

**RSX-11M-PLUS  
Batch and Queue Operations  
Manual**

Order No. AA-JS18A-TC

RSX-11M-PLUS Version 4.0

---

**First Printing, November 1981**  
**Revised, August 1987**

The information in this document is subject to change without notice and should not be construed as a commitment by Digital Equipment Corporation. Digital Equipment Corporation assumes no responsibility for any errors that may appear in this document.

The software described in this document is furnished under a license and may be used or copied only in accordance with the terms of such license.

No responsibility is assumed for the use or reliability of software on equipment that is not supplied by Digital Equipment Corporation or its affiliated companies.

Copyright ©1981, 1987 by Digital Equipment Corporation

All Rights Reserved.  
Printed in U.S.A.

The postpaid READER'S COMMENTS form on the last page of this document requests the user's critical evaluation to assist in preparing future documentation.

The following are trademarks of Digital Equipment Corporation:

DEC	EduSystem	UNIBUS
DEC/CMS	IAS	VAX
DEC/MMS	MASSBUS	VAXcluster
DECnet	MicroPDP-11	VMS
DECsystem-10	Micro/R SX	VT
DECSYSTEM-20	PDP	
DECUS	PDT	
DECwriter	RSTS	
DIBOL	RSX	

**digital**

ZK3073

---

**HOW TO ORDER ADDITIONAL DOCUMENTATION**  
**DIRECT MAIL ORDERS**

**USA & PUERTO RICO\***

Digital Equipment Corporation  
P.O. Box CS2008  
Nashua, New Hampshire 03061

**CANADA**

Digital Equipment  
of Canada Ltd.  
100 Herzberg Road  
Kanata, Ontario K2K 2A6  
Attn: Direct Order Desk

**INTERNATIONAL**

Digital Equipment Corporation  
PSG Business Manager  
c/o Digital's local subsidiary  
or approved distributor

In Continental USA and Puerto Rico call 800-258-1710.

In New Hampshire, Alaska, and Hawaii call 603-884-6660.

In Canada call 800-267-6215.

\* Any prepaid order from Puerto Rico must be placed with the local Digital subsidiary (809-754-7575).

Internal orders should be placed through the Software Distribution Center (SDC), Digital Equipment Corporation, Westminister, Massachusetts 01473.

---

This document was prepared using an in-house documentation production system. All page composition and make-up was performed by T<sub>E</sub>X, the typesetting system developed by Donald E. Knuth at Stanford University. T<sub>E</sub>X is a trademark of the American Mathematical Society.

# Contents

---

Preface	vii
---------	-----

---

Summary of Technical Changes	xi
------------------------------	----

---

## Chapter 1 Introduction to the Queue Manager

---

1.1 Queue Assignments	1-1
1.2 Printing Files	1-3
1.3 Batch Processing	1-7
1.4 Monitoring the Progress of Jobs in Queues	1-7
1.5 Deleting Jobs from Queues	1-9

## Chapter 2 Queuing Jobs

---

2.1 Spooling Output	2-1
2.2 PRINT (PRI) Command	2-2
2.3 DELETE (QUE /DEL) Command	2-10
2.4 Print Jobs Queued by System Tasks	2-12
2.4.1 Maps, Logs, and Listings	2-12
2.4.2 PIP, CLIs, and Output Files	2-12
2.5 Print Jobs Queued by User Tasks	2-13
2.5.1 Opening a File on Disk and Using the .PRINT Subroutine	2-13
2.5.2 Opening a File on Disk and Using the PRINT Command	2-13
2.5.3 Opening a File on LPn:	2-14
2.6 SHOW QUEUE (QUE /LI) and SHOW PROCESSOR (QUE /LI:DEV) Commands	2-14
2.6.1 SHOW QUEUE Command	2-14
2.6.2 SHOW PROCESSOR Command	2-17
2.7 SET QUEUE (QUE /MOD) Command	2-18
2.8 Holding and Releasing Jobs	2-21
2.8.1 HOLD (QUE /HO) Command	2-22
2.8.2 RELEASE (QUE /REL) Command	2-22

## Chapter 3 Preparing a User Batch Job

---

3.1	Commands . . . . .	3-2
3.2	Data . . . . .	3-2
3.3	Labels . . . . .	3-3
3.4	Batch-Specific Commands . . . . .	3-4
3.4.1	JOB . . . . .	3-4
3.4.2	EOJ . . . . .	3-6
3.4.3	DATA . . . . .	3-6
3.4.4	EOD . . . . .	3-8
3.4.5	STOP . . . . .	3-8
3.4.6	CONTINUE . . . . .	3-9
3.4.7	GOTO . . . . .	3-9
3.4.8	ON . . . . .	3-10
3.4.9	SET . . . . .	3-11
3.4.10	IF . . . . .	3-11
3.4.11	Comments . . . . .	3-12
3.5	Indirect Command Files . . . . .	3-13
3.6	Allocating Devices and Mounting Volumes from Batch Jobs . . . . .	3-13
3.7	SUBMIT (SUB) Command and Qualifiers . . . . .	3-15
3.8	The Batch Log File . . . . .	3-19
3.9	Preparing Batch and Print Jobs from Card Decks . . . . .	3-23
3.9.1	Card Deck Format for Batch and Print Jobs . . . . .	3-24
3.9.1.1	The \$SUBMIT Card . . . . .	3-24
3.9.1.2	The \$PRINT Card . . . . .	3-25
3.9.1.3	The \$PASSWORD Card . . . . .	3-26
3.9.2	Operating the Card Reader . . . . .	3-26

## Chapter 4 At-a-Glance Guide

---

4.1	PRINT (PRI) Command . . . . .	4-2
4.2	Batch-Specific Command Summary . . . . .	4-2
4.2.1	Opening and Closing Batch Jobs . . . . .	4-3
4.2.2	Data in Batch Jobs . . . . .	4-3
4.2.3	Sequence Control Commands . . . . .	4-4
4.3	SUBMIT (SUB) Command . . . . .	4-5
4.4	SHOW QUEUE (QUE /LI) Command . . . . .	4-6
4.5	SHOW PROCESSOR (QUE /LI:DEV) Command . . . . .	4-6
4.6	SET QUEUE (QUE /MOD) Command . . . . .	4-7
4.7	HOLD (QUE /HO) and RELEASE (QUE /REL) Commands . . . . .	4-8
4.7.1	HOLD (QUE /HO) . . . . .	4-8
4.7.2	RELEASE (QUE /REL) . . . . .	4-8

4.8	DELETE (QUE /DEL) Command . . . . .	4-8
-----	-------------------------------------	-----

## Appendix A Error Messages

---

A.1	Batch Processing Error Messages . . . . .	A-1
A.1.1	Error Messages in Batch Logs . . . . .	A-1
A.1.2	Error Messages to the Operator's Console . . . . .	A-3
A.1.3	Error Messages from the Card Reader Processor . . . . .	A-4
A.2	Output Despooler Error Messages . . . . .	A-7
A.3	DCL and MCR Error Messages from Queue Manager Commands . . . . .	A-8

## Glossary

---

## Index

---

## Examples

---

1-1	SHOW QUEUE Display . . . . .	1-8
3-1	Batch Job Oriole . . . . .	3-21
3-2	Log File for Batch Job Oriole . . . . .	3-21
3-3	Batch Job Friend . . . . .	3-22
3-4	Log File for Batch Job Friend . . . . .	3-23
3-5	A Batch Job to Be Submitted by the Card Reader Processor . . . . .	3-24

## Figures

---

1-1	Queue Assignments to Output Devices . . . . .	1-2
1-2	How the PRINT (PRI) Command Is Processed . . . . .	1-4

## Tables

---

3-1	Batch-Specific Commands . . . . .	3-4
-----	-----------------------------------	-----



# Preface

---

## Manual Objectives

The *RSX-11M-PLUS Batch and Queue Operations Manual* explains the functions of the RSX-11M-PLUS Queue Manager and batch processor and provides information on preparing batch jobs.

A batch job is one or more files containing commands and data that are submitted to the Queue Manager for execution by the batch processor. Batch processing permits users to do other work at their terminals while their batch jobs are executing.

The Queue Manager is a system task that maintains waiting lines of jobs that have been submitted by users, or by system tasks that produce output. It passes the jobs to the batch processor or to output devices, such as line printers. The commands that control the functions of the Queue Manager, batch processor, and printers are explained in detail in this manual.

This manual provides a general description of batch and queue operations on an RSX-11M-PLUS system. Special applications of the Queue Manager or batch processor at a particular installation may affect the functions of these facilities. System managers can provide information on special applications.

## Intended Audience

This manual is intended for all RSX-11M-PLUS users and is designed to meet the needs of users new to operating systems.

System managers should consult the *RSX-11M-PLUS and Micro/RSX System Management Guide* for information on setting up and running the Queue Manager.

## Structure of This Document

Chapter 1 provides introductory information on printing files, batch processing, monitoring the progress of jobs in queues, and deleting jobs from queues.

Chapter 2 presents descriptions of the commands associated with the Queue Manager: PRINT (PRI), DELETE (QUE /DEL), SHOW QUEUE (QUE /LI), SHOW PROCESSOR (QUE /LI:DEV), SET QUEUE (QUE /MOD), HOLD (QUE /HO), and RELEASE (QUE /REL).

Chapter 3 gives instructions on preparing batch jobs and descriptions of the batch-specific commands and the SUBMIT (SUB) command.

Chapter 4 provides a list of all commands associated with batch and queue operations, including batch-specific commands.

Appendix A includes explanations of all error messages and suggestions for appropriate user actions.

The Glossary provides definitions of terms used in batch and queue operations.

## Associated Documents

- *RSX-11M-PLUS Command Language Manual*
- *RSX-11M-PLUS MCR Operations Manual*
- *RSX-11M-PLUS and Micro/RSX System Management Guide*

## Conventions Used in This Document

The DIGITAL Command Language (DCL) and the Monitor Console Routine (MCR) are the two command language interpreters (CLIs) used in the examples in this manual.

DCL Queue Manager commands are generally in the following format:

```
DCL>command/commandqualifier file/filequalifier
```

The DCL qualifiers either define the operation (command) or modify that operation for a file.

MCR commands, unlike DCL commands, do not follow one general format. The position of command elements and parameters varies from one command to another.

DCL is presented as the primary CLI in this manual. When the names of DCL commands or qualifiers are used in the text, they are followed by the MCR names in parentheses.

The `RETURN` symbol has been omitted from the examples. You can assume that every command is terminated by pressing the RETURN key.

A number of conventions are used in the command descriptions in this manual. With the exception of /qualifier (DCL) and /switch (MCR), the terms and conventions are the same for DCL and MCR.



Convention	Meaning
. . .	The ellipsis ( . . . ) denotes batch-stream elements without significance to the point being discussed.
. . . . . .	The vertical ellipsis shows where elements of command input have been omitted because they are irrelevant to the point being discussed.
[field]	Any command field enclosed in brackets is optional.
{ element 1 . . . element n }	Two or more command elements enclosed in braces indicate that you must choose one of the elements when you issue the command.
lowercase	Any command field in lowercase must have a substitute. Usually the lowercase word will identify what kinds of substitutions are expected.
/qualifier (DCL)	A command element preceded by a slash (/) is either a DCL qualifier or an MCR switch (for information on MCR switches, see the next paragraph). Command qualifiers alter the action of a command they follow. File qualifiers modify the action of the command as it affects that file. If a file qualifier is used as a command qualifier, it affects all files in the command line.
/switch (MCR)	A command element preceded by a slash (/) is either a DCL qualifier or an MCR switch. Switches alter the action of a command. You can apply a switch directly to the command or to a command parameter. If a switch follows the file, it modifies the attributes of that file and all other files that follow in the command line.
parameter	Any required field for a command is called a parameter. The most common parameters are file specifications. Parameters are preceded by blanks.
filespec	A full file specification includes device, directory, file name, file type, and version number, as in the following example:  DB1 : [303,5] JONES . TXT ; 2  Full file specifications are rarely needed. If you do not give a version number, the highest numbered version will be used. Some files may not have a file type. Many commands will accept a wildcard character (*) in place of the file type. See the command descriptions in the <i>RSX-11M-PLUS Command Language Manual</i> .
red ink	In command examples, commands or other input that the user types at the terminal are printed in red. System output is printed in black.
<span style="border: 1px solid black; padding: 2px;">CTRL/N</span>	The symbol <span style="border: 1px solid black; padding: 2px;">CTRL/N</span> indicates that you must press the key labeled CTRL while you simultaneously press another key, for example, CTRL/C, CTRL/Z, or CTRL/O.



## Summary of Technical Changes

---

This manual reflects changes and additions in Version 4.0 of the RSX-11M-PLUS operating system.

The following technical changes are described in this manual:

1. A new DCL qualifier, `/[NO]ADJACENT`, and MCR job switch, `/[NO]AD`, have been added to the description of the PRINT command in Chapter 2. The `/[NO]WIDE` qualifier, which is now invalid, has been deleted from the description of the PRINT command.
2. Information on submitting batch jobs that are on private disks has been added to the description of the `/[NO]PRINT (/NOPRIN)` qualifier of the SUBMIT command in Chapter 3.
3. A restriction on the use of the `/DELETE (/DEL)` file qualifier and `/RESTART (/RES)` qualifier together has been added to the descriptions of the PRINT and SUBMIT commands in Chapters 2 and 3.
4. A restriction on the use of the HOLD command (`QUE /HO`) has been added to Chapter 2.



# Chapter 1

---

## Introduction to the Queue Manager

The Queue Manager is a system task that provides for the orderly processing of print and batch jobs. Jobs are sent to the Queue Manager by users and by system tasks that produce output, such as the MACRO-11 Assembler and the Task Builder. Every PRINT command, every SUBMIT command, and every request for a printed listing or map of a program is passed to the Queue Manager for processing. The Queue Manager places the requests in a queue, a waiting line of jobs whose order is determined by system or user priorities. Jobs are then passed to the batch processor or appropriate output device, such as a line printer, as soon as the device becomes available.

