority is 1 and the lowest priority is 9. If the job is not on the job queue, no job priority is shown.

G *Output Priority:* The number shown indicates the priority of output files from the job on the output queue. The highest priority is 1 and the lowest priority is 9.

H *Cancel Severity:* If a message is generated during processing of the job that is equal to or greater than a specified severity level (see *Severity Codes* in Chapter 11), the job will be canceled. The severity level specified for this job is shown here.

I *Logging Message Level:* The number shown indicates which types of messages are to be included in the job log. The number can range from 0 (no messages) through 4 (all messages).

Logging Message Severity: Only messages having a severity equal to or greater than the level shown are sent to the job log.

Logging Text Level: The value shown indicates what message text levels are sent to the job log (see *Message Text* in Chapter 11).

- If *MSG is shown, only the first-level text is sent.
- If *SECLVL is shown, both first- and second-level text are sent.

The job start message (CPF1124) and job completion message always appear on the job log regardless of what logging level is specified.

J *Default Output Queue Name:* The name shown is the output queue on which any output files resulting from the job are to be placed if the output queue in the device file is (*JOB). If a queue name is in the file or override, then it goes to that queue. The name of the library where the queue is located is shown on the next line.

K *Job Date:* This is the date specified for the job in the JOB or SBMJOB command when the job was submitted.

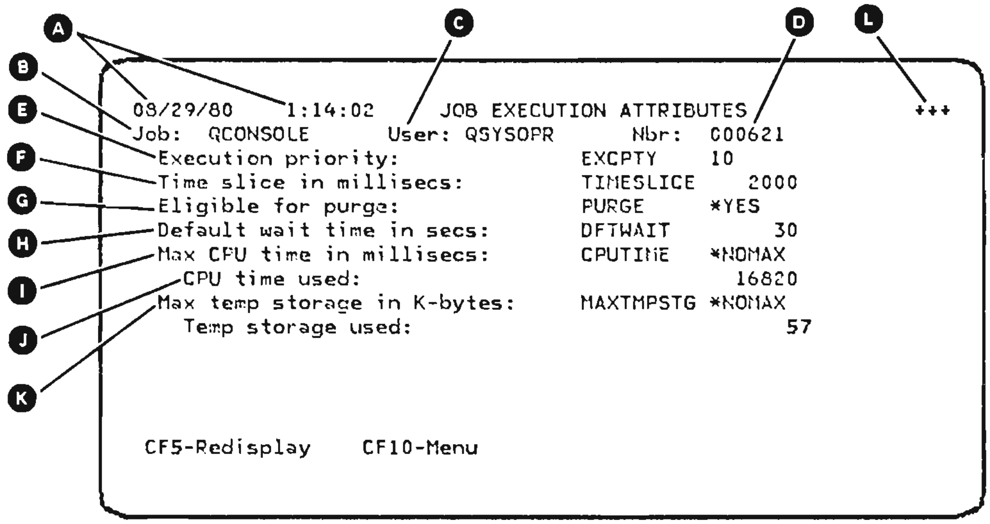
L *Job Switches:* These eight switch settings that affect the job can be specified when the job is submitted. The setting of each switch will be either '1' or '0'.

M If three plus signs appear in this field, you can press the Enter key and you will get the next job display.

N *CF Keys:*

- CF5: Redisplay with updated values
- CF10: Return to Job Display Menu

If the job to be displayed is active, selection of option 4 from the job display menu produces the following Job Execution Attributes display:



```
 08/29/80    1:14:02     JOB EXECUTION ATTRIBUTES                    +++
 Job:  QCONSOLE      User: QSYSOPR      Nbr:  C00621
   Execution priority:                 EXCPTY     10
   Time slice in millisecs:            TIMESLICE    2000
   Eligible for purge:                 PURGE      *YES
   Default wait time in secs:          DFTWAIT        30
   Max CPU time in millisecs:          CPUTIME    *NOMAX
     CPU time used:                                16820
   Max temp storage in K-bytes:        MAXTMPSTG *NOMAX
     Temp storage used:                                57



  CF5-Redisplay    CF10-Menu
```

A *Date/Time:* The date and the time shown are the job date and the system time when you requested the display.

B *Job Name:* This is the name of the job.

- For autostart jobs, it is the name provided through the autostart job entry of the subsystem description.

- For batch jobs, it is the name specified when the job was submitted.

- For interactive jobs, it is the name of the work station being used for the interactive session.

C *User:* This is the name of the user who submitted the job.

- For an autostart job, it is the name defined in the job description referenced by the autostart job entry.

- For a batch job, this is the name given in the job description. The job description is specified in the JOB command, such as on the first card of card input, or in the SBMJOB command that you or another user enter when you submit a job from a work station.

- For an interactive job, it is usually the name of the user profile associated with the user password entered at sign-on. If the work station was signed on without a password, the name shown is the name specified in the job description referenced by the work station entry of the subsystem description.

D *NBR:* This is the job number assigned to the job by the system when it entered the system. No other job in the system has the same job number (see *Job Numbers* in this chapter).

E *Execution Priority:* The execution priority for the job.

F *Time Slice:* The maximum amount of processor time used by this job before another job, waiting to use the same storage pool, can execute.

G *Eligible for Purge*: The job is eligible to be moved out of main storage at the end of a time slice or upon entering a long wait.

H *Default Wait Time:* The default maximum wait time.

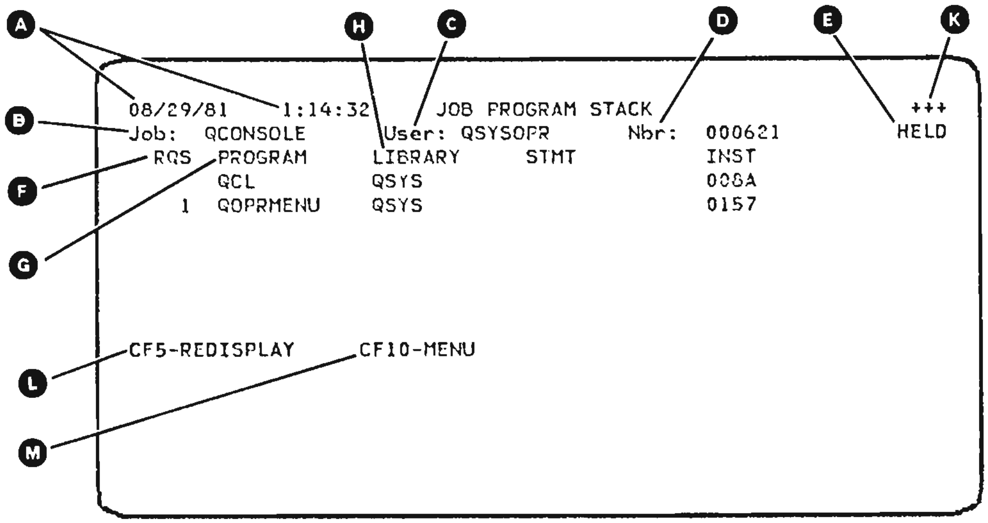
I *Max CPU Time:* The maximum CPU time that this job's current routing step can take to completely execute.

J *CPU Time Used:* The field shows the CPU time used by the job. If you suspect a job is looping or running too long, press the CF5 key to update the value. This will allow you to determine if the job is making normal progress.

K *Max Temp Storage:* The maximum amount of temporary storage this job's current routing step can use for processing.

L If three plus signs (+++) appear in this field, pressing the Enter key will display the next job.

Selection of option 5 from the job display menu produces the Program Stack Display:



```
08/29/81     1:14:32    JOB PROGRAM STACK                     +++
Job:  QCONSOLE         User: QSYSOPR     Nbr:   000621        HELD
  RQS  PROGRAM         LIBRARY      STMT        INST
       QCL             QSYS                     008A
    1  QOPRMENU        QSYS                     0157



CF5-REDISPLAY       CF10-MENU
```

A *Date/Time*: The date and the time shown are the job date and the system time when you requested the display.

🅑 *Job Name:* This is the name of the job.

- For autostart jobs, it is the name provided through the autostart job entry of the subsystem description.
- For batch jobs, it is the name specified when the job was submitted.
- For interactive jobs, it is the name of the work station being used for the interactive session.

🅒 *User:* This is the name of the user who submitted the job.

- For an autostart job, it is the name defined in the job description referenced by the autostart job entry.
- For a batch job, this is the name given in the job description. The job description is specified in the JOB command, such as on the first card of card input, or in the SBMJOB command that you or another user enter when you submit a job from a work station.
- For an interactive job, it is usually the name of the user profile associated with the user password entered at sign-on. If the work station was signed on without a password, the name shown is the name specified in the job description referenced by the work station entry of the subsystem description.

🅓 *NBR:* This is the job number assigned to the job by the system when it entered the system. No other job in the system has the same job number (see *Job Numbers* in this chapter).

🅔 *Status Field:* This field indicates if the job is being held or suspended because of a system request.

🅕 *RQS:* This field shows the request level of the request processing program.

🅖 *Program:* This is the program name.

🅗 *Library:* The library containing the program.

🅘 *Stmt:* This is the high-level language statement ID. This will be displayed only if the program is observable and not suspended and if the breakpoint offset mapping table exists.

🅙 *Inst:* This is the hexadecimal representation of the current or next MI instruction number in the program. This will not be displayed if the program is not observable or if the program is suspended.

🅚 If you selected option 1 or took the default from the job display menu, three pluses (+++) will appear in this field and pressing the Enter key will take you to the next display.

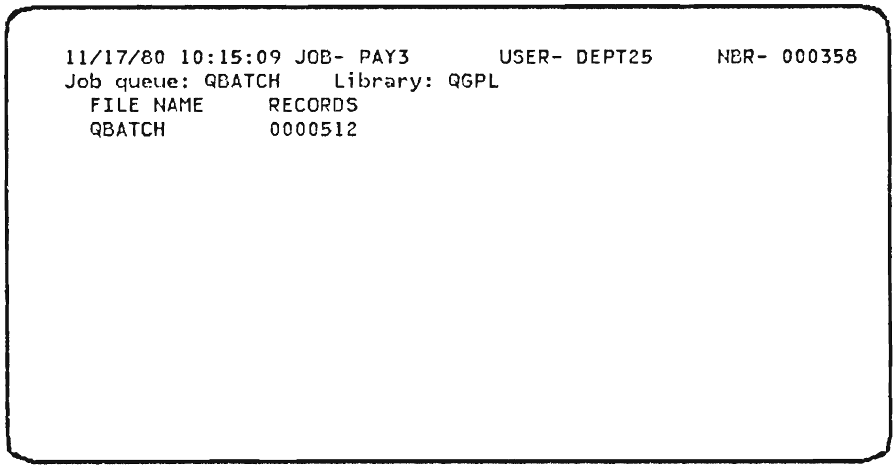
Ⓛ *CF5:* Pressing the CF5 key will redisplay the job Program Stack Display with updated information.

Ⓜ *CF10:* Pressing the CF10 key will return you to the job display menu.

If the job is held, suspended by system request, or cannot be interrupted by the request level, the HLL statement ID and MI instruction number are not displayed.

Selecting option 6 from the job display menu gives you a display of spooled input files for the job that:

- Is on the job queue
- Has not been placed on the job queue because of a Transfer Job (TFRJOB) command



```
11/17/80 10:15:09 JOB- PAY3       USER- DEPT25      NBR- 000358
Job queue: QBATCH    Library: QGPL
  FILE NAME     RECORDS
  QBATCH        0000512
```

The job name, user name, job number, and job queue shown on this display are the same as those shown on the other job displays. In addition, this display shows the following details:

- File Name: The name of each inline data file that was read in for the job.
  - If a name was specified for the data file (in the FILE parameter of the DATA command), the specified name is shown.
  - If no name was specified for the data file, QINLINE is shown.
- Records: The number of records contained in the inline data file.

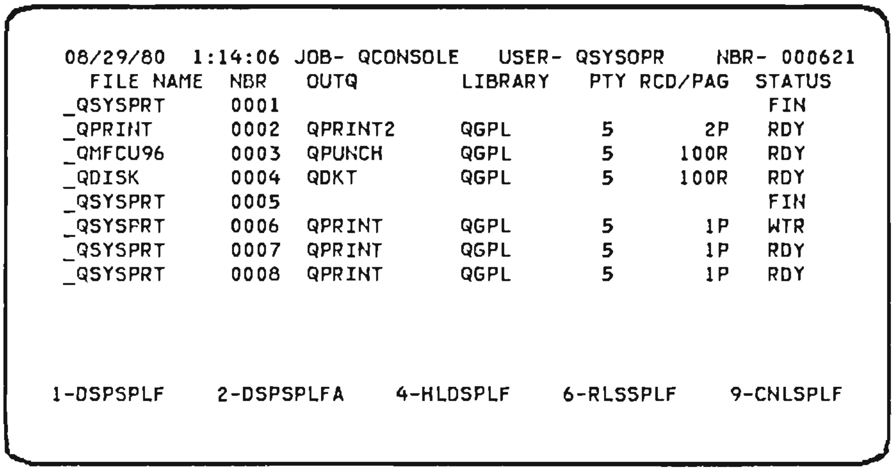
If there are more lines of input file information than can be shown on your screen at one time, a + appears to the right of the last line currently shown. You obtain the additional lines by using the Roll Up key.

Pressing the Enter key from this display takes you back to the job display menu.

Pressing CF2 from this display, if you used option 6 on the job display menu, takes you back to the job display menu.

Pressing CF2 from this display, if you used option 1 from the display job menu, takes you back to the job execution attributes display (option 4).

Interpreting the Output Files Display: If you select option 6 from the job display menu for a job that is being processed, or has completed processing and still has spooled output to be produced, you receive a display of spooled output files from the job, such as



```
08/29/80  1:14:06 JOB- QCONSOLE   USER- QSYSOPR    NBR- 000621
  FILE NAME  NBR   OUTQ        LIBRARY    PTY RCD/PAG  STATUS
_QSYSPRT     0001                                        FIN
_QPRINT      0002  QPRINT2     QGPL        5       2P   RDY
_QMFCU96     0003  QPUNCH      QGPL        5     100R   RDY
_QDISK       0004  QDKT        QGPL        5     100R   RDY
_QSYSPRT     0005                                        FIN
_QSYSFRT     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
```

The job name, user name, and job number shown on the top line are the same as shown on the job display. In addition, this display shows the following details about each output file for the job:

- File Name: The name of the spooled output file. Each spooled output file derives its name from the name of the device file that was used to create it.

- Nbr: The spooled file number assigned to the file when it was created.

- Outq: The name of the output queue on which the spooled output file has been placed.

- Library: The name of the library where the output queue is located.

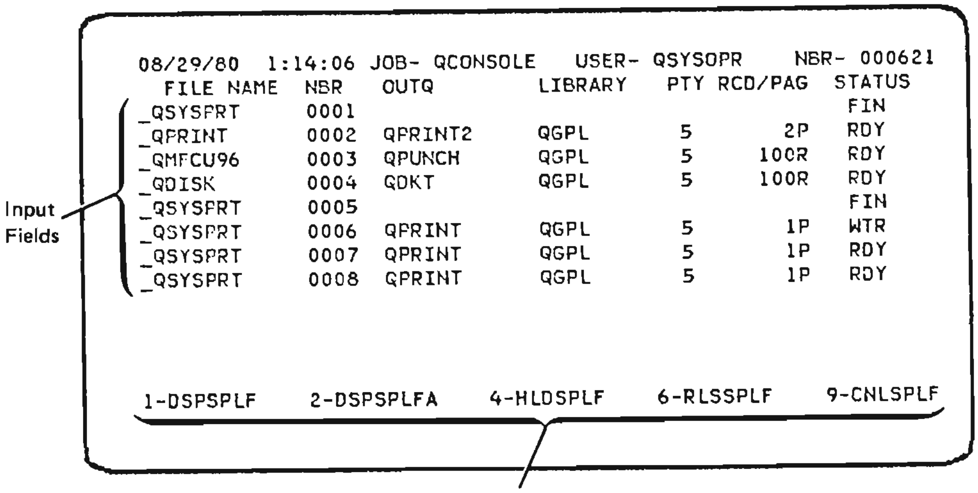
- Pty: The output priority assigned to the file. The output priority of each spooled output file for a job is the same as the output priority specified for the job.

- RCD/PAG: The number of output records followed by R for diskette and card files or pages, followed by a P for printer files. This field is blank if the file has not been closed.

- Status: The current status of the file. There are several possibilities:
  - RDY: The file is ready to be produced by a spooling writer. A writer will select a file from an output queue only if the file has a RDY status.
  - OPN: The file is still open; it is currently being processed and is not available to be produced by a writer.
  - CLO: The file is closed; it has completed processing, but SCHEDULE(*JOBEND) was specified in the associated device file and the job that produced the file is not yet finished.
  - HLD: The file has been held on the output queue by either HLDSPLF or by HLDJOB SPLFILE(*YES) and cannot be produced until it is released.
  - WTR: The file is currently being produced by a writer.
  - FIN: A writer has finished producing the file on an output device, or the file has been canceled.

If there are more lines of output file information than can be shown on your screen at one time, a + appears to the right of the last line currently displayed. You obtain the additional lines by using the Roll Up key.

Using the Output Files Display: An input field is provided to the left of each file shown on the output files display. At the bottom of the display, five command names are shown, each preceded by a number. By entering one of these numbers in the input field next to a file name, you can execute the command associated with the number; the command is executed only for the file where you entered the number.



```
08/29/80  1:14:06 JOB- QCONSOLE   USER- QSYSOPR    NBR- 000621
  FILE NAME   NBR   OUTQ        LIBRARY    PTY RCD/PAG  STATUS
_QSYSPRT      0001                                       FIN
_QPRINT       0002  QPRINT2     QGPL        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
```

To execute one of these commands for a particular file, enter the indicated number in the input field next to the file name.

If you key a number into several input fields and then press the Enter key, all of the associated commands are executed one at a time in the same order that the files are displayed. Note that if you enter a 1 (DSPSPLF) or a 2 (DSPSPLFA) for a particular file, you receive a separate spooled file display for that file. If you press the CF1 key while you are viewing a separate spooled file display (requested through the DSPSPLF or DSPSPLFA function on the output files display), you return directly to your basic working display, bypassing the output files display. If you had specified options on the output files display for files that follow the file you were displaying, none of those commands are executed.

The following chart summarizes the functions you can perform directly from the output files display:

To do this:	**Enter this:**	**For details, see:**
Display the data in a spooled output file (DSPSPLF command)	1	DSPSPLF command in *CL Reference Manual*
Display the output characteristics of a spooled output file (DSPSPLFA command)	2	*Determining Output Characteristics* in Chapter 10
Hold a spooled output file (HLDSPLF command)	4	*Holding a Spooled File* in Chapter 10
Release a spooled output file (RLSSPLF command)	6	*Releasing a Spooled File* in Chapter 10
Cancel (remove) a spooled output file (CNLSPLF command)	9	*Canceling a Spooled File* in Chapter 10

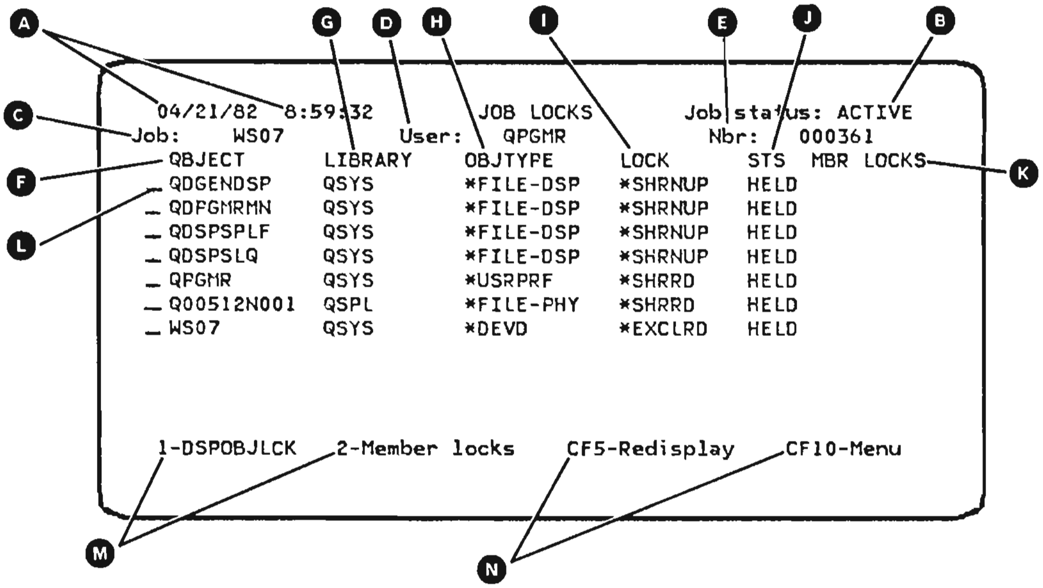
After the commands have been executed, the output files display is reshown with the current status of all output files on the display, including the files whose status you changed by executing the DSPSPLFA with CF3 option to the CHGSPLFA, HLDSPLF, RLSSPLF, or CNLSPLF command from the display. Any messages related to the execution of CHGSPLFA will be shown on the DSPSPLFA screen. Execution messages from other options will be shown on the output files display. If there are more messages that can be shown on your screen at one time, a + appears to the right of the last message currently displayed. You obtain the remaining messages by positioning the cursor on any of the message lines and pressing the Roll Up key.

When you are finished with the output files display, you return to your basic working display (such as the system operator menu) by pressing the Enter key or the CF1 key. If you press the CF1 key while you are viewing a separate spooled file display (requested through the DSPSPLF or DSPSPLFA function on the output files display), you return directly to your basic working display, bypassing the output files display. If you specified commands on the output files display for files that follow the file you were displaying, none of those commands are executed.

For an example of executing spooled file commands from a display, see *Output Queue Displays* in Chapter 10.

Interpreting the Job Locks Display

Selecting option 7 from the job display menu for an active job produces the Job Lock display in the following form:



```
04/21/82  8:59:32         JOB LOCKS          Job status: ACTIVE
Job:    WS07          User:   QPGMR            Nbr:   000361
  QBJECT        LIBRARY    OBJTYPE     LOCK       STS  MBR LOCKS
_ QDGENDSP      QSYS       *FILE-DSP   *SHRNUP    HELD
_ QDFGMRMN      QSYS       *FILE-DSP   *SHRNUP    HELD
_ QDSPSPLF      QSYS       *FILE-DSP   *SHRNUP    HELD
_ QDSPSLQ       QSYS       *FILE-DSP   *SHRNUP    HELD
_ QPGMR         QSYS       *USRPRF     *SHRRD     HELD
_ Q00512N001    QSPL       *FILE-PHY   *SHRRD     HELD
_ WS07          QSYS       *DEVD       *EXCLRD    HELD




 1-DSPOBJLCK  2-Member locks    CF5-Redisplay   CF10-Menu
```

A *Date/Time:* The date and time shown are the job date and the system time when you requested the display.

B *Job Status:* This is the status of the job. The status can be:

- ACTIVE: The job is eligible for execution.
- HELD: The job is suspended as the result of a HLDJOB.
- SYSREQ: The job is suspended as the result of system request.

C *Job:* This is the simple job name of the job for which locks are being displayed.

D *User:* The name of the user who submitted the job.

E *NBR:* The job number assigned to the job.

F *Object:* The name of the object for which locks are being displayed.

G *Library:* The library where the object resides.

H *Object Type:* The object type, such as *FILE or *PGM. If the object is a file, it is further defined, such as:

- *FILE–PHY: Physical file
- *FILE–LGL: Logical file
- *FILE–xxx: Device file, where xxx is the abbreviation of the specific device file type.

Ⓘ *Lock:* The type of lock being displayed. This can be:

- Exclusive (*EXCL): The object is reserved for the exclusive use of the requesting routing step; no other routing steps can use the object. However, if the object is already allocated to another routing step, your routing step cannot get exclusive use of the object.

- Exclusive allow read (*EXCLRD): The object is allocated to the routing step that requested it, but other routing steps can read the object if they request a shared for read lock state for the same object.

- Shared for update (*SHRUPD): The object can be shared either for update or read with another routing step. That is, another routing step can request either a shared for read lock state or a shared for update lock state for the same object.

- Shared no update (*SHRNUP): The object can be shared with another routing step if the routing step requests either a shared no update lock state, or a shared for read lock state. This lock state is appropriate when a routing step does not intend to change an object but wants to ensure that no other routing step changes the object.

- Shared for read (*SHRRD): The object canbe shared with another routing step if the routing step does not request exclusive use of the object. That is, another routing step can request an exclusive allow read, shared for update, shared for read, or shared no update lock state.

Ⓙ *STS:* The status of the lock:

- HELD: The lock is currently held by the job.

- WAIT: The job is in a synchronous wait for the lock.

- REQ: The job has a lock request outstanding for the object.

Ⓚ *MBR LOCKS:* For data base files there may be member and shared member locks in addition to file level locks. When data base locks exist, the MBR LOCKS column is included in the display and '2-member locks' is displayed at the bottom of the screen. The value in the MBR LOCKS column will be YES if additional locks exist for a data base file, WAIT if at least one of the additional locks is being waited on synchronously, and NO if there are additional lock requests. For object types other than data base files, the field will be blank. If the field is set to YES or WAIT, the user can display the shared member by entering a 2 in the input field to the left of the lock entry. The locks displayed does not include data base record locks.

Ⓛ *Input field:* Enter the option Ⓜ you want to take in the input field.

Ⓜ *Options:* The options available from this display are:

1 DSPOBJLCK: Entering option 1 in the input field next to the object name results in a display of all the lock requests in the system for that object. Held Locks and locks being waited for are displayed. This display does not show record locks for data base files.

2 Memberlocks: Entering a 2 in the input field next to the object name that is a data base file, results in a display of the lock entries for each member of the file.

Ⓝ *CF Keys:*

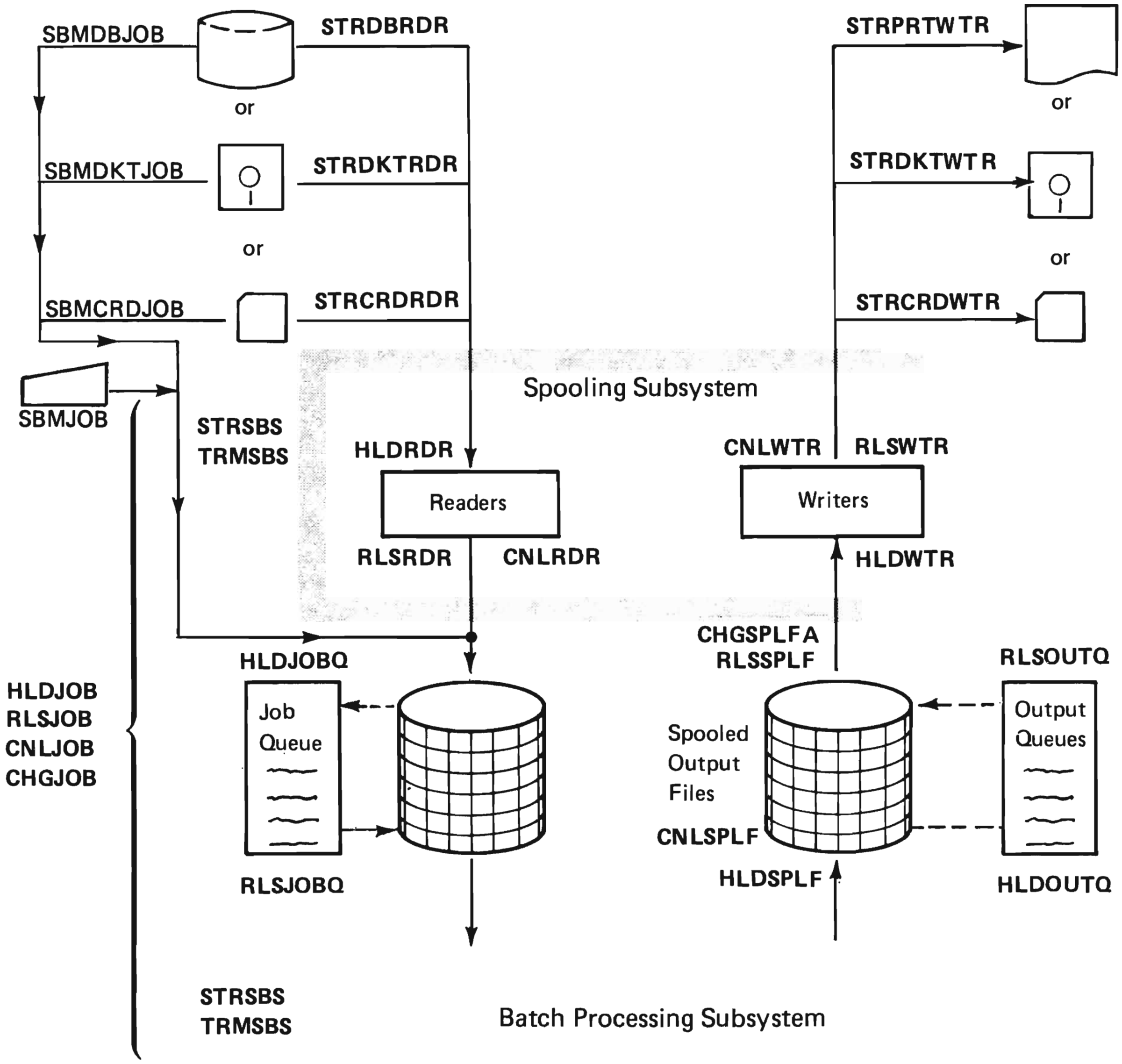
- CF5: Redisplays the display showing current values
- CF10: Returns you to the job display menu

  Note: CF10 is not valid if you obtained the Job Locks display from the Active Jobs display.