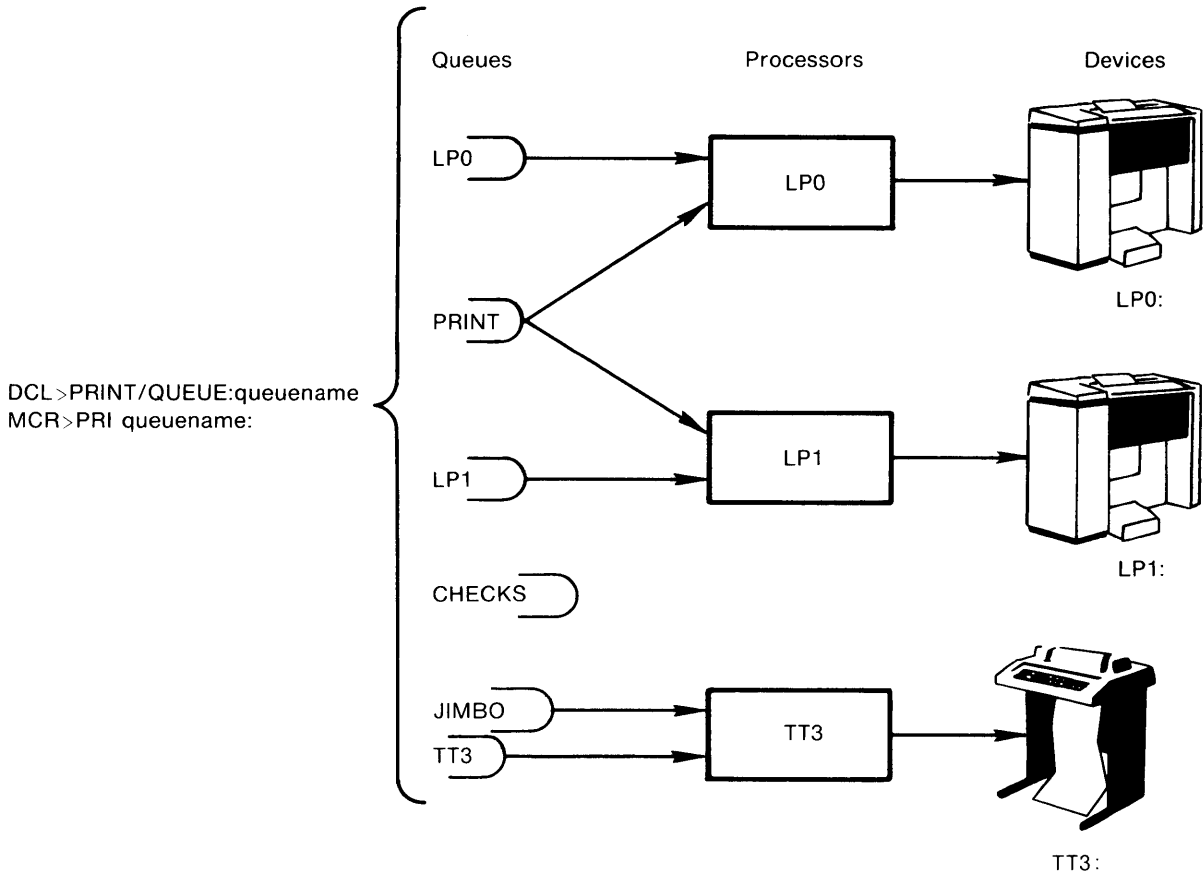
The RSX-11M-PLUS Queue Manager records all its queues in the file SP0:[1,7]QUEUE.SYS, which is kept on a mass-storage disk. This means that jobs will not be lost if a system failure occurs.

### 1.1 Queue Assignments

System managers choose the number of queues to be set up and the devices to which the jobs are directed. Certain printers can be designated to receive output for special types of paper, such as printed or numbered forms. Consult your system manager for information on such forms and the devices on which they are used.

A single queue can be assigned to one or more output devices or to no device. A single device may have one or more queues assigned to it or no queue. Figure 1-1 shows some of the possible assignments of queues to devices.

**Figure 1-1: Queue Assignments to Output Devices**



ZK-238-81

The /QUEUE:queuename qualifier (queuename:) to the PRINT command directs output to the designated queue. When you do not specify a particular queue in the PRINT command, output goes into the default queue PRINT. In Figure 1-1 the jobs on the PRINT queue go to one of two possible printers, LP0: or LP1:. The printers, LP0:, LP1:, and TT3:, are also served by their device-specific queues, LP0, LP1, and TT3. The queue CHECKS has not been assigned to any output device. Jobs can be placed in this queue, but they will not be taken out of the queue until it is assigned to an output device by the system manager. The queues JIMBO and TT3 are both assigned to the same output device.

The batch queues function in the same way, except that batch jobs are queued first to the batch processor. When the batch processor executes a batch job, it produces a file containing a log of the job, which is passed to the Queue Manager to be queued to a line printer. Any files created by the batch job are appended to the log file. An RSX-11M-PLUS operating system may include as many as 16 batch processors and as many as 16 batch queues. The default batch queue is called BATCH.

When a line printer or other output device is under control of the Queue Manager (QMG), it is said to be a spooled device. Spooled devices have been initialized with certain attributes. These attributes are defined by the PRINT command qualifiers (see Chapter 2). Figure 1-2 shows how the PRINT (PRI) command is executed.

## 1.2 Printing Files

The Queue Manager handles the orderly printing of files for an RSX-11M-PLUS system through software tasks called print processors. Every spooled line printer and hardcopy terminal on the system is associated with a software print processor. An RSX-11M-PLUS operating system may include as many as 15 print processors associated with output devices. A line printer and its associated processor are usually referred to as a printer.

The PRINT command supplies most entries to the PRINT queue. For example:

```
DCL>PRINT/LOWER/AFTER:14:00 PETER.TXT/COPIES:3
MCR>PRI /LOWER/AF:14:00=PETER.TXT/COP:3
```

This command queues a job that prints three copies of the file PETER.TXT on a line printer initialized as /LOWERCASE after 2 P.M. today.

Here is an example of a PRINT command naming several files:

```
DCL>PRINT/NAME:KIDS/FLAG_PAGE MARY.TXT,JOHN,PETER
MCR>PRI KIDS/FL=MARY.TXT,JOHN,PETER
```

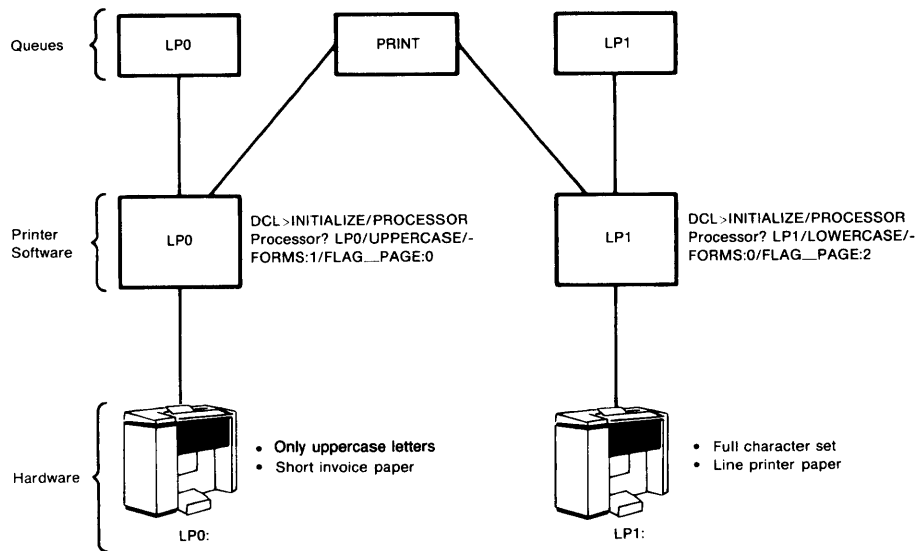
More than one file queued with a single print command constitutes one print job. The /NAME:jobname qualifier (jobname) gives the print job the name KIDS and causes a page bearing the job name to be printed at the beginning of the job. The default job name is the name of the first file submitted for printing. The Queue Manager accepts job names up to nine characters in length. The files are printed in the order listed, on the same line printer, and without interruption. The /FLAG\_PAGE qualifier (/FL) causes each file to be separated by a flag page bearing the name of the file. These file flag pages are in addition to the page bearing the name of the job that appears at the beginning of the job.

## Figure 1-2: How the PRINT (PRI) Command Is Processed

The Queue Manager controls printers through output processors associated with the devices. The printer and its associated processor are often referred to collectively as the printer. The system manager initializes queues and thereby makes it possible for the Queue Manager and the printer hardware and software to interact with each other.

The system manager also initializes print processors to establish the characteristics of the printer, such as the character set or type of paper it will use. You can then specify those characteristics for your job, using the qualifiers to the PRINT command.

This figure illustrates the relationship between the PRINT command and two line printers.



Printer LP0: prints only uppercase letters. It is loaded with short invoice paper and has been initialized with the /FORMS:1 qualifier. The system manager has defined FORMS:1 for the width and length of the short invoice paper. This printer is also initialized with the /FLAG\_PAGE:0 because the invoices are numbered, and flag pages should not be printed on them.

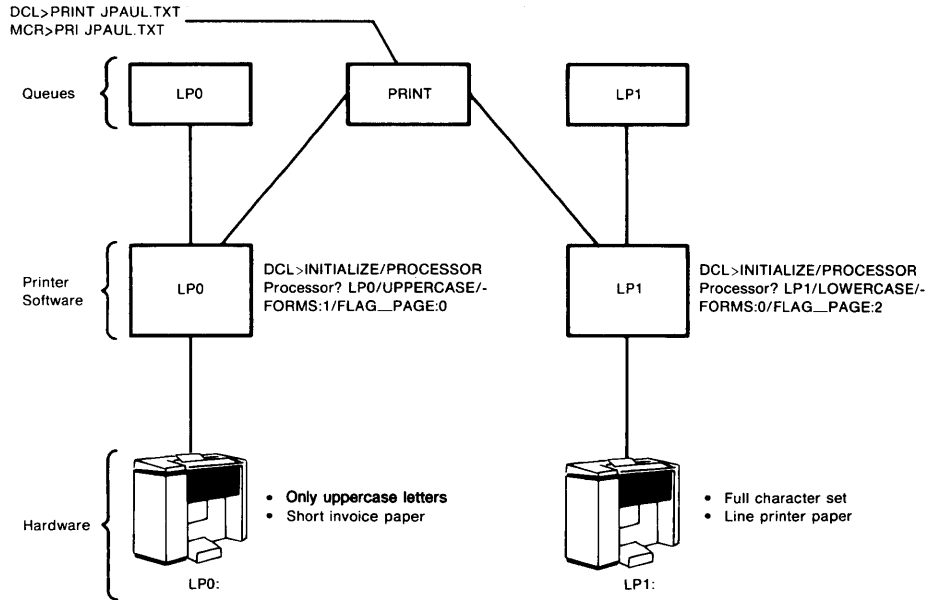
Printer LP1: has the full character set and prints both uppercase and lowercase letters. It is loaded with standard line-printer paper. Since system output and text files are queued to this printer, LP1: has been initialized with the /FLAG\_PAGE:2 qualifier, so that any file printed will be identified by a flag page.

ZK-239/1-81

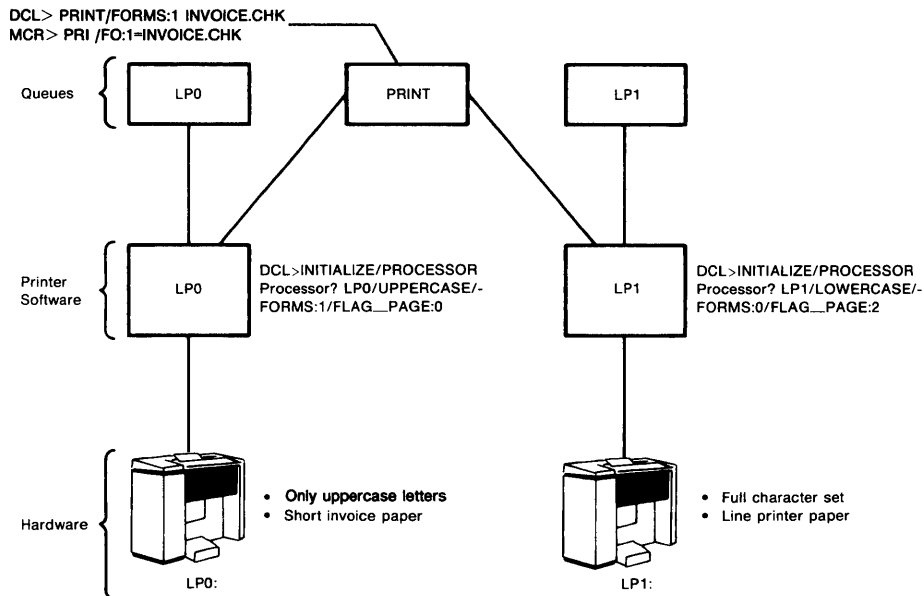
(Continued on next page)



**Figure 1-2 (Cont.): How the PRINT (PRI) Command Is Processed**



The PRINT command specifies no queue, so the job goes into the default queue PRINT. Since the default /FORMS qualifier (/FO:) is /FORMS:0 (/FO:0), the job goes from the PRINT queue to processor LP1 and thence to device LP1: to be printed.

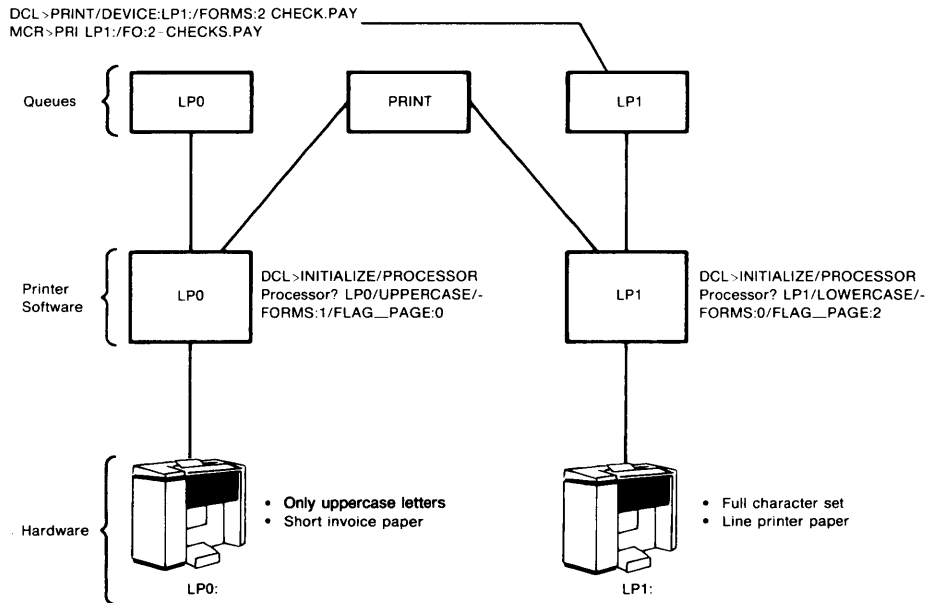


The PRINT command specifies no queue, so the job goes into the default queue PRINT. Since the /FORMS qualifier (/FO:) is /FORMS:1 (/FO:1), the job goes from the PRINT queue to processor LP0 and thence to device LP0: to be printed.

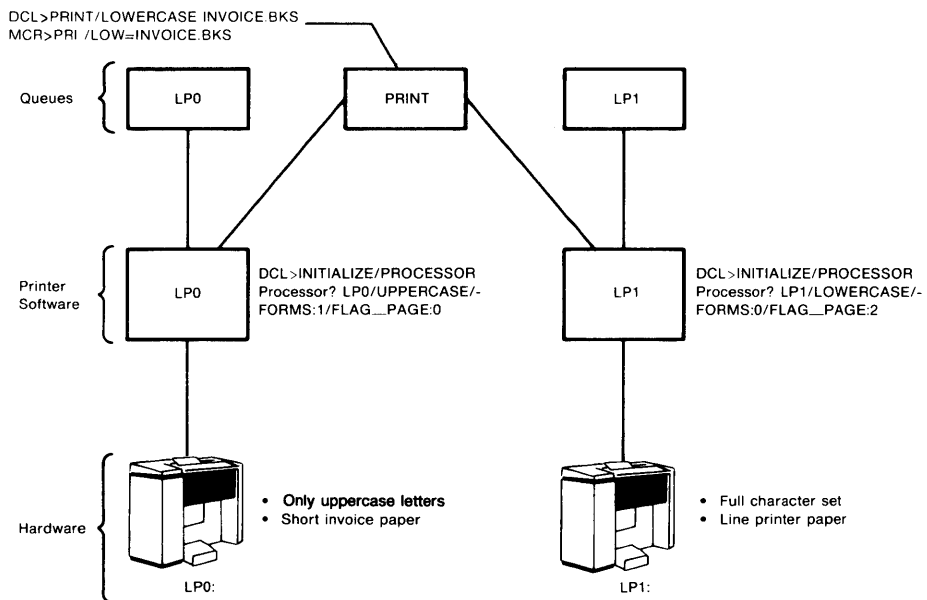
ZK-239/2-81

**(Continued on next page)**

**Figure 1-2 (Cont.): How the PRINT (PRI) Command Is Processed**



The PRINT command specified device LP1, so the job goes to the device-specific queue LP1. The PRINT command specified the /FORMS:2 qualifier (/FO:2), however, so the job will not be printed until LP1 is reinitialized to /FORMS:2 (/FO:2). In addition, the paper in the machine must be changed.



The PRINT command specified no queue, so the job goes to the default queue PRINT. The /FORMS qualifier (/FO:) is /FORMS:0 (/FO:0) by default. The user specified a printer initialized with the /LOWERCASE qualifier (/LOW), so the job goes to processor LP1 and thence to device LP1 to be printed.

You should note that initialization with the /LOWERCASE qualifier (/LOW) does not necessarily mean that the device itself has the lowercase character set. In most cases, it has, but in emergencies, devices without the lowercase character set may be initialized with the /LOWERCASE qualifier (/LOW) so that jobs whose PRINT commands included the /LOWERCASE qualifier (/LOW) will not be blocked in their queues.

## 1.3 Batch Processing

A batch job is one or more files containing commands and data that are submitted to the Queue Manager for execution by the batch processor. Most operations that you can perform interactively can also be executed in a batch job, including compiling, assembling, building, and running tasks.

Each batch job is executed as a separate terminal session; thus, you may use your terminal to do other work while the batch job is executing. You may also determine when a batch job will execute. For example, you may wish to run batch jobs at night, when the central processing unit (CPU) may be less active.

Because a batch job is executed as a separate terminal session, it must log itself on to the system, provide any information that the system requires in order to complete the job, and log itself off the system. For information on preparing a batch job, see Chapter 3.

The SUBMIT command is used to send the batch file to the Queue Manager for processing, in the following format:

```
DCL>SUBMIT filename.bat,filename.bat
MCR>SUB filename.bat,filename.bat
```

More than one batch file submitted with a single SUBMIT command constitutes a single batch job. A batch job containing more than one batch file may also be referred to as a batch chain.

The following is an example of a SUBMIT command:

```
DCL>SUBMIT/NAME: JONES/PRIO:150 JOE.BAT, PETER.BAT
MCR>SUB JONES/PRIO:150=JOE.BAT, PETER.BAT
```

This command enters the batch job named PETER in the default batch queue BATCH at a queue priority of 150. (The queue priority setting has no effect on the running priority of any tasks run by the batch job. It only affects the job's position in the queue.) The batch job is given the name JONES and is composed of the batch files JOE.BAT and PETER.BAT.

## 1.4 Monitoring the Progress of Jobs in Queues

The SHOW QUEUE (QUE /LI) command displays information on the status of jobs in print and batch queues. Its format is as follows:

```
DCL>SHOW QUEUE
MCR>QUE /LI
```

The qualifiers to the SHOW QUEUE command are described in Section 2.6.1.

Example 1-1 shows the information that appears on your terminal when you use the SHOW QUEUE command.

### Example 1-1: SHOW QUEUE Display

```

$ SHOW QUEUE
  QUE/LI
  ① ** PRINT QUEUES **
PRINT => ② LPO LP2 TT20 TT12 TT30
  ③ [7,11] ④ KIDS      ENTRY:602                ⑤ COP:2      ACTIVE ON LP2
          { 1 DR5:[DUNNE]MARY.TXT;1
          ⑥ { 2 DR5:[DUNNE]JOHN.TXT;2
          { 3 DR5:[DUNNE]PETER.TXT;1
  [7,11]  ORIOLE     ENTRY:601                HELD
          1 DR5:[DUNNE]ORIOLE.TXT;18
LXY11 => LP1
LN03  => TT12
LA50  => TT30
LPO   => LPO
LP1   => LP1
  [7,11]  BOOK      ENTRY:603                ⑦ FORM:2
          1 DR5:[DUNNE]BOOK.TXT;1
LP2   => LP2
LQP   => TT20
TT20  => TT20
TT12  => TT12
TT30  => TT30
LA75  => TT30
** BATCH QUEUES **
BATCH => BAPO BAP1
  [7,11]  EXAMPLE ⑧ ENTRY:604                ⑨ ACTIVE ON BAPO
          > 1 DR5:[DUNNE]EXAMPLE.BAT;43
LMPO  =>
$

```

ZK-240-81

The circled numbers following identify the elements of the display that appears on your terminal when you enter the SHOW QUEUE (QUE /LI) command:

- ①. The header for the print queues, which indicates that the queues listed below it are print queues.
- ②. The name of the queue (PRINT) and the spooled devices or batch processors assigned to it.
- ③. The User Identification Code (UIC) of the job.
- ④. The job name, the name of the first file in the job or the name assigned with the /NAME:jobname (jobname) qualifier of the PRINT command.
- ⑤. The number of copies of the file MARY.TXT that will be printed.
- ⑥. The entry order of a file in a job and its full file specification.
- ⑦. The form number assigned to the printer to which the job is queued.
- ⑧. The job entry number (the range is 1 to 999). The Queue Manager assigned this number when it placed the job in queue.
- ⑨. The queue status of a job. A job may be active, held, or blocked until a particular time.

## 1.5 Deleting Jobs from Queues

The `/ENTRY:nnn (/EN:nnn)` qualifier of the `DELETE` command (`QUE /DEL`) deletes a job from a queue by means of its entry number in the queue. Example 1-1 shows the position of the entry number in the display issued in response to the `SHOW QUE (QUE /LI)` command. The format of the `DELETE` command is as follows:

```
DCL>DELETE/ENTRY:nnn
```

```
MCR>QUE /EN:nnn/DEL
```

You can also delete a job from a queue by specifying a job name with the `DELETE` command. For more information on deleting jobs from queues, see Chapter 2.



## Chapter 2

---

### Queuing Jobs

The Queue Manager is a system task that maintains a waiting line of jobs and distributes the jobs to batch processors and output devices, such as line printers. Jobs are queued by users and by system tasks that produce output, such as the MACRO-11 Assembler and the Task Builder. A waiting line of jobs is called a queue, and sending a job to the Queue Manager is called queuing a job.

The SHOW QUEUE (QUE /LI) command displays information regarding jobs in queues. The DELETE (QUE /DEL) command removes jobs from queues. The SET QUEUE (QUE /MOD) command alters the status of jobs in queues. The HOLD (QUE /HO) command holds jobs in queues, and the RELEASE command (QUE /REL) releases jobs that have been held. These commands and their qualifiers are described in this chapter.

#### 2.1 Spooling Output

The term *spool* is the acronym for Shared Peripheral Operations On Line. Spooling is the process of sending output to a peripheral device, such as a line printer, in an orderly fashion. On the RSX-11M-PLUS system, spooling is accomplished by means of the Queue Manager.

A spooled output device is a hardware device and associated processor that are under the control of the Queue Manager. The spooled device services only one request at a time and processes only those jobs that have been passed to it by the Queue Manager.

The most common output device for the Queue Manager is a line printer. Most RSX-11M-PLUS systems use either a line printer or some other device that has been designated as the system output device and assumes the role of the line printer.

You can print files by substituting the device name of a spooled output device for the output file specification in any system task or user-written task that creates an output file. This is called transparent spooling. Thus, the commands in the following examples will send files to the Queue Manager to be printed:

```
DCL>COPY JONES.MAC LPO:
MCR>PIP LPO:=JONES.MAC
```

You can also send a file to the Queue Manager to be printed by using the EDT WRITE or PRINT commands and substituting the name of the spooled output device for the name of the output file. The format for these commands is shown next.

### Format

```
*COMMAND ddnn: [range]
```

### Example

```
*WRITE LPO: 200:400
```

This example creates a file consisting of the text between lines 200 and 400 and queues it for printing on LPO:. It is not necessary to exit from the editor to do this. You can type the command and parameters at the EDT asterisk (\*) prompt. No file remains on the disk.

When you successfully submit a job for queuing with either the PRINT or SUBMIT command, you will receive a submission notification message on your terminal. For example:

```
DCL>PRINT JONES.DMP
```

```
MCR>PRI JONES.DMP
```

The following message will appear:

```
PRI - JOB 123, name "JONES  ", submitted to queue "PRINT "
```

PRI The command used to submit the job. If you use the SUBMIT command, SUB will take the place of PRI.

123 The unique job entry number given to the job (1 to 999).

JONES The name of your job. Job names can be up to nine characters in length. If you do not specify a job name, the job name will be derived from the first file name in your job.

PRINT The queue into which the job is submitted.

## 2.2 PRINT (PRI) Command

The PRINT (PRI) command queues files for printing on a line printer. The file you specify is first placed in a queue by the Queue Manager (QMG). The QMG places the print job on the queue according to the job's print status, which is determined by PRINT command qualifiers and switches described in this section.

The PRINT command can also be used to queue jobs to other output devices and applications processors. These jobs are referred to as print jobs even if they do not involve any hardware printing.

The PRINT command gives you control over how, when, and where your job will be printed through the use of PRINT command qualifiers and job switches.



## Format

DCL> PRINT[/commandqualifier[s]] filespec[s][/filequalifier[s]]

MCR> PRI[[queue:][jobname][/jobswitch]=]filespec[s][/fileswitch[es]]

### DCL Command Qualifiers

/JOB\_COUNT:n  
/QUEUE:queuename  
/LOWERCASE  
/UPPERCASE  
/[NO]HOLD  
/PAGE\_COUNT:n  
/NAME:jobname  
/PRIORITY:n  
/FORMS:n  
/LENGTH:n  
/[NO]RESTART  
/[NO]FLAG\_PAGE  
/AFTER:(dd-mmm-yy hh:mm)  
/AFTER:TOMORROW[:hh:mm]  
/DEVICE:ddnn:  
/[NO]JOB\_PAGE  
/[NO]ADJACENT

### DCL File Qualifiers

/COPIES:n  
/[NO]DELETE  
/[NO]TRANSFER

### MCR Job Switches

/CO:n  
queue:queuename:  
/LOW  
/[NO]LOW  
/[NO]HO  
/PA:n  
jobname  
/PRIO:n  
/FO:n  
/LE:n  
/[NO]RES  
/[NO]FL  
/AF:hh:mm:dd-mmm-yy  
/AF:[hh]:[mm]:TOMORROW  
queue:queuename:  
/[NO]JO  
/[NO]AD

### MCR File Switches

/CO:n  
/[NO]DEL  
/[NO]TR

## Parameter

### filespec[s]

Specifies the file or files to be printed.