CONTROLLING JOB OPERATIONS

Once jobs, batch jobs in particular, are submitted to the system for processing by you or other users, you can control the timing of their processing by using the various hold, release, and cancel commands. By holding or canceling certain spooling functions, you can hold all jobs using those functions or make the functions unavailable for use by jobs. In addition, you can control the processing of a particular job, regardless of what stage it is in within the system, by using the Hold Job, Release Job, and Cancel Job command. The relationship of these job control commands is summarized below.

You can change the way a job is handled by the system by using the Change Job command.



Commands to Control Multiple Jobs

Once the system is prepared for operation, there are various commands you can use to control one or more jobs.

Start Subsystem (STRSBS) Command: When you enter this command, all jobs under control of the subsystem become available for processing and will be processed unless they are suspended by one of the following commands.

Terminate Subsystem (TRMSBS) Command: You can use this command to stop processing on all jobs under control of a subsystem.

Hold Reader (HLDRDR) Command: Using this command or option 4 from all the readers display, you can prevent jobs from being read from a specific device to a job queue for processing.

Hold Writer (HLDWTR) Command: You can use this command or option 4 from the all writers display, to prevent output files on an output queue from being produced on a specific output device.

Release Reader (RLSRDR) Command: Using this command or option 6 from the all readers display, you can release a reader that was held.

Release Writer (RLSWTR) Command: Using this command or option 6 from the all writers display, you can release a writer that was held.

Cancel Reader (CNLRDR) Command: You can use this command or option 9 from the all readers display, to make a reader unavailable for reading job input into the system.

Cancel Writer (CNLWTR) Command: You can use this command or option 9 from the all writers display, to make a writer unavailable for producing output from jobs.

Hold Job Queue (HLDJOBQ) Command: Using this command or option 4 from the all job queues display, you can suspend processing of all jobs on a specific job queue. Jobs may still accumulate on the queue, but none may be selected by a subsystem for execution.

Hold Output Queue (HLDOUTQ) Command: Using this command or option 4 from the all output queue display, you can suspend output operations on all files on a particular output queue. When you do this more files may not be selected by a writer, but the file the writer is currently processing will continue to be written and new files can accumulate on the queue.

Release Job Queue (RLSJOBQ) Command: This command or option 6 from the all job queues display, allows you to resume processing of jobs on a job queue that was held.

Release Output Queue (RLSOUTQ) Command: This command or option 6 from the all output queue display, allows you to resume output operations on files on an output queue that was held.

Hold Spooled File (HLDSPLF) Command: Using this command, option 4 on display specific writer, option 4 on display specific output queue, or option 4 on display of jobs output files, you can suspend output operations on a specific spooled file.

Release Spooled File (RLSSPLF) Command: This command, option 6 on display specific output queue, or option 6 on display of jobs output files, allows you to resume output operations on a spooled file that was held.

Cancel Spooled File (CNLSPLF) Command: Using this command, option 9 on display specific writer, option 9 on display specific output queue, or option 9 on display specific job output files, you can remove a specific spooled output file from the system.

Change Spool File Attributes (CHGSPLFA) Command: This command or CF3 from the DSPSPLFA display, allows you to move a file from one queue to another and change its SCHEDULE or other attributes.

Note: DSPSPLFA display is available from the display specific writer display, display specific output queue display, or display specific job output files which can be obtained from option 6 of the display job menu.

Most of these job and file control commands are available for execution (with default values for optional parameters) from displays.

Holding a Job

Using the Hold Job (HLDJOB) command or option 4 from the DSPSBS display, or option 4 from the DSPJOBQ display, you can stop processing on a job regardless of whether it is on a job queue, on an output queue, or active in the system. If you hold a job while it is active, all system resources being used in processing the job will also be held. Those system resources will be *unavailable* for other uses until the job is either canceled or released and completed. If you hold a job while it is on a job queue, it will be bypassed and any other jobs on the queue will be selected for processing instead. Similarly, if you hold output files from the job that are on an output queue, those files will be bypassed and other files on the queue will be produced instead. The job will not be selected from the job queue for processing, or its output files will not be produced from the output queue, until the job is released.

When entering the command, you must include the JOB parameter. There is only one additional parameter.

JOB Parameter: Use this parameter to specify the job to be held. If there is a possibility that another job exists on the system with the same name, you should also include the user name and job number, such as

```
JOB(PAY3.DEPT25.000358)
```

If you do not enter the fully qualified job name (job name, user name, and job number) and more than one job in the system has the same job name, or job name and user name, you receive messages that list the job name, user name, and job number for all jobs that have the name you specified. You can then correct the command to include the fully qualified name of the job you want.

SPLFILE Parameter: Use this parameter when you also want to hold spooled output files created by the job.

- If you do not enter the parameter, SPLFILE(*NO) is assumed and the files are not held.

- If you specify SPLFILE(*YES) all output files from the job are held. You should specify *YES if you want to suspend all operations on an active job or to hold all the files of a job on the output queues they are on. These files cannot be released separately using the RLSSPLF command or release options from the displays.

Examples:

1. To suspend processing of any requests associated with the job WS001, but not hold any output files that have already been produced from the job, enter

```
HLDJOB  JOB(WS001)
```

2. To suspend processing, including any output files produced, on the job STAR with the job number 111737 that was submitted by the user DAC, enter

```
HLDJOB JOB(STAR.DAC.111737)
  SPLFILE(*YES)
```

Releasing a Job

Using the Release Job (RLSJOB) command or option 6 from the DSPSBS display, or option 6 from the DSPJOBQ, you can resume processing of a job that was previously held, regardless of whether it had been active in the system, on a job queue, or on an output queue. When entering the command, specify the name of the job to be released in the JOB parameter, which is the only parameter.

If you do not enter the fully qualified job name (job name, user name, and job number) and more than one job in the system has the same job name, or job name and user name, you receive messages that list the job name, user name, and job number for all jobs that have the name you specified. You can then correct the command to include the fully qualified name of the job you want.

Examples:

1. To resume processing of the job INVTRY that was previously held, enter

```
RLSJOB JOB(INVTRY)
```

2. To resume processing of the job STAR of user DAC that was previously held, enter

```
RLSJOB JOB(STAR.DAC)
```

Canceling a Job

To remove a job from the system regardless of whether it is on a job queue, on an output queue, or active in the system, enter the Cancel Job (CNLJOB) command or option 9 from the DSPSBS, or option 9 from the DSPOUTQ display. There are five parameters; you must enter the JOB parameter.

JOB Parameter: Use this parameter to specify the name of the job to be canceled. Include the user name and job number if there is a possibility other jobs with the same name exist on the system. If you do not enter the fully qualified job name (job name, user name, and job number) and more than one job in the system has the same job name, or job name and user name, you receive messages that list the job name, user name, and job number for all jobs that have the name you specified. You can then correct the command to include the fully qualified name of the job you want.

OPTION Parameter: This parameter determines whether the job is canceled immediately or allowed time to perform cleanup.

- If you do not enter the parameter, OPTION(*CNTRLD) is assumed and the job is allowed time for cleanup activities before being canceled. The amount of time allowed is determined by the DELAY parameter.

- If you specify OPTION(*IMMED) the job is canceled immediately with only limited termination processing.

CAUTION
You should use the CNLJOB command with OPTION(*IMMED) only after a controlled cancel has been tried unsuccessfully, because an immediate cancel could result in damage to system or user objects being used for the canceled job.

DELAY Parameter: If you allowed the job time for cleanup by not entering the OPTION parameter, you use this parameter to specify the amount of time allowed.

- If you do not enter this parameter, 30 seconds are allowed.

- If you specify a time, such as DELAY(60) that number of seconds is allowed.

SPLFILE Parameter: This parameter determines whether spooled output files created by the job are to be retained for writing to an output device or deleted.

- If you do not enter this parameter, SPLFILE(*NO) is assumed and the output files are retained.

- If you specify SPLFILE(*YES) the files are deleted; however, the job log is retained.

LOGLMT Parameter: This parameter determines whether a limit is placed on the size of the job log for the canceled job. You would use this parameter, for example, to prevent the printing of a large job log from a job that is being (or was previously) canceled.

- If you do not enter this parameter, LOGLMT(*SAME) is assumed and the limit on the size of the job log is not changed; however, if no previous limit was specified, *NOMAX is assumed.

- If you specify LOGLMT(*NOMAX) no limit is placed on the job log.

- If you specify a particular limit, such as LOGLMT(50) the job log will be limited to the specified number of entries (messages). Note that if you specify LOGLMT(0) none of the job log will be printed.

You can enter the CNLJOB command and specify a LOGLMT even if the job was canceled previously. The limit you specify will have no effect if the job log was already printed. If the log entries are currently being written, or have not yet been written, to a spooled file, the writing of the entries will stop when the limit is reached, or immediately if the limit has already been reached.

Examples:

1. To cancel the job WS001 in a controlled manner such that it is allowed 30 seconds for cleanup before processing is stopped, but save any spooled output created by the job, enter

```
CNLJOB JOB(WS001)
```

2. To cancel the job STAR with the job number 111737 that was submitted by the user DAC, stopping all processing immediately and not print the job log, enter

```
CNLJOB JOB(STAR.DAC.111737) OPTION(*IMMED)
 LOGLMT(0)
```

3. To cancel the job INVTRY in a controlled manner such that it is allowed 120 seconds for cleanup before processing is stopped, and delete any spooled output created, enter

```
CNLJOB JOB(INVTRY) DELAY(120) SPLFILE(*YES)
```

Changing a Job

You can change the priority of a specific job, control which messages are sent to its job log, and assign the job output to a different output queue. The job can be on a JOBQ, on an output queue, or active. To change the job, enter the Change Job (CHGJOB) command and specify the job name in the JOB parameter. Six other parameters determine specific job characteristics. New characteristics remain in effect for the duration of the job or another CHGJOB command. If you do not enter a parameter, the job characteristic it defines is not changed.

JOB Parameter: Use this parameter to specify which job is to be changed.

- If you do not enter the parameter, the job being changed is assumed to be your own interactive job at the work station from which you are entering this command.

- Usually, you will want to enter this parameter and specify a job, such as JOB(WRITE.DEN.715475).

If other jobs in the system might have the same name, you should specify the fully qualified job name that includes the user name (DEN) and job number (715475) along with the job name (WRITE). If you do not enter the fully qualified job name and more than one job in the system has the same job name, or job name and user name, you receive messages that list the job name, user name, and job number for all jobs that have the name you specified. You can then correct the command to include the fully qualified name of the job you want.

JOBPTY Parameter: This parameter determines the priority of the job on the job queue, and is likely to be one of the most frequently used parameters of the command. By changing the priority, you can affect how soon a job is processed (see *Input/Output Priorities* in Chapter 10).

- If you do not enter the parameter, JOBPTY(*SAME) is assumed and the priority stays the same.

- If you specify a number from 1 through 9, such as JOBPTY(5) the priority is changed to that value. The lower the number, the higher the priority.

OUTPTY Parameter: This parameter determines the output priority of the output files produced for the job (see *Input/Output Priorities* in Chapter 10).

- If you do not enter the parameter, OUTPTY(*SAME) is assumed and the priority stays the same.

- If you specify a number from 1 through 9, such as OUTPTY(3) the output priority is changed to that value. The lower the number, the higher the priority.

LOG Parameter: This parameter determines what information is sent to the job log.

- If you do not enter the parameter, LOG(*SAME) is assumed and there is no change.

- If you enter the parameter, you must specify a set of three values, such as LOG(2 30 *MSG).

  Logging Level: The first value specifies the level of messages to be logged. The possible levels are:

  0 No data is to be logged.

  1 The only messages to be logged are those sent to the job's external message queue that have a severity equal to or greater than the severity specified in the second value. Messages of this type indicate when a job started, when it ended, and its status at completion.

  2 The following information is to be logged:
  - Logging level 1 information
  - Any requests for which high-level messages are issued with a severity code greater than or equal to the severity specified
  - Any high-level messages with a severity code greater than or equal to the severity specified

  3 The following information is to be logged:
  - Logging level 1 information
  - All requests
  - Any high-level messages with a severity code greater than or equal to the severity specified

  4 All requests and all messages, including messages related to lower level programs that were invoked.

  Message Severity: The second value specifies the severity of messages to be logged. The value can be from 00 through 99 (see *Severity Codes* in Chapter 11). Any message whose severity is equal to or greater than the value specified will be logged.

  Text Level: The third value specifies whether just the first-level text or both the first- and second-level text of each message is logged (see *Message Text Levels* in Chapter 11).
  - If you specify *MSG, only the first level text is logged.
  - If you specify *SECLVL, both levels of the text are logged.

OUTQ Parameter: Use this parameter when needed to specify that the spooling files from the processed job are to be placed on a different output queue or when they are created (opened) if they have OUTQ(*JOB).

- If you do not enter this parameter, OUTQ(*SAME) is assumed and the output queue is not changed.

- If you specify a different output queue name, such as OUTQ(QPRINT) spooled files created by the processed job will be placed on that queue.

You might use this parameter to put new spooled output files of the job:

- On an output queue from which a writer is currently producing output

- On an output queue reserved for low- or high-priority jobs

- On an output queue for a different type of device so that the output is redirected to that device

This command does not move files that already exist for the job (use CHGSPLFA to do that). It also does not affect files that have an output queue explicitly given in the device file or are overriden to a particular queue in the job.

Note: Redirecting output to the MFCU results in punch checks if the data being redirected contains characters other than the 64 characters supported by the MFCU. Job logs, in particular, should not be redirected to the MFCU. QPJOBLOG is OUTQ(*JOB) as shipped from IBM. OUTQ should not be directed to the MFCU at the time the job ends.

DATE Parameter: Use this parameter when needed to change the value to be used as the job date.

- If you do not enter the parameter, DATE(*SAME) is assumed and the date remains the same.

- If you specify a different value, such as DATE(120479) that value is assigned to the job date.

SWS Parameter: This parameter determines the settings of eight one-digit job switches. You should not change the settings unless you are instructed to do so.

- If you do not enter the parameter, SWS(*SAME) is assumed and the switch settings remain the same.

- If you specify an eight-digit code, such as SWS(00101010) the eight switches are changed to those settings. The only valid setting for each switch is 1 or 0. Usually the 1 indicates an on or true state and the 0 an off or false state. You can also specify some of the switch positions and put an X in the other positions to indicate that the settings for those positions should not be changed.

Examples:

1. To change the output priority of your interactive job so that its output is produced before any of the other jobs, whose priorities range from 4 through 7, enter

```
CHGJOB  OUTPTY(3)
```

2. To put the lowest priority on the job STAR with the job number 111737 that was submitted by the user DAC and change its output queue to QPRINT2, enter

```
CHGJOB  JOB(STAR.DAC.111737)  JOBPTY(9)  OUTPTY(9)
  OUTQ(QPRINT2)
```

3. To change the output priority to 4 for the job PAY2 of DEPT22, without changing the scheduling priority, and modify the job log such that both the basic and secondary text (*SECLVL) of diagnostic messages (message level 2) greater than severity 40 are sent to the log, enter

```
CHGJOB JOB(PAY2.DEPT22)  OUTPTY(4)  LOG(2  40
  *SECLVL)
```

Handling High-Priority Jobs

By using the job control commands described in this chapter, you can affect which jobs are completed first. You can, for example:

- Specify the highest scheduling and output priority when you submit a job using the SBMJOB command, or change the priorities using the CHGJOB command after it had already been submitted.

- Hold all other jobs on the same job queue or output queue or active in the system by using the HLDJOB command; although this could be a tedious operation if there are many other jobs on the queues or being processed, because you can only hold one job at a time using the HLDJOB command.

- Hold all other job queues and output queues by using the HLDJOBQ and HLDOUTQ commands.

- Cancel the writer producing other output on the device to be used for the high-priority job, then if the new job's output is all on a different output queue, restart the writer to the new job's output queue.

Procedures could also be established to allow for fast processing of high-priority jobs. For example:

- The highest priority levels (1, 2, and 3) could be reserved only for high-priority jobs.

- Special output queues for each type of output could be reserved for only urgent jobs. Writers to other output queues could then be canceled when a high-priority job is submitted, so that only output from the urgent queue is produced by a new writer.

Example:

You receive an urgent request to process a job in the fastest possible manner. The job will use the job description PAY in the library PAYROLL and be run under control of the program PAYREC. It will be processed in the QBATCH subsystem, as specified in the job description, but its output will be printed from the QPRINTS output queue instead of the QPRINT output queue that is specified in the job description.

Therefore, you do the following:

- Submit the job, specifying job and output priority of 3, which have been reserved for urgent jobs (priorities 1 and 2 are reserved for system functions on your system):

```
SBMJOB JOBD(PAY.PAYROLL) JOBPTY(3) OUTPTY(3)
 RQSDTA('CALL PAYREC') OUTQ(QPRINTS)
```

- Display the QPRINTS output queue, DSPOUTQ OUTQ(QPRINTS) which you find contains two other jobs, TSTPGM and PRTVOUCH.

- Hold the two other jobs:

```
HLDJOB JOB(TSTPGM) SPLFILE(*YES)

HLDJOB JOB(PRTVOUCH) SPLFILE(*YES)
```

- Cancel the writer QSYSPRT that is currently producing output from a different output queue on the system printer:

```
CNLWTR WTR(QSYSPRT) OPTION(*CNTRLD)
```

- Load the special forms required for the high-priority job into the system printer.

- Restart the writer from the QPRINTS output queue to the system printer:

```
STRPRTWRT DEV(QSYSPRT) OUTQ(QPRINTS)
```

By submitting the job with a job priority (JOBPTY) higher than any other job, you ensure that it will be taken off the QBATCH job queue for processing before any other job. Similarly, giving it the highest output priority (OUTPTY) ensures it will be produced before any other output on the queue.

However, if you start a writer from the queue and none of the output from the new job is ready to be produced, the writer will start producing the next highest priority output files. The output from the new job will then have to wait until the lower priority output that was already started is completed. To prevent this from happening, you hold the jobs currently on the queue or hold the files of these jobs using option 4 on the display writer display.

By canceling the writer that was currently producing output on the device to be used by the new job, you make the device available. The output files that were being produced remain on the output queue from which the writer had been producing the output. No information is lost.

By restarting the writer from the output queue on which the files from the new job are to be placed, you ensure that the device will start producing the output as soon as the output files are available.

HANDLING FILES

Data in the system is stored in files, each of which is an object that can be separately accessed. There may be times when you will be instructed to move files from one location to another or to reorganize the members of a file.

Copying Files

You copy files from one place to another by using the Copy File (CPYF) command. As its name implies, the CPYF command copies the data by duplicating it. The data remains in its original location and a copy is made of it in the new location. Using this command, you can copy from:

- One location to another in the data base
- A device to the data base
- The data base to a device
- A device to a device

A file located in the data base (auxiliary storage) is referred to as a *data base file.* When data is read from or written to system devices, it is handled as a file and referred to as a *device file.* The CPYF function can handle card (MFCU) and diskette device files.

When the CPYF command is used to copy data from one file to another, it can also be used to rearrange that data in the new file, or copy only some of the data (records) in a file.

Example:

You are instructed to copy all inventory records with inventory key numbers through 9999 to the diskette file QDKT whose label is INVEN1. You are to provide a list of the copied records. Therefore, you enter

```
CPYF FROMFILE(INVENTORY.RECORDS) TOFILE(QDKT)
 TOMBR(INVEN1) TOKEY(1 9999) PRINT(*COPIED)
```

Reorganizing Files

When records are deleted from data base files, the space in which they were located still remains. You can compress the files, and rearrange the storage sequence of members in a data base file, by using the Reorganize Physical File Member (RGZPFM) command. If you use this command, you should be given specific instructions on which parameters and values to enter.

If a system failure occurs while you are reorganizing a file, you will receive a message instructing you how to recover from the failure. It is important that you follow the instructions to avoid the possible loss of data from the file. Press the Help key to obtain the second-level text of the message, and do exactly what is indicated by the second-level text.

Example:

You are instructed to reorganize the member CLMBR1 of the file QCLSRC so that it is in keyed sequence. The file is to have sequence numbers resequenced and a null date field (000000) inserted in all records when the original member is replaced. Therefore, you enter

```
RGZPFM  FILE(QCLSRC)  MBR(CLMBR1)
 SRCOPT(*NUM *DATE)  KEYFILE(QCLSRC)
```

Chapter 13. Diskette, Tape, and Save/Restore Operations

Save/restore operations provide the means for you to transfer information between internal storage and external diskettes or magnetic tape. The saved information can also be used for backup. When information is saved, a copy of the information is written onto one or more diskettes or reels of tape. The diskettes or tape can then be removed and stored for future use on the same system or another system. When information is restored, a copy of that information is written from diskettes or tape into internal storage where it can be accessed by system users. The information restored could have been on internal storage previously, or it could be new information to be placed in the system for the first time.

You perform save/restore operations and prepare diskettes or tape for those operations by using commands.

Before handling diskettes, review the diskette handling precautions in Chapter 3.

DISKETTE CONSIDERATIONS

Save/restore operations involving diskettes use the diskette magazine drive. As described in Chapter 3, this device can hold two magazines containing 10 diskettes each, as well as individual diskettes in three I/O slots. Although the device can handle a variety of diskettes in various record formats, diskettes used for save/restore must be a specific type and conform to a specific format.

Commands involving diskettes contain parameters that specify which magazine and/or diskette position is to be used and, as necessary, which particular volume is to be used. The ways of identifying individual diskettes, slots, and magazines in commands are indicated in Figure 3/13-1.

The tracks or cylinders, on a diskette are divided into portions of equal size that are called sectors. Generally, for the programming support available with System/38, each sector can contain only one record; therefore, the maximum record length is the sector size. The single exception is for diskettes in the 256-byte format, where two 128-byte records can be read from (but not written to) each 256-byte sector. The sector format is established when the diskette is initialized (see *Initializing Diskettes* in Chapter 13). For additional information on diskette formats, see the section on *Diskette Addressing and Layout* in *The IBM Diskette General Information Manual.*

The System/38 diskette magazine drive can use the following types of diskettes:

- IBM diskette 1 (single-sided)

- IBM diskette 2 (double-sided)

- IBM diskette 2D (double-sided, double density)

The CPF programming available with System/38 supports all three types of diskettes for the basic exchange of data. The sector size must be 128 bytes for diskettes 1 and 2, and 256 bytes for diskette 2D. To format a diskette 1 or 2, you specify the FMT(*DATA) option of the Initialize Diskette (INZDKT) command. Diskettes with this format can be written with Basic Data Exchange files. To format a diskette 2D, you specify the FMT(*DATA2) option of the INZDKT command. Diskettes with this format can be written with Type H Basic Exchange files.

Save/restore operations require a diskette 2D that has a sector size of 1024 bytes. To format a diskette for save/restore, you specify the FMT(*SAVRST) option of the INZDKT command.

It is not necessary to initialize a diskette before each use. Each diskette is initialized to a specific sector size before it is shipped. If the sector size is correct for the intended use, you can use the Rename Diskette (RNMDKT) command to write a volume identifier on the diskette when needed. Diskettes used for save/restore operations, for example, require a specific type of volume identifier. If a diskette has the correct sector size for the intended use, but contains unneeded files, you can remove the files by using the Clear Diskette (CLRDKT) command. You can save about 2 minutes per diskette by using either of these commands instead of the INZDKT command. The initialization process involves rewriting every sector on every track, which clears any files from the diskette, and allows the option of adding specific identifiers on the diskette.

The procedure for using these commands are discussed under *Preparing Diskettes* in this chapter.

Volume Label: All diskettes must have a volume label, which identifies them to the system. A volume label is given to a diskette when you initialize it or rename it. The identifier that is used for the volume label of each save/restore diskette should be unique (not more than one diskette with the same identifier) and must follow these conventions:

- The volume identifiers of all diskettes used in a magazine must begin with the same combination of letters and numbers up to a maximum of five characters. When a magazine of diskettes is initialized, the user can specify a maximum of five characters and the system adds a final character that is determined by the position of each diskette in the magazine:

Diskette Location	**Final Character**
1 through 9	1 through 9
10	0

  The number that is the final character in the volume identifier indicates the position of the diskette in the magazine. The diskette must not be moved to another position in a magazine while it has that identifier. That is, volume ABCDE1 must be in position 1, volume ABCDE2 must be in position 2, and volume ABCDE0 must be in position 10 in the same magazine (see Figure 13-1).

- The volume identifier of each diskette placed in an I/O slot, S1, S2, or S3, can be any combination of letters and numbers up to a maximum of six characters. When initialized, the user can specify a maximum of six characters. The system does not add the last character. If a diskette was initialized in a magazine and you use it in an I/O slot, you must specify the entire volume identifier, including the final character that was added by the system to indicate the magazine position.

When specifying the volume label, you have the option of indicating an owner identifier along with the required volume identifier. The owner identifier can be any combination of letters and numbers up to a maximum of 14.

Diskettes for Save: For a Save Object or Save Library operation, the save will begin in the available space on the first diskette, but all the following diskettes must be cleared unless CLEAR(*YES) is specified in the command. For a Save System operation, all diskettes used must be cleared unless CLEAR(*YES) is specified on the SAVSYS command. (See *Clearing Diskettes* in this chapter.)

Indicating Diskette Positions in Commands

When entering a diskette or save/restore command, you specify the diskette position or positions in the *location* parameter. If you want to specify that only a particular diskette volume is to be used, you identify that volume in the *volume* parameter. The diskette positions as you would enter them in a command are summarized in Figure 13-1. The format of volume identifiers for individual diskettes in a magazine is also indicated.

The following topics summarize the use of the location and volume parameters of the commands described in this chapter.

Location (LOC) Parameter

This parameter determines which magazines or I/O slots are to be used and, in some commands, which diskettes in those magazines or slots are to be used. Depending on the command, you can specify up to three values in each LOC parameter, such as

```
LOC(*M1 3 9)
      | | |
      A B C
```

A *Magazine/Slot*: The first value specifies the magazines or I/O slots to be used. The possible values are:

*M12	Use magazine 1 followed by magazine 2
*M1	Use magazine 1 only
*M2	Use magazine 2 only
*S1	Use slot 1 only
*S2	Use slot 2 only
*S3	Use slot 3 only
*S12	Use slot 1 followed by slot 2
*S23	Use slot 2 followed by slot 3
*S123	Use slot 1 followed by slot 2 followed by slot 3

Not all commands allow all of these alternatives. Some commands allow only a specific magazine or I/O slot, but not a combination of magazines or slots.