Separate multiple file specifications with commas. You can use the asterisk wildcard character (\*) in the directory and file name, type, and version fields of the file specification.

If your file specification includes no file type, the default file type is LST.

## DCL Command Qualifiers and MCR Job Switches

/JOB\_COUNT:n

/CO:n

Specifies the number of copies (n) of the print job that are produced. A QMG print job consists of one or more files that are printed in the order they were listed.

### Note

The /JOB\_COUNT:n (/CO:n) qualifier to the PRINT command controls the number of printed copies of the job. Please do not confuse this job qualifier with the /COPIES:n (/CO:n) file qualifier, which controls the number of printed copies of a file.

**/QUEUE:queuename**

**queuename:**

Specifies the name of the print queue in which the job is to be placed. The default queue is PRINT.

**/LOWERCASE**

**/LOW**

Specifies that the Queue Manager must queue the print job to a line printer initialized with the /LOWERCASE qualifier. The output file will appear in both uppercase and lowercase letters if that is how it was written and if the printer has been correctly initialized.

**/UPPERCASE**

**/[NO]LOW**

Specifies that the Queue Manager may queue the print job to any line printer, regardless of whether or not the printer has been initialized with the /LOWERCASE qualifier. This is the default. The printer may or may not print the output file in uppercase and lowercase letters. If it has the capability of printing in uppercase and lowercase, it will do so. If the printer has only the capability of printing in uppercase, it will print all letters in the file in uppercase.

**/[NO]HOLD**

**/[NO]HO**

Specifies that the job be held in its queue and is therefore not eligible for printing. You can release the job with the RELEASE/JOB (QUE /REL) command described in Section 2.8.2.

**/PAGE\_COUNT:n**

**/PA:n**

Sets the limit on the number of pages a job may produce. The range is 1 to 65535. Without this qualifier, the number of pages is unlimited (n = 0).

**/NAME:jobname**

**jobname**

Specifies a job name for the print job. The job name may not exceed nine alphanumeric characters in length.

If you specify a job name, that name appears on the flag page at the beginning of the printed output. Otherwise, the name of the first file in the job is used as the job name.

**/PRIORITY:n**

**/PRIO:n**

Sets the queue priority of the print job. Nonprivileged users can set n to 0 to 150. Privileged users can set n to 0 to 250. The default is 50.

The Queue Manager runs the job with the highest priority first. If two jobs have the same priority, the job that has been waiting longer runs first.

**/FORMS:n**

**/FO:n**

Specifies the form, such as an invoice blank, on which the job is printed. The value of n can be from 0 to 255. The default is 0. See your system manager for information on the forms used on your printers.

**/LENGTH:n**

**/LE:n**

Sets the length of a logical page; n can be any number from 0 to 255. The default is 0. If you set a length, a form feed will be automatically generated if one is not found within n lines.

This qualifier has no effect on how the printer handles form-feed characters. A form feed still moves the printer to the top of the next physical form, such as the perforation in a sheet of line-printer paper. This qualifier simply adds the requirement that the printer move to the top of the next physical page if no form-feed character is encountered within n lines.

When you set the length at 0 (the default), no form feeds will be inserted.

**/[NO]RESTART**

**/[NO]RES**

Controls requeuing of your job if it is stopped or aborted while being processed. **/NORESTART (/NORES)** is the default. If the printer is stopped while your job is active, when the printer is restarted the job will be restarted at the point at which printing stopped.

If **/RESTART (/RES)** is specified, your job will be processed again from the beginning if it is stopped or aborted.

In neither case will you lose any of your job, but with **/RESTART (/RES)** your job will be printed on consecutive pages.

### Notes

The **/RESTART (/RES)** qualifier and the **/DELETE (/DEL)** file qualifier cannot be used together. The system deletes the file before the job can be restarted.

If you use the **/RESTART (/RES)** qualifier in printing a file that is on a privately mounted disk, you must also use the **/NOTRANSFER (/NOTR)** file qualifier in order for the job to restart if it is aborted. The **/NOTRANSFER** qualifier is required because the system deletes the transferred file before the job can be restarted. In addition, using the **/NOTRANSFER** qualifier necessitates leaving your private disk attached to the system, so that the system can access the job without transferring it.

**/[NO]FLAG\_PAGE**

**/[NO]FL**

Adds flag pages before each file in your print job. The default is **/NOFLAG\_PAGE (/NOFL)**.

With **/NOFLAG\_PAGE (/NOFL)**, your job is still preceded by a job flag page, but the files in the job are printed without file flag pages. (See **/[NO]JOB\_PAGE**.)

**/AFTER:(dd-mmm-yy hh:mm)**

**(mm/dd/yy hh:mm)**

**/AF:hh:mm:dd-mmm-yy**

Blocks the job until after the specified time. Depending on the number of jobs on the print queues at that time, your job may be run immediately or later when its turn comes up in the queue.

The time and date are in the same format as that generated by the SET TIME (DCL) and TIM (MCR) commands.

If you do not specify /AFTER (/AF), your job is immediately eligible for printing.

`/AFTER:TOMORROW[:hh:mm]`  
`/AF:[hh]:[mm]:TOMORROW`

Blocks the print job until the next day at the specified time. If no time is specified, the job is blocked until midnight.

`/DEVICE:ddnn:`  
`queuename:`

Specifies the device on which you want your output to appear. The default is the device or devices served by the PRINT queue. (The /DEVICE qualifier is a DCL synonym for /QUEUE.)

`/[NO]JOB_PAGE`  
`/[NO]JO`

Controls the printing of a job flag page at the beginning of the job. The default is /JOB\_PAGE (/JO). (See also the /[NO]FLAG\_PAGE and /[NO]ADJACENT qualifiers.)

`/[NO]ADJACENT`  
`/[NO]AD`

The /ADJACENT (/AD) qualifier causes print jobs to begin at the point at which the preceding job finished; that is, if the preceding job finished in midpage, the new one will commence on the next line. In print jobs that include multiple files, each file will begin at the point at which the preceding file finished. The /ADJACENT qualifier will override any command qualifiers for job flag pages or file flag pages, regardless of whether or not they are the default. (See the /[NO]FLAG\_PAGE and /[NO]JOB\_PAGE qualifiers, described in this section.)

The default is /NOADJACENT (/NOAD), which causes a print job to begin on a new page, regardless of where the previous job finished.

### Restrictions

LN01 and LN03 printers will begin print jobs on a new page whether or not the /[NO]ADJACENT qualifier is used, as these printers require a form feed at the end of each print job in order to eject the last sheet in the job.

Whether or not the /ADJACENT qualifier is used, LA50, LA100, and LA120 printers will begin a print job on a new page if the /FORMS:n qualifier specified for a particular job is different from that specified for the preceding job.

### DCL File Qualifiers and MCR File Switches

You may specify a file qualifier as a command qualifier in DCL. If you do, that qualifier applies to all files submitted with that PRINT command. You can, however, override that qualifier by using the NO form of it as a file qualifier to the file you wish to exempt, as shown in the following example:

```
DCL>PRINT/DELETE JULIAN.TXT,MURIEL,ALEX/NODELETE,DAD.MEE
```

In this example the files JULIAN.TXT, MURIEL, and DAD.MEE will be printed and then deleted from the user's directory. The file ALEX will be printed but not deleted.

If you use the /COPIES:n file qualifier as a command qualifier, you can alter its effect on an individual file by also using it as file qualifier and changing the value of n.

With MCR, any file specifications that follow a file specification that is altered by a switch is altered as well. In the following example, you must use the /CO:1 switch after ALEX if you do not want four copies of this file printed:

```
MCR>PRI =JULIAN.TXT,MURIEL/CO:4,ALEX/CO:1,DAD.MEE
```

With DCL, only the file followed by the file qualifier is affected.

```
DCL>PRINT JULIAN.TXT,MURIEL/COPIES:4,ALEX,DAD.MEE
```

In this example, four copies of the file MURIEL are printed and one of all other files.

```
/COPIES:n  
/CO:n
```

Sets the number of copies for a particular file in a job. The default is one copy. This qualifier does not affect the number of copies of the job file.

```
/[NO]DELETE  
/[NO]DEL
```

Deletes the files after they are printed. The default is /NODELETE.

#### **Note**

Please note that the /DELETE file qualifier and the /RESTART qualifier cannot be used together, because the system deletes the file before the job can be restarted.

```
/[NO]TRANSFER  
/[NO]TR
```

Allows the copying of a file from a private device. The default is /TRANSFER. If a file is transferred from a private device and then printed, the copy of the file is deleted. The original file on the private device is not deleted.

The following PRINT command examples appear in both DCL and MCR syntax:

#### **Examples**

```
DCL>PRINT JULIAN.TXT
```

```
MCR>PRI JULIAN.TXT
```

This example prints the most recent version of JULIAN.TXT on the line printer.

```
DCL>PRINT/NOJOB_PAGE MURIEL.TXT
```

```
MCR>PRI /NOJO=MURIEL.TXT
```

This example prints MURIEL.TXT without a job flag page.

```
DCL>PRINT/JOBCOUNT:4 JULIAN.TXT,MURIEL
```

```
MCR>PRI /CO:4=JULIAN.TXT,MURIEL
```

This example prints four copies of the print job JULIAN, each containing one copy of the file JULIAN.TXT and one copy of the file MURIEL.TXT. The copies are printed end to end with a job flag page preceding each copy.

```
DCL>PRINT/JOBCOUNT:4/FLAG_PAGE MURIEL.TXT
```

```
MCR>PRI /CO:4/FL=MURIEL.TXT
```

This example prints four copies of the print job MURIEL, each containing one copy of the file MURIEL.TXT preceded by a job flag page and a file flag page.

```
DCL>PRINT/FLAG_PAGE JULIAN.TXT,MURIEL.TXT,JIMBO.TXT
```

```
MCR>PRI /FL=JULIAN.TXT,MURIEL.TXT,JIMBO.TXT
```

This example prints one copy of each file, each preceded by a file flag page.

```
DCL>PRINT/NAME:FATHER/FLAG_PAGE JULIAN.TXT,MURIEL.TXT,ALEX.TXT,PETER.TXT
```

```
MCR>PRI FATHER=JULIAN.TXT,MURIEL.TXT,ALEX.TXT,PETER.TXT
```

This example prints one copy of each file preceded by a file flag page and gives the name FATHER to the job as a whole. The name FATHER appears on the job flag page.

```
DCL>PRINT/LENGTH:55 DAD.MAC
```

```
MCR>PRI /LE:55=DAD.MAC
```

This example prints a single copy of the source program file DAD.MAC. This source program file does not contain form-feed characters. To avoid printing over the perforations, the user has specified a logical printed page length of 55 lines. Each time 55 lines are printed, the logical page is complete and the printer moves to the top of the next physical page.

If you specify a file with form feeds in it, the printer would move to the top of a physical page each time it encountered a form feed and each time it printed 55 lines before encountering a form feed.

```
DCL>PRINT/JOBCOUNT:4 JIMBO.DOC,MOUTH.DOC,QUACK.DOC
```

```
MCR>PRI /CO:4=JIMBO.DOC,MOUTH.DOC,QUACK.DOC
```

This example prints the QMG print job four times. Each copy of the print job contains one copy of the file JIMBO.DOC, one copy of the file MOUTH.DOC, and one copy of the file QUACK.DOC.

```
DCL>PRINT/JOBCOUNT:3 LIT.MAC,FRENCH.FRY,APPLE.PIE/COPIES:2,MOUTH.WSH
```

```
MCR>PRI /CO:3=LIT.MAC,FRENCH.FRY,APPLE.PIE/CO:2,MOUTH.WSH/CO:1
```

This example prints three copies of the print job named LIT. Each copy of the print job contains one copy of the files LIT.MAC and FRENCH.FRY, two copies of the file APPLE.PIE, and one copy of the file MOUTH.WSH.

Note the difference between the /JOB COUNT:3 qualifier (/CO:3) to the PRINT command, which controls the number of printed copies of the job, and the /COPIES:2 file qualifier (/CO:2), which controls the number of printed copies of the file APPLE.PIE.

If you are using MCR as your command line interpreter, the differences between the two /CO:n switches may not appear distinct. When you use /CO:n as a switch to the PRINT command, you are instructing the Queue Manager to print n copies of the print job as separate print jobs. When you use /CO:n as a switch to a file specification, contained within a print job, you are instructing the Queue Manager to print n copies of that file end to end within that print job.

```
DCL>PRINT/NAME:SAILOR/JOB COUNT:2/DELETE JIMBO.DOC,MOUTH/COPIES:3,QUACK
MCR>PRI SAILOR/CO:2=JIMBO.DOC/DE,MOUTH/CO:3,QUACK/CO:1
```

The print job has the name SAILOR and is printed twice. Each of the two printed listings contains one copy of the file JIMBO.DOC, three copies of the file MOUTH.DOC, and one copy of the file QUACK.DOC. All the files are deleted after they are printed.

```
DCL>PRINT/AFTER:(12:15 10-AUG-83) JIMBO.TXT
MCR>PRI /AF:12:15:10-AUG-83=JIMBO.TXT
```

This example blocks the job in queue until the specified date and time.

```
DCL>PRINT/AFTER:(10:30) MURIEL.TXT
MCR>PRI /AF:10:30=MURIEL.TXT
```

This example blocks the job in queue until 10:30 on the date the PRINT command was entered. If you entered this command at any time after 10:30, the job MURIEL would have been immediately eligible for printing.

```
DCL>PRINT DX0:[303,14]MURIEL.TXT
MCR>PRI DX0:[303,14]MURIEL.TXT
```

You can print jobs from a private device using the /TRANSFER switch.