B *Starting Position*: The second value specifies where in a magazine, magazines, or multiple I/O slots the operation will begin. If you do not enter this value, the operation will begin with the leftmost diskette of the magazines or slots specified in the first value A, such as (see Figure 13-1):

- Position 1 of magazine 1 if you specified *M12 or *M1 in the first value

- Position 1 of magazine 2 if you specified *M2 in the first value

- I/O slot 1 if you specified *S12 or *S123 in the first value

- I/O slot 2 if you specified *S23 in the first value

Provided you specified a magazine or magazines in the first value A, you can use this second value to specify that the operation is to start with a diskette other than the first (leftmost) diskette. For example, if you specified 3 for the second value, the operation would begin with:

- Position 3 of magazine 1 if you specified *M12 or *M1 in the first value

- Position 3 of magazine 2 if you specified *M2 in the first value

Ⓒ *Ending Position*: The third value specifies where in a magazine, magazines, or multiple I/O slots the operation will end. If you do not enter this value, the operation will end with the rightmost diskette of the magazines or slots specified in the first value Ⓐ, such as (see Figure 13-1):

- Position 0 (10) of magazine 2 if you specified *M12 or *M2 in the first value
- Position 0 (10) of magazine 1 if you specified *M1 in the first value
- I/O slot 3 if you specified *S23 or *S123 in the first value
- I/O slot 2 if you specified *S12 in the first value

If you specify *ONLY for this value, only the single diskette in the position specified in the second value Ⓑ will be used.

Provided you specified a magazine or magazines in the first value Ⓐ, you can use this third value to specify that the operation should end with a diskette other than the last (rightmost) diskette. For example, if you specified 8 for this value, the operation would end with:

- Position 8 of magazine 2 if you specified *M12 or *M2 in the first value
- Position 8 of magazine 1 if you specified *M1 in the first value

Save/restore commands allow you to specify the location in the diskette magazine drive and a starting position in magazines or multiple slots, but not an ending position. When you save objects or libraries, the ending position is determined by the amount of space required on the diskettes to store the specified information. If additional diskettes or magazines of diskettes are required to complete a save or restore operation, you will receive a message (in the system operator message queue) to load them in the diskette magazine drive.

In save/restore commands, you can also specify a value for the starting position that instructs the system to start with the diskette in the location where the last save/restore ended (*CURRENT), or to search for the first diskette (*SEARCH) that is cleared or has available space (save operation) or contains the specified libraries or objects (restore operation).

Volume (VOL) Parameter

This parameter determines whether a diskette having a specific volume identifier (volume label) is to be used.

- If you do not enter this parameter, the system will
  - Use the diskette currently located in each position you specified in the LOC parameter if the default value is either *LOC or *NONE.
  - Use the diskette volumes mounted in the location specified in the LOC parameter, starting with the diskette position specified, if the default value is *MOUNTED.
  - Use the diskette containing the most recently saved version of an object or library if the default value is *SAVVOL (occurs only in commands that restore objects or libraries to the system).

- If you enter the parameter and specify one or more volume identifiers, such as VOL(ABCDE3) the system will use only the specified volumes and will send you a message to mount those volumes if they are not in the positions specified in the LOC parameter. One volume identifier should be supplied for each magazine/slot used in the save/restore operation. Note, however, that when magazines of diskettes are used in save/restore operations, you can specify only the overall volume identifier of all diskettes in the magazine (up to five characters), not the specific volume identifier of an individual diskette in the magazine. For example, you can specify a volume identifier of ABCDE but not ABCDE3, where 3 is the final character assigned by the system to indicate the position of the diskette in the magazine (see the discussion of volume labels under *Diskette Requirements for Save/Restore* in this chapter).

DETERMINING DISKETTE CONTENTS

Two displays are available that you can request to determine what information is contained on a diskette. The *diskette volume display* provides only the information contained on the diskette labels and indicates the amount of space remaining on the diskette where additional files could be added. You will most likely want to request the *save/restore volume display*, which lists the objects or libraries contained on each diskette and indicates when they were saved.

Obtaining the Displays

To obtain either display, enter the Display Diskette (DSPDKT) command.

LOC Parameter: The LOC parameter is required. You must at least specify a magazine, I/O slot, or combination of magazines or slots. All of the values discussed for the LOC parameter in *Diskette Considerations* are valid, including the values that specify the starting and the ending diskette positions.

The remaining parameters are optional; however, if you want the save/restore volume display, you must include the DATA parameter.

LABEL Parameter: This parameter determines which files are to be displayed. An identifier for each file on a diskette is stored in the volume label area of the diskette. Using this parameter, you can specify whether information is to be displayed on all stored files of each diskette specified in the LOC parameter or only on a particular file.

- If you do not enter this parameter, LABEL(*ALL) is assumed and all files are displayed.

- If you specify a particular identifier, such as LABEL(SAVE) information is displayed only for that identifier.

DATA Parameter: This parameter determines which display you receive.

- If you do not enter this parameter, DATA(*LABELS) is assumed and you receive the diskette volume display.

- If you specify DATA(*SAVRST) you receive the save/restore volume display.

OUTPUT Parameter: This parameter determines how the information is provided.

- If you do not enter this parameter, the information is displayed on your work station screen.
- If you specify OUTPUT(*LIST) the information is written to a spooled output file and placed on an output queue for printing on the system printer.

Examples:

1. To display only the volume and file labels of all diskettes in both magazines and have that display appear on the screen of your work station, enter

```
DSPDKT LOC(*M12)
```

2. To display at your local work station the save/restore information for the file named TOTAL on the third diskette of magazine 1, enter

```
DSPDKT LOC(*M1 3 *ONLY) LABEL('TOTAL.Q001')
  DATA(*SAVRST)
```

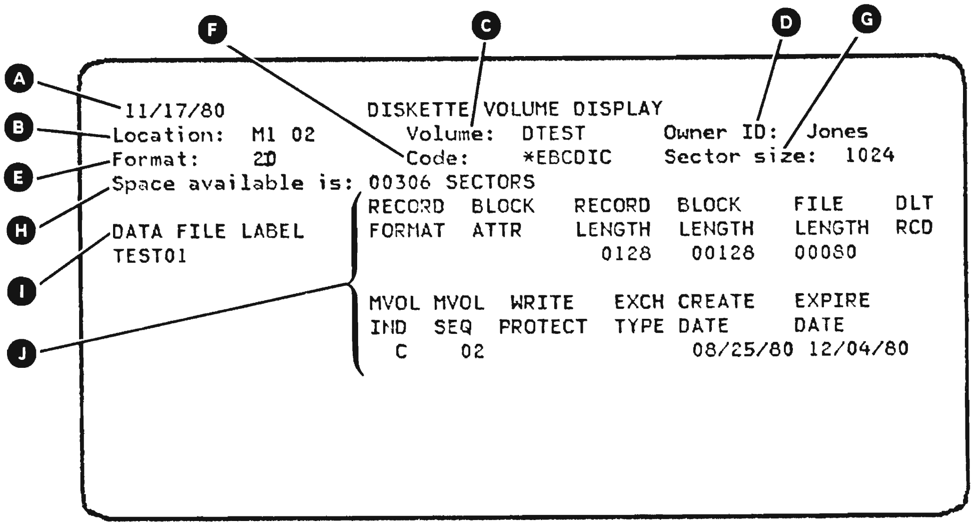
Note: The qualified save/restore file label 'TOTALQ001' must be in quotes and in uppercase.

3. To obtain a printout of all save/restore information from the third diskette of magazine 1 through the seventh diskette of magazine 2, enter

```
DSPDKT LOC(*M12 3 7) DATA(*SAVRST) OUTPUT(*LIST)
```

Interpreting the Diskette Volume Display

If you enter the DSPDKT command without the DATA parameter, you receive a display of the form:



```
11/17/80                  DISKETTE VOLUME DISPLAY
Location:  M1 02             Volume:  DTEST       Owner ID:  Jones
Format:    2D                Code:    *EBCDIC     Sector size:  1024
Space available is:  00306 SECTORS
                          RECORD  BLOCK   RECORD  BLOCK    FILE    DLT
DATA FILE LABEL           FORMAT  ATTR    LENGTH  LENGTH   LENGTH  RCD
TESTO1                                    0128    00128    00080

                          MVOL MVOL  WRITE    EXCH CREATE   EXPIRE
                          IND  SEQ  PROTECT   TYPE DATE     DATE
                           C    02                 08/25/80 12/04/80
```

A *Date*: The date shown is the date when you requested the display.

B *Location*: The identifier shown indicates the location of the diskette volume in the diskette magazine drive (see Figure 13-1).

- If the diskette is in one of the three I/O slots, the identifier will be either S1, S2, or S3.
- If the diskette is in a magazine, the identifier will be M1 or M2 plus a number from 1 to 10 that indicates the position in the magazine.

C *Volume*: This field gives the volume identifier of the diskette whose file label is being displayed.

D *Owner ID*: The characters shown identify the owner of the diskette.

E *Format*: This field indicates the type of the diskette. The type can be:

- 1: One-sided diskette
- 2: Two-sided diskette
- 2D: Two-sided double density

F *Code*: This indicates the code used on the diskette. The value can be EBCDIC or ASCII.

G *Sector Size*: This field indicates the number of bytes per sector that was specified when the diskette was initialized.

H *Space Available*: This field shows the number of sectors of space available after the last usable (unexpired) file on the diskette.

I *Data File Label*: The label shown is the name that was assigned to the file when it was created.

J *Label Characteristics*: For each file, the following label information is displayed:

- Record Format: An F in this field or a blank field indicates that the records are fixed length. The length is shown in the RECORD LENGTH field.

- Block Attr: If nothing is shown in this field, the records are unblocked and unspanned. The other possibilities are:

  B = records blocked, unspanned
  S = records unblocked, spanned
  R = records blocked, spanned

- Record Length: The number of bytes in each record. If nothing is shown, the record length is the same as the block length.

- Block Length: The number of bytes in each block (sector). For save/restore files, the length is 1024.

- File Length: The number of sectors of data contained in the file.

- Dlt Rcd: This is the logical record delete indicator and applies to only I-format files. If the field is blank, the file does not contain logically deleted records. If a character is displayed, that character is the last position in a logically deleted record.

- Mvol Ind: If nothing is shown, the data file is complete on this volume. The other possibilities are:

  C = continued on another volume
  L = last volume of multivolume data file

- Mvol Seq: If a number is shown, it indicates the order of the volume relative to the first volume of a multivolume data file.

- Write Protect: If a P is shown, the data file can be read but not altered. If nothing is shown, there is no write protection.

- Exch Type: Indicates the type of data exchange format.

  Blank= Basic Exchange Format (Type 1 or 2; 128 bytes per sector)
  H = (Type 2D; 256 bytes per sector)
  I = (Type 1 or 2; 128, 256, or 512 bytes per sector) or
  (Type 2D; 256, 512, or 1024 bytes per sector)
  E = Save/Restore (To determine the diskette
  contents, use DSPDKT with the DATA(*SAVRST) parameter)

  Note: This could also be an exchange type unknown to System/38.

- Create Date: The date shown is the date the file was created. If nothing is shown, the create date is not significant.

- Expire Date: The date shown is when the file can be deleted. The word *PERM instead of a date indicates a permanent file. If nothing is shown, the file has already expired.

You can display additional files by pressing the Enter key. Press the CF1 key to return to the display from which you requested the label information.

Interpreting the Save/Restore Volume Display

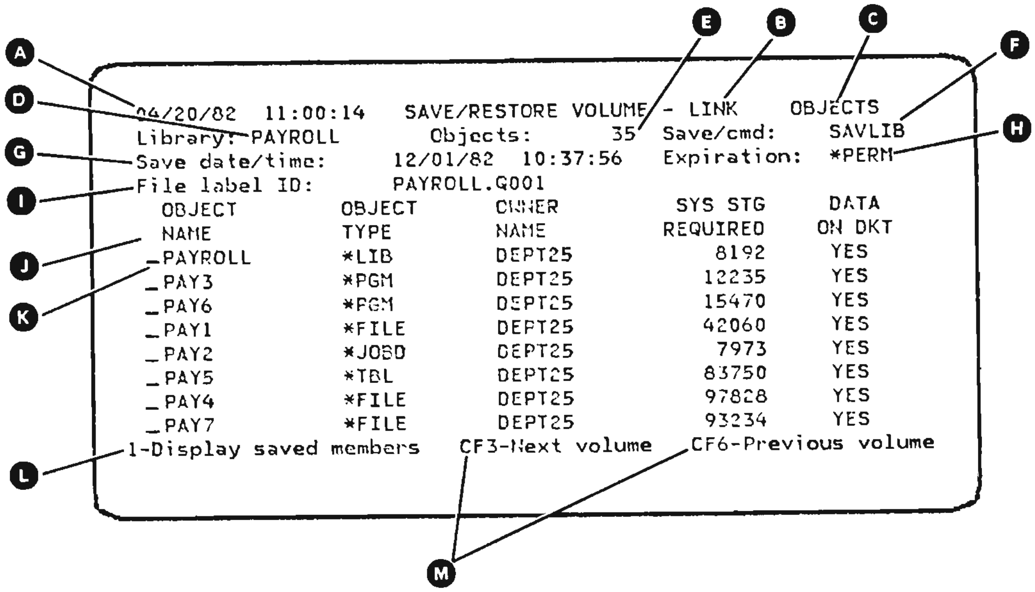
If you specify DATA(*SAVRST) when you enter the DSPDKT command, you receive one or more displays that identify the libraries, objects, and data base file members saved on each diskette for each file specified in the command. The type and number of displays you receive depends on whether you specified LABEL(*ALL) or a single file identifier in the command, whether one or multiple files are contained on the specified diskettes, and the type of command that was used to save the libraries or objects on the diskettes (see *Saving Information* in this chapter).

- You receive an object-oriented display that provides details on all objects in a library if
  - The objects on the specified diskettes were saved by the SAVOBJ or SAVCHGOBJ command or were in a single library that was saved individually by the SAVLIB command.

- You receive two types of displays if you specified LABEL(*ALL) and
  - The objects on the specified diskettes were saved by the SAVLIB command with the *NONSYS option (which saves all user libraries including QGPL).
  - The objects on the specified diskettes were saved by the SAVSYS command (which saves system libraries and objects).

  You first receive a library-oriented display that lists the libraries saved. Following the library-oriented display, you can obtain object-oriented displays that list details on the objects in each library.

The following display is an example of an object-oriented display. The library-oriented display is similar, except that it lists libraries instead of objects.



```
04/20/82  11:00:14   SAVE/RESTORE VOLUME - LINK     OBJECTS
Library: PAYROLL        Objects:        35  Save/cmd:      SAVLIB
Save date/time:      12/01/82  10:37:56  Expiration:    *PERM
File label ID:       PAYROLL.G001
  OBJECT            OBJECT        OWNER          SYS STG      DATA
  NAME              TYPE          NAME           REQUIRED     ON DKT
_PAYROLL            *LIB          DEPT25             8192      YES
_PAY3               *PGM          DEPT25            12235      YES
_PAY6               *PGM          DEPT25            15470      YES
_PAY1               *FILE         DEPT25            42060      YES
_PAY2               *JOBD         DEPT25             7973      YES
_PAY5               *TBL          DEPT25            83750      YES
_PAY4               *FILE         DEPT25            97828      YES
_PAY7               *FILE         DEPT25            93234      YES
1-Display saved members   CF3-Next volume   CF6-Previous volume
```

Ⓐ *Date/Time:* The date shown is the date and time when you requested the display.

Ⓑ *Volume:* The volume identifier of the diskette you requested is shown following the title of the display.

Ⓒ *Display Type:* The words OBJECTS indicate that this is an object-oriented display containing details on the objects saved from a specific library. A library-oriented display that contains a list of libraries would be indicated by the words LIBRARIES. A member-name-oriented display that contains a list of members in a specific file, would be indicated by the word MEMBERS.

Ⓓ *Library:* This is the name of the library where the objects on the volume were saved.

Ⓔ *Objects*: This is the number of objects in the file.

Ⓕ *Save Cmd*: The name shown is that of the CL command or other command that was used to save the objects.

- If SAVLIB is indicated, the entire library was saved, including all objects in the library and their PC (programming change) status. Each object saved is listed on the display. You can restore all objects in the library and their PC status to the system by using the Restore Library command. Or you can restore one or more individual objects in the library without their PC status by using the Restore Object command.
- If SAVOBJ or SAVCHGOBJ is indicated, it is likely that not all objects in a library were saved. You can restore those objects listed by using the Restore Object command.
- If SAVSYS is indicated, basic system information was saved. This information can only be restored by reinstalling CPF.
- If CPF is indicated, the objects or libraries were not saved directly by an individual save command, but by an internal CPF program that invoked the save command. You cannot restore objects from this file.

Ⓖ *Save Date/Time*: The date and the time shown are the system date and the time when the objects or libraries were saved.

Ⓗ *Expiration*: This is the expiration date for the diskette file.

Ⓘ *File Label ID*: The name shown is the file label on the diskette that is currently being displayed. The label consists of the library name followed by a .Q followed by a 3 digit sequence number.

J *Objects/Libraries List*: Basic information is listed here on objects or libraries saved.

- An object-oriented display (as in the example shown) will list each object saved or all the objects of the library saved. For each object listed, the display will show:
  - The name of the object
  - The type of object

*CLS	Class
*CMD	Command
*DTAARA	Data area
*EDTD	Edit description
*FCT	Forms control table
*FILE	File
*JOBD	Job description
*JRN	Journal
*JRNRCV	Journal receiver
*MSGF	Message file
*PGM	Program
*PRTIMG	Print image
*SBSD	Subsystem description
*SSND	Session description
*TBL	Table

If the SAVSYS command was used to save the system on the diskette volume being displayed, the following object types can also be displayed:

*CUD	Control unit description
*DEVD	Device description
*LIND	Line description
*USRPRF	User profile

  - Further defines the object type by subtype, such as PGM RPG.
  - The name of the object owner
  - The size of the object in terms of how many bytes it occupied on the system at the time of the save
  - Data on DKT: This field indicates if the data portion of the object exists on diskette. Program device files, and journal receivers whose storage had been freed before the save operation will have only the definition of the objects saved on diskette.

- A library-oriented display would list only the names of the libraries saved.

K *Input Field:* This is displayed for data base files only and is the one-character input field where you enter the option to display the members of the data base file.

Ⓛ *Option:* The option is:

1 Display Saved Members: All saved members are displayed.

Ⓜ *CF Keys:*

- CF3: Display next diskette
- CF6: Display previous diskette

To return to the display from which you requested the save/restore information, press the CF1 key.

TAPE CONSIDERATIONS

Certain tape functions are considered long-running functions. These are:

- INZTAP CLEAR(*YES)
- Rewinding a tape, depending on current tape position
- Searching for a particular file on a mounted tape when there is a considerable distance between the current tape position and the file being searched for

If a job or program that uses tape is abnormally terminated during a long running tape function, CPF will wait up to four minutes for these functions to complete. If the four-minute limit is reached before the function completes, a device-description partial-damage error will occur (CPF8105 message). The recovery action for CPF8105, including varying off and on, must be followed before the tape drive may be used again.

The Reset key on the tape drive can be used to terminate the long-running function within the four-minute time period to avoid the partial damage error and to minimize the termination processing time.

If the tape movement still continues after the four-minute limit, it may be terminated by using the Reset key on the tape unit. The tape drive or the system must never be powered down with a tape in the loaded state, as this may damage the tape and make it unreadable.

DETERMINING TAPE CONTENTS

You can obtain two displays that summarize the information contained on a reel of magnetic tape. The *tape volume display* shows the volume label and data file label information on the tape. The *save/restore volume display* lists the objects or libraries contained on the tape and indicates when they were saved; it is the same as the display you receive when you request save/restore information on diskettes. The two displays can be obtained only for tape having standard IBM tape labels. You will receive an error message if you attempt to display a tape that is unlabeled or contains nonstandard labels.

Obtaining the Display

To obtain either display, enter the Display Tape (DSPTAP) command. You must specify at least the DEV parameter. The remaining parameters are optional; however, you must include the DATA parameter if you want the save/restore volume display.

DEV Parameter: Use this parameter to specify the name of the magnetic tape unit that contains the tape files you want displayed, such as

```
DEV(QTAPE1)
```

LABEL Parameter: Use this parameter to specify which files on the tape are to be displayed.

- If you specify LABEL(*ALL) information is displayed for all files on the tape mounted in the specified device.

  If there is not room for all files on the first display, +++ will appear in the upper right corner of the first display. To obtain additional file information, press the Enter key. When all of the file information has been displayed, the +++ disappears and pressing the Enter key returns you to the display from which you requested the tape file display. Pressing the CF1 key at any time during the tape display returns you to the display from which you requested the tape file display.

- If you specify a data file identifier, such as LABEL(SAVE1) information is displayed only for the specified file.

SEQNBR Parameter: Use this parameter when you need to specify the sequence number of the file to be displayed.

- If you do not enter the parameter, SEQ(1) is assumed and the file is assumed to be the first file on the tape. If you also specified LABEL(*ALL) for the LABEL parameter, all files on the tape are displayed, starting with the first file.

- If you specify a sequence number, such as SEQNBR(3) the file in the specified position on the tape is displayed. If you specified *ALL for the LABEL parameter, all files following the file in the specified position are also displayed.

DATA Parameter: This parameter determines which type of information is displayed.

- If you do not enter this parameter, DATA(*LABELS) is assumed and you receive the tape volume display.

- If you specify DATA(*SAVRST) you receive the save/restore volume display.

OUTPUT Parameter: This parameter determines how the tape information is provided.

- If you do not enter this parameter, the information is displayed on your work station screen.

- If you specify OUTPUT(*LIST) the information is written to a spooled output file and placed on an output queue for printing on the system printer.

Examples:

1. To display at your work station the volume label and data file labels of all files on the tape mounted in the tape unit QTAPE1, enter

   ```
   DSPTAP DEV(QTAPE1) LABEL(*ALL)
   ```

2. To display at your work station the save/restore information for all files on the same tape, enter

   ```
   DSPTAP DEV(QTAPE1) LABEL(*ALL) DATA(*SAVRST)
   ```

3. To display at your work station the save/restore information for the third file, identified as SAVES, on the tape mounted in the tape unit QTAPE2, enter

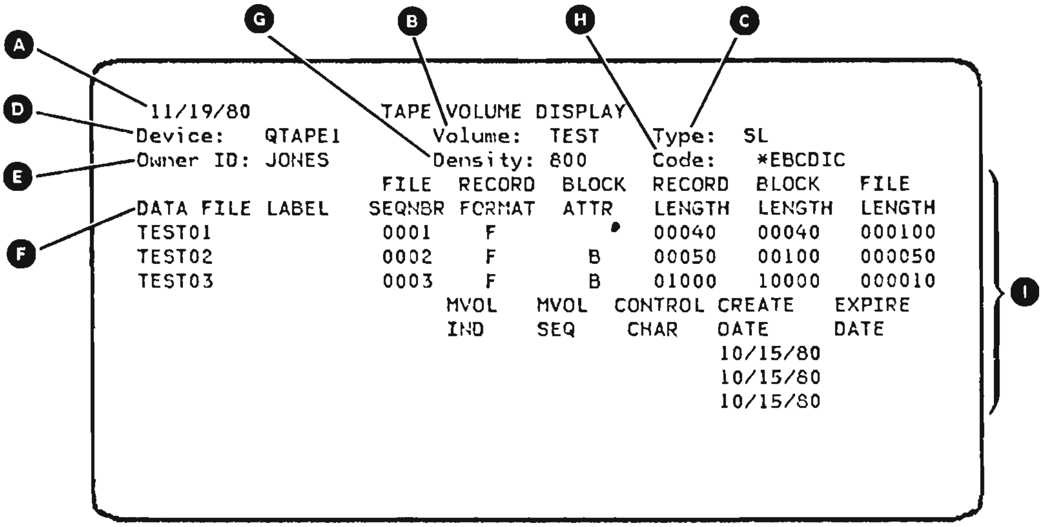
   ```
   DSPTAP DEV(QTAPE2) LABEL(SAVES) SEQNBR(3)
    DATA(*SAVRST)
   ```

4. To print the save/restore information for the second and all subsequent files on the tape mounted in the tape unit MAGTAP, enter

   ```
   DSPTAP DEV(MAGTAP) LABEL(*ALL) SEQNBR(2)
    DATA(*SAVRST) OUTPUT(*LIST)
   ```

Interpreting the Tape Volume Display

If you enter the DSPTAP without the DATA parameter, you receive a display of the following form:



```
 11/19/80               TAPE VOLUME DISPLAY
Device:   QTAPE1           Volume:  TEST      Type:  SL
Owner ID: JONES            Density: 800       Code:   *EBCDIC
                        FILE   RECORD  BLOCK  RECORD  BLOCK   FILE
DATA FILE LABEL        SEQNBR  FORMAT  ATTR   LENGTH  LENGTH  LENGTH
TESTO1                  0001     F            00040   00040   000100
TESTO2                  0002     F        B   00050   00100   000050
TESTO3                  0003     F        B   01000   10000   000010
                           MVOL    MVOL  CONTROL CREATE    EXPIRE
                           IND     SEQ     CHAR  DATE      DATE
                                                 10/15/80
                                                 10/15/80
                                                 10/15/80
```

- **A** *Date*: The date shown is the date when you requested the display.
- **B** *Volume*: This field shows the name of the volume being displayed.
- **C** *Type*: This field shows the type of tape labels. Because only standard labeled tapes can be displayed, SL will always be shown in this field.
- **D** *Device:* This is the name of the device that contains the tape files.
- **E** *Owner ID*: This field shows the name of the owner of the tape.
- **F** *Data File Label*: This field shows the data file identifier of each file being displayed. The identifier was assigned by the user when the file was created.
- **G** *Density:* This field shows the density (800 or 1600) that was used when the tape was initialized.
- **H** *Code:* This field shows the code (*EBCDIC or *ASCII) used for data files on tape.

1 *Label Characteristics*: For each file, the following label information is displayed:

- File SEQNBR (file sequence number): Indicates the sequence of the file on the tape volume.

- Record Format: The value shown indicates the format of the file. If the field is blank, the records are fixed length. The other possibilities are:

  F = fixed-length records (same as a blank)

  V = variable-length records in EBCDIC format

  D = variable-length records in ASCII format

  U = undefined

- Block Attr: If this field is blank, the records in the file are unblocked. If B is shown in the field, the records are blocked.

- Record Length: Indicates the number (for fixed-length records) or maximum number (for variable-length records) of bytes in each record.

- Block Length: Indicates the number of bytes in each block.

- File Length: Indicates the number of records in the file.

- Mvol Ind: If this field is blank, the file either is complete on the volume (tape reel) being displayed or is the last volume of multivolume file. If CONT is shown, the file is continued on another tape volume.

- Mvol Seq: The number shown, if any, indicates the sequential order of this tape volume relative to other volumes on which the file is stored.

- Control Character: If nothing is shown in this field, there is no control character as the first data character in each record of the file being displayed. An M indicates that there is a machine code control character for each record. The character provides for carriage control and card punch stacker selection.

- Create Date: Indicates the date on which the data file was created.

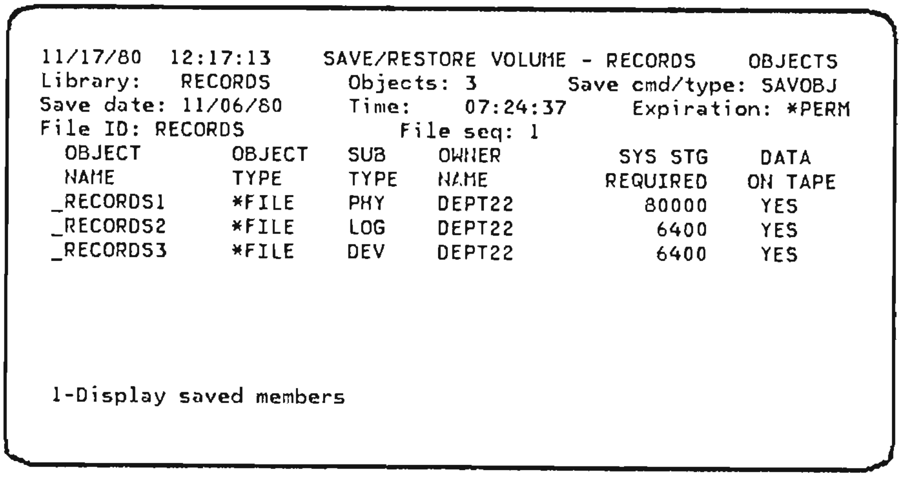
- Expire Date: Indicates the date, if any, after which the file can be deleted. If no data is shown, as in the example, the expire date has passed and the file can be deleted.