The /TRANSFER qualifier (/TR) does not have to be specified because it is the default. The privately allocated device DX0: is mounted with a volume containing the file MURIEL.TXT. Before this file is printed on a line printer, it is copied to the pseudo device SP:. Therefore, the volume may be dismounted and the device deallocated before the file is printed on the line printer. The file will remain on the dismounted volume but will automatically be deleted from SP:[1,7] after printing.

```
DCL>PRINT/NOTTRANSFER DX0:[303,14]MURIEL.TXT
MCR>PRI =DX0:[303,14]MURIEL.TXT/NOTR
```

In this example, the file is not copied to a pseudo device, therefore the volume must remain mounted until the print job is completed.

## Notes

1. Once your job is in a queue, you can modify some job attributes with the SET QUEUE (QUE /MOD) command described in Section 2.7.
2. Regardless of whether you specify the /[NO]RESTART qualifier (/[NO]RES), your system manager may restart your job by commands to the Queue Manager over which you have no control. For information on these commands, see the *RSX-11M-PLUS and Micro/RSX System Management Guide*.
3. The job flag pages and the file flag pages have different formats. The job flag page shows on the first line the User Identification Code (UIC) the PRINT command was issued under (or the user name on RSX-11M-PLUS systems with CPU accounting) and a 9-character job name. The job name is either a specified job name or is derived from the file name of the first file in the job. The file flag page shows the file name on the first line and the file type and version number on the second line.
4. One or more files queued with a single PRINT command constitute a single print job.
5. If you use the /NOTTRANSFER qualifier (/NOTR), you must leave the volume mounted until the print job is finished. Under the default /TRANSFER (/TR), your files are copied to SP:[1,7] and the volume can be dismounted. If you do not wish your files to be accessible to other system users, or if there is no space on SP:, use the /NOTTRANSFER (/NOTR) qualifier.

You may prefer not to use the /NOTTRANSFER (/NOTR) qualifier with slower devices, such as cassettes, DECtapes, or diskettes.

## 2.3 DELETE (QUE /DEL) Command

The DELETE (QUE /DEL) command is used to delete jobs in a queue or files in a queued job.

### Formats

```
DCL> DELETE/ENTRY:nnn[/FILE_POSITION:n]
DCL> DELETE/JOB queueName jobName[/FILE_POSITION:n]
MCR> QUE /EN:nnn[/FI:n]/DEL
MCR> QUE queueName:jobName[/FI:n]/DEL
```

You can delete a job from a queue by using the DELETE command and qualifiers and specifying either the job's entry number or job name. Specifying the entry number of a job is preferable to specifying its job name. As jobs may share a job name, it is possible to delete the wrong job when the job name is specified. Deleting a job by its unique entry number protects you from such mistakes. Use the SHOW QUEUE (QUE /LI) command to determine the entry number of a job in a queue. You can also delete a single file in a job by specifying the /FILE\_POSITION:n qualifier (/FI:n).

If several jobs in a queue have the same job name and you enter a DELETE command specifying that job name, only the first job with that name is deleted from the queue.



## Parameters

### queuename

Specifies the name of the queue containing the job you wish to delete.

### jobname

Identifies a job to be deleted from the specified queue.

HELD or ACTIVE jobs will be deleted before WAITING jobs.

## DCL Command Qualifiers and MCR Job Switches

**/ENTRY:nnn**

**/EN:nnn**

Deletes a queue entry by number. The number is unique and is therefore the preferable means by which to delete a queue entry.

**/FILE\_POSITION:n**

**/FI:n**

Identifies the file by the file's position in the job.

## Examples

```
DCL>DELETE/ENTRY:301
```

```
MCR>QUE /EN:301/DEL
```

This command deletes the job from a queue by referencing the job's unique entry number (in this example, job number 301). Specifying the entry number of a job is preferable to specifying its job name. As jobs may share a job name, it is possible to delete the wrong job when the job name is specified. Deleting a job by its unique entry number protects you from such mistakes. Use the SHOW QUEUE (QUE /LI) command to determine the entry number of a job in a queue.

```
DCL>DELETE/JOB LPO MURIEL
```

```
MCR>QUE LPO:MURIEL/DEL
```

This command deletes the job MURIEL by means of the job's queue name, LP0.

```
DCL>DELETE/ENTRY:301/FILE_POSITION:2
```

```
MCR>QUE /EN:301/FI:2/DEL
```

This example deletes only the second file appearing in job 301. You may refer to the file you wish to delete only by the file's position in the print job (determined by the order in which you entered the file specifications). You cannot delete a file by specifying its file name.

## Notes

1. Specifying the entry number of a job is preferable to specifying its job name. As jobs may share a job name, it is possible to delete the wrong job when the job name is specified. Deleting a job by its unique entry number protects you from such mistakes. Use the SHOW QUEUE (QUE /LI) command to determine the entry number of a job in a queue.

2. DELETE/QUEUE (QUE /DEL) works on all categories of jobs in any queue. You can delete ACTIVE jobs, WAITING jobs, TIME-BLOCKED jobs, or HELD jobs. There can be more than one job in a queue with the same name and the same UIC. The DELETE/QUEUE (QUE /DEL) command deletes the first job of a given name from the queue.

## 2.4 Print Jobs Queued by System Tasks

Several system tasks can pass print jobs to the Queue Manager. Any task that creates an output file can do so.

### 2.4.1 Maps, Logs, and Listings

Print jobs queued by system tasks include listings from language compilers, maps from the Task Builder, and batch log files. The tasks follow their own rules for spooling output, but their requests for spooled output are handled by the Queue Manager.

You cannot control the printing of such output directly, as you can through the PRINT command qualifiers. Once the listings and maps appear in a queue, however, you can alter their attributes by using the SET QUEUE (QUE /MOD) command (see Section 2.7). If the batch log file is printed, any /[NO]JOB\_PAGE and /[NO]FLAG\_PAGE qualifiers specified with the PRINT command within a batch job are overridden by the corresponding PRINT characteristics of the batch log file, which are set by the system manager in initializing the batch queue.

### 2.4.2 PIP, CLIs, and Output Files

You can use PIP, the Peripheral Interchange Program, to queue print jobs.

The following PIP command causes the file NAME.TYP to be queued for printing:

```
PIP>NAME.TYP/SP
```

The print job will have a name derived from the name of the file, and the SHOW QUEUE (QUE /LI) display will give the file specification.

You can also send files directly to a line printer by using PIP, as shown in the following example:

```
PIP>LP1:=NAME.TYP
```

The print job will have a name derived from the name of the task, such as PIPT10, and the SHOW QUEUE (QUE /LI) display will not give the file specification, but only the file ID number, labeled FID. The job will appear in the device-specific queue corresponding to the device you specify (in this case, LP1).

Because of the Queue Manager queue system, your files will not be interspersed with someone else's files.

You can also use the DCL command COPY to send files to the line printer, as shown in the following example:

```
DCL>COPY  
From? NAME.TYP  
To? LP1:
```

The print job will have a name derived from the command and the terminal, such as COPT32, and only the file ID number, labeled FID, and not the file specification, will appear in the display from SHOW QUEUE (QUE /LI).

On RSX-11M-PLUS systems, any system or applications task that creates an output file can send that file to a printer or other spooled output device, rather than writing it to the disk. Simply replace the output file specification with the name of the device to which you wish to send the output. This is called transparent spooling.

## 2.5 Print Jobs Queued by User Tasks

There are several methods for spooling output from user-written tasks. When you spool output from within a task, you cannot control the printing directly as you can through the PRINT command. Once the print job appears in queue, however, you can alter its attributes through the SET QUEUE (QUE /MOD) command (see Section 2.7).

### 2.5.1 Opening a File on Disk and Using the .PRINT Subroutine

Your task can open a file on disk, send output to it, and close it using the PRINT\$ macro call. You can also call the .PRINT subroutine to spool the output (see the *RSX-11M-PLUS and Micro/RSX I/O Operations Reference Manual*). The file will be placed in the default queue PRINT.

In FORTRAN IV, FORTRAN IV-PLUS, and FORTRAN-77 on RSX-11M-PLUS operating systems, the equivalent is to use DISP= 'PRINT' in your OPEN or CLOSE statement.

### 2.5.2 Opening a File on Disk and Using the PRINT Command

Your task can open a file on disk, send output to it, and close it. Once your task exits, you can print the file with the PRINT command.

While this is not the simplest method of spooling, it is the only method that gives you access to the PRINT command qualifiers. You can accomplish much the same thing using the two methods described in Section 2.5.1 by waiting until the job is in the queue and then altering its attributes with the SET QUEUE command.

If you run your task from an indirect command file or batch job that includes a PRINT command after task exit, the difference between spooling from within the task or from outside it will probably not be noticeable.

You can also have the task issue the PRINT command by using the SPWN\$ directive. (Consult the *RSX-11M-PLUS Command Language Manual*.)

### 2.5.3 Opening a File on LPn:

Your task can use an OPEN\$ macro and name the output device. File Control Services (FCS) will open the file on pseudo device SP0:. When your task has finished writing to this file, you can close it with a CLOSE\$ macro.

The file will be placed in the device-specific queue for the device you named. The file will be deleted after it is printed. This file is not in any directory but is identified only by a file ID number, which will be labeled FID in the SHOW QUEUE display.

In FORTRAN IV, FORTRAN IV-PLUS, and FORTRAN-77 on RSX-11M-PLUS systems, the equivalent is to use CALL ASSIGN and CALL CLOSE or the OPEN and CLOSE statements, using a file specification of LPn:.

## 2.6 SHOW QUEUE (QUE /LI) and SHOW PROCESSOR (QUE /LI:DEV) Commands

You may use the commands described in this section to display information about queues, the jobs in the queues, spooled devices, and batch processors.

The SHOW QUEUE (QUE /LI) command, described in Section 2.6.1, shows all jobs in all queues. Qualifiers (switches) can be used to limit the display to specific queues, specific jobs, and the like.

You may examine the contents of your system's queues (in decreasing degree of detail) by entering one of the following command qualifiers (switches):

- /FULL (/FU)
- /FILES (/LI)
- /BRIEF (/BRIEF)

The SHOW PROCESSOR (QUE /LI:DEV) command, described in Section 2.6.2, displays information about spooled devices and batch processors under the control of the Queue Manager.

### 2.6.1 SHOW QUEUE Command

SHOW QUEUE (QUE /LI) displays information about print jobs and batch jobs.

#### Format

```
DCL> SHOW QUEUE [queuename][/qualifier[s]]
MCR> QUE/LI [queuename:][[uic]][jobname]/switch[es]
```

DCL Command Qualifiers	MCR Job Switches
/FULL	/FU
/FILES	/LI
/BRIEF	/BR
/DEVICE	/LI:P
/ENTRY:nnn	/EN:nnn
/FORMS[:n]	/FO[:n]
/NAME:jobname	jobname
/OWNER_UIC:uic	[uic]
/PRINT	/LI:P
/BATCH	/LI:B

## DCL Command Qualifiers and MCR Job Switches

**/FULL**

**/FU**

Displays detailed information about queues, queue assignments, jobs, the attributes of jobs in queues, and files that compose jobs in queues.

**/FILES**

**/LI**

Displays information about queues, queue assignments, jobs in queues, and files within jobs in queues. The attributes of the jobs are not displayed. The form number of the job and the attributes of the files are only displayed if they differ from the default values. This display format is the default of SHOW QUEUE (QUE /LI).

**/BRIEF**

**/BR**

Displays only queues, queue assignments, and jobs in queues.

You may examine specific attributes of queues by entering one of the following qualifiers (switches).

**SHOW QUEUE queuename**

**QUE queuename:/LI**

Displays information for a single queue. The information displayed may be for either print queues or batch queues.

**/DEVICE**

**/LI:P**

Limits information to print queues.

**/ENTRY:nnn**

**/EN:nnn**

Lists information regarding a particular job entry, referenced by the job's unique entry number.

**/FORMS[:n]**

**/FO[:n]**

Lists information on jobs that are to be printed on a specified form. If n is omitted, the display shows all jobs that are other than Form 0.

**/NAME:jobname**

**jobname**

Lists information on jobs with the specified job name.

**/OWNER\_UIC:uic**

**[uic]**

Lists information on jobs owned by the specified UIC.

```
/PRINT
/LI:P
```

Lists information on print queues. This qualifier produces the same information as the /DEVICE qualifier (DCL).

```
/BATCH
/LI:B
```

Lists information on batch queues.

The following examples show the three output displays FULL, FILES, and BRIEF.

### Examples

1. DCL>SHOW QUEUE/FULL

```
MCR>QUE /FU
```

```
** PRINT QUEUES **
```

```
PRINT => LPO (LP1)
```

```
[7,25] DRGCL   ENTRY:22   FORM:0   ACTIVE ON LPO
      PRI:50  LEN:0  PAGE:0  NORESTART  FLAG:JOB  NOLOWER  COP:1
      > 1 DBO:[11,34]DRGCL.LST;1  COP:1   NODELETE
[7,40] LOGIN   ENTRY:24   FORM:0
      PRI:50  LEN:0  PAGE:0  NORESTART  FLAG:JOB  NOLOWER  COP:1
      1 DB7:[7,40]LOGIN.CMD;120  COP:1   NODELETE
```

```
LPO => LPO
```

```
LP1 => (LP1)
```

```
** BATCH QUEUES **
```

```
BATCH => BAPO
```

```
[7,40] X      ENTRY:23           ACTIVE ON BAPO
      PRI:50  NORESTART  LOG PRINT:PRINT
      > 1 DB7:[7,40]X.BAT;1           NODELETE
```

```
>
```

This display shows all information about the queues and jobs in queues. In this example, the Queue PRINT can send jobs to LP0: or LP1:. LP1: has been stopped by the system manager and is therefore not eligible to accept jobs. (This is indicated by the processor name enclosed in parentheses.)