Interpreting the Save/Restore Volume Display

If you specify

DATA(*SAVRST)

when you enter the DSPTAP command, you receive a display of the following form:



```
11/17/80  12:17:13    SAVE/RESTORE VOLUME - RECORDS    OBJECTS
Library:   RECORDS      Objects: 3       Save cmd/type: SAVOBJ
Save date: 11/06/80     Time:     07:24:37     Expiration: *PERM
File ID: RECORDS               File seq: 1
  OBJECT        OBJECT     SUB     OWNER          SYS STG     DATA
  NAME          TYPE       TYPE    NAME          REQUIRED    ON TAPE
 _RECORDS1      *FILE      PHY     DEPT22           80000      YES
 _RECORDS2      *FILE      LOG     DEPT22            6400      YES
 _RECORDS3      *FILE      DEV     DEPT22            6400      YES




 1-Display saved members
```

This display is similar to the display you receive when you request save/restore information on diskettes (see the description of the save/restore volume display under *Determining Diskette Contents* in this chapter). As described for the diskette save/restore display, you receive either a library-oriented display followed by object-oriented displays or only an object-oriented display, depending on what you specified in the DSPTAP command and which command was used to save the information being displayed. The library-oriented display lists the libraries saved. The object-oriented display, as in the example shown here, lists details on the objects in each library.

PREPARING DISKETTES

Before a diskette can be used on System/38, it must be an appropriate type and have the appropriate sector format for the intended use. These considerations are discussed under *Basic Concepts* in Chapter 3.

If a diskette has the wrong sector format, it must be initialized. This is done by the Initialize Diskette (INZDKT) command. You should also reinitialize a diskette when:

- The diskette is exposed to a strong magnetic field.
- A defect has occurred in one or two tracks.

The INZDKT command erases all information from the diskette and writes a new volume label on it. If you want to change the volume label without affecting the data on it, you can use the Rename Diskette (RNMDKT) command. The Clear Diskette (CLRDKT) command, on the other hand, allows you to erase the data from the diskette without changing the volume label that identifies it.

You can delete individual files from a diskette by using the Delete Diskette Label (DLTDKTLBL) command, or you can copy an entire diskette onto one or more other diskettes by using the Duplicate Diskette (DUPDKT) command.

Whenever you change the information on a diskette, such as by initializing it or renaming it, you should indicate the change on the labels attached to the diskette jacket. It is especially important that the current sector size and volume identifier be shown, because these characteristics frequently restrict how the diskette can be used. For example, only diskettes with a 1024-byte sector size can be used for save/restore operations. Specific jobs or procedures may require a diskette with a particular volume identifier.

Initializing Diskettes

When you initialize a diskette, identification information is written on it and it is given a specific format for storing information. The initialization process includes:

- Optionally checking for files that are still active and should not be cleared.

- Testing each track for physical defects on the recording surface. A diskette is unusable if more than two defective cylinders are found, cylinder 0 is defective, or the track identifier of a defective track cannot be read.

- Formatting each track to a specified sector size (128, 256, 512, or 1024 bytes) and a recording mode (single density or double density). This is explained under the FMT and SCTSIZ parameters.

You initialize diskettes by placing them in the diskette magazine drive and entering the Initialize Diskette (INZDKT) command. You must include the LOC parameter and at least specify the magazine or slot position of each diskette to be initialized.

When initializing diskettes for save/restore operations, you are restricted to some extent in your choice of parameters and values for this command. The restrictions are:

- You cannot specify

  ```
  LOC(*M12)
  LOC(*S12)
  LOC(*S23)
  LOC(*S123)
  ```

- You must specify the NEWVOL parameter.

- You must specify FMT(*SAVRST).

- You cannot specify CODE(*ASCII).

These restrictions are discussed further in the following description of the parameters of the INZDKT command.

LOC Parameter: This parameter determines which diskettes are to be initialized. All of the values described for the location parameter in *Diskette Considerations* can be used except combinations of magazines or I/O slots.

The value *M12, *S12, *S23, or *S123 cannot be specified when diskettes are being initialized for save/restore because diskettes, in a magazine, that are being initialized for save/restore operations should have a volume identifier that is different from the volume identifier of any diskette in any other magazine. If both magazines were initialized at the same time, the diskettes in the two magazines would have the same volume identifiers. Similarly, if you specified more than one I/O slot at a time, the diskettes in those slots would have the same volume identifier. Therefore, the values *M12, *S12, *S23, and *S123 cannot be specified when diskettes are being initialized for save/restore.

NEWVOL Parameter: This parameter defines what volume identifier is to be assigned to each diskette being initialized.

- If you do not enter this parameter, no volume identifier is assigned.

- You must enter this parameter and specify a volume identifier if diskettes are being initialized in the save/restore format (see the following discussion of the FMT parameter). However, you can also specify a volume identifier for a diskette being initialized in any other format.
  - For diskettes being initialized in the save/restore format, the volume identifier can be any combination of letters and numbers up to a maximum of five for diskettes in a magazine and six for diskettes in manual slots. If the diskettes to be initialized are in a magazine, the system adds a final character that indicates the position of each diskette in the magazine (see *Diskette Considerations* in this chapter).
  - For diskettes being initialized in a format other than save/restore, the volume identifier can be any combination of letters and numbers up to a maximum of six. All diskettes being initialized are given the same identifier.

Note: If you initialize only some of the diskettes in a magazine for save/restore, the volume identifier you specify must be the same as the volume identifier specified when the other diskettes were initialized.

NEWOWNID Parameter: This parameter is used to specify an owner of each diskette being initialized. You do not have to specify an owner identifier; if you do, however, that identifier is written in the volume label along with the volume identifier.

- If you do not enter this parameter, blanks are written in the owner identifier field.

- If you enter this parameter, you can specify any combination of letters and numbers up to a maximum of 14, such as

```
NEWOWNID(DEPT22ISOWNER)
```

FMT Parameter: This parameter determines how each diskette is formatted.

- If you do not enter the parameter, the diskette is formatted with single density recording. This format can be used only for diskettes 1 and 2.
- If you specify FMT(*DATA2) the diskette is formatted with double density recording. Diskette 2D must be used.
- If you specify FMT(*SAVRST) the diskette is given the save/restore format with double density recording. You *must* enter the parameter and specify this value if the diskette is to be used for save/restore. The *SAVRST format can be used only with diskette 2D.
- If you specify FMT(1) a diskette 1 is required. The diskette will be formatted with single density recording.
- If you specify FMT(2) a diskette 2 is required. The diskette will be formatted with single density recording.
- If you specify FMT(2D) a diskette 2D is required. The diskette will be formatted with double density recording.

When you specify this format, you will receive an error message if you
- Specified *M12, *S12, *S23, or *S123 in the LOC parameter
- Did not include the NEWVOL parameter with a volume identifier (see the NEWVOL parameter)
- Specified CODE(*ASCII)
- Specify a value other than *STD or 1024 for the SCTSIZ parameter

SCTSIZ Parameter: This parameter determines the number of bytes per sector the diskette is initialized to.

- If you do not enter this parameter, the diskette is initialized to the standard sector size for the value entered in the FMT parameter. This value is:

FMT Parameter	
Value	**Sector Size**
*DATA	128
1	128
2	128
2D	256
*DATA2	256
*SAVRST	1024

The following chart shows the valid sector sizes for each value of the FMT parameter.

SCTSIZ	*DATA	1	2	2D	*DATA2	*SAVRST
*STD	X	X	X	X	X	X
128	X	X	X			
256	X	X	X	X	X	
512	X	X	X	X	X	
1024				X	X	X

CHECK Parameter: This parameter determines whether a check is made for active files (files whose expiration date is greater than the current date) on each diskette being initialized.

- If you do not enter the parameter, the diskette is checked for active files and you receive a message if active files are found, giving you the option of continuing or terminating the initialization. If you continue, the active files are destroyed.

- If you specify CHECK(*NO) the initialization is performed without checking for active files.

CODE Parameter: This parameter determines whether the volume label is written in EBCDIC or ASCII code. All subsequent data written on the diskette must have the same code.

- If you do not enter the parameter, EBCDIC code is used. Do not enter the parameter if you are initializing the diskette in the save/restore format because EBCDIC code is required for that format.

- If you specify CODE(*ASCII) the diskette is written in ASCII code.

Note: When you initialize diskettes with a volume identifier, you should write that volume identifier on the label on the diskette jacket of each diskette for later reference. Remember to use only a fiber-tip pen and press lightly so that you do not damage the diskette.

Examples:

1. To initialize all diskettes in magazines 1 and 2 with the 2D format, sector size 512, after checking that they contain no active files, enter

   ```
   INZDKT LOC(*M12) FMT(2D) SCTSIZ(512)
   ```

   These diskettes cannot be used for save/restore because they are I exchange type.

2. To initialize the diskettes in magazine 1 with the save/restore format, without checking for active files, and to label them with the volume identifier MAST but no owner identifier, enter

   ```
   INZDKT LOC(*M1) NEWVOL(MAST) FMT(*SAVRST) CHECK(*NO)
   ```

   When the diskettes are labeled, a number is added to the volume identifier of each diskette that indicates which position in the magazine it is in. That is, the diskette in position 2 is labeled MAST2, the diskette in position 5 is labeled MAST5, and the diskette in position 10 is labeled MAST0.

3. To initialize only the second diskette in magazine 2 with the save/restore format, after checking that it contains no active files, and to label it with the volume identifier PAY and the owner identifier SMITH, enter

   ```
   INZDKT LOC(*M2 2 *ONLY) NEWVOL(PAY)
     NEWOWNID(SMITH) FMT(*SAVRST)
   ```

   The second diskette becomes identified as volume PAY2 with an owner named SMITH.

4. To initialize the third through sixth diskettes of magazine 2 as in example 3, but not check for active files, enter

   ```
   INZDKT LOC(*M2 3 6) NEWVOL(PAY) NEWOWNID(SMITH)
     FMT(*SAVRST) CHECK(*NO)
   ```

   The third through sixth diskettes are labeled PAY3 through PAY6 with an owner named SMITH.

Clearing Diskettes

When you clear a diskette, all files are deleted from the diskette by deleting the file labels from the diskette label. However, the volume identifier that identifies the diskette to the system still remains.

To clear a diskette, enter the Clear Diskette (CLRDKT) command.

LOC Parameter: You must enter this parameter and at least specify the position in a magazine or slot of each diskette to be cleared. You can use any of the values described for the LOC parameter in *Diskette Considerations*.

VOL Parameter: This parameter determines whether a check is made for a specific diskette volume.

- If you do not enter the parameter, VOL(*LOC) is assumed and no check is made and each position specified in the LOC parameter is cleared.

- If you specify a volume identifier, such as VOL(SAVE) the diskette in each position specified in the LOC parameter is matched to the volume identifier. If the volume identifiers do not match, you receive a message that allows you to either mount the correct volume and try again or proceed to the next diskette position.

CHECK Parameter: This parameter determines whether a check is made for active files (files whose expiration date is greater than the current date) on each diskette being cleared.

- If you do not enter the parameter, CHECK(*YES) is assumed; the diskette is checked for active files, and you receive a message if active files are found, giving you the option of continuing or terminating the clear operation. If you continue, the active files are destroyed.

- If you specify CHECK(*NO) the diskettes are cleared without checking for active files.

Note: You cannot use this command to clear diskettes having an extended label area. Those diskettes can be cleared only by reinitializing them, using the INZDKT command.

Examples:

1. To clear all diskettes in both magazines, enter

```
CLRDKT LOC(*M12)
```

2. To clear all diskettes from the third position in magazine 1 through the seventh position in magazine 2, enter

```
CLRDKT LOC(*M12 3  7)
```

3. To clear the diskettes in the second through last positions (positions 2 through 10) of magazine 1 only if their volume identifier is SAV, enter

```
CLRDKT LOC(*M1 2) VOL(SAV)
```

Renaming Diskettes

When you rename a diskette, you change only the volume identifier and/or owner identifier on the volume label. The files on the diskette are not affected.

To rename a diskette, enter the Rename Diskette (RNMDKT) command and include at least the LOC parameter. There are four parameters:

LOC Parameter: This parameter determines which diskette is to be renamed. You must specify a magazine or I/O slot. The valid values are *M1, *M2, *S1, *S2, or *S3 (see Figure 13-1). If you specified a magazine, you can also specify which diskette in the magazine is to be renamed:

- If you do not specify a diskette number, only the first diskette in the magazine is renamed.

- If you specify a diskette number along with the magazine identifier, such as LOC(*M2 5) only that diskette is renamed.

VOL Parameter: This parameter determines whether a check is made for a specific diskette volume in the same way as the VOL parameter of the CLRDKT command (see *Clearing Diskettes* in this chapter). If you do not enter the parameter, VOL(*LOC) is assumed and no check is made.

NEWVOL Parameter: This parameter defines the new volume identifier for the diskette. If you do not enter the parameter, NEWVOL(*SAME) is assumed and the volume identifier is not changed.

If you are renaming a save/restore diskette to be used in a magazine, you must specify the entire volume identifier, including the final number that indicates the position of the diskette in the magazine.

NEWOWNID Parameter: This parameter defines the new owner identifier for the diskette. If you do not enter the parameter, NEWOWNID(*SAME) is assumed and the owner identifier is not changed.

Examples:

1. To change the volume identifier of the first diskette currently mounted in magazine 1 to SAV, enter

   ```
   RNMDKT LOC(*M1) NEWVOL(SAV)
   ```

2. To change the volume identifier of the fifth diskette in magazine 2 to MAST and the owner identifier to DEPT22, only if the current volume identifier is SAV, enter

   ```
   RNMDKT LOC(*M2 5) VOL(SAV) NEWVOL(MAST)
    NEWOWNID(DEPT22)
   ```

Deleting Diskette Labels

When you delete a file identifier from a diskette label, you in effect delete the file from the diskette because the area occupied by the actual file data can be overwritten the next time information is placed on the diskette.

To delete a file identifier, enter the Delete Diskette Label (DLTDKTLBL) command and specify the file identifier to be deleted (LABEL parameter) along with the diskette location (LOC parameter). There are five parameters.

LABEL Parameter: You must include this parameter to specify what file identifier is to be deleted.

LOC Parameter: You must include this parameter and specify the diskette or diskettes containing the file identifier to be deleted. Any magazine, slot, or combination of magazines or slots as described in *Diskette Considerations* is valid. As an option, you can specify the starting diskette position.

- If you do not specify a starting diskette position, all diskettes starting with the first (leftmost) position in the specified magazine or slot locations (see Figure 13-1) are used.

- Provided you specified one or both magazines, you can specify a starting position other than the first, as described for the LOC parameter in *Diskette Considerations*.

VOL Parameter: Use this parameter to define whether a check is to be made for a specific diskette volume, as described for the VOL parameter in *Clearing Diskettes*. If you do not enter the parameter, VOL(*LOC) is assumed and no check is made before the file identifier is deleted.

CRTDATE Parameter: This parameter determines whether a check is made of a specific creation date on the file before the file identifier is deleted.

- If you do not enter the parameter, no check is made.

- If you specify a creation date such as CRTDATE(102079) or CRTDATE('10/20/79') the specified date is checked against the creation date on the file. If the dates do not match, you receive a message that allows you to either retry the operation or terminate the command. Note that you must enclose the value in apostrophes (quotes) if you enter the special character /.

OPTION Parameter: This parameter determines how the file is to be deleted from the diskette after its file identifier is deleted.

- If you do not enter the parameter, OPTION(*SCRATCH) is assumed and the expiration date of the file is changed to the current system date so that the space occupied by the file can be overwritten. The file can still be referenced until it is actually overwritten.

- If you specify OPTION(*RMV) the file label is deleted so that it no longer can be referenced. The data in the file remains until it is overwritten.

- If you specify OPTION(*ERASE) the file label is deleted and the file data is overwritten with binary zeros.

CHECK Parameter: Specifies whether a check for active files will be performed. Enter CHECK(*NO) or let CHECK(*YES) be assumed.

Examples:

1. To delete any file named RECORDS that exists on any diskette in either magazine by making the file location available for overwriting, enter

```
DLTDKTLBL LABEL(RECORDS) LOC(*M12)
```

2. To delete any file named FILEA on the fifth through the last diskette of magazine 2 by removing the file label, enter

```
DLTDKTLBL LABEL(FILEA) LOC(*M2 5) OPTION(*RMV)
```

3. To erase both the label and the data of a file named INVENTORY only if it is located on volume SAVE3 of magazine 1 and has a create date of 10/12/79, enter

```
DLTDKTLBL LABEL(INVENTORY) LOC(*M1) VOL(SAVE3)
 CRTDATE('10/12/79') OPTION(*ERASE)
```

Duplicating Diskettes

When you duplicate a diskette, the contents of that diskette are copied onto one or more other diskettes. Diskette data in either the basic data exchange format or the save/restore format (see *Initializing Diskettes* in this chapter) can be copied. If the diskettes on which the information is to be copied do not have the same sector size as the diskette being copied, you receive a message that allows you either to terminate the copy operation or to specify that the receiving diskettes are to be initialized with the proper sector size.

To copy a diskette, enter the Duplicate Diskette (DUPDKT) command and specify which diskette is to be copied onto which diskette or diskettes. You do this by the values you specify in the three parameters.

FROMLOC Parameter: Use this parameter to specify the location of the diskette to be copied. You must at least specify a magazine or I/O slot. The valid values are *M1, *M2, *S1, *S2, or *S3 (see Figure 13-1). If you specify a magazine, you can also specify a particular position in the magazine.

- If you specify only a magazine, the first diskette in the magazine is copied.

- If you specify a magazine and diskette position, such as FROMLOC(*M2 5) the diskette in that position is copied.

TOLOC Parameter: Use this parameter to specify the location of the diskette or diskettes on which the copy is to be made. As in the FROMLOC parameter, you must specify a magazine or an I/O slot. If you specify a magazine, you have the option of specifying both a starting and ending position in the magazine.

- If you specify only a magazine, the copy is made only on the first diskette in the magazine (see Figure 13-1).

- If you specify a magazine plus a diskette position, such as TOLOC(*M1 7) the copy is made only on the diskette in that position of the magazine.

- If you specify a magazine plus a starting and an ending position, such as TOLOC(*M2 3 8) a copy is made on all of the diskettes from the specified starting to the ending positions. Note that you could also specify *FIRST and *LAST for the starting and the ending positions to have the copy made from the first and/or to the last position in the magazine.

RGZVOL Parameter: Use this parameter if you want to utilize unused space between files. Enter RGZVOL(*YES) or let RGZVOL(*NO) be assumed.

Examples:

1. To copy the information on the diskette in I/O slot 1 to the first diskette in magazine 1, enter

   ```
   DUPDKT FROMLOC(*S1) TOLOC(*M1)
   ```

2. To copy the information on the diskette in the fifth position of magazine 1 to the eighth, ninth, and tenth diskettes in magazine 2, enter

   ```
   DUPDKT FROMLOC(*M1 5) TOLOC(*M2 8 *LAST)
   ```

   Note: You could also enter the TOLOC parameter as TOLOC(*M2 8 0). In this example, you are making multiple copies of the information on a diskette.

PREPARING TAPE

All magnetic tape to be used on the system must be initialized. Even if the tape has been initialized, you may need to initialize it again to add or change a volume identifier on the tape. If you are unable to display the label information on a tape (see *Determining Tape Contents* in this chapter), it is likely the tape is not in a form usable on the system and needs to be initialized.

You initialize a magnetic tape by placing the tape reel in a tape unit connected to the system and entering the Initialize Tape (INZTAP) command. You must include the DEV parameter. All other parameters are optional.

DEV Parameter: Use this parameter to specify the name of the tape unit into which you have placed the tape that is to be initialized; for example

```
DEV(QTAPE1)
```

NEWVOL Parameter: Use this parameter to specify the volume identifier for the tape being initialized (labeled tapes are required for save/restore operations).

- If you specify NEWVOL(*NONE) the tape will be initialized as non-labeled.

- If you specify a volume identifier (any combination of letters and numbers up to maximum of six), such as NEWVOL(LABEL1) the specified identifier is written in the volume label.

NEWOWNID Parameter: Use this parameter when you want to specify the name of the tape owner to be written in the volume label.

- If you do not enter this parameter, NEWOWNID(*BLANK) is assumed and blanks are written in the owner identifier field of the volume label.

- If you specify an owner identifier (any combination of letters and numbers up to a maximum of 14), such as NEWOWNID(JONES) the specified characters are written in the owner identifier field of the volume label. NEWOWNID is ignored when initializing a non-labeled tape.

VOL Parameter: Use this parameter if you want to check the volume identifier before initializing.

- If you do not enter this parameter, VOL(*MOUNTED) is assumed and the tape that is mounted is initialized.

- If you specify a volume identifier such as VOL(LABEL2) the mounted tape is checked for a volume identifier LABEL2. If a different volume or a non-labeled volume is mounted, the initialization is terminated and a message is sent to the user who issued the command.

Note: VOL(*MOUNTED) or the default must be used to initialize a new or completely erased tape.

CHECK Parameter: This parameter determines whether a check is made for active files (files having an expiration date greater than the system date).

- If you do not enter this parameter, CHECK(*YES) is assumed and the file labels are checked. If any active files are found, the initialization is terminated and a message indicating the termination is sent to the user who entered the command.

- If you specify CHECK(*NO) the tape is initialized without checking for active files. Any active files on the tape are destroyed.

  Note: CHECK(*NO) must be specified to initialize a new or completely erased tape.

- If you specify CHECK(*FIRST), only the first file is checked.

DENSITY Parameter: This parameter specifies the density in which data is to be written on the tape after it is initialized.

- If you do not enter this parameter, the data density is 1600 bytes per inch.

- If you specify DENSITY(800) the data density is 800 bytes per inch.

Note: The density of a non-labeled tape may be changed when the first file is written on the tape.

CODE Parameter: This parameter determines the character code in which the volume label is to be written. All data subsequently written on the tape must be in the same code.

- If you do not enter this parameter, the volume label is written in EBCDIC code and is an IBM standard label. All subsequent data must be written in EBCDIC. A tape to be used for save/restore operations must be initialized with EBCDIC code.

- If you specify CODE(*ASCII) the volume label is written in ASCII code and is an IBM standard label in the same format as the EBCDIC label. All subsequent data will be written in ASCII. CODE is ignored when initializing a non-labeled tape.

ENDOPT Parameter: This parameter specifies whether the tape is to be rewound and unloaded or only rewound after it has been initialized.

- If you do not enter this parameter, ENDOPT(*REWIND) is assumed and the tape is rewound to the load point, which is the beginning of tape (BOT) marker, after it is initialized.

- If you specify ENDOPT(*UNLOAD) the tape is rewound and then unloaded so that the tape reel can be removed from the tape unit.

CLEAR Parameter: This parameter determines whether the tape volume is erased from the load point (BOT marker) to the end of tape (EOT marker) when it is initialized.

- If you do not enter this parameter, CLEAR(*NO) is assumed and the tape is not erased. For most practical purposes, however, the data cannot be accessed by the system.

- If you specify CLEAR(*YES) the tape is erased.

Examples

1. To initialize the tape in the device QTAPE1 after checking for active files, and write the volume identifier SAVE1 on it, enter

   ```
   INZTAP DEV(QTAPE1) NEWVOL(SAVE1)
   ```

2. To initialize the tape in the device QTAPE2 without checking for active files and to write both the volume identifier SAVE2 and the owner identifier DEPT22 on it, enter

   ```
   INZTAP DEV(QTAPE2) NEWVOL(SAVE2) NEWOWNID(DEPT22)
    CHECK(*NO)
   ```

   In both examples 1 and 2, a tape density of 1600 bytes per inch is specified, the tape volume label is written in EBCDIC code, and the tape is rewound but not unloaded after initialization.

3. To initialize the tape in the device MAGTAP with the volume identifier TAPE1 in ASCII code, specify a density of 800 bytes per inch, and unload the tape after initialization, enter

   ```
   INZTAP DEV(MAGTAP) NEWVOL(TAPE1) DENSITY(800)
    CODE(*ASCII) ENDOPT(*UNLOAD)
   ```

SAVE/RESTORE GUIDELINES

General Considerations

Save/restore operations are based on copying one or more objects from one storage medium to another. (The concept of an object is described in Chapter 6.) The following summarizes the points you should remember when saving or restoring information. Keep in mind that guidelines given for an *object* can also apply to a group of objects or to an entire library.

- In general, only you or another person with system operator authority can save or restore all objects in the system. Users can save or restore objects to which they have authority, but normally only you do it.

- When an object is saved, a *definition* of the object is saved along with the object itself. This definition consists of:
  - Object name
  - Object type/subtype
  - Date and time of the save
  - For data base files, a description of the file members
  - Owner name

- All object types can be saved except message queues (*MSGQ), job queues (*JOBQ), and output queues (*OUTQ).

- Logical files can be saved, although the system will only save the description. The access path will be rebuilt automatically on the restore operation if you did not specify *REBLD for the MAINT option when the file was created.

- An object that is saved is not removed from the system. That is, it still occupies storage on the system and can be used normally. Only a copy of it is saved.

- If you do not want to keep the object on the system, you can make the storage it occupies available for other use. This is called freeing storage. However, only files, programs, and journal receivers can have their storage freed.

- When you free the storage of an object, the object is removed from internal storage, but its definition remains; the object is still known to the system, and an operation can still be performed on it (such as moving it or renaming it). However, the object cannot be used unless it is restored (copied back into internal storage).

- When you save an object whose storage was previously freed, only the definition of the object is saved because the object itself does not exist on internal storage. A data base file whose storage was previously freed, cannot be saved.

- The IBM-supplied system device files for save/restore media, either QSYSDKT or QSYSTAP, cannot be overriden. The save/restore operation ignores override commands for these files.

- You can save and restore objects while normal system operations continue. However, if an object is being used for updating or is allocated exclusively to a job, it cannot be saved or restored while being used. Also, you cannot free the storage of an object that is being used in a job.

- If you are saving or installing the system, the system must be dedicated to the operation. Other functions cannot be performed at the same time.

- You can restore only those objects that were previously saved on your system or another System/38.

- You can restore objects to a system whether those objects still exist on the system or not. The object might have had its storage freed or deleted, or the object might have never existed on the system you are restoring it to.

- When you restore an object whose storage has not been freed, the original object still existing on the system is replaced by the saved object. If the object does not exist on the system, the system allocates storage for it and restores it.

- If only selected objects from a library are saved, only those objects, not the entire library, can be restored.

- With the exception of journals and journal receivers, you can restore objects to a library other than the library from which they were saved.

- If you restore an object to the system and its owner no longer exists on the system, the system security officer becomes its owner.

- If the original object still exists on the system, but has a different owner, it will not be restored.

- Objects that have been saved by separate commands must also be restored by separate commands. If you try to restore them together in a single command, not all of them will be restored.

- The user profile for the person running the restore is charged for certain auxiliary storage used on the restore operation.

- Save and restore operations will take less time if the system is dedicated and the user has save system authority because individual object locking and authority checking will not be performed. The performance improvement is noticeable only when you save and restore a large number of objects.

Save/Restore Status

The system maintains save/restore information that includes the following data about each object:

- System date and time of the last save

- System date and time of the last restore

- Volume identifiers of volumes containing the most recently saved version

- Location of the saved object

- Size of the saved object

- Whether the data portion of the object resides on the media

You can display the save/restore history of an object by entering the Display Object Description (DSPOBJD) command and specifying DETAIL(*FULL). For example, you can display the save/restore history of the file named TALLY in the library named RECORDS by entering

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

There are three data areas (object type *DTAARA) in the library QSYS that contain information about the system. They are:

- QSAVLIBALL: Contains save/restore history of all the user libraries.

- QSAVUSRPRF: Contains save/restore history of all the user profiles.

- QSAVSYS: Contains save/restore history of the CPF.