There are two print jobs in the Queue PRINT:

- [7,25]DRGCL, entry number 22, which is currently being printed on LP0:
- [7,40]LOGIN, entry number 24, which is waiting to be printed

The batch job X, entry number 23, is running on batch processor BAPO:.

The attributes of the job are indicated in the display, and the files that make up each job are also listed. The right angle bracket ( > ) indicates which file of a job is currently being processed.

2. DCL>SHOW QUEUE [/FILES]

```
MCR>QUE /LI
** PRINT QUEUES **
PRINT => LPO (LP1)
  [7,25] DRGCL      ENTRY:22          ACTIVE ON LPO
        > 1 DBO:[11,34]DRGCL.LST;1
  [7,40] LOGIN     ENTRY:24
        1 DB7:[7,40]LOGIN.CMD;120
LPO    => LPO
LP1    => (LP1)
** BATCH QUEUES **
BATCH => BAPO
  [7,40] X         ENTRY:23          ACTIVE ON BAPO
        > 1 DB7:[7,40]X.BAT;1
```

This display, the default format, shows the queues in the same state. It does not show the attributes of the job. It only shows the form number of the job and the attributes of the files if other than the default values were specified.

3. DCL>SHOW QUEUE/BRIEF

```
MCR>QUE /BR
** PRINT QUEUES **
PRINT => LPO (LP1)
  [7,25] DRGCL      ENTRY:22          ACTIVE ON LPO
  [7,40] LOGIN     ENTRY:24
LPO    => LPO
LP1    => (LP1)
** BATCH QUEUES **
BATCH => BAPO
  [7,40] X         ENTRY:23          ACTIVE ON BAPO
```

This display only shows the names, entry numbers, and status of the jobs in the queues. Also, if a form other than Form 0 is specified for a print job, the form will appear in the display.

## 2.6.2 SHOW PROCESSOR Command

SHOW PROCESSOR (QUE /LI:DEV) displays information about the initialized characteristics of spooled devices and batch processors, printers, and other output devices under control of the Queue Manager.

### Format

```
DCL> SHOW PROCESSOR [processername]
MCR> QUE [processername:]/LI:DEV
```

Displays information about all processors, including batch processors and spooled input devices, as follows:

```
DCL>SHOW PROCESSOR/BATCH
MCR>QUE /LI:DEV:B
```

Shows all batch processors.

```
DCL>SHOW PROCESSOR/PRINT or SHOW PROCESSOR/DEVICE
```

```
MCR>QUE /LI:DEV:P
```

Shows all print processors.

```
DCL>SHOW PROCESSOR/INPUT or SHOW PROCESSOR/CARD_READER
```

```
MCR>QUE /LI:DEV:I
```

Shows spooled card reader devices, described in Section 3.9.

### Example

```
DCL>SHOW PROCESSOR
```

```
MCR>QUE /LI:DEV
```

If you enter one of the preceding commands, information about print processors and batch processors and spooled input devices will be displayed at your terminal, as shown in the following example:

```
** SPOOLED DEVICES **
LP0  <= PRINT LP0
      LOWER      FORM:0  FLAG:2
      CURRENT JOB: [301,31]CAESAR  ENTRY:59
LP1  (STOPPED) <= LP1
      NOLOWERFORM:1  FLAG:1
**BATCH PROCESSORS **
BAPO  <= BATCH
```

This display shows all information about spooled devices and batch processors. In this example, spooled device LP0: can receive jobs from queues PRINT and LP0.

Spooled device LP1: has been stopped by the system manager and is therefore not eligible for receiving jobs for printing.

The batch processor, BAP0, can only receive jobs from the queue BATCH.

LP0: is currently printing job CAESAR, entry number 59.

## 2.7 SET QUEUE (QUE /MOD) Command

SET QUEUE (QUE /MOD) modifies attributes given to either print jobs, batch jobs, or files within jobs in queues. Such jobs and files have been entered in queues by either the PRINT command or the SUBMIT command, and their attributes have been set by the qualifiers to these commands. You cannot change the attributes of an active job, a job on which processing has already begun.



## Job Formats

```
DCL> SET QUEUE queueName jobName/qualifier[/qualifier[s]]
DCL> SET QUEUE/ENTRY:nnn/qualifier[/qualifier[s]]
MCR> QUE queueName:jobName/MOD/switch[/switch[es]]
MCR> QUE /EN:nnn/MOD/switch[/switch[es]]
```

### DCL Job Qualifiers

```
/AFTER:(dd-mmm-yy hh:mm)
/JOBCOUNT:n
/FORMS:n
/LENGTH:n
/LOWERCASE
/PAGE_COUNT:n
/PRIORITY:n
/[NO]RESTART
/UPPERCASE
```

### MCR Job Switches

```
/AF:hh:mm:dd-mmm-yy
/CO:n
/FO:n
/LE:n
/LOW
/PA:n
/PRIO:n
/[NO]RES
/[NO]LOW
```

## File Formats

```
DCL> SET QUEUE queueName jobName/FILE_POSITION:n/qualifier[/qualifier[s]]
DCL> SET QUEUE/ENTRY:nnn/FILE_POSITION:n/qualifier[/qualifier[s]]
MCR> QUE queueName:jobName/MOD/FILE:n/switch[/switch[es]]
MCR> QUE /EN:nnn/MOD/FILE:n/switch[/switch[es]]
```

### DCL File Qualifiers

```
/FILE_POSITION:n
/COPIES:n
/[NO]DELETE
```

### MCR File Switches

```
/FILE:n
/CO:n
/[NO]DEL
```

## DCL Job Qualifiers and MCR Job Switches

**/AFTER:(dd-mmm-yy hh:mm)**

**/AF:hh:mm:dd-mmm-yy**

Changes the time after which your job will be printed. The job will be BLOCKED until the time and date you specify. The job will not necessarily be printed at exactly the time you state but will be eligible after the time you state.

If you do not supply the calendar field, the default is the current date. If you do not supply the clock field, the default is midnight on the date given in the calendar field.

If you supply both the clock and calendar fields, you must separate them with a space (DCL) or a colon (MCR).

**/JOBCOUNT:n**

**/CO:n**

Changes the number of copies of a job you wish to have printed. See File Qualifiers in this section for an explanation of how to modify the number of copies of a file within a job. See Section 2.2 for an explanation of the /COPIES file qualifier as it applies to print jobs.

**/FORMS:n**

**/FO:n**

Changes the FORMS attribute of your print job. See Section 2.2 for an explanation of the FORMS attribute as it applies to print jobs.

**/LENGTH:n**

**/LE:n**

Changes the length of a logical page (number of lines per printed page) of your print job. A line printer will move to the top of a physical page every time n lines have been printed on a page. See Section 2.2 for an explanation of the /LENGTH qualifier as it applies to print jobs.

**/LOWERCASE**

**/LOW**

Queues your job to a line printer initialized with the /LOWERCASE qualifier.

**/PAGE\_COUNT:n**

**/PA:n**

Changes the limit on the number of pages your job may produce. See Section 2.2 for an explanation of the /PAGE\_COUNT:n qualifier as it applies to print jobs.

**/PRIORITY:n**

**/PRIO:n**

Changes the queue priority of a print job. Nonprivileged users may set priorities from 0 to 150. Privileged users may set priorities from 0 to 250. See Section 2.2 for an explanation of the /PRIORITY qualifier as it applies to print jobs.

**/[NO]RESTART**

**/[NO]RES**

Changes the restart attribute of your job. If you specify /RESTART (/RES), your job will start again from the beginning when it is interrupted while ACTIVE. If you specify /NORESTART (/NORE), the default, your job will start where it stopped when interrupted while ACTIVE.

**/UPPERCASE**

**/NOLOW**

Queues the job to a line printer that may or may not be initialized with the /LOWERCASE qualifier.

## **DCL File Qualifiers and MCR File Switches**

**/FILE\_POSITION:n**

**/FI:n**

Changes the operation of /COPIES:n (/CO:) or /[NO]DELETE (/[NO]DEL) as they apply to a file contained in a job in queue. The number n refers to the file position in the job. Use SHOW QUEUE to determine its position.

`/COPIES:n`

`/CO:n`

Changes the number of copies of a file you wish to have printed within a single printing of a print job.

`/[NO]DELETE`

`/[NO]DEL`

Changes the delete status of a single file contained in a print job.

The example following shows the display issued in response to the `SHOW QUEUE/FULL (QUE /FU)` command:

```
** PRINT QUEUES **
PRINT => (LP0)
  [303,5] MURIEL  ENTRY:696  FORM:0
        PRI:50 LEN:0 PAGE:0 NORESTART FLAG:JOB NOLOWER COP:2
          1 DB7:[303,5]MURIEL.TXT;1 COP:1  NODELETE
          2 DB7:[303,5]ALEX.TXT;1  COP:3  NODELETE
LP0    => LP0
LP1    => LP1
```

To print two copies of the file `ALEX.TXT` instead of the three copies specified in the initial print command, you would enter one of the following `SET QUEUE (QUE /MOD)` commands:

```
DCL>SET QUEUE/ENTRY:696/FILE_POSITION:2/CO:2
```

```
MCR>QUE /EN:696/MOD/FI:2/CO:2
```

The `SHOW QUEUE/FULL (QUE /FU)` command output on your terminal would look like this:

```
** PRINT QUEUES **
PRINT => (LP0)
  [303,5] MURIEL  ENTRY:696  FORM:0
        PRI:50 LEN:0 PAGE:0 NORESTART FLAG:JOB NOLOWER COP:2
          1 DB7:[303,5]MURIEL.TXT;1 COP:1  NODELETE
          2 DB7:[303,5]ALEX.TXT;1  COP:2  NODELETE
LP0    => LP0
LP1    => LP1
```

Note that the file in position 2, `ALEX.TXT`, will now be printed twice instead of three times.

## Notes

1. When you issue the `PRINT` command or the `SUBMIT` command, you can set attributes of your job through command qualifiers (switches). `SET QUEUE (QUE /MOD)` command qualifiers change these attributes.
2. You cannot change the attributes of an active job, a job on which processing has begun.
3. You can delete files or change the number of copies to be printed by specifying `/FILE_POSITION:n (/FI:n)` in the `SET QUEUE` command line.

## 2.8 Holding and Releasing Jobs

You can specify that a print job be held in queue when you issue the `PRINT` command with the `/HOLD` qualifier.

You can release held jobs with the `RELEASE (QUE /REL)` command.

## 2.8.1 HOLD (QUE /HO) Command

HOLD (QUE /HO) blocks a job in its queue until it is released with the RELEASE command described in the next section.

### Formats

```
DCL> HOLD/JOB queueName jobName
DCL> HOLD/ENTRY:nnn
MCR> QUE queueName:jobName/HO
MCR> QUE /EN:nnn/HO
```

Specifying the entry number of a job is preferable to specifying its job name. As jobs may share a job name, it is possible to hold the wrong job when the job name is specified. Holding a job by its unique entry number protects you from such mistakes. Use the SHOW QUEUE (QUE /LI) command to determine the entry number of a job in a queue. If you name the queue and job, do not supply the entry number.

### Note

Please note that you cannot use the HOLD command (QUE /HO) with jobs that are active.

## 2.8.2 RELEASE (QUE /REL) Command

RELEASE (QUE /REL) releases a job that has been held in queue.

### Formats

```
DCL> RELEASE/JOB queueName jobName
DCL> RELEASE/ENTRY:nnn
MCR> QUE queueName:jobName/REL
DCL> QUE /EN:nnn/REL
```

The following example shows the display issued in response to a SHOW QUEUE/FULL (QUE /FU) command:

```
** PRINT QUEUES **
PRINT => LPO
  [303,5] MURIEL  ENTRY:698  FORM:0  HELD
        PRI:50  LEN:0  PAGE:0  NORESTART  FLAG:JOB  NOLOWER  COP:1
        1 DB7: [303,5]ALEX.TXT;1  COP:1  NODELETE
LPO    => LPO
LP1    => LP1
```

To release the job MURIEL, you would type one of the following RELEASE (QUE /REL) commands on your terminal:

```
DCL>RELEASE/ENTRY:698
DCL>RELEASE/JOB PRINT MURIEL
MCR>QUE /EN:698/REL
MCR>QUE PRINT:MURIEL/REL
```

After the job MURIEL is released, the display issued in response to the SHOW QUEUE/FULL (QUE /FU) command looks like the following:

```
** PRINT QUEUES **  
PRINT => LPO  
  [303,5] MURIEL  ENTRY:698  FORM:0  ACTIVE ON LPO  
        PRI:50  LEN:0  PAGE:0  NORESTART  FLAG:JOB  NOLOWER  COP:1  
        > 1 DB7:[303,5]ALEX.TXT;1  COP:1  NODELETE  
LPO    => LPO  
LP1    => LP1
```

### Note

The HELD status of the job in the previous example was first changed to WAITING, and then, since there were no jobs in queue, became ACTIVE.



## Chapter 3

---

### Preparing a User Batch Job

A batch job is one or more files containing commands and data that are submitted to the Queue Manager (QMG) for execution by the batch processor. The batch processor uses a software "terminal," called the virtual terminal (VTnn:), to process the batch job. Thus, you may use your terminal to do other work while the batch job is executing. You may also determine when the batch job will execute. For example, you may wish to run batch jobs late at night, when the central processing unit (CPU) may be less active.