You can display these data areas using the DSPOBJD command. For example, to obtain the save/restore history of the user libraries, enter

```
DSPOBJD OBJ(QSAVLIBALL.QSYS) OBJTYPE(*DTAARA) DETAIL(*FULL)
```

SAVING INFORMATION

There are separate commands that allow you to save individual objects (SAVOBJ), saving only changed objects (SAVCHGOBJ), user libraries (SAVLIB), and basic system information (SAVSYS). You can also save the entire system by using a combination of the SAVLIB and SAVSYS commands (see *Saving and Restoring the System* later in this chapter). Objects, user libraries, and basic system information can be saved on either diskettes or magnetic tape.

The following topics discussed in this chapter are:

1. Saving Objects
2. Saving Changed Objects
3. Saving Libraries
4. Saving System Information

Saving Objects

When you save an object, a copy of that object is made on one or more diskettes or reels of magnetic tape. You can save:

- A single object
- Several individual objects
- A group of objects by generic name
- A group of objects by generic name and object type

However, all objects saved at one time must be in the same library.

To save an object or objects, enter the Save Object (SAVOBJ) command. You must enter the OBJ and LIB parameters.

You can also save an object or group of objects that have changed by using the Save Changed Objects (SAVCHGOBJ) command. You can specify a date and time on the command and only objects that have changed after the date and time specified will be saved. For more information on the SAVCHGOBJ command, see the *CL Reference Manual*.

OBJ Parameter: Use this parameter to specify the object or objects to be saved.

- If you specify OBJ(*ALL) all objects of the specified types in the specified library are saved.
- If you specify one or more generic object names (up to 50), such as OBJ(PAY*) all objects of the specified types in the specified library whose names start with those characters are saved.

Q*

- If you specify one or more object names (up to 50), such as

  OBJ(PAY1 PAY2)

  only those objects of the specified type are saved.

LIB Parameter: Use this parameter to specify the library containing the objects to be saved. Only objects in the specified library are saved.

OBJTYPE Parameter: This parameter determines whether all types of objects are saved or only those of a specific type.

- If you do not enter this parameter, OBJTYPE(*ALL) is assumed and all types of objects having the names specified in the OBJ parameter are saved.
- If you specify one or more object types, such as OBJTYPE(*PGM) only objects of the specified type are saved. The object types that you can save and the value you enter in the command for each are:

Value	Type of CPF Object
*CLS	Class
*CMD	Command
*DTAARA	Data area
*EDTD	Edit description
*FCT	Forms control table
*FILE	File
*JOBD	Job description
*JRN	Journal
*JRNRCV	Journal receiver
*MSGF	Message File
*PGM	Program
*PRTIMG	Print image
*SBSD	Subsystem description
*SSND	Session description
*TBL	Table

SAVOBJ - Q* + *File

--ODKT

vol(*Mounted)

DEV Parameter: This parameter determines which device is used.

- If you do not enter this parameter, the diskette magazine drive as identified by the name QDKT is used. QDKT is the name of the IBM-supplied device description for the diskette magazine drive.

- If you specify another device description name, such as DEV(QTAPE1) the identified device is used. The device could be the diskette magazine drive as identified by a device description name other then QDKT, or it could be a magnetic tape unit. For magnetic tape, you can specify up to four tape device names. This allows overlapping of tape rewind-unload with read/write operations.

LOC Parameter: If the diskette magazine drive was specified in the DEV parameter, this parameter determines on which diskette or diskettes the specified objects are to be saved. It consists of two values.

Device Location: The first value specifies the magazine or slot location in the diskette device. The possible values are *M12, *M1, *M2, *S123, *S12, *S23, *S1, *S2, and *S3 (see Figure 13-1).

- If you do not enter this parameter, *M12 is assumed.

- If you specify another location, such as LOC(*S12) the diskette or diskettes in the specified location are used.

Starting Position: The second value specifies the starting position in the specified location.

- If you do not enter this value, the operation begins with the first (leftmost) position. In a magazine, this would be position 1 (see Figure 13-1).

- If you specify *CURRENT, the diskette at which the diskette drive is currently positioned is used. You would specify this value, when you want to begin the operation where the previous save/restore operation ended. If the current diskette position is not located anywhere within the location specified in the first value, a diagnostic message is issued and the command is terminated.

- If you specify *SEARCH, the diskettes in the location specified by the first value are searched for the first diskette having cleared space, where the operation will begin. The search begins with the leftmost diskette in the specified location and ends with the first diskette on which cleared space is found.

- If you specify the number of a particular diskette position, the operation begins with the diskette in that position. The number can be from 1 to 3 for diskettes in I/O slots or from 1 to 10 for diskettes in magazines (see Figure 13-1). The number 1 indicates the diskette in the first (leftmost) position in the slot or magazine.

If the specified diskettes do not contain enough storage to save all of the specified objects, you will receive a message to mount additional diskettes.

VOL Parameter: This parameter determines whether the object or objects are to be saved only on a specific volume.

- If you do not enter this parameter, VOL(*MOUNTED) is assumed, which means that the objects are saved on the diskettes currently loaded in the diskette device that are in the position specified in the LOC parameter, or on the tape reel currently loaded in the magnetic tape unit.

- If you specify one or more volume identifiers, such as VOL(SAVE) the objects are saved on the diskettes or tape having those identifiers (see the discussion of the VOL parameter in *Diskette Considerations* in this chapter). If the specified volume is not in the specified location or not in the specified device, you will receive a message to mount the correct volume in the device.

SEQNBR Parameter: This parameter is used for tape only and specifies the file sequence number where the save is to begin.

- If you do not specify this parameter, the file will be saved after the last sequence number on the tape.

- If you specify a sequence number, the file will be saved at that sequence number.

Note: If a file on a tape is overwritten, the files that follow on that tape are destroyed.

CLEAR Parameter: This parameter determines whether data on diskette or tape should be overwritten.

For diskette: The default is *NO, specifying that diskettes are not to be cleared during the save operation but should be cleared by the user prior to using the SAVE command.

*yes

If *YES is specified for the CLEAR keyword, and an unexpired file on diskette is encountered while the operator is attempting to save objects, the diskette will be cleared and the save operation will continue. If *NO has been specified, an inquiry message will be sent to the system operator message queue if an uncleared diskette is encountered while the system is continuing a save operation. If an ignore ('I') response is given to the inquiry message, the diskette will be cleared and the save operation will continue.

The CLEAR keyword relates only to *clearing* the diskette, not to *initializing* it. The diskette must still be initialized in the save/restore format before beginning the save operation.

There is some interaction between the CLEAR parameter and the second parameter of the LOC keyword. If the second parameter of the LOC keyword is *CURRENT or *SEARCH, clearing only applies to *subsequent* diskettes involved in the save operation (not to the diskette on which the save operation begins). For other values of the second parameter of the LOC keyword (*FIRST or starting diskette position), the CLEAR parameter applies also to the diskette on which the save operation begins.

For tape: The default is *NO, specifying that tapes are not to be cleared during the save operation but should be initialized by the user prior to using the SAVE command.

If *YES is specified for the CLEAR keyword, and an unexpired file is encountered while the operator is attempting to save objects, the file will be overwritten and the save operation will continue. If *NO has been specified, an inquiry message will be sent to the system operator message queue if an unexpired file is encountered while the system is continuing a save operation. If an ignore ('I') response is given to the inquiry message, the file will be overwritten and the save operation will continue.

For tape, there is interaction between the SEQNBR parameter and the CLEAR parameter. The tape overwrites files beginning only at the sequence number specified.

EXPDATE Parameter: This parameter specifies the date that the file to be saved will expire. The file is protected and cannot be written over until the expiration date or a specific request occurs to clear or initialize the media.

- If you do not enter this parameter the file will be permanently protected.

- If you enter an expiration date, the file will be protected until the expiration date.

The CLEAR parameter on the SAVE commands check the expiration date before it writes over an existing file.

STG Parameter: This parameter determines whether the storage occupied by each object being saved is freed. When storage is freed, the data part of each object is removed from internal storage, although the object definition still remains. The object remains known to the system, but it cannot be used unless it is restored.

- If you do not enter the parameter, storage is not freed; that is, the object remains on internal storage and can still be used.

- If you specify STG(*FREE) the object is removed from internal storage. Only objects of object type (OBJTYPE), file (*FILE), journal receiver (*JRNRCV), or program (*PGM) can have their storage freed.

ENDOPT Parameter: When a tape device is specified in the DEV parameter, this parameter determines how the tape is positioned after the operation ends. If the file is contained on more than one tape reel, the ENDOPT parameter applies to only the last reel; all preceding reels are automatically rewound and unloaded.

- If you do not enter this parameter, the tape is rewound to the load point, which is the beginning of tape (BOT) marker.

- If you specify ENDOPT(*UNLOAD) the tape is rewound and then unloaded so that you can remove it from the device.

- If you specify ENDOPT(*LEAVE) the tape is not moved from the position where the last operation ended. You would use this option, for example, when you are planning to save additional information on the tape.

Examples:

1. To save the object PAY1 of the library PAYROLL on the diskette used for the most recent save/restore operation in the device QDKT, without freeing the object's storage, enter

   ```
   SAVOBJ  OBJ(PAY1) LIB(PAYROLL) LOC(*M12 *CURRENT)
   ```

2. To save the objects PAY1 and PAY2 of the library PAYROLL on the diskette labeled MAST in I/O slot 2 of the device DKT, and free their storage, enter

   ```
   SAVOBJ  OBJ(PAY1 PAY2) LIB(PAYROLL) DEV(DKT) LOC(*S2)
    VOL(MAST) STG(*FREE)
   ```

3. To save the same objects on the tape volume SAVTAP that is mounted in the device QTAPE1, without freeing their storage, and to leave the tape where it is (not rewind it) after the save operation, enter

   ```
   SAVOBJ  OBJ(PAY1 PAY2) LIB(PAYROLL) DEV(QTAPE1)
    VOL(SAVTAP)  ENDOPT(*LEAVE)
   ```

4. To save all *files* in the library PAYROLL whose generic names start with PAY, placing them on the first cleared diskette and any subsequent diskette needed in magazine 2 of device QDKT, without freeing their storage, enter

   ```
   SAVOBJ  OBJ(PAY*) LIB(PAYROLL) OBJTYPE(*FILE)
    LOC(*M2 *SEARCH)
   ```

Saving Libraries

When you save a library, a copy of the entire library, including all objects in it and their PC status, is made on diskettes or magnetic tape.

To save a library, enter the Save Library (SAVLIB) command. Using this command, you can save:

- A single library

- All user libraries (including the IBM-supplied user library QGPL)

The SAVLIB command has seven parameters:

LIB Parameter: You must enter this parameter and specify whether all libraries are to be saved or only a particular library.

- If you specify LIB(*NONSYS) all user libraries are saved, including the general purpose library, QGPL, and the program product libraries such as QRPG and QIDU. Note that before you use the SAVLIB command with the LIB(*NONSYS) option, you must terminate all subsystems by entering TRMSBS SBS(*ALL) (see *Terminating All Subsystems* in Chapter 8).

- If you specify a library name, such as LIB(MYLIB), only that library is saved.

DEV Parameter: Use this parameter as described in *Saving Objects* to specify the devices to be used. If you do not enter this parameter, the device QDKT (the diskette magazine drive) is used.

LOC Parameter: If diskettes are to be used, use this parameter as described in *Saving Objects* to specify the diskette locations and the starting position. If you do not enter this parameter, the operation starts with the first diskette in magazine 1 and continues through magazine 2 if needed. You will receive a message to mount additional diskettes if more storage space is required.

VOL Parameter: Use this parameter as described in *Saving Objects* to specify the particular diskette or tape volumes to be used. If you do not enter this parameter, the currently mounted tape or the diskettes currently mounted in the specified location are used.

SEQNBR Parameter: This parameter is used for tape only and specifies the sequence number where the save is to begin.

- If you do not specify this parameter, the file will be saved after the last sequence number on the tape.

- If you specify a sequence number, the file will be saved at that sequence number.

Note: If a file on a tape is overwritten, the files that follow on that tape are destroyed.

CLEAR Parameter: This parameter determines whether data on diskette or tape should be overwritten.

For diskette: The default is *NO, specifying that diskettes are not to be cleared during the save operation but should be cleared by the user prior to using the SAVE command.

If *YES is specified for the CLEAR keyword, and an unexpired file on diskette is encountered while the operator is attempting to save objects, the diskette will be cleared and the save operation will continue. If *NO has been specified, an inquiry message will be sent to the system operator message queue if an uncleared diskette is encountered while the system is continuing a save operation. If an ignore ('I') response is given to the inquiry message, the diskette will be cleared and the save operation will continue.

The CLEAR keyword relates only to *clearing* the diskette, not to *initializing* it. The diskette must still be initialized in the save/restore format before beginning the save operation.

There is some interaction between the CLEAR parameter and the second parameter of the LOC keyword. If the second parameter of the LOC keyword is *CURRENT or *SEARCH, clearing only applies to *subsequent* diskettes involved in the save operation (not to the diskette on which the save operation begins). For other values of the second parameter of the LOC keyword (*FIRST or starting diskette position), the CLEAR parameter applies also to the diskette on which the save operation begins.

For tape: The default is *NO, specifying that tapes are not to be cleared during the save operation but should be initialized by the user prior to using the SAVE command.

If *YES is specified for the CLEAR keyword, and an unexpired file is encountered while the operator is attempting to save objects, the file will be overwritten and the save operation will continue. If *NO has been specified, an inquiry message will be sent to the system operator message queue if an unexpired file is encountered while the system is continuing a save operation. If an ignore ('I') response is given to the inquiry message, the file will be overwritten and the save operation will continue.

For tape, there is interaction between the SEQNBR parameter and the CLEAR parameter. The tape overwrites files beginning only at the sequence number specified.

EXPDATE Parameter: This parameter specifies the date that the file to be saved will expire. The file is protected and cannot be written over until the day after the expiration date unless a specific request occurs to clear or initialize the media.

- If you do not enter this parameter the file will be permanently protected.

- If you enter an expiration date, the file will be protected until the expiration date.

The CLEAR parameter on the Save commands can check the expiration date before an existing file on tape is overwritten.

STG Parameter: Use this parameter as described in *Saving Objects* to specify whether storage occupied by objects in the library or libraries is to be freed. If you do not enter the parameter, storage is not freed.

ENDOPT Parameter: If magnetic tape is to be used, use this parameter to specify how the tape is to be positioned after the save operation ends. If you do not enter this parameter, the tape is rewound to the load point (BOT marker).

Examples:

1. To save user libraries on both magazines, if needed, in the device QDKT without freeing the storage of the objects in the libraries, enter

   ```
   SAVLIB LIB(*NONSYS)
   ```

2. To save all user libraries on the tape mounted in the device QTAPE1, without freeing their storage, and unload the tape so that it can be removed, enter

   ```
   SAVLIB LIB(*NONSYS) DEV(QTAPE1) ENDOPT(*UNLOAD)
   ```

3. To save the library named TRANS on the diskettes labeled SAVE in magazine 2 of the device DKTMD, starting at the diskette location where the diskette drive is currently positioned, and free the storage of the objects in the library, enter

   ```
   SAVLIB LIB(TRANS) DEV(DKTMD) LOC(*M2 *CURRENT)
    STG(*FREE)
   ```

Saving System Information

CPF information that is required for system operation can be saved on diskettes or tape. You save the information by using the Save System (SAVSYS) command. When you enter the command, the following information is copied onto diskettes or tape:

- All CPF-required libraries, including QSYS, but not including the contents of QGPL
- The system input/output configurations
- All user profiles
- All object authorizations

If you are saving the system on tape, the system information will be written to standard labeled tape and the diskette needed to restore the system information will be written to a clear diskette initialized to *SAVRST format in *S1.

Note that this command *does not* save user libraries or program product libraries such as QRPG. To save all user libraries, including QGPL, you use the SAVLIB command with the LIB(*NONSYS) option.

Note: You should ensure that the QSYSOPR message queue is in *BREAK delivery mode so you will receive all of the messages during the system save.

Before entering the SAVSYS command, you must terminate all subsystems by entering TRMSBS SBS(*ALL) or TRMCPF (see *Terminating All Subsystems* in Chapter 8).

The SAVSYS command has the following parameters:

DEV Parameter: This parameter determines the device to be used.

- If you do not enter this parameter, the IBM-supplied device description QDKT is used.
- If you specify another device description name, such as DEV(QTAPE1), that tape device is used for system information and the install programs will be written to *S1 of the diskette device.

LOC Parameter: This parameter determines which magazine or magazines are to be used if the system is saved on diskette. The only valid values are *M12, *M1, and *M2 (see Figure 13-1). If you do not enter this parameter, the save will begin with magazine 1 and proceed to magazine 2 as needed. If additional magazines are needed, you will receive a message to mount them.

VOL Parameter: This parameter determines whether specific diskette or tape volumes are to be used. If you do not enter the parameter, the diskettes or tape currently mounted will be used. If you specify volume identifiers, you must not enter more than five characters for diskette. The last character is determined by the system, and you cannot restrict the operation to a specific position in a magazine.

CLEAR Parameter: This parameter determines whether data on diskette or tape should be overwritten.

For diskette: The default is *NO, specifying that diskettes are not to be cleared during the save operation but should be cleared by the user prior to using the SAVE command.

If *YES is specified for the CLEAR keyword, and an unexpired file on diskette is encountered while the operator is attempting to save objects, the diskette will be cleared and the save operation will continue. If *NO has been specified, an inquiry message will be sent to the system operator message queue if an uncleared diskette is encountered while the system is continuing a save operation. If an ignore ('I') response is given to the inquiry message, the diskette will be cleared and the save operation will continue.

The CLEAR keyword relates only to *clearing* the diskette, not to *initializing* it. The diskette must still be initialized in the save/restore format before beginning the save operation.

For tape: The default is *NO, specifying that tapes are not to be cleared during the save operation but should be initialized by the user prior to using the SAVE command.

If *YES is specified for the CLEAR keyword, and an unexpired file is encountered while the operator is attempting to save objects, the file will be overwritten and the save operation will continue. If *NO has been specified, an inquiry message will be sent to the system operator message queue if an unexpired file is encountered while the system is continuing a save operation. If an ignore ('I') response is given to the inquiry message, the file will be overwritten and the save operation will continue.

EXPDATE Parameter: This parameter specifies the date that the file to be saved will expire. The file is protected and cannot be written over until the day after the expiration date or a specific request occurs to clear or initialize the media.

- If you do not enter this parameter the file will be permanently protected.

- If you enter an expiration date, the file will be protected until the expiration date.

The CLEAR parameter on the Save commands checks the expiration date before it writes over an existing file.

ENDOPT Parameter: If magnetic tape is to be used, use this parameter to specify how the tape is to be positioned after the save operation ends. If you do not enter this parameter, the tape is rewound to the load point (BOT marker).

Examples:

1. To save basic system information on the diskettes currently mounted in the two magazines of the device QDKT, enter

   ```
   SAVSYS
   ```

2. To save basic system information on the diskettes labeled SYSAV in magazine 2 of the device DKTMD, enter

   ```
   SAVSYS DEV(DKTMD) LOC(*M2) VOL(SYSAV)
   ```

3. To save basic system information on tape mounted on device QTAPE1, enter

   ```
   SAVSYS DEV(QTAPE1)
   ```

   and insert a clear diskette initialized to *SAVRST format in *S1. This diskette is used when you restore the system information from tape.

RESTORING INFORMATION

There are separate commands that allow you to restore objects (RSTOBJ), libraries (RSTLIB), user profiles (RSTUSRPRF), and authority to use specific objects (RSTAUT). The RSTOBJ, RSTUSRPRF, and RSTLIB commands can be used to restore information from either diskettes or magnetic tape. User profiles are saved as a part of saving system information (SAVSYS command). The RSTLIB, RSTUSRPRF, and RSTAUT commands are used in restoring the system.

Restoring Objects

When you restore an object, you copy it from a diskette or tape into internal storage. To restore an object or objects, enter the Restore Object (RSTOBJ) command. Using this command, you can restore:

- A single object
- Several individual objects
- A group of objects by generic name
- A group of objects by generic name and object type

However, all objects restored at one time must be in the same library.

If you use the RSTOBJ command to restore individual objects that were saved as a group by the SAVLIB command, the program change (PC) objects' status that was saved with the objects is not restored and the PC information in the QCHG log may no longer be accurate for that library. You should use the RSTOBJ command instead of the RSTLIB command, therefore, when you want to restore objects without replacing the associated PC status information currently in the system (see *Restoring Libraries* in this chapter).

The RSTOBJ command has the following parameters. You must include the OBJ and SAVLIB parameters.

OBJ Parameter: You must enter this parameter and specify the object or objects to be restored.

- If you specify OBJ(*ALL) all objects of the specified types that were saved from the library specified in the SAVLIB parameter are restored.
- If you specify one or more generic names (up to 50), such as OBJ(PAY*) all objects of the specified types whose name starts with those characters that were saved from the library specified in the SAVLIB parameter are restored.
- If you specify one or more object names (up to 50), such as OBJ(ORDERS BILLS) the only objects of the specified types that are restored are those with the specified name that were saved from the library specified in the SAVLIB parameter.

SAVLIB Parameter: You must enter this parameter and specify the library from which the objects were saved, such as

SAVLIB(MYLIB)

OBJTYPE Parameter: Use this parameter as described in *Saving Objects* to specify whether all types of objects or only specific types are restored. If you do not enter this parameter, all object types having the specified names are restored.

DEV Parameter: Use this parameter as described in *Saving Objects* to specify the device to be used. If you do not enter this parameter, the device QDKT (the diskette magazine drive) is used.

LOC Parameter: If the information to be restored is on diskettes, use this parameter as described in *Saving Objects* to specify the diskette locations to be used. If you do not enter this parameter, the restore begins with the first diskette in magazine 1 and proceeds through magazine 2 as necessary. You specify *SEARCH for the starting position if you want the diskettes in the specified magazines or slots to be searched for the objects specified by the OBJ and SAVLIB parameters (and for a specific version if a SAVDATE and SAVTIME are specified).

VOL Parameter: Use this parameter to specify which diskette or tape volumes are to be used.

- If you do not enter this parameter, the most recently saved version of the specified objects is restored; this version is based on the internal save/restore history. If the wrong volume is mounted in the device, a message that instructs you to load the correct volume is sent to the system operator message queue.

- If you specify VOL(*MOUNTED) the objects are restored from the volume currently mounted in the device.

- If you specify a volume identifier, only the version of each object on the diskette volume is restored. If the specified volume is not mounted in the specified location, you will receive a message to mount the correct volume.

If you are restoring objects that were saved on a different system or have been deleted from your system, you must specify *MOUNTED or a volume identifier because the default for this parameter is *SAVVOL and your system will not have a record of objects deleted or saved on another system.

SEQNBR Parameter: This parameter is used for tape only and specifies the sequence number where the restore is to begin.

- If you do not specify this parameter, the file to be restored will be searched for starting with the current position on the tape. The first file found that matches the selection criteria specified by the other parameters will be restored.

- If you specify a sequence number, the file will be restored from that sequence number.

OPTION Parameter: This parameter determines how the restored objects are to be handled.

- If you do not enter this parameter, all of the specified objects are restored regardless of whether an object of the same name and type exists in the system or not. The restored objects replace any existing objects having the same name and type.

- If you specify OPTION(*NEW) the only objects restored are those that do *not* exist in the current version of the library in which they are to be placed. None of the objects already existing in the system are replaced.

- If you specify OPTION(*OLD) the only objects restored are those that have a current version already existing in the library in which they are to be placed. These old objects thus *replace* the current objects existing in the system.

- If you specify OPTION(*FREE) the only objects restored are those whose space was freed when they were saved previously. That is, this option puts objects back in the system so they can be used.

MBROPT Parameter: This parameter determines what members in a file are restored. This parameter applies only for data base files currently on the system. The creation date of the member on the system must match the creation date of the member on the media for the restore of the file to be successful.

- If you specify MBROPT(*MATCH), the members that exist in the current file on the system are restored if the member lists of the file on the system and the file on the media are identical.

- If you specify MBROPT(*ALL), all members on the media are restored.

- If you specify MBROPT(*NEW), members not on the system are restored.

- If you specify MBROPT(*OLD), only members that are on the system are restored.

SAVDATE Parameter: If you specified a volume identifier or *MOUNTED in the VOL parameter, you can use this parameter to specify that only a version saved on a specific date is to be restored, such as

```
SAVDATE(111779) or SAVDATE('11/17/79')
```

Note that you must enclose the value in apostrophes (quotes) to use the special character /.

If you do not enter this parameter, the first version found in the specified volume is restored.

SAVTIME Parameter: If you specified a volume identifier or *MOUNTED in the VOL parameter, you can use this parameter in combination with the SAVDATE parameter to specify that only a version saved on a specific date and time is to be restored, such as

```
SAVDATE(111779) SAVTIME(043057)

    or

SAVDATE('11/17/79') SAVTIME('04:30:57')
```

You must enclose the values in apostrophes to use the special characters / and :.

In this value for SAVTIME, the first two digits (04) are hours, the second two digits (30) are minutes, and the final two digits (57) are seconds.

If you do not enter this parameter, the first version found on the specified volume is restored.

RSTLIB Parameter: Use this parameter when you want the objects to be restored to a different library than the library from which they were saved.

- If you do not enter this parameter, the objects are restored to the same library from which they were saved, provided that the library exists in your system.

- If you specify a different restore library, such as RSTLIB(LIB1) the specified objects are restored to that library.

ENDOPT Parameter: If magnetic tape is to be used, use this parameter to specify how the tape is to be positioned after the save operation ends. If you do not enter this parameter, the tape is rewound to the load point (BOT marker).

Examples:

1. To restore the most recent version of the object PAY1 to the library PAYROLL from which it was originally saved on the diskettes mounted in the device QDKT, enter

```
RSTOBJ OBJ(PAY1) SAVLIB(PAYROLL)
```

2. To restore the objects PAY1 and PAY2 that were originally saved from the library PAYROLL onto diskette volume MAST in magazine 2 of device QDKT at 13:50:09 on date 09/17/79, writing them from the same diskettes into the library RECORDS so that they replace any existing objects by the same name, enter

```
RSTOBJ OBJ(PAY1 PAY2) SAVLIB(PAYROLL) LOC(*M2)
 VOL(MAST) OPTION(*OLD) SAVDATE('09/17/79')
 SAVTIME('13:50:09') RSTLIB(RECORDS)
```

3. To restore *programs* whose names begin with the characters PAY that were originally saved from the library PAYROLL, writing them into the library RECORDS from the tape volume REC1 in the device TAP1 such that any previous programs of the same name in RECORDS are not replaced but any programs not existing in RECORDS are added, enter

```
RSTOBJ OBJ(PAY*) SAVLIB(PAYROLL) OBJTYPE(*PGM)
 DEV(TAP1) VOL(REC1) OPTION(*NEW) RSTLIB(RECORDS)
```

Restoring Libraries

When you restore a library, a copy is made of the entire library from diskettes or tape to internal storage. The information restored to internal storage includes all objects in the library and their PC status.

To restore a library, enter the Restore Library (RSTLIB) command. Using this command, you can restore:

- A single library

- All user libraries (including the IBM-supplied user library QGPL and program product libraries such as QRPG and QIDU)

When you use the RSTLIB command, the PC status information currently in the system for all objects in the library is destroyed. Therefore, you should use the RSTOBJ command instead of the RSTLIB command if you want to restore objects from a particular library to the system without destroying the PC status information that is already in the system for the library. For example, you would use the RSTOBJ command instead of the RSTLIB command to install a program product such as RPG III into a library that contains other program products and their PC status.

SAVLIB Parameter: You must enter this parameter and specify the name of the saved library that is to be restored.

- If you specify SAVLIB(*NONSYS) or TRMCPF all saved user libraries are restored. Note that before you use the RSTLIB command with the LIB(*NONSYS) option, you must terminate all subsystems by entering TRMSBS SBS(*ALL) (see *Terminating All Subsystems* in Chapter 8).

- If you specify a library name, such as SAVLIB(MYLIB) only that library is restored.

DEV Parameter: Use this parameter as described in *Saving Objects* to specify the device to be used. If you do not enter this parameter, the device QDKT (the diskette magazine drive) is used.

LOC Parameter: If the library or libraries are to be restored from diskettes, use this parameter as described in *Saving Objects* to specify the diskette location or locations to be used. If you do not enter this parameter, the restore begins with the first diskette in magazine 1 and proceeds through magazine 2 as necessary. You would specify *SEARCH for the starting position if you want the diskettes in the specified magazines or slots to be searched for the objects specified by the SAVLIB parameter (and for a specific version if a SAVDATE and SAVTIME are specified).

VOL Parameter: Use this parameter as described in *Restoring Objects* to specify whether particular diskette volumes are to be used. If you do not enter this parameter, the most recently saved version of the library or libraries is restored. Note that you must specify a volume identifier or *MOUNTED if the SAVLIB-library does not currently exist on the system.

SEQNBR Parameter: This parameter is used for tape only and specifies the sequence number where the restore is to begin.

- If you do not specify this parameter the file to be restored will be searched for starting with the current position on the tape. The first file found that matches the selection criteria specified by the other parameters will be restored.

- If you specify a sequence number, the file will be restored from that sequence number.

OPTION Parameter: Use this parameter as described in *Restoring Objects* to specify how objects in the restored library are to be handled. If you do not enter this parameter, all of the specified objects are restored regardless of whether an object of the same name and type exists in the library or not. The restored objects replace any existing objects having the same name and type.

MBROPT Parameter: This parameter determines what members in a file are restored. This parameter applies only for data base files currently on the system. The creation date of the member on the system must match the creation date of the member on the media for the restore of the file to be successful.

- If you specify MBROPT(*MATCH), the members that exist in the current file on the system are restored if the member lists of the file on the system and the file on the media are identical.

- If you specify MBROPT(*ALL), all members on the media are restored.

- If you specify MBROPT(*NEW), members not on the system are restored.

- If you specify MBROPT(*OLD), only members that are on the system are restored.

SAVDATE/SAVTIME Parameters: Use these parameters as described in *Restoring Objects* to specify that only the version of the library saved on a particular date and time is to be restored from the volume specified in the VOL parameter. If you do not enter these parameters, the first version found on the specified volume is restored.

RSTLIB Parameter: Use this parameter when you want to specify that the information in the restored library is to be placed in a different library from the library that was saved. If the specified library does not exist on the system, it will be created.

ENDOPT Parameter: If magnetic tape is to be used, use this parameter to specify how the tape is to be positioned after the save operation ends. If you do not enter this parameter, the tape is rewound to the load point (BOT marker).

Examples:

1. To restore the most recent version of the library RECORDS from the diskettes in the device QDKT, enter

   ```
   RSTLIB SAVLIB(RECORDS)
   ```

2. To restore the version of the library PAYROLL that was saved at 16:30:27 on 11/24/79, using the tape volume SAVE in the device TAP2, and replace only those objects in the library whose storage had been freed, enter

   ```
   RSTLIB SAVLIB(PAYROLL) DEV(TAP2) VOL(SAVE)
    OPTION(*FREE) SAVDATE('11/24/79') SAVTIME('16:30:27')
   ```

Restoring User Profiles

To restore the user profiles that were saved on diskettes or tape by the SAVSYS command, enter the Restore User Profiles (RSTUSRPRF) command. Before using this command, you must terminate all subsystems by entering

```
TRMSBS SBS(*ALL) or TRMCPF
```

(see *Terminating All Subsystems* in Chapter 8).

The RSTUSRPRF command has the following optional parameters:

Dev Parameter: This parameter determines the diskette or tape device to be used.

- If you do not enter this parameter, the IBM-supplied device description QDKT is used.
- If you specify another device description name, such as DEV(QTAPE2), that tape device is used.

LOC Parameter: If the user profiles were saved on diskette, use this parameter when needed to specify the diskettes that are to be used. It consists of two values:

Device Location: The first value specifies the magazine or slot location in the diskette device. The possible values are *M12, *M1, *M2, *S123, *S12, *S23, *S1, *S2, and *S3 (see Figure 13-1).

- If you do not enter this parameter, *M12 is assumed.
- If you specify another location, the diskette or diskettes in the location are used.

Starting Position: The second value specifies the starting position in the specified location.

- If you do not enter this value, the restore operation begins with the diskette in the position where the last save/restore ended. However, if the current position is not located anywhere within the location specified in the first value, the first (leftmost) position in the specified location is used.
- If you specify *FIRST for this value, the restore operation begins with the first (leftmost) position. In a magazine, this is position 1 (see Figure 13-1).
- If you specify the number of a particular diskette position, the operation begins with the diskette in that position. The number can be from 1 to 3 for diskettes in I/O slots or from 1 to 10 for diskettes in magazines (see Figure 13-1). The number 1 indicates the diskette in the first (leftmost) position in the slot or magazine.

VOL Parameter: Use this parameter as described in *Saving Objects* to specify whether particular diskette volumes are to be used. If you do not enter this parameter, the diskette in the position currently selected or the tape currently on the device is used.

Notes:

1. You can determine the location of the user profiles on the set of system diskettes by entering the Display Diskette (DSPDKT) command with the parameters LOC(*M12) DATA(*SAVRST). See *Determining Diskette Contents* in this chapter. Examine the displayed contents of each save/restore file until you find the file containing object type *USRPRF. On this display, the last digit of the VOLUME name is the position in the magazine where the user profiles reside.
2. Depending on the security restrictions of your system, you may not be allowed to enter this command. If you find that you cannot, ask the system security officer to enter it.

SEQNBR Parameter: This parameter is used for tape only and specifies the sequence number where the restore is to begin.

- If you do not specify this parameter, the file to be restored will be searched for starting with the current position on the tape.

- If you specify a sequence number, the file will be restored from that sequence number.

ENDOPT Parameter: If magnetic tape is to be used, use this parameter to specify how the tape is to be positioned after the save operation ends. If you do not enter this parameter, the tape is rewound to the load point (BOT marker).

Examples:

1. To restore the user profiles, starting at the diskette where the last save/restore ended, enter

   ```
   RSTUSRPRF
   ```

2. To restore user profiles that had been previously saved on the diskette volume SAVE, starting at the first diskette in magazine 2, enter

   ```
   RSTUSRPRF LOC(*M2 *FIRST) VOL(SAVE)
   ```

3. To restore user profiles that had been previously saved on tape mounted on QTAPE1, enter

   ```
   RSTUSRPRF DEV(QTAPE1)
   ```

Restoring Object Authority

When user objects and profiles are restored to the system, the authority of users to use the objects they had used previously is not automatically restored. You restore that object authority by using the Restore Authority (RSTAUT) command.

To execute the RSTAUT command, enter

```
RSTAUT
```

This is the only possible form of the RSTAUT command. It has no parameters.

This command cannot be used until user profiles are restored to the system. Before using this command, you must terminate all subsystems by entering

```
TRMSBS SBS(*ALL) or TRMCPF
```

(see *Terminating All Subsystems* in Chapter 8).

Note: Depending on the security restrictions of your system, you may not be allowed to enter this command. If you find that you cannot, ask the system security officer to enter it.

SAVING AND RESTORING THE SYSTEM

You cannot save or restore the entire system by using a single command. You must use a combination of commands or a combination of commands and other procedures.

When either saving or restoring the system, no other operation can be performed.

Saving the System

Saving the entire system is a two-step process:

1. Save system libraries using the SAVSYS command (see *Saving System Information* in this chapter).

2. Save all user libraries, including QGPL and program product libraries such as QRPG, using the SAVLIB command with the LIB(*NONSYS) option (see *Saving Libraries* in this chapter).

Restoring the System

To restore the system, you must reinstall the CPF. The restore operation is a four-step process that should be done in coordination with the system security officer or an equivalent authority:

1. Install the CPF:
   a. Mount the diskettes or tapes containing basic system information (saved by the SAVSYS command). If the basic system information was saved on tape, mount the diskette containing the install programs in position 1 of magazine 1.
   b. Start the system using the IMPL-with-AIPL setting and procedure described in *Settings to Install CPF* in Chapter 1.
   c. Press the Enter key without changing either the install type prompt or the install prompt, unless you need to recreate damaged system objects or restore the I/O configuration (see *Installing CPF* in Chapter 8).
   d. Sign on and start CPF as described in *Preparing System/38 for Operation* in Chapter 8. If you are installing CPF after auxiliary storage initialization, refer to the note regarding system values containing library lists under *Installing CPF* in Chapter 8.

2. Restore user profiles using the RSTUSRPRF command (see *Restoring User Profiles* in this chapter).

3. Restore all user libraries using the RSTLIB command with the SAVLIB(*NONSYS) option (see *Restoring Libraries* in this chapter).

4. Restore user authority using the RSTAUT command (see *Restoring Object Authority* in this chapter).

The steps must be done in the sequence shown.

SYSTEM BACKUP

The System/38 save/restore functions provide the capability for copying vital system information onto diskettes or tape, which can then be removed and stored as a backup. Should CPF programming or system microcode, or user programs or data, be lost as the result of a system failure, there are procedures for restoring them to the system by using the operator/service panel and the diskette magazine drive or tape drives. If a failure occurs that terminates the system, the effect on jobs being processed depends on where they were in the system.

To provide a backup for the system, you use the Save commands (SAVOBJ, SAVLIB, SAVSYS) described in this chapter to copy vital system information onto diskettes or tape.

CPF Information

When CPF was installed on your system, the SAVSYS command was used to copy essential CPF information on diskette or tape. When programming changes (PCs) are applied to the CPF programming in your system, the SAVSYS command should be used again to save the information so that you always have an up-to-date version of CPF available as a backup.

User Information

Individual objects and special libraries created for the needs of users of your system are copied onto diskette or tape by means of the SAVOBJ and SAVLIB commands. Individual users can save the objects or libraries they own; but you must save other objects and libraries. Vital information that is continually changing should be saved on a regular basis. Your operating procedures should describe what objects and libraries should be saved and at what interval, whether that be hourly, daily, weekly, or some longer period.

Diskette Supply

As more and more information is backed up on diskettes, the time may come when your supply of diskettes is exhausted. You should keep a record of how many diskettes you have available and take the necessary steps to obtain additional diskettes when you have only a few remaining. Of course, if updated versions of the same information are continually being backed up, you can use the CLRDKT command to clear diskettes containing previous versions (see *Clearing Diskettes* in this chapter). The cleared diskettes can then be used to store new information.

If you find that all diskettes have been used and it is not possible to obtain additional diskettes, contact your manager or the data processing manager, whichever is appropriate, and suggest the possibility of clearing some of the less vital information from the diskettes until additional diskettes can be obtained.

STAND-ALONE DUMPS

When indicated by the problem determination procedures or requested by your service representative, you take the stand-alone dumps to provide information on the operation of system microcode for service personnel.

Diskette Requirements

The stand-alone dumps require specially prepared diskettes in a magazine identified as the CE Tools magazine.

You receive two sets of Service Library magazines with your system. At least one of the sets always contains the latest version of the microcode used on your system. The magazines are periodically updated by your service representative. For the stand-alone dump procedures, you use the Service Library Vol 1 magazine that has the highest periodic level (Per. Lvl.) number and the CE Tools magazine.

The CE Tools magazine is shipped to you with at least one diskette for a main storage dump. This diskette, which is labeled DMS, is in position 1 in the magazine (see Figure 13-1 in this chapter). If your system has more than a million bytes of main storage, there is a second DMS diskette in position 2 of the magazine. If you are not sure how much main storage is on your system, use two diskettes so you will be sure to get all the data. It is advisable to prepare additional DMS diskettes in the event more than one main storage dump might be needed. You can then replace the existing DMS diskette in position 1 (and 2) of the magazine by the diskette or diskettes you have prepared.

You prepare the diskettes for a main storage dump by placing them in the diskette magazine drive and either initializing them or renaming them, depending on their current sector (record) format. If they do not have the 1024-byte record format, such as the one used for the save/restore operations, you must initialize them by using the INZDKT command. You specify the *SAVRST format and a volume identifier of DMS in the command. The diskette must be type 2D (double-sided, double density). For example, to prepare a diskette as a main storage dump diskette, place it in I/O slot 1 of the diskette magazine drive and enter the command

```
INZDKT LOC(*S1) NEWVOL(DMS) FMT(*SAVRST)
```

If a diskette to be used already has the 1024-byte record format (as indicated on the diskette label), you can prepare it as a main storage dump diskette by using the RNMDKT command to rename it. For example, to prepare a magazine of 1024-byte diskette for main storage dump, place the diskette in slot 1 of the diskette drive and enter the command

```
RNMDKT LOC(*S1) NEWVOL(DMS)
```

If you do prepare diskettes for use in a main storage dump, ensure that the correct sector size (1024 bytes) and the volume identifier (DMS) is shown on the label on the diskette jacket.

Main Storage Stand-alone Dump

The first dump is from main storage.

1. Ensure that a DMS diskette is in position 1 (and position 2 if you have more than one million bytes of main storage) of the CE Tools magazine.

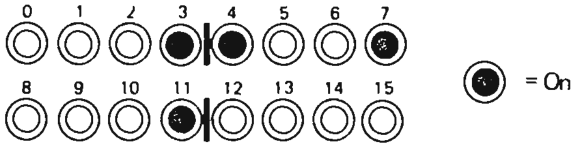
2. Place the Service Library Vol 1 magazine in position 1 of the diskette magazine drive and the CE Tools magazine in position 2.

3. Set the two rotary switches to the Dump MS setting shown below:

SW1	SW2
9	0

4. Press the Load switch.

When the dump is completed, the condition indicators will display the hexadecimal code 1910; that is, they will come on in the following pattern:



Remove the Service Library Vol 1 magazine and the CE Tools magazine and save the dump diskette(s) on the CE Tools magazine for the service representative.

The next step is to continue following the procedure in the PDP Guide that told you to take the main storage dump. The procedure in the PDP Guide may direct you to perform a virtual storage dump, do an IMPL, do a re-install, or call your IBM service representative.

Chapter 14. System Updating

INSTALLING PROGRAMMING CHANGES

IBM periodically does system updating by distributing programming changes (PCs) to correct logic problems within a particular program product. A PC is usually generated to close an acceptable authorized problem analysis report (APAR). An APAR is originated primarily by IBM service representatives who have encountered logic problems after performing in-depth problem determination at a System/38 user location. IBM customer personnel can also originate an APAR.

The System/38 program product code is written in modules and distributed as a complete program. When the program product is installed, it resides in the auxiliary disk storage. When a PC is created, it is designed to fully replace specific objects in a particular program product. The PC contains the object replacements as well as a PC description object. A PC should be applied only to a program product that is inactive.

There are two categories of PCs: deferred and immediate.

Because some CPF functions are active whenever CPF is active, PCs to these functions must be *deferred* to the next time CPF is started. When CPF is being started and deferred PCs are in a temporary state, you are notified of the pending deferred PCs (through the deferred PC display, prior to the start CPF prompt).

PCs to program products, other than CPF, can be applied *immediately* if the particular program product is not in use.

PCs can be applied (made active) to a program product on either a *temporary* or *permanent* basis. If the PC is applied temporarily, a copy of the old version of the object(s) being changed is saved and managed by the service function of CPF. If the PC is applied permanently, the old version of the object(s) is destroyed. Permanently applied PCs cannot be removed.

DETERMINING PROGRAM CHANGE (PC) STATUS

There may be times when you want to determine the status of program changes and program patches on your system. This can be done by using the DSPPGMCHG command.

The Display Program Change (DSPPGMCHG) command displays or lists each programming change (PC) and locally generated programming change (PATCH) for a specified program product and library. For a PC to be displayed, it must have been loaded on the system.

Obtaining the Displays

To obtain the Program Change display, enter the Display Programming Change (DSPPGMCHG) command. The command has the following parameters:

PGMID Parameter: This parameter specifies the program product for which PCs are to be displayed.

- If you do not enter this parameter, PCs for all program products are displayed.
- If you enter the name of a program product, the PCs for that program product are displayed.

LIB Parameter: This parameter specifies the name of the library from which PCs are to be displayed.

- If you do not enter this parameter, all the libraries on the system are searched for PCs to display.
- If you enter a library name, the PCs from that library are displayed.

SELECT Parameter: This parameter specifies which PC is to be displayed.

- If you do not enter this parameter, all PCs and patches for the specified program product in the specified library are displayed.
- If you enter a PC identification number, that PC is displayed.

OUTPUT Parameter: This parameter specifies whether the output is to be displayed at the requesting work station or listed with the job's spooled printer output.

- If you do not enter this parameter, the output will be displayed at the requesting work station.
- If you enter *LIST, the output will be listed with the job's spooled output.

Examples:

1. To display at your work station the status of all PCs for all program products in all libraries, enter

   ```
   DSPPGMCHG
   ```

2. To list on a printer the status of PCs for the program product 5714SS1 in the QSYS library, enter

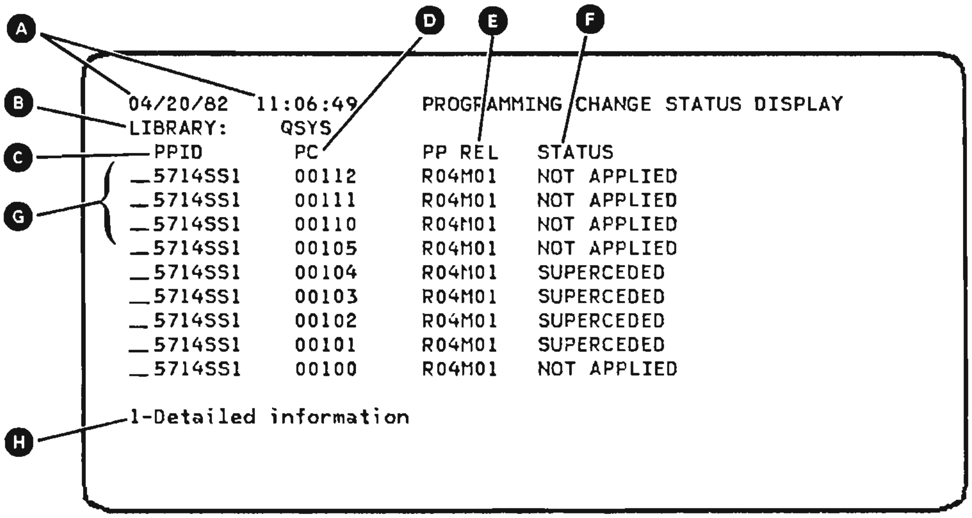
   ```
   DSPPGMCHG PGMID(5714SS1) LIB(QSYS) OUTPUT(*LIST)
   ```

3. To list detailed information about PC 00034 for program product 5714SS1 in QSYS library, enter

   ```
   DSPPGMCHG PGMID(5714SS1) LIB(QSYS) SELECT(00034)
   OUTPUT(*LIST)
   ```

Interpreting the Program Change Display

If you enter DSPPGMCHG, you receive a display of the following form:



```
04/20/82  11:06:49       PROGRAMMING CHANGE STATUS DISPLAY
LIBRARY:     QSYS
  PPID        PC         PP REL    STATUS
_5714SS1    00112        R04M01    NOT APPLIED
_5714SS1    00111        R04M01    NOT APPLIED
_5714SS1    00110        R04M01    NOT APPLIED
_5714SS1    00105        R04M01    NOT APPLIED
_5714SS1    00104        R04M01    SUPERCEDED
_5714SS1    00103        R04M01    SUPERCEDED
_5714SS1    00102        R04M01    SUPERCEDED
_5714SS1    00101        R04M01    SUPERCEDED
_5714SS1    00100        R04M01    NOT APPLIED

1-Detailed information
```

Ⓐ *Date/Time:* This is the date and time you requested the display.

Ⓑ *Library:* This is the name of the library in which the PC resides.

Ⓒ *PPID:* This the program product identification number that the PC applies to.

Ⓓ *PC:* This is the programming change identification number.

Ⓔ *PP Rel:* This is the current level of the program that is installed.

Ⓕ *STATUS:* This is the current status of the PC. The following values may be displayed:

Damaged

Superceded

Temporarily applied

Not applied

Permanently applied

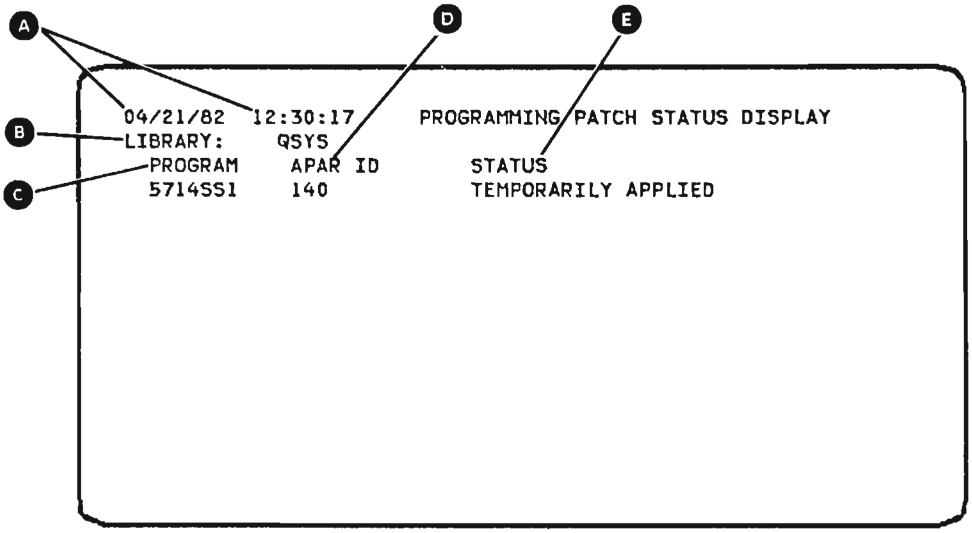
Ⓖ *Input field:* Enter the option Ⓗ that you want to take here.

Ⓗ *Options*: If you enter a 1 in the option fields Ⓖ you will receive a detailed display for the PC.

If you want to display program patches, press the Enter key.

Interpreting the Program Patch Display

If you enter the DSPPGMCHG command and there are patches in the libraries, you receive a display of the following form:



```
04/21/82  12:30:17      PROGRAMMING PATCH STATUS DISPLAY
LIBRARY:    QSYS
  PROGRAM    APAR ID        STATUS
  5714SS1    140            TEMPORARILY APPLIED
```

Ⓐ *Date/Time:* This is the date and time you requested the display.

Ⓑ *Library:* This is the name of the library in which the patch resides.

Ⓒ *Program:* This the name of the patched program.

Ⓓ *APAR ID:* This is the APAR number assigned to the patch.

Ⓔ *STATUS:* This is the status of the patch. This can be:

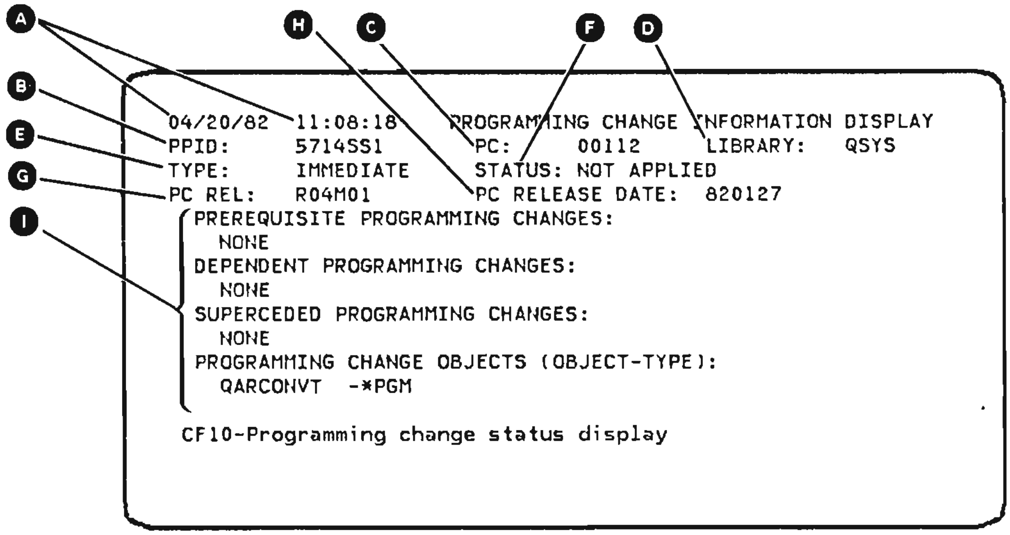
Damaged

Temporarily applied

Not applied

Interpreting the Detailed Programming Change Display

If you enter the DSPPGMCHG command and take option 1, you receive a display of the following form:



```
04/20/82   11:08:18    PROGRAMMING CHANGE INFORMATION DISPLAY
PPID:      5714SS1       PC:      00112      LIBRARY:   QSYS
TYPE:      IMMEDIATE     STATUS: NOT APPLIED
PC REL:    R04M01        PC RELEASE DATE:  820127
 PREREQUISITE PROGRAMMING CHANGES:
   NONE
 DEPENDENT PROGRAMMING CHANGES:
   NONE
 SUPERCEDED PROGRAMMING CHANGES:
   NONE
 PROGRAMMING CHANGE OBJECTS (OBJECT-TYPE):
   QARCONVT  -*PGM

 CF10-Programming change status display
```

Ⓐ *Date/Time:* This is the date and time you requested the display.

Ⓑ *PPID:* This the program product identification number that the PC applies to.

Ⓒ *PC:* This is the programming change identification number.

Ⓓ *Library:* This is the name of the library in which the PC resides.

Ⓔ *Type:* This is the PC type. It can be:

- Deferred
- Immediate

Ⓕ *STATUS:* This is the current status of the PC. The following values can be displayed:

Damaged

Superceded

Temporarily applied

Not applied

Permanently applied

Ⓖ *PC Rel:* This is the release level of the program product that the PC applies to.

Ⓗ *PC release date:* This is the date the PC was released.

Ⓘ *Prerequisite programming changes:* A list of PCs that are prerequisites for this PC.

Prerequisite engineering changes: A list of ECs that are prerequisites for this PC.

Dependent programming changes: A list of PCs that are dependent on this PC.

Superceded programming changes: A list of PCs that have been superceded by this PC.

Programming change objects: A list of CPF objects that are affected by this PC.

Special handling programs: A list of exit programs invoked by this PC.

APARs fixed: A list of APAR IDs that this PC fixes.

Descriptive text: A description of the PC.

Instructive text: Instructions on how to apply and use the PC.

Programming Change Installation

Basically, installing a PC is a two-step process:

1. Load the PC into the system by using the Load Programming Change (LODPGMCHG) command.

2. Apply the PC to the appropriate program product.
   a. Deferred PCs are applied through the deferred programming changes display that appears only when CPF is started.
   b. Immediate PCs are applied through the Apply Programming Change (APYPGMCHG) command.

CPF maintains a record of all programming changes that are installed. When PCs are applied, the system determines whether any prerequisite PCs have been applied. The following conditions must exist before PCs can be applied:

- If you are applying a PC temporarily, prerequisite PCs must have been applied as either permanent or temporary.

- If you are applying a PC permanently, prerequisite PCs must have been applied as permanent.

If the PC you are loading supersedes previous PCs, the new PC replaces the previous PC. If the previous PC was applied as temporary, you must remove it before you can load the new PC. You remove a deferred PC by using the deferred programming changes display that appears during the system startup process; you remove other PCs by entering the Remove Programming Change (RMVPGMCHG) command.

Loading the Programming Change

The Load Programming Change (LODPGMCHG) command is used to load immediate and deferred PC objects from a diskette into the specified library for a particular program product (for example, QSYS for CPF). PCs are not made active at the time they are loaded into a library.

- Immediate PCs are made active through the Apply Programming Change (APYPGMCHG) command.

- Deferred PCs are made active through the programming changes display.

The PC diskette(s) can contain many PCs. You have the option of loading all of them or selecting or omitting one or a few of them.

As PCs are loaded, the proper release level of the program product library is verified. If the release level prerequisite is met, then (for each PC being loaded) a check is made for prerequisite PCs and ECs. The check is satisfied if the prerequisite ECs are installed and the prerequisite PCs are either loaded, applied temporarily, or applied permanently. If the prerequisites are not on the system, the loading of the PC is terminated.