The SUBMIT command is used to send the batch file or files to the Queue Manager, in the following format:

```
DCL>SUBMIT filename.bat,filename.bat
```

```
MCR>SUB filename.bat,filename.bat
```

When more than one file is submitted with a single SUBMIT command, the virtual terminal processes all the files as a single batch job. By default, the batch processor creates and prints a log file when it executes the batch job. A log file is a record of the virtual terminal's activity in processing the job. When you submit more than one batch file with a single SUBMIT command, the logs of all the files appear in one log file. If you wish to have a copy of the log file placed in your directory instead, you may use the /NOPRINT qualifier with the SUBMIT command. For more information on qualifiers to the SUBMIT command, see Section 3.7.

For purposes of identification, the virtual terminal processes batch jobs under the User Identification Code (UIC) of the user who submitted the job. Privileged users may submit batch jobs under UICs other than their own.

Because a batch job is executed on a virtual terminal, it must log itself in and out and must include all the data a user would supply in an interactive terminal session. It must include responses to system prompts and instructions to the batch processor on how to proceed if a WARNING, ERROR, or SEVEREERROR exit status code is returned by a task it is executing. The batch job therefore performs the following steps:

- Logs itself in
- Issues commands
- Supplies data for tasks it uses

- Provides responses to exit status codes
- Logs itself out

The following is an example of a simple batch file:

```
$JOB
$ON WARNING THEN STOP
$CREATE ORIOLE.TXT
These lines of text are data. They will be inserted into the
file ORIOLE.TXT by the batch processor.
$PRINT ORIOLE.TXT
$EOJ
```

This batch job uses the DCL command CREATE to create a file named ORIOLE.TXT, supplies data to be inserted into the file, and prints the file. It also uses the batch-specific command ON to direct the batch processor to stop executing the job if a WARNING is encountered. (ON WARNING THEN STOP is the batch processor's default setting.) The batch processor automatically simulates a CTRL/Z, which causes CREATE to close the file, when it encounters the next command in the batch file.

## 3.1 Commands

The commands in a batch job may be either command language interpreter (CLI) commands (such as DCL or MCR) or batch-specific commands, which are described in Section 3.4. All commands must be preceded by the dollar sign (\$). The batch processor reads lines not preceded by the dollar sign as data. The commands JOB and EOJ are batch-specific login and logout commands, respectively. The DCL commands LOGIN and LOGOUT and their MCR equivalents, HELLO and BYE, cannot be substituted for JOB and EOJ. The format of the batch command line is shown next.

### Format

\$command

DCL and MCR commands that do not fit on a single line can be continued by placing a hyphen (-) in the last position of the first line. The continued portion of the line should not be preceded by a dollar sign and should not begin with a space, tab space, or label. (Labels are explained in Section 3.3.) Batch-specific commands cannot be continued.

## 3.2 Data

Data is any information exclusive of commands that the batch processor requires in order to process the batch file. Lines of data in a batch file are not preceded by dollar signs. You can use the DATA command to mark the beginning of a block of data in a file if you wish to use one of the DATA command qualifiers. If you do not use a DATA command qualifier, you can omit the DATA command. For more information on DATA command qualifiers, see Section 3.4.3. If you use the DATA command, you must also use the EOD command to mark



the end of the block of data. Since DATA and EOD are commands, they must be preceded by the dollar sign. Following is the previous example with the beginning and end of the data block marked:

```
$JOB
$ON WARNING THEN STOP
$CREATE ORIOLE.TXT
$DATA
These lines of text are data. They will be inserted into ORIOLE.TXT by the batch
processor.
$EOD
$PRINT ORIOLE.TXT
$EOJ
```

### 3.3 Labels

Labels mark places in batch files. They are used with the batch-specific command GOTO and direct the batch processor to continue processing at the place marked. If a WARNING, ERROR, or SEVEREERROR exit status code is returned by a task in a batch job, a GOTO command followed by a label can direct the batch processor to continue executing the remainder of the job rather than stop. Labels are preceded by the dollar sign and followed by a colon (:). Spaces (including tab spaces) preceding the colon are ignored. Labels are user defined and are limited to six alphanumeric characters in length.

#### Format

\$label:

In the following example, the batch job performs two operations: it creates a file and runs the program HI. If the batch processor encounters an error in creating ORIOLE.TXT, it will still run the HI program because the ON statement directs it to the label PUFFIN.

```
$JOB
$ON ERROR THEN GOTO PUFFIN
$CREATE ORIOLE.TXT
$DATA
These lines of text are data. They will be inserted into the
file ORIOLE.TXT by the batch processor.
$EOD
$PRINT ORIOLE.TXT
$PUFFIN: RUN HI
$EOJ
```

## 3.4 Batch-Specific Commands

Table 3-1 lists the batch-specific commands and their functions. Batch-specific commands may be abbreviated to the first three letters.

**Table 3-1: Batch-Specific Commands**

Function	Command
Login and logout commands	JOB EOJ
Data commands	DATA EOD
Sequence Control (error handling)	STOP CONTINUE GOTO ON SET IF
Comment	!

Each command is described individually in the following sections.

### 3.4.1 JOB

The JOB command marks the beginning of a user batch file.

#### Format

`$JOB[/qualifier] [loglabel] [[uic]]`

#### Command Qualifier

`/TIME:(hh:mm)`

or

`/TIME:m`

#### Parameters

##### loglabel

When more than one batch file at a time is submitted to the batch processor, the activity of all the files appears in one log file. The loglabel is therefore useful in differentiating between the jobs in the log file. Note that the loglabel appears only in the log file and is not included in the display issued in response to the SHOW QUEUE (QUE /LI) command. The loglabel cannot exceed six alphanumeric characters in length.

##### [uic]

In executing a batch job, the batch processor by default uses the current protection UIC of the user who submitted the job. Privileged users may submit batch jobs under any UIC, but nonprivileged users must use the UIC under which the job was submitted. It is not necessary to include a UIC within the batch file, but doing so prevents unauthorized users from running the job. If you do not use a UIC, users who have access to the file may run

the batch job. If you are a nonprivileged user, you may use only your own UIC in the JOB command. UICs must always be enclosed in brackets.

If you use the slash format for the UIC [g/m], you can suppress most login messages. Login messages are the system notices that appear whenever a user logs in.

### Command Qualifier (Job Switch)

**/TIME:(hh:mm)**

or

**/TIME:m**

Limits the job to a specified number of hours (hh) and minutes (mm) of CPU time. The default is 3 minutes. Note that this qualifier pertains to CPU time only and not to the duration of the batch job. The /TIME qualifier is useful in preventing a batch job from being caught in an infinite loop. Batch processing allows backward referencing, which increases the likelihood that infinite loops will occur. Backward referencing in a batch file is the placing of a label at a point in the file before the GOTO command that directs the batch processor to that label.

### Examples

**\$JOB**

Shows the simplest form of the JOB command. It logs the user on to the virtual terminal. The UIC is the same as the UIC under which the SUBMIT command was issued.

**\$JOB [20/20]**

Logs the batch job in under the UIC [20,20] and suppresses login messages. Nonprivileged users must have the UIC [20,20] in order to run this job. Privileged users may run this job regardless of their UIC.

**\$JOB ORIOLE**

Shows the loglabel ORIOLE in the job heading of the log file.

**\$JOB/TIME:5**

Specifies that the job must not exceed 5 minutes of CPU time. The job is ended automatically after 5 minutes of CPU time have elapsed.

**\$JOB/TIME:(1:25)**

Specifies that the job must not exceed 1 hour and 25 minutes of CPU time.

### 3.4.2 EOJ

The EOJ command marks the end of a user batch file.

#### Format

`$EOJ`

If a batch file ends without an EOJ command, the batch processor stops processing the job as if the command were present.

If the batch job includes two or more batch files, the batch processor logs out the first before it begins processing the second.

### 3.4.3 DATA

The DATA command marks the beginning of a data block in the batch file. Data is any user input except commands that is required to process the batch file. The DATA command should immediately follow the command that requests input. The DATA command is optional unless one of its qualifiers is needed. The EOD command is required to end the data block.

#### Format

`$DATA[/qualifier[s]]`

#### Command Qualifiers (Job Switches)

`/NOCOPY`

`/DOLLARS[:"string"]`

`/NOCOPY`

This qualifier prevents printing of the data blocks in the log file. The default is `/COPY`.

`/DOLLARS[:"string"]`

This qualifier permits you to include data preceded by the dollar sign (\$) in the data block. Otherwise, the batch processor interprets the first line beginning with a dollar sign as marking the end of the data block.

The `/DOLLARS` qualifier also allows you to use an argument string (1 to 15 characters in length) as a terminator for a data block. The batch processor ignores `$EOD` until it encounters the argument string.

#### Examples

```
$JOB
$RENAME
$DATA
SLEEP.TXT
MARK.LIS
$EOD
$PRINT MARK.LIS
$EOJ
```

Renames the file SLEEP.TXT to MARK.LIS and prints MARK.LIS. The DATA command marks the beginning of the data block SLEEP.TXT and MARK.LIS, and `$EOD` marks its end. This

batch job would run successfully without the DATA and EOD commands.

```
$JOB
$RUN BILL.TSK
$DATA/DOLLARS
$1
$2
$3
$5
$10
$20
$50
$EOD
$EOJ
```

Shows that the task named BILL reads data with the dollar sign (\$) in the first position. The /DOLLARS qualifier allows this data to be read as data instead of being read as commands. The EOD command ends the data block.

```
$JOB
$CREATE CHECKBOOK.BAT
$DATA/DOLLARS:"LIST"
$JOB
$RUN CHECKBOOK.TSK
FOOD
$20.98
CAR PAYMENT
$148.58
$EOD
$EOJ
LIST
$SUBMIT CHECKBOOK.BAT
$EOJ
```

Shows a batch file that creates another batch file, CHECKBOOK.BAT, and submits it for processing. The batch processor reads \$EOD and the first \$EOJ as data because of the /DOLLARS:"LIST" line, which ensures that the batch processor will read all lines as data until it reaches the word LIST.

```
$JOB
$!
$!
$!This example creates a batch file
$!within a batch file
$!
$PIP CAT.BAT=TI:
$DATA/DOLLARS:"PURR"
$JOB
$RUN MEOW.TSK=yes
$DATA/DOLLARS
$3.00 tuna
$2.98 liver
$1.58 catnip
$EOD
$EOJ
PURR
$EOJ
```

Shows a batch file in which PIP reads lines of data from TI: (the user's terminal) and copies them to the file CAT.BAT. The batch processor does not terminate data until it encounters PURR. The file CAT.BAT includes \$EOD and the first \$EOJ.

The MCR /EST=yes qualifier in the preceding example ensures that the MEOW task will return exit status codes to the batch processor. If this qualifier is not specified, the task may or may not return exit status codes. The equivalent qualifier in DCL is /STATUS:TASK.

### Notes

1. You can omit the DATA command in a batch file if you do not need to use its qualifiers. Any line without the dollar sign (\$) in the first position will be interpreted as data if the preceding command called for user input. The batch processor continues to read lines as data until it encounters a dollar sign.
2. Qualifiers to the DATA command affect only the data block with which they are used, not subsequent data blocks.
3. The DATA command does not appear in the log file.

### 3.4.4 EOD

The EOD command marks the end of a data block.

#### Format

```
$EOD
```

#### Note

The EOD command must be used if you have specified either DATA/DOLLARS or DATA/DOLLARS:"\$EOD".

### 3.4.5 STOP

The STOP command stops execution of the batch job. It can be used alone or with the ON or IF commands.

#### Formats

```
$STOP  
$ON statuscode THEN STOP  
$IF statuscode THEN STOP
```

#### Notes

1. The STOP command without a qualifier is a batch-specific command. A STOP command followed by a qualifier or any other command line element is passed to the current CLI. The batch-specific STOP command is equivalent to the EOJ command. A batch file can include more than one STOP command used with the GOTO, ON, or IF commands.
2. If more than one batch job is submitted to the batch processor at a time, the batch processor continues with the next batch job upon encountering a STOP command.

3. To stop a batch job interactively, use the DELETE/ENTRY:nnn (QUE /DEL) command, described in Section 2.3.

### 3.4.6 CONTINUE

CONTINUE is a no-operation command. It can be used alone or with the ON or IF commands.

#### Formats

```
$CONTINUE
$ON WARNING THEN CONTINUE
```

#### Note

For the most part, this command is used with the ON or IF commands. Its main use is to keep the batch processor from stopping execution of a job when it encounters a WARNING or more severe error.

### 3.4.7 GOTO

The GOTO command directs the batch processor to continue processing at a line beginning with a specified label and to ignore any intervening lines. It can be used alone or with the ON or IF commands.

#### Format

```
$GOTO label
.
.
.
$label:[command]
```

#### Parameter

##### label

The label cannot exceed six alphanumeric characters in length and must be terminated by a colon (:).

The line with the referenced label must have the following form:

```
$label:[command]
```

#### Examples

```
$GOTO HANDLE
$HANDLE:DEV
```

Directs the batch processor to continue processing at the label HANDLE, ignoring intervening lines. Note that when the name of the label is used with the GOTO command, it does not include the colon (:). When the batch processor reaches the label HANDLE, it executes the MCR command DEV to display all the devices on the system.

**\$ON WARNING THEN GOTO HANDLE**

Directs the batch processor to continue execution at the label HANDLE and ignore intervening lines if a task returns a WARNING or more severe error.

### 3.4.8 ON

The ON command specifies an exit status code and the action to be taken by the batch processor when that code or a more severe code is returned by subsequent commands and tasks. The default setting is: \$ON WARNING THEN STOP.