You can determine the history of PC activity by displaying the programming change log (QCHG). You can determine the current status of and attributes of all PCs loaded on the system by using the Display Programming Change (DSPPGMCHG) command.

To determine the current EC status of the microcode, note which service library magazines show the highest periodic level.

If the PC being loaded supersedes an existing PC, the following considerations apply:

- If the superseded PC is permanently applied, it is considered to be a part of the base system. The load PC operation is completed. The new PC can now be applied or removed.

- If the superseded PC is not applied to the system, the load operation is completed. The new PC can now be applied or removed.

- If the superseded PC is temporarily applied, the load operation is terminated. The superseded PC must first be removed or applied permanently.

Use the following procedure to load the PC:

1. Place the PC diskette (or diskettes) in the diskette magazine drive.

2. Key in the LODPGMCHG command, specifying the parameters as follows:

Parameter	**Value Required**
PGMID	5714SS1 (if the PC is for CPF) 5714RG1 (if the PC is for RPG III) 5714CB1 (if the PC is for COBOL) 5714UT1 (if the PC is for IDU) 5714CV2 (if the PC is for the Conversion Reformat Utility)
LIB	Name of the library containing the program product
VOL	Volume identifier of the diskette or diskettes containing the PCs to be loaded
DEV	Name of the diskette device
LOC	Location in the diskette drive where you placed the PC diskette or diskettes (see Figure 13-1 in Chapter 13). Do not use *CURRENT if the programming change is on more than one diskette.
SELECT	PC numbers (if only specified PCs in the diskette volume are to be loaded)
OMIT	PC numbers (if specified PCs in the diskette volume are *not* to be loaded)

   To ensure that you are using the correct library name and volume identifier, refer to the *Memo to Users* that accompanied the programming change. The DEV, LOC, SELECT, and OMIT parameters are optional. If you do not specify the DEV parameter, QDKT is assumed. If you do not specify the LOC parameter, slot 1 (*S1) is assumed. You cannot specify both the SELECT and the OMIT parameters. If you do not specify either the SELECT or the OMIT parameters, all PCs in the specified volume are loaded.

3. Press the Enter key. The system loads all the PCs from the diskette or diskettes that pertain to the specified program product.

4. Remove the diskette or diskettes from the diskette magazine drive.

Applying an Immediate Programming Change

The Apply Programming Change (APYPGMCHG) command is used to apply (make active) an immediate PC. The program elements to which the PC is being applied must be inactive at the time.

An immediate PC can be applied either temporarily or permanently. If the PC is applied temporarily, the replaced objects are saved and managed by the service function of CPF. If the PC is applied permanently, the replaced objects are destroyed.

As a PC is applied, prerequisite checking is again performed to ensure that any PC that is prerequisite to the PC being applied has been correctly applied.

Use the following procedure to apply the PC:

1. Key in the APYPGMCHG command, specifying the parameters as follows:

Parameter	**Value Required**
PGMID	5714SS1 (if the PC is for CPF) 5714RG1 (if the PC is for RPG III) 5714CB1 (if the PC is for COBOL) 5714UT1 (if the PC is for IDU) 5714CV2 (if the PC is for the Conversion Reformat Utility)
LIB	Name of the library containing the program product
SELECT	PC numbers (if only the specified PCs are to be applied)
OMIT	PC numbers (if the specified PCs are *not* to be applied)
APY	*TEMP (to apply the PC temporarily) *PERM (to apply the PC permanently)

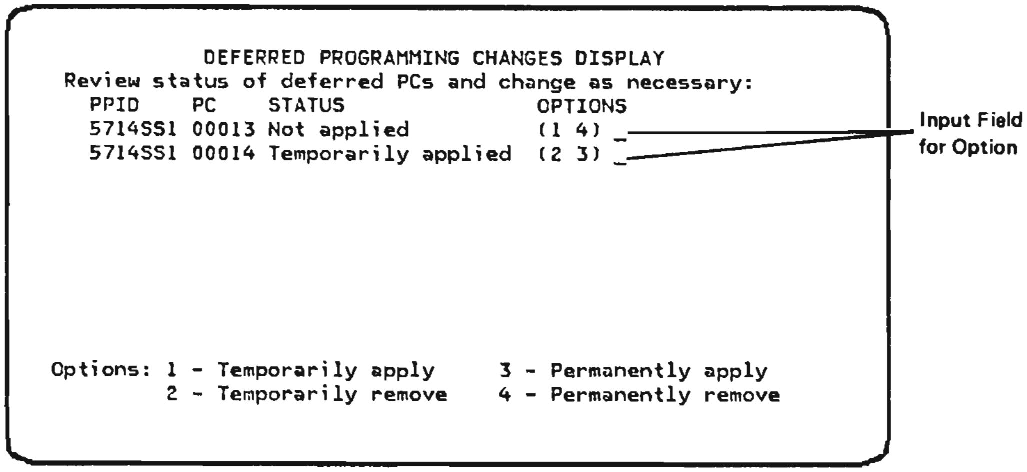
   The SELECT, OMIT, and APY parameters are optional. You cannot specify both the SELECT and OMIT parameters. If you do not specify either the SELECT or the OMIT parameters, all of the PCs are applied. If you do not specify the APY parameter, the PCs are applied temporarily.

   Note: PCs applied temporarily can be removed; PCs applied permanently cannot be removed (see *Removing a Programming Change* in this chapter).

2. Press the Enter key. The system applies the PCs to the specified program product.

Applying a Deferred Programming Change

If deferred PCs have been loaded or temporarily applied but not permanently applied, the next time you start the system a deferred programming changes display is shown prior to the start CPF prompt.



```
                DEFERRED PROGRAMMING CHANGES DISPLAY
    Review status of deferred PCs and change as necessary:
      PPID     PC    STATUS                  OPTIONS
      5714SS1 00013 Not applied              (1 4) _
      5714SS1 00014 Temporarily applied      (2 3) _







   Options: 1 - Temporarily apply     3 - Permanently apply
            2 - Temporarily remove    4 - Permanently remove
```

The STATUS column indicates the current status of the PC. You can change the status by entering one of the two numbers shown in the OPTIONS column; that is,

1. Key one of the option numbers into the input field provided.

2. Press the Enter key.

If you press the Enter key without keying in an option, the status of the PC is not changed.

Note: Two start CPF operations are necessary to permanently apply a deferred PC. For example, the following sequence of actions might be used to permanently apply a deferred PC:

1. Once CPF is started, use the Load Program Change (LODPGMCHG) command to load the PC.

2. Use the Power Down System (PWRDWNSYS) command to power down the system.

3. Set the rotary switches on the operator/service panel for an IMPL Abbr (see *Normal Settings* in Chapter 1). Press the Power On switch.

4. Sign on to the system.

5. Apply the PC temporarily by keying in option 1 (options 1 and 4 would be offered). Press the Enter key.

6. Repeat steps 2, 3, and 4.

7. Apply the PC permanently by keying in option 3 (options 2 and 3 would be offered). Press the Enter key.

8. Continue with system operations.

If this display indicates that a PC has been temporarily removed, you can use the display to either temporarily reapply the PC or permanently remove it.

Removing a Programming Change

You use the RMVPGMCHG command to remove immediate programming changes that have been applied as temporary, or loaded but not applied. When the PC is removed, you can retain it on the system or delete it from the system. All the temporary PCs that have been applied to a program product can be removed in a single operation, or specific PCs can be removed. Permanently applied PCs cannot be removed.

Immediate PCs can be removed either temporarily or permanently. When the PC is removed permanently, the objects are destroyed. When the PC is removed temporarily, the objects are saved and managed by the service function of CPF. A temporarily removed PC can be reapplied at a later time.

When a temporarily applied PC is removed, the original objects that were replaced by the PC are restored to the program product library. Prerequisite checking is again performed to ensure the PC being removed is not a prerequisite to any other applied PC. If the PC being removed is found to be a prerequisite, the other PC (to which it is a prerequisite) must be removed first.

Use the following procedure to remove an immediate PC:

1. Key in the RMVPGMCHG command, specifying the parameters as follows:

Parameter	**Value Required**
PGMID	5714SS1 (if the PC is for CPF) 5714RG1 (if the PC is for RPG III) 5714CB1 (if the PC is for COBOL) 5714UT1 (if the PC is for IDU) 5714CV2 (if the PC is for the Conversion Reformat Utility)
LIB	Name of the library containing the program procuct
SELECT	PC numbers (if only specified PCs are to be removed)
OMIT	PC numbers (if only specified PCs are *not* to be removed)
RMV	*TEMP (if the PCs are to be retained on the system) *PERM (if the PCs are to be deleted from the system)

The SELECT, OMIT, and RMV parameters are optional. You cannot specify both the SELECT and the OMIT parameters. If you do not specify either the SELECT or the OMIT parameter, all of the PCs in the specified program product that were temporarily applied or not applied are removed. If you do not specify the RMV parameter, the PCs are removed, but retained in the system for possible reapplication later.

2. Press the Enter key. The system removes the PCs.

You remove a deferred programming change by selecting the appropriate option on the deferred programming changes display (see the previous discussion of *Applying a Deferred Programming Change* in this chapter).

INSTALLING MICROCODE

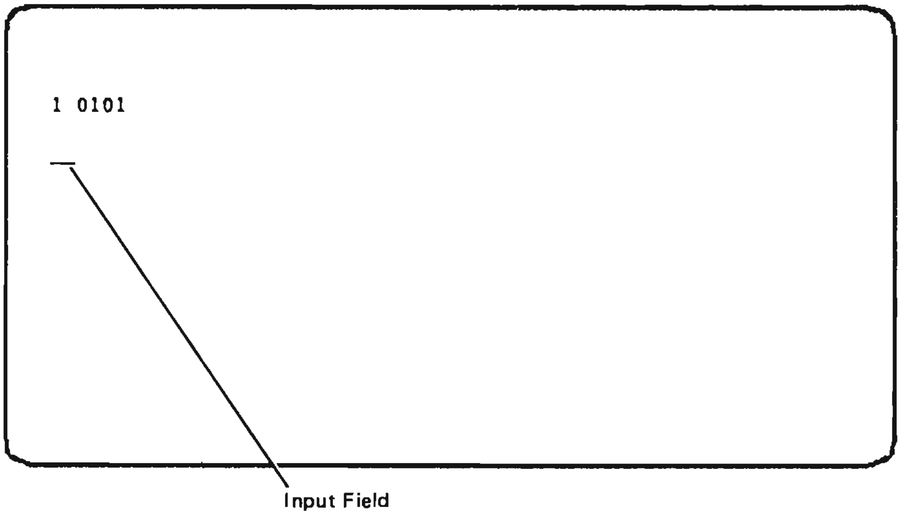
Two sets of diskette magazines are supplied with your system to provide a backup for the microcode on your system. Each set consists of two magazines, one labeled Service Library Vol 1, the other labeled Service Library Vol 2. When the system microcode is changed, the IBM service representative will incorporate the new version of the microcode into one set of these Service Library magazines. Using the procedure described here, you can install the new version at a convenient time when the system is not being used, such as before normal operations begin or after they have ended. All system activity must be terminated before the microcode is installed.

The IBM service representative will incorporate the new version of the microcode into the magazines that do not contain the most recent version, so that one set of Service Library magazines will contain the new version and the other set will contain the previous version. The newest version will be in the Service Library magazines whose labels show the highest periodic level (Per. Lvl.) number. If you are unable to install the new version of the microcode successfully, or if you have a problem with the new version, you can usually reinstall the previous version and resume normal operations until the problem can be corrected.

Should a system failure occur that results in the loss of the microcode, you would also use the procedure described here to reinstall the system microcode.

Using the Service Monitor for Microcode Installation

Part of the microcode installation procedures involve using the dedicated service monitor that is activated by the Alt IMPL or Alt IMPL Abbr setting on the operator/service panel (see *Settings to Activate the Service Monitor Mode* in Chapter 1). When the service monitor mode is activated, the following display appears on the system console screen:



```
1 0101

_
```

The service monitor mode operates through numeric codes. You enter a specific code in the input field provided on the left side of the screen, and one or more different codes are displayed in response. Some of these codes that are displayed will require a response from you; others will not. The first digit of the code indicates whether a response is required and the type of response requried:

- If the first digit is 1, such as 1 0120 you must respond by entering a code in the input field provided.
- If the first digit is 2, such as 2 0200 you must respond by pressing the Enter key.
- If the first digit is 3, such as 3 0220 no response is required.

The actual codes you enter and receive on the screen are described in the following stand-alone dump procedures. Whether using the stand-alone dump functions described here or another service monitor function, you will see one or more general service monitor codes that require a response from you. the codes and their meanings are:

Code	Meaning
1 0102	An error occurred while the data was being loaded from the diskette. Enter the code 05 to eject the diskette and to reset the diskette magazine drive.
1 0103	The funciton you requested has ended. Select another function or enter 01 to end the service monitor mode.
1 0104	The diskette has been ejected. Select another function.
1 0105	Your entry was not valid. enter a valid code.
1 0120	You have requested that the service monitor mode be ended. Enter the code 01 to complete the termination.
1 0121	Your entry was not valid. Enter the code 01 to end the service monitor mode or the code 02 to return to the previous function.
1 0130	The service monitor is unable to allocate sufficient main storage space for the requested function to run at its best performane. Enter the code 01 to end the service monitor mode or the code 02 to continue the service function with the available space.

Starting the Microcode Installation Process

Use the following procedure to start the microcode installation:

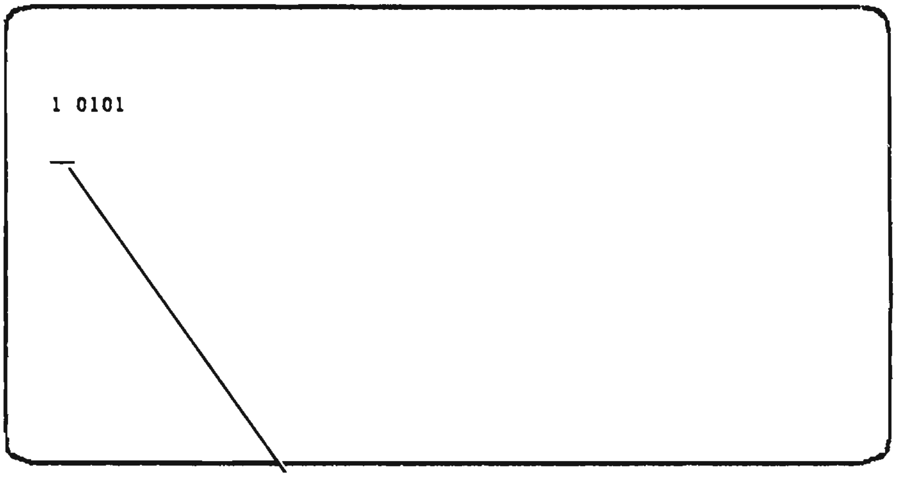
1. If the system is currently in operation, enter the TRMCPF command to terminate operations.

2. Place the Service Library Vol 1 magazine into magazine position 1 and the Service Library Vol 2 magazine into magazine position 2 of the diskette magazine drive (see Chapter 3 for the procedures for loading the magazines).

3. Set the two rotary switches to the Alt IMPL setting described under *Settings to Activate the Service Monitor Mode* in Chapter 1.

4. Press the Power On switch if system power is off (Power On switch is not lighted), or press the Load switch if system power is on (Power On switch is lighted).

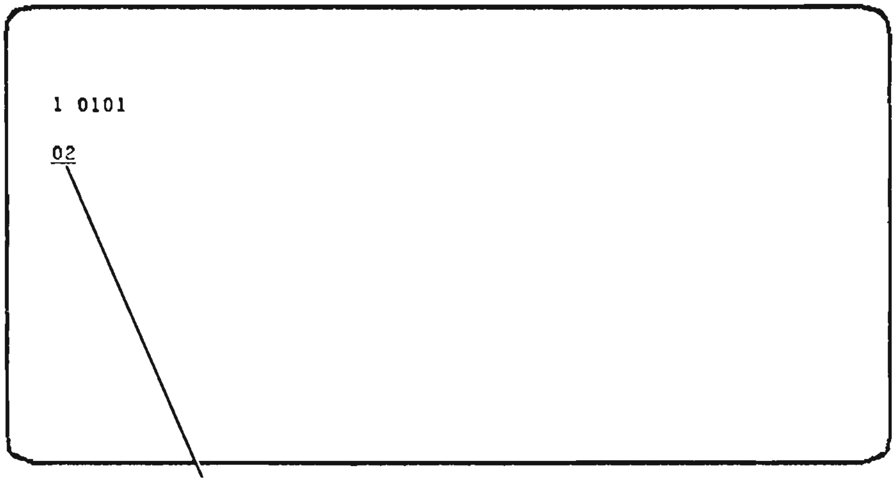
   When the Alt IMPL sequence is completed, you will see the following display on the system console screen:



```
1 0101

_
```

Input Field

5. Enter the code 02 in the input field.



```
1 0101

02
```

Enter 02 here.

| *Interacting with the Displays During the Microcode Installation*

If the installation process is proceeding normally, the following codes will be displayed on your screen to indicate the current stage of the process:

Code	**Meaning**
3 0220	Activation
3 0240	Loading
3 0260	Nucleus build
3 0280	Address resolution
3 0290	Shut down

These codes are for your information only. No action is required by you. If an error occurs during the process, you may see only some of these codes.

Either during or at the end of the installation process, you will see a code beginning with a 2. In response to this type of code, you must press the Enter key. Any other actions you need to take will depend on whether an error occurred during the process.

Successful Completion: If the microcode is installed successfully, you will see the code

```
2 0200
```

Press the Enter key. The code

```
1 0103 0000
```

will be displayed in response. This indicates that you have returned to the basic service monitor mode.

Successful Completion, Data Set Missing: If you see the code

```
2 0240
```

followed by a string of characters such as

```
dsxxxx1 dsxxxx2 . . . dsxxxxn
```

the diskette volume for the interim/patch data set is missing or defective. You should do the following:

1. Write down what is shown on your screen.

2. Press the Enter key to continue the installation.

3. When you see the code

   ```
   2 0200
   ```

   press the Enter key again. The code

   ```
   1 0103 0000
   ```

   will be displayed, indicating that you have returned to the basic service monitor mode.

4. As soon as possible, contact your service representative and report what happened.

Installation Aborted: If you see a code beginning with a 2 that is different from the two codes described above, the installation is unsuccessful because of an error. You should do the following:

1. Write down all of the information shown on the screen, exactly as it appears.

2. Press the Enter key to return to the basic service monitor mode. The code

   ```
   1 0103 0000
   ```

   will be displayed in response.

3. Contact your service representative and report what happened.

If the microcode installation process fails, the cause of the failure will have to be determined and the microcode reinstalled before you can resume normal operations. However, if you were attempting to install a new version of the microcode when the failure occurred, you may be able to reinstall the previous version and continue operations until the problem with the new version of the microcode is corrected.

Ending the Service Monitor Mode

Whether the microcode installation was completed or aborted, you return to the basic service monitor mode, as indicated by the code

```
1 0103 0000
```

on your screen. Enter 01 in the input field to end the service monitor mode. The code

```
1 0120
```

will be displayed in response. Enter 01 again to complete the termination. You can now remove the Service Library magazines.

To return the system to normal operations after installing the microcode, follow the procedures for *Preparing System/38 for Operation* in Chapter 8. If the system power is on, press the Load switch instead of the Power On switch to start the IMPL sequence.

If you will not be resuming operations, you can power down by using the Force Pwr Off setting (see *Settings Activated by Load Switch* in Chapter 1).

DISK MIGRATION UTILITY

The disk migration utility provides a means to transfer data from disk drives being removed from a system to disk drives being added to a system. This utility is used if you change the storage capacity of your system and you must copy your system to new disk drives. A typical use of the disk migration utility is shown in the following example.

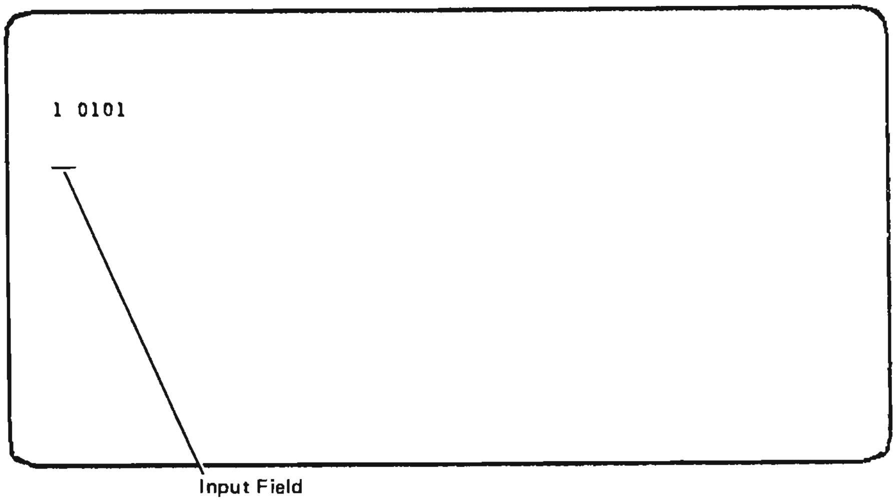
Example

You want to expand the storage capacity of your system by adding a 3370 Disk Storage Unit and discontiniuing 62PC disk drives that are presently on your system.

1. The IBM service representative physically attaches the 3370 to your system and performs auxiliary storage initialization to prepare the 3370 for use. If you choose to, at this point, you can return to normal processing and postpone the copying of your system to the 3370 to a more convenient time.

2. When you decide it is a convenient time, copy your system to the 3370 using the disk migration utility as shown below:

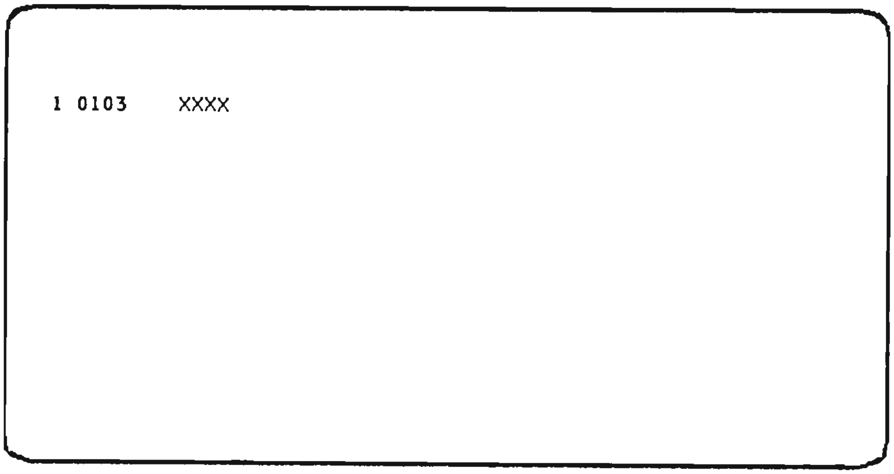
   Activate the service monitor mode as described in Chapter 1. You will see the following display on the system console screen:



```
1 0101

_
```

Enter the code 05 in the input field and press the Enter key.

The migration utility does not display any screens of its own. At completion you will be returned to the basic service monitor mode as indicated by the following system console screen:



```
1 0103    XXXX
```

XXXX represents the return code.

0000 in the return code indicates that the migration utility has completed successfully. Do an IMPL or an IMPL abbreviated, as described in Chapter 1, immediately. Do not attempt to run another service monitor function.

Any return code other than 0000 indicates an error. Record the code and call the service representative. You may do an IMPL or an IMPL abbreviated and run with the old system configuration until the problem is fixed.

3. If the disk migration utility was successful, you have now copied your system to your new storage configuration and you can resume normal processing even though the 62PCs are still in the machine.

4. At a convenient time, the IBM service representative will initialize the 62PCs that are to be removed so there is no data on them. The IBM service representative can then remove the 62PCs, and the installation is complete.

Appendix A. CPF Command Function Key Assignments

The following tables summarize the CPF command functions available to assist you in performing system operator tasks. Note, however, that you may not be able to use these functions while your work station is under control of another program, such as a program you have called by using the CALL command.

Functions Valid for All CPF

Key	Function	Description
CF1	Exit function (return to original display)	Provides a permanent exit from a function. If you entered a command that requests a separate display, pressing CF1 signals the end of the request so that you return to the original display on which you entered the command (such as the system operator menu). If you are in the process of selecting or entering a command through a menu or prompt display, pressing CF1 terminates the prompting sequence and returns you to your original display to start a new entry.
CF2	Backup to previous display	When viewing a series of displays, you can press CF2 to back up to the previous display in the series. If the first display in the series is being displayed when you press the CF2 key, you are returned to the caller.
CF24	Print (system console)	Serves as the Print key for the system console. When you press CF24, the display on the system console screen is printed on the system printer or an alternative printer assigned by the system.

Functions Valid on the System Operator Menu		
Key	**Function**	**Description**
CF3	Command entry	Requests the command entry display.
CF4	Prompt key	Requests prompting assistance. Pressing CF4 after selecting option 5 on the system operator menu, but without making any other entry, results in a menu display of command categories. Pressing CF4 after selecting option 5, and entering at least a command name, results in a prompt display for the parameters of the selected command.
CF6	Display messages in QSYSOPR	Requests a display of all messages currently in the system operator message queue.
CF7	Display subsystems	Requests a display of all jobs in all subsystems currently active in the system.
CF8	Display system	Requests a display of general information about all subsystems that are currently active in the system or being terminated.
Help	Second-level display	Requests the second-level message display for the message on the line where the cursor is positioned.

Functions Valid on the Command Entry Display		
Key	**Function**	**Description**
CF3	Duplicate command	Duplicates any previously entered command into the current input field. Position the cursor on the line containing the command to be duplicated and press CF3. The command then appears in the input field following the :: symbol. If the cursor is in the input field, the last command entered is duplicated.
CF4	Prompt key	Requests prompting assistance. Pressing CF4 without making an entry on the command entry display results in a menu display of command categories. Pressing CF4 after entering a partial command, or at least a command name, results in a prompt display for that command. If a parameter on a prompt can contain another complete command (an embedded command), pressing CF4 again results in a separate prompt display for the embedded command.
CF7	Display detail messages	Requests a display of detail messages not normally shown on command entry display. If the cursor is in the input field, only detail messages related to the last command entered are shown. If the cursor is positioned at a previously entered command, detail messages related to that command and all following commands are shown.
Help	Second-level display	Requests the second-level message display for the message on the line where the cursor is positioned.

Functions Valid During Prompting		
Key	**Function**	**Description**
CF13	Display key assignments	Displays all key functions available to assist you during prompting.
CF14	Display command string	When using a prompt display to enter values for parameters of a command, you can press CF14 to display the command in string form as you would enter it on the command entry display. The command string shown contains all parameters for which you have entered a value.
CF15	Display error messages	Obtains a display of all error messages you received since you signed on.
CF16	Enter command—command complete	Signals the system to consider the command complete as entered and attempt to execute it, even if there are prompt displays to follow that contain additional parameters for the command. The command will be executed successfully only if a default value exists for all parameters you have not entered.
CF18	Revert to command defaults	Returns a prompt display to its original form with all default values restored. Any values you had previously entered are destroyed.

Functions Valid on Message and System Log Displays		
Key	**Function**	**Description**
CF6	Remove a message	Removes from the message queue the message on the line of the display where the cursor is positioned (not valid on system log displays).
CF7	Display all messages	Requests that all messages be shown on the display, including messages that had been filtered out by the DSPMSG or DSPLOG command that originally requested the message display.
CF8	Remove all messages	Removes all messages from the message queue (not valid on system log displays).
Help	Second-level display	Requests the second-level message display for the message on the line where the cursor is positioned.

Functions Valid on Work Management and Spool Displays		
Key	**Function**	**Description**
CF5	Update display	Updates a display with current values as if the command was reissued.

Glossary

This glossary includes terms and defintions from the *IBM Vocabulary for Data Processing, Telecommunications, and Office Systems*, GC20-1699.

abnormal termination: System termination by a means other than the successful execution of the Power Down System (PWRDWNSYS) command. See also *system termination* and *normal 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*.

active file: A diskette file, or tape file whose expiration date is greater than the system date.

activity level: An attribute of a storage pool or the system that specifies the maximum number of jobs that can execute concurrently in the storage pool or in the system.