#### Format

\$ON statuscode THEN { STOP  
CONTINUE }  
GOTO label }

#### Parameter

##### statuscode

In order of increasing severity, the status codes returned by tasks in a batch job are WARNING, ERROR, SEVEREERROR, and SUCCESS. However, the batch processor does not recognize the SUCCESS exit status code, so it does not appear in batch logs. If you specify ERROR, the batch processor will ignore all WARNINGS. If you specify SEVEREERROR, the batch processor will ignore both ERRORS and WARNINGS. (SEVEREERROR is one word and may be abbreviated to SEV.)

#### Examples

**\$ON WARNING THEN STOP**

Shows the default setting of the ON command. If you do not override the default with some other ON command, then any WARNING or more severe exit status code will stop execution of the batch job.

**\$ON ERROR THEN GOTO NEXT**

Specifies that the batch processor should ignore all WARNINGS, and when it receives an ERROR or SEVEREERROR code go directly to the line beginning \$NEXT: and process the command following the label and all subsequent commands.

#### Notes

1. Once the condition specified by the ON command is encountered, the ON command is canceled and the default setting, ON WARNING THEN STOP, is restored. You can reset the ON command at any point in the batch job.
2. The conditions that cause a particular exit status code to be returned are determined individually by each system task.
3. The ON command remains set until superseded by another ON command, by the specified error condition, by a SET NO ON command (see Section 3.4.9), or by the end of the user batch file. Compare with the IF command.



4. Directory [USER] on the system disk contains tasks that do nothing but exit with status. You can copy these tasks and use them to test uses of the sequence control commands. They are: SUCCESS.TSK, WARNING.TSK, ERROR.TSK, and SEVERE.TSK.

### 3.4.9 SET

The SET command enables or disables the ON command setting.

#### Format

```
$SET [NO] ON
```

#### Parameters

##### NOON

Disables the ON command, including its default setting. The batch processor ignores exit status codes (unless an IF command follows a particular command).

The space is optional. Either NO ON or NOON is acceptable.

##### ON

Reinstates the setting of the ON command in effect before the SET NO ON command was executed.

#### Notes

1. Although the SET NO ON command disables ON, any error messages returned by commands or tasks are included in the batch log.
2. The SET NO ON command has no effect on any IF command in a batch file.

### 3.4.10 IF

The IF command directs the batch processor to check for a specified exit status code following execution of the preceding command in a batch file. The THEN part of the IF command indicates the action to be taken by the batch processor when the specified code is returned.

#### Format

```
$IF statuscode THEN { STOP  
                     CONTINUE  
                     GOTO label }
```

#### Parameter

##### statuscode

The exit status codes returned in batch jobs are WARNING, ERROR, SEVEREERROR, and SUCCESS. The batch processor does not recognize the SUCCESS code, so it does not appear in the batch log file. IF SUCCESS is not executable.

## Example

```
$JOB AONE[303,14]
$!
$! IF command example
$!
$RUN WONDER.TSK/EST=yes
$IF SEVEREERROR THEN GOTO BOMB
$IF ERROR THEN GOTO ALAS
$IF WARNING THEN GOTO OKAY
$GOTO REST ! SUCCESS ASSUMED
$BOMB:
$BRO TT10:SEVERE ERROR. RUN ENDS
$STOP
$ALAS:
$BRO TT10:ERROR. RUNNING WONDER.TSK
$RUN OLD.TSK/EST=yes
$GOTO REST
$OKAY:
$BRO TT10:WARNING RECEIVED
$REST:EOJ
```

## Note

If the status code specified in the IF command matches the task's exit status, the specified action is taken and any ON command in effect is ignored. If the status code in the IF command does *not* match the task's exit status code, then the ON command takes effect if it is enabled.

## 3.4.11 Comments

The exclamation point (!) is used to include comments in the batch log file. It is the only comment character that should be used in batch files. The batch processor does not send text preceded by the exclamation point to the command line interpreter (CLI). Comments beginning with any other characters will be passed to the CLI and could cause problems with batch processing.

### Formats

!*comment*

!*[label:]batch-specific command comment*

The comment can include any character. The batch processor does not process anything that appears after the exclamation point. However, comments following CLI commands are sent to the CLI. Consult the appropriate command language manual for more information on this use of a comment. Comments appear in the batch log file.

### Example

```
!$BIG QUESTION?
```

Indicates a comment line. The second dollar sign (\$) has no effect.

## 3.5 Indirect Command Files

The at sign (@) invokes an indirect command file from within a batch job. When the command is executed, the indirect command file is accessed and the commands in it are executed. The at sign command is both a DCL and an MCR command, but it may not be a valid command in user-written CLIs.

Note that the default file type for batch files is not BAT but CMD. Therefore, it is unwise to have batch and indirect command files with the same name because you could inadvertently submit an indirect file as a batch file. Giving batch files a file type of BAT also will help to eliminate confusion.

### Format

```
$@indirect[.CMD]
```

### Examples

**\$@BACK**

This command line finds the file named BACK.CMD on the default device and User File Directory (UFD) and invokes it.

**\$@DBO: [20,20]BACK**

This command line finds the file named BACK.CMD at the specified location and passes it to the indirect command file facility.

## 3.6 Allocating Devices and Mounting Volumes from Batch Jobs

In interactive mode, you usually must allocate a device, physically load the device on a drive, and mount the volume before you can issue commands involving that volume.

In order to allocate a device through a batch job that may be run in your absence, you can use the /TYPE qualifier with the ALLOCATE command to allocate the first available device of the particular type that you specify. You can also specify a logical name to be assigned to the device. The command format is shown next, where dd represents the name of the physical device and ddnn: the logical name.

### Format

(DCL) \$ALLOCATE [/TYPE:devtype] dd ddnn:

(MCR) \$ALL dd=ddnn: [/TYPE:devtype]

You may also use a logical name for the device, since you may not know which device will be available when the batch job is processed. The following example illustrates the format, in

which RK07 is the device type, DM are the first two letters of the name of the physical device, and IN: is the logical name of the device:

```
$JOB
$!
$! This batch job does a validity check on an RK07
$! disk pack labeled CHECKME.
$!
$ALLOCATE/TYPE:RK07 DM IN:
$MOUNT/WAIT IN: CHECKME
$RUN $VFY/EST=yes
IN:
$DISMOUNT/NOUNLOAD IN:
$EOJ
```

You can mount a volume through a batch job by using the /WAIT qualifier to the MOUNT command. This qualifier sends a message to the console terminal, asking the operator to mount the volume with the specified label on the drive specified. The command format is shown next.

#### Format

```
(DCL) $MOUNT/WAIT [logical_name] [volume_label]
(MCR) $MOU [logical_name] [volume_label]/WAIT
```

After sending the message, the MOUNT command stops executing. The operator then loads the device and uses the UNSTOP command to restart the MOUNT command. When the MOUNT command has completed, the remaining commands in the batch file are executed. All devices privately allocated are automatically deallocated when the batch job is finished.

If you yourself intend to load the device on a specific drive, you need only use the standard form of the ALLOCATE and MOUNT commands as shown in the following examples:

```
$JOB
$!
$! This batch job will get a full directory listing
$! of an RLO2 disk that is already loaded and spun up.
$!
$SET TERMINAL MCR
$ALL DL1:
$MOU DL1:IMREADY
$PIP LPO:=DL1:[*,*]/FU
$DMO DL1:
$DEA DL1:
$EOJ
```

### 3.7 SUBMIT (SUB) Command and Qualifiers

The SUBMIT command is used to send batch files to the Queue Manager (QMG) for execution by the batch processor. The format of the SUBMIT command and a list of command qualifiers, file qualifiers, and log file qualifiers follow. Command qualifiers modify the action of the command as it affects the batch job in the queue. File qualifiers modify the action of the command as it affects the file submitted, log file qualifiers modify the action of the command as it affects the log file.

#### Format

```
DCL> SUBMIT[/qualifier] filespec[/filequalifier/logfilequalifier]
MCR> SUB [queuename:]jobname/jobswitch/logfileswitch]=filespec[/fileswitch]
```

DCL Command Qualifiers	MCR Job Switches
/AFTER:(dd-mmm-yy hh:mm) (mm/dd/yy hh:mm)	/AF:hh:mm:dd-mmm-yy
/AFTER:TOMORROW	/AF::TOMORROW
/[NO]HOLD	/[NO]HO
/NAME:jobname	jobname
/[NO]RESTART	/[NO]RES
/PRIORITY:n	/PRIO:n
/QUEUE:queuename	queuename:
DCL File Qualifiers	MCR File Switches
/[NO]DELETE	/[NO]DEL
/[NO]TRANSFER	/[NO]TR
DCL Batch Log File Qualifiers	MCR Batch Log File Switches
/NAME:jobname	jobname
/[NO]LOGFILE	/[NO]LO
/[NO]PRINT:printqueue	/[NO]PRIN:printqueue

#### Note

The MCR jobname and queuename: switches do not require a preceding slash, and do not execute if a slash is used.

#### Parameter

##### filespec

The file specification of the batch file or files.

If you specify multiple files, you must separate them by commas. You can use the asterisk wildcard character (\*) in the directory, name, type, and version fields of the file specifications.

If the batch file contains no UIC after the JOB command, any user with access to the file can run the batch job.

If your file specification includes no file type, the default file type is CMD.

## DCL Command Qualifiers and MCR Job Switches

**/AFTER:(dd-mmm-yy hh:mm)**  
**(mm/dd/yy hh:mm)**

**/AF:hh:mm:dd-mmm-yy**

Ensures that the job is not executed until after the specified time. Depending on the number of jobs in the queues at that time, your job may be run immediately or later, when its turn comes up in the queue.

If your current CLI is DCL, the slashes or hyphens, colons, and the space between the calendar and clock fields are mandatory. The parentheses may be omitted if either the calendar or clock time is omitted.

If your current CLI is DCL, you can specify the calendar field in either of the following formats:

**dd/mmm/yy** Uses a 1- or 2-digit number for the day, the first three letters for the month, and two numbers for the year.

**mm/dd/yy** Uses a 1- or 2-digit number for the month, a 1- or 2-digit number for the day, and two numbers for the year.

If you omit the calendar time, the day defaults to the current date.

If you omit the clock time, the time defaults to 00:00:00.

If your system CLI is MCR and you wish to omit the clock time, you must use two additional colons (::) after the /AF switch. For an example of this, see the end of this section.

**/AFTER:TOMORROW**

**/AF:::TOMORROW**

Ensures that a batch job is run after midnight of the day the job is submitted.

**/[NO]HOLD**

**/[NO]HO**

Specifies that the job be held in its queue. You can release the job with the **RELEASE** command (**QUE /REL**), described in Section 2.8.2.

**/NAME:jobname**

**jobname**

Specifies a name for the batch job. The job name cannot exceed nine alphanumeric characters in length.

If you specify a job name, that name appears in the output from **SHOW QUEUE (QUE /LI)** commands. If you do not specify a job name, the file name of the first batch file submitted is used as the job name.

If you specify **/NOPRINT (/NOPRIN)** and a job name, the job name is used as the name of the the batch log file, which has the file type **LOG**.

Note that individual jobs in the log file can be distinguished by the name provided in the **loglabel** field of the **JOB** command in each batch file.

**/[NO]RESTART**  
**/[NO]RES**

Controls requeuing of a job if it is aborted while being processed.

If **/RESTART** (**/[NO]RES**) is specified, the job will be processed again from the beginning if the system manager has set up the system to make this possible. If a job does not restart automatically, you can restart it by means of the **RELEASE (QUE /REL)** command (see Section 2.8.2).

The default is **/NORESTART** (**/NORE**). If a job submitted without the **/RESTART** qualifier is aborted, it is deleted from the queue.

#### **Notes**

Please note that the **/RESTART** (**/RES**) qualifier and the **/DELETE** (**/DEL**) file qualifier cannot be used together. The system deletes the batch file before the job can be restarted.

Please note also that if you use the **/RESTART** (**/RES**) qualifier in submitting a batch file that is on a privately mounted disk, you must also use the **/NOTRANSFER** (**/NOTR**) qualifier in order for the job to restart if it is aborted. The **/NOTRANSFER** qualifier is required because the system deletes the transferred file before the job can be restarted. In addition, using the **/NOTRANSFER** qualifier necessitates leaving your private disk attached to the system, so that the system can access the job without transferring it.

**/PRIORITY:n**  
**/PRIO:n**

Sets the queue priority of the batch job. Nonprivileged users can set *n* to 1 to 150. Privileged users can set *n* to 1 to 250. The default is 50. The Queue Manager runs the job with highest priority first. If two jobs have the same priority, the job that has been in the queue longer runs first.

This qualifier and switch have no effect on the running priority of any tasks in a batch job. They only affect the job's position in the queue.

**/QUEUE:queuename**  
**queuename:**

Specifies the name of the batch queue in which the job is to be placed.

The default is the **BATCH** queue.

#### **DCL File Qualifiers and MCR File Switches**

**/[NO]DELETE**  
**/[NO]DEL**

Specifies whether the batch file or files will be deleted after processing.

The default is **/NODELETE**. For more information on **/[NO]DEL**, see the last example at the end of this section.

#### **Note**

Please note that the **/DELETE** and **/RESTART** qualifiers cannot be used together, because the system deletes the batch file before the job can be restarted.

**/[NO]TRANSFER**  
**/[NO]TR**

Specifies whether or not the system should make temporary copies of batch files submitted from a volume mounted on a privately allocated device. The default is **/TRANSFER (/TR)**. This permits you to dismount the volume and deallocate the device as soon as the prompt returns after the **SUBMIT** command.

If you specify **/NOTRANSFER (/NOTR)**, no copies will be made, and you must keep the volume mounted and the device allocated until the batch processing is completed.

### **DCL Batch Log File Qualifiers and MCR Log File Switches**

**/NAME