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.

AIPL: See *alternative initial program load*.

allocate: To assign a resource for use in performing a specific task. Contrast with *deallocate*.

allowable resources: Attributes of a process that identify which resources the process is allowed to allocate. Some of these attributes are obtained at process creation time from the user profile (maximum resources).

alphabetic character: Any one of the letters A through Z (uppercase and lowercase) or one of the characters #, $, or @.

Alt IMPL: See *alternative initial microprogram load*.

Alt IMPL Abbr: See *alternative initial microprogram load abbreviated*.

alternative initial microprogram load: The process of loading the System/38 microprogramming code from diskettes (rather than auxiliary storage) and then activating the code. Abbreviated Alt IMPL on the operator/service panel.

alternative initial microprogram load abbreviated: The process of loading the System/38 microprogramming code from a diskette (rather than auxiliary storage and then activating the code to perform system startup, bypassing certain hardware tests. Abbreviated Alt IMPL Abbr on the operator/service panel.

alternative initial program load: A process, when combined with the IMPL sequence, that prepares the system for operation and installs CPF from the diskette magazine drive. Abbreviated AIPL on the operator/service panel.

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.

APAR: See *authorized program analysis report*.

application: (1) A particular data processing task, such as an inventory control application or a payroll application. (2) In IDU, specialized program created by IDU from user input.

application program: A program used to perform a particular data processing task such as inventory control or payroll.

arithmetic instruction: An instruction that specifies an arithmetic operation.

arithmetic operation: An operation such as addition, subtraction, multiplication, or division that is performed only on numeric fields.

arithmetic operator: A symbol that indicates the arithmetic operation to be performed. In CL, the arithmetic operators are + (addition), - (subtraction), * (multiplication), and / (division). In COBOL, the arithmetic operators include: + (addition), - (subtraction), * (multiplication), / (division), and ** (exponentiation).

arrival sequence access path: An access path that is based on the order in which records are stored in a physical file. See also *keyed sequence access path* and *access path*.

ascending key: The values by which data is ordered from the lowest value to the highest value of the key in accordance with the rules for comparing data items. Contrast with *descending key*.

ascending key sequence: The arrangement of data in an order from the lowest value of the key field to the highest value of the key field. Contrast with *descending key sequence*.

ascending sequence: The arrangement of data in an order from low to high based on the contents of a specific field or fields. Contrast with *descending sequence*.

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.

attribute character: A character associated with a field in a display file that defines how the field is displayed (such as underlined, blinking, or intensified).

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.

authorized program analysis report: A request for correction of a problem caused by a defect in a current unaltered release of a program. Abbreviated APAR.

autoanswer: See *automatic answer*.

autocall: See *automatic call*.

automatic answer: A machine feature that permits a station to respond to a call it receives over a switched line without operator action. Abbreviated autoanswer.

automatic call: A machine feature that permits a station to initiate a connection with another station over a switched line without operator action. Abbreviated autocall.

automatic variable: A variable that is allocated during the invocation of the program containing the variable. Every time a program is invoked a new copy of the variable is placed in storage. Contrast with *static variable*.

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.

base storage pool: A storage pool that contains all unassigned main storage on the system and whose minimum size is specified in the system value QBASPOOL. The system-recognized identifier is *BASE.

basic data exchange: A format for exchanging data on diskettes between systems or devices.

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*.

batch subsystem: A subsystem in which batch jobs are to be processed. IBM supplies one batch subsystem: QBATCH.

binary: Relating to, being, or belonging to a numbering system with a base of 2. Valid digits are 0 and 1.

binary format: Representation of a decimal value in which each field must be 2 or 4 bytes long. The sign (+ or -) is in the leftmost bit of the field, and the integer value is in the remaining bits of the field. Positive numbers have a 0 in the sign bit and are in true form. Negative numbers have a 1 in the sign bit and are in twos complement form.

binary synchronous communications: A form of communications line control that uses transmission control characters to control the transfer of data over a communications line. Abbreviated BSC.

block: A set of adjacent logical records recorded as a unit on a diskette or magnetic tape.

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.

break delivery: The method of delivering messages to a message queue in which the job associated with that message queue is interrupted as soon as the message arrives.

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.

BSC: See *binary synchronous communications*.

BSC file: A device file created by the user to support BSC. Contrast with *communications file*.

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).

CF key: See *command function key*.

character: Any letter, digit, or other symbol in the data character set that is part of the organization, control, or representation of data.

character field: An area that is reserved for a particular unit of information and that can contain any of the characters in the data character set. Contrast with *numeric field*.

character literal: A symbol, quantity, or constant in a source program that is itself data, instead of a reference to data. Contrast with *numeric literal*.

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*.

CNP: See *communications statistical network analysis procedure*.

cold start: A process in which all noninstalled objects (CPF objects created by CPF after installation) are deleted and re-created as a group.

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 analyzer: An IBM-supplied program that processes commands. Command processing includes validity checking, transferring control to a command processing program (CPP), and returning to the caller of the command analyzer.

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 definition statement: A source statement used in creating a command definition. Command definition statements define keywords and parameter values, qualified names, elements in a list, parameter dependencies and interrelationships, and prompt text for a command.

command file: In System/38 RJEF, a remote job input stream that can contain host system commands and job control language (JCL), data, and RJEF control statements (READFILE or EOF). Contrast with *data file.*

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 key indicator: In RPG, an indicator that is set on when a system operator or work station user presses a valid corresponding command key. Valid command key indicators are KA through KN and KP through KY.

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.

comment: A word or statement in a program, command, or file that serves as documentation instead of as instructions. A comment is ignored by a compiler.

common carrier: A government or private company that furnishes the general public with telecommunication facilities. Examples are: the government-regulated telephone and telegraph companies in the USA, the General Post Office in the United Kingdom, the Bundespost in Germany, and Nippon Telephone and Telegraph Public Corporation (NTT) in Japan.

communications adapter: A hardware feature that enables System/38 to become part of a data communications network.

communications file: A device file created by the user to support LU1 SDLC communications. Contrast with *BSC file.*

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.*

communications statistical network analysis procedure: A procedure that allows the service personnel to obtain statistics on communications line activity. Abbreviated CSNAP or CNP.

compilation: Translation of a source program (such as RPG or COBOL specifications) into an executable program.

compile: To translate a source program into an executable program (an object).

compiler: A program that translates a source program 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.

composite key: A key for a file or record format that is composed of more than one key field.

concatenated field: Two or more fields from a physical file record format that have been combined to make one field in a logical file record format.

consecutive processing: A method of file processing in which records are read in the order in which they exist in the file.

constant: Data that has an unchanging, predefined value to be used in processing. A constant does not change during the execution of a program, but the contents of a field or variable can. See also *literal.*

constant field: In an externally described display or printer file, an unnamed field that contains actual data that is passed to the display or printer but is unknown to the program passing it.

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 storage: High-speed memory containing microcode. This storage can be implemented as read only or direct access.

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*. Abbreviated CTL or CTLU.

control unit description: An object that contains a description of the features of a control unit that is either directly attached to the system or attached to a communications line. The system-recognized identifier for the object type is *CUD. Abbreviated CUD.

controlling subsystem: An interactive subsystem that is started automatically when the system is started and through which the system operator controls the system. IBM supplies one controlling subsystem: QCTL.

CPF: See *Control Program Facility.*

CPP: See *command processing program.*

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.

CSNAP: See *communications statistical network analysis procedure.*

CTLU: See *control unit.*

CUD: See *control unit description.*

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: (1) Any nonsource file. A data file is created by the specification of FILETYPE(*DATA) on a create file command. (2) In System/38 RJEF, a remote job input stream that can contain host system commands and job control language (JCL) as well as data. Contrast with *command file.*

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 rights: The authority to read, add, update (modify), or delete data contained in an object. See also *add rights*, *delete rights*, *read rights*, and *update rights.*

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.*

debug mode: An environment in which programs can be tested.

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 program: A user-specified program that is assumed when no other program is specifically named on a debug command, or a special program defined for handling error messages.

default record: A record in which numeric fields are initialized with zeros and character fields are initialized with blanks.

default reply: A system-assigned reply to an inquiry or notify message that is used when the message queue at which the message arrives is in default delivery mode.

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.

delay maintenance: A method of maintaining keyed access paths for data base files. This method does not update an access path when the file is closed, but it retains updates in a *delayed* form so that they can be quickly applied at the next open, avoiding a complete rebuild. Contrast with *rebuild maintenance* and *immediate maintenance*.

delete: (1) To remove an object or a unit of data (such as character, a field, or a record). (2) The SEU operation in which existing records can be removed from a source member.

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*.

deleted record: A record that has been initialized or removed so that it is not eligible for access. A deleted record holds a place in a physical file member and can be replaced with a data record by an update operation.

delimiter: A character or a sequence of contiguous characters that identifies the end of a string of characters. A delimiter separates a string of characters from the following string of characters. A delimiter is not part of the string of characters that it delimits.

descending key: The values by which data is ordered from the highest value to the lowest value of the key, in accordance with the rules for comparing data items. Contrast with *ascending key*.

descending key sequence: The arrangement of data in order from the highest value of the key field to the lowest value of the key field. Contrast with *ascending key sequence*.

descending sequence: The arrangement of data in an order from high to low based on the contents of a specific field. Contrast with *ascending sequence*.

DEVD: See *device description*.

device description: An object that contains information describing a particular device that is attached to the system. The system-recognized identifier for the object type is *DEVD. Abbreviated DEVD.

device file: An object that contains a description of how input data is to be presented to a program from an external device and/or how output data is to be presented to the external device from the program. External devices can be work stations, card devices, printers, diskette magazine drives, magazine tape drives, or a communications line.

device name: The symbolic name of an individual device. The name is specified when the device is defined to the system by the Create Device Description (CRTDEVD) command.

device type: The generic name for a group of device names. All the devices in a device type must have the same physical attributes and functions.

DFU: See *data file utility*.

diagnostic message: A message that contains information about errors in the execution of an application program or a system function.

diskette drive: The mechanism used to seek, read, and write data on diskettes. See also *diskette magazine drive*.

diskette file: A device file created by the user to support a diskette device.

diskette location: The slot into which the diskette is inserted before being read or written.

diskette magazine drive: A diskette drive that can hold two magazines, each containing 10 diskettes, plus individual diskettes in three separate slots. It is used to transfer information between system internal storage and removable diskettes.

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. Sometimes called a screen.

display file: A device file created by the user to support a display work station or console.

display format: The name of the device file and the name of the record format to be used when the subsystem obtains routing data from the user.

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.

EBCDIC: Extended binary-coded decimal interchange code. A coded character set consisting of 8-bit coded characters.

embedded blank: A blank that appears between characters.

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*.

error log: A record of machine checks, device errors, and volume statistical data.

escape message: A message that can be monitored for and that describes a condition for which a program terminates without completing the requested function.

exclusive-allow-read lock state: The allocation that a routing step has for an object in which other routing steps can read the object if they request a shared for read lock state for the same object. The predefined value for this lock state is *EXCLRD.

exclusive lock state: The allocation that a routing step has for an object in which no other routing steps can use the object. The predefined value for this lock state is *EXCL.

execute: To cause a program, command, utility, or other machine function to be performed.

execution: The carrying out of the instructions of a computer program by a processing unit.

external message queue: A message queue that is part of the job message queue and is used to send messages between an interactive job and the work station user. For batch jobs, messages sent to the external message queue only appear in the job log.

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.*

externally described file: A file for which the fields in the records are described to CPF, through 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 file.*

field: An area that is reserved and used for a particular item of information.

field reference file: A physical file that contains no members and whose record format describes the fields used by a group of files.

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 description: (1) The information contained in the file that describes the file and its contents. (2) In COBOL, an entry in the File Section of the Data Division that provides information about the identification and physical structure of a file.

file overrides: The file attributes specified at execution time that will override the attributes specified in the file description or in the program.

file reference function: A CPF function that lets the user track file usage on the system.

file separator: The pages or cards to be produced at the beginning of each output file to separate the file from the other files being spooled to an output device.

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*.

fold: To continue data for a line on the following printed or displayed line. Contrast with *truncate*.

function check: A notification (by a message) that an unexpected condition has stopped the execution of a program.

function key: A keyboard key that is used to request a specific system function. See also *command function key*.

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.

generic name: The initial characters common to object names that can be used to identify a group of objects. A generic name ends with an * (asterisk). For example, ORD* identifies all objects whose names begin with the characters ORD.

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.

hexadecimal number: The 1-byte hexadecimal equivalent of an EBCDIC character.

high-level language: A programming language that relieves the programmer from the rigors of machine level or assembler level programming; for example, RPG III, CL, and COBOL. Abbreviated HLL.

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. The user is not notified when a message arrives at a message queue that is in hold delivery.

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*.

immediate maintenance: A method of maintaining keyed access paths for data base files. This method updates the access path whenever changes are made to the data in the access path. Contrast with *rebuild maintenance* and *delay maintenance*.

IMPL: See *initial microprogram load*.

IMPL Abbr: See *initial microprogram load abbreviated*.

impromptu message: A message that is created when it is sent. Contrast with *predefined message*.

indexed file: A data base file whose access path is built on key values. Each record in the file is identified by a key field.

indicator: (1) A 2-character entry on a specification form that is used to test a field or record or to tell when certain operations are to be performed. (2) An internal switch used by a program to remember when a certain event occurs and what to do when the event occurs.

information display: A display that presents information such as the status of the system to a user, but that rarely requests a response.

informational message: A message that conveys information about the normal condition of a function.

initial microprogram load: The process that loads the system microprogram code from the system auxiliary storage, then checks system hardware and prepares system programming for user operations. Abbreviated IMPL.

initial microprogram load abbreviated: A shorter version of the IMPL sequence that bypasses certain hardware tests. Abbreviated IMPL Abbr.

initial program: A program, specified in a user profile, that is to be executed when the user signs on and the command processor program QCL is invoked. QCL invokes the initial program.

initialize: To set to a starting position or value.

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 in a display file that can receive input from a user.

input field: A field in a display file into which a work station user can key in data. An input field is passed from the device to the program when the program reads the record containing that field.

input file: A data base or device file that has been opened with the option to allow records to be read.

input stream: (1) 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. (2) In RJEF, data sent to the host system. Contrast with *output stream*.

inquiry: A request for information from a data file usually made against one record.

inquiry message: A message that conveys information and that requests a reply.

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*.

interactive subsystem: A subsystem in which interactive jobs are to be processed. IBM supplies three interactive subsystems: QCTL, QINTER, and QPGMR.

I/O slot: One of three locations in the diskette magazine drive where individual diskettes can be inserted for input/output operations. Same as *manual slot*.

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 control rights: The authority to change, cancel, display, hold, and release all jobs and, optionally, job and output queues and entries on them.

job date: The date associated with a job. The job date usually defaults to the system date.

job description: An object that contains information defining the attributes of a job. The system-recognized identifier for the object type is *JOBD.

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 message queue: A message queue that is created for each job. A job message queue is used for receiving requests to be processed (such as commands) and for sending messages that result from processing the requests. A job message queue consists of an external message queue and a set of program message queues. See also *external message queue* and *program message queue*.

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 priority: The order in which batch jobs on a job queue are selected for execution by CPF. More than one job can have the same priority.

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.

job separator: The pages or cards placed at the beginning of the output for each job that has spooled file entries on the output queue. Each separator contains information that identifies the job such as its name, the job user's name, the job number, and the time and date the job was executed.

job stream: See *input stream*.

Julian date: A date format that contains the year in positions 1 and 2, and the day in positions 3 through 5. The day is represented as 1 through 366, right-adjusted, with zeros in the unused high-order positions.

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: The order in which records appear in an access path. The access path is based on the contents of one or more key fields contained in the records.

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. See also *arrival sequence access path* and *access path*.

keyword: (1) A name that identifies a parameter. Keywords are used in CL commands and in DDS. (2) In RPG, a word whose use is essential to the meaning and structure of a statement in a programming language.

label: The name of a file on a diskette or tape. (2) An identifier of a command generally used for branching.

level checking: A function that compares the record format level identifiers of a file to be opened with the file description that is part of a compiled program to determine if the file record format has changed since the program was compiled.

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.

listing: A printout usually containing the input and output of the compilation of a program, the creation (compilation) of an object, or the execution of a program. See also *compiler listing*.

literal: A character string whose value is given by the characters themselves. For example, the numeric literal 7 has the value 7, and the character literal 'CHARACTERS' has the value CHARACTERS. See also *character literal*, *constant*, and *numeric literal*.

local work station: A work station that is connected directly to System/38 without need for data transmission facilities. Contrast with *remote work station*.

lock state: The definition of how an object is allocated, how it is used (read or update), and whether the object can be shared (used by more than one job).

log: See *error log, history log, job log, programming change log,* and *service log*.

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 file member: A named logical grouping of data records from one or more physical file members. See also *member*.

logical unit description: An MI object that is created as the result of executing the Create Device Description (CRTDEVD) command. Abbreviated LUD.

machine interface: The instruction set and interface to the machine. The instruction set is called the System/38 instruction set. Abbreviated MI.

magazine: A container that holds up to 10 diskettes and is inserted into a diskette magazine drive.

main storage: All storage in a computer from which instructions can be executed directly.

manual answer: Operator actions to make a station ready when a station receives a call on a switched line.

manual call: Operator actions to make a connection with a station on a switched line.

manual slot: See *I/O slot*.

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 identifier: A 7-character code that identifies a predefined message and is used to retrieve its message description from a message file.

MI: See *machine interface*.

microcode: The instructions that provide the basic machine functions and support the 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.

multifunction rotary switches: Two switches on the operator/service panel, each of which can be set to one of 16 different positions by rotating them in either a clockwise or counterclockwise direction.

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.

nested do group: A do group that is contained within another do group.

network: A configuration in which two or more stations can communicate.

nonswitched line: A connection between systems or devices that does not have to be made by dialing. Contrast with *switched line*.

normal termination: System termination that results from the successful execution of the Power Down System (PWRDWNSYS) command. See also *abnormal termination* and *system 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.

numeric character: Any one of the digits 0 through 9.

numeric field: An area that is reserved for a particular unit of information and that can contain only the digits 0 through 9. Contrast with *character field*.

numeric literal: The actual numeric value to be used in processing, instead of the name of a field containing the data. A numeric literal can contain any of the numeric digits 0 through 9, a sign (plus or minus), and a decimal point. Contrast with *character literal*.

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 description: The attributes (such as name, type, and owner name) that describe an object.

object existence rights: The authority to delete, save, free the storage of, restore, and transfer ownership of an object.

object management rights: The authority to move, rename, grant authority to, revoke authority from, and change the attributes of an object.

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 type: The attributes that define the purpose of an object within the system. Each object type has associated with it a set of commands with which to process that type of object.

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.

ODP: See *open data path*.

open: The function that connects a file to a program for processing. Contrast with *close*.

open data path: The path through which all I/O operations for the file are performed. Abbreviated ODP.

operational rights: The authority to use an object and to look at its description.

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: Data transferred from storage to an output device.

output field: A field in a device file in which data can be modified by the program and sent to the device during an output operation.

output file: (1) A data base or device file that has been opened with the option to allow records to be written. (2) In COBOL, a file that is opened in either output mode or extend mode.

output priority: The priority used to determine the order in which spooled output files produced by the job are to be written. More than one file can have the same priority.

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.

output stream: In RJEF, data received from the host system (for example, control characters, data files, and messages). Contrast with *input stream*.

packed decimal format: Representation of a decimal value in which each byte within a field represents two numeric digits except the rightmost byte, which contains one digit in bits 0 through 3 and the sign in bits 4 through 7. For all other bytes, bits 0 through 3 represent one digit; bits 4 through 7 represent one digit. For example, the decimal value +123 is represented as 0001 0010 0011 1111. Contrast with *zoned decimal format*.

packed field: A field that contains data in the packed decimal format.

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 frame: A 512-byte block of main storage that contains a 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.

parameter list: A list of values that provides a means of associating addressability of data defined in a called program with data in the calling program. It contains parameter names and the order in which they are to be associated in the calling and called program.

password: A unique string of characters that a system user enters to identify himself to the system.

PC: See *programming change.*

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.*

physical file membur: A named subset of the data records in a physical file. See also *member.*

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.*

print image: An object that contains a description of the print belt or train on a printer. The system-recognized identifier for the object type is *PRTIMG.

printer file: A device file created by the user to support a printer device.

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.

processing: The action of performing operations on input data.

processor: The functional unit that interprets and executes instructions. Same as *CPU.*

production library: A library containing objects needed for normal processing. Contrast with *test library.*

program: An object that contains a set of instructions that tell a computer where to get input, how to process it, and where to put the results. A program is created as a result of a compilation. The system-recognized identifier for the object type is *PGM.

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.*

program-described file: A file for which the fields in the records are described only in the program that processes the file. To CPF, the record is viewed as a character string. Contrast with *externally described file.*

program message queue: A message queue used to hold messages that are sent between program invocations of a routing step. The program message queue is part of the *job message queue.*

programming change: A modification to an IBM-supplied program. Abbreviated PC.

programming change log: A log of information about the application of program changes and patches to IBM products. Named QCHG.

prompt: A displayed request for information or user action. The user must respond to allow the program to proceed.

public: The collection of all system users.

public authority: The authority to an object granted to all users.

QCL: The IBM-supplied control language processor that accepts CL commands so that they can be interpreted and executed by the system.

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.*

qualifier: A name used to uniquely identify another name. In CPF, for example, a library name can be used to qualify an 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.

query application: See *application.*

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: (1) A program that reads jobs from an input device or a data base file and places them on a job queue. (2) In RJEF, a program that reads jobs from a data base file and sends them to the host system.

rebuild maintenance: A method of maintaining keyed access paths for data base files. This method updates the access path only while the file is open, not when the file is closed; the access path is rebuilt when the file is opened. Contrast with *immediate maintenance* and *delay maintenance.*

record: An ordered set of fields that make up a single occurrence of the basic unit of data transferred between a file and a program.

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.

recovery: The act of resetting the system, or data stored in the system, to an operable state following damage.

remote device: A device whose control unit is connected to a System/38 through a data link.

Remote Job Entry Facility: A System/38 licensed program that provides a data link with a remote host system. Abbreviated RJEF.

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.*

response indicator: A 1-character field passed with an input record from CPF to a program to provide information about the data record or actions taken by the work station user.

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.*

RJEF: See *Remote Job Entry Facility.*

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.

sector: The addressable unit into which each track on a diskette is divided.

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.

separator: A punctuation character used to delimit character strings. See also *file separator* and *job separator*.

sequential file: A file in which records are processed in the order that they are stored in the file.

service library: The library provided in CPF that is used temporarily for loading IBM-supplied programming changes and assembling data for APAR submission. Named QSRV.

service log: A log of information about errors detected in IBM program products. Named QSRV.

SEU: See *source entry utility*.

severity code: A code that indicates how important a message is. The higher the code, the more serious the condition is.

shared access path: An access path used by more than one file to provide access to data common to the files.

shared file: A file whose open data path can be shared between two or more programs executing in the same routing step.

sign off: To enter a command or to select an option from a menu at a work station that instructs the system to end an interactive job.

sign on: To enter a password that identifies the user to the system and instructs the system to establish an interactive job at a work station.

simple list: A list of like values, for example, a list of user names. Contrast with *mixed list*.

single-level storage: The technique of addressing multiple levels of storage through a single addressing structure.

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 optionally diagnostics. See also *compiler listing*.

source member: A member of a data base source file that contains source statements such as RPG, COBOL, or DDS specifications. See also *member*.

source program: A set of instructions, written in a programming language such as RPG or COBOL, that represents a particular job as defined by a programmer. A source program is used as input to the compiler to create an executable program.

source statement: A statement written in symbols of a programming language. For example, RPG, COBOL, or DDS specifications are source statements.

special character: A character other than a digit, a letter, or #, $, or @. For example, *, +, and % are special characters.

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.

spooled input file: See *inline data file.*

spooled output file: A device file that causes output data to be saved for later processing by a writer.

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.

spooling subsystem: A subsystem that provides the operating environment needed by the CPF programs that read jobs onto job queues and write files from the output queues. IBM supplies one spooling subsystem: QSPL.

static variable: A variable that is allocated when a program is first invoked in a routing step and exists in storage for subsequent invocations of the same program until the program is de-activated. Contrast with *automatic variable.*

storage pool: A logical segment of main storage reserved for executing a group of jobs.

subsystem: An operating environment, defined by a subsystem description, through which CPF coordinates work flow and resource usage.

subsystem attributes: Specifications in a subsystem description that specify the amount of main storage available to the subsystem and the number of jobs that can execute concurrently in the subsystem.

subsystem description: An object that contains information defining a subsystem and that CPF uses to control the subsystem. The system-recognized identifier for the object type is *SBSD.

switched line: A connection between two stations that is established by dialing. Contrast with *nonswitched line.*

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 arbiter: A system job that provides overall control of the work being done on the system.

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 date: The date established for the system when it is started.

system library: The library provided by CPF to contain system-oriented objects provided as part of CPF. Named QSYS.

system log message queue: A message queue used for sending information to the system history log, service log, or programming change log, from any job in the system.

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 operator user profile: The CPF-supplied user profile that has the authority necessary for the system operator and the special authorities of save system rights and job control rights. 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.*

system time: The elapsed time from the point where the system was started to the current time. If the system time is changed to the local time when the system is started, the current system time is the local time of day.

system unit: The main unit of the system, which contains the processing unit, the system console keyboard/display, the operator/service panel, the diskette magazine drive, main storage, auxiliary storage, the work station controller, and the communications subsystem.

system value: A value that contains control information for the operation of certain parts of the system. A user can change the system default value to tailor the system to his working environment. System date and library list are examples of system values.

tape file: A device file created by the user to support a tape device.

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.*

test library: A library to be used in debug mode and that does not contain objects needed for normal processing. Contrast with *production library.*

time slice: The quantity of processor time (specified in milliseconds) allowed for a routing step before other waiting routing steps are given the opportunity to execute.

time stamp: (1) To apply the current system time. (2) The value on an object that is an indication of the system time at some critical point in the object's history. (3) In query, the identification of the day and time a query report was created that query automatically provides on each report.

transaction: (1) In a batch or remote batch entry, a job or job step. (2) An exchange between a work station and another device that accomplishes a particular action or result; for example, the entry of a customer's deposit and the updating of the customer's balance. (3) A specific set of input data that triggers the execution of a specific processor job; a message destined for an application program. (4) A unit of processing (consisting of one or more application programs) initiated by a single request. In many cases, the request will originate at a work station.

truncate: To drop data that cannot be printed or displayed in the line width specified or available. Contrast with *fold.*

uninterruptible power supply: A buffer between the utility power (or other power source) and a load that requires uninterrupted, precise power. Abbreviated UPS.

update rights: The authority to change the entries in an object. Contrast with *add rights, delete rights,* and *read rights.*

UPS: See *uninterruptible power supply.*

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 message queue: A user-created message queue used to send messages to system users and between application programs.

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.

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 controller: A device in the system unit that provides for a direct connection of local work stations to the system.

work station entry: A work entry in a subsystem descripticn 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.

work station user: A person who uses a work station to communicate with System/38.

working display: See *basic working display*.

writer: (1) A CPF program that writes spooled output files from an output queue to an external device, such as a printer. (2) In RJEF, a program that receives output data (files) from the host system.

zoned decimal format: Representation of a decimal value by 1 byte per digit. Bits 0 through 3 of the rightmost byte represent the sign; bits 0 through 3 of all other bytes represent the zone portion; bits 4 through 7 of all bytes represent the numeric portion. For example, in zoned decimal format, the decimal value of +123 is represented as 1111 0001 1111 0010 1111 0011. Contrast with *packed decimal format*.

Index

IBM System/38
Operator's Guide

READER'S COMMENT FORM

SC21-7735-3

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 ADDRESSEE

IBM CORPORATION
Publications, Dept. 245
Rochester, Minnesota 55901

Fold and tape

Please do not staple

Fold and tape

IBM

IBM System/38 Operator's Guide (File No. S[illegible]0) Printed in U.S.A. SC21-7735-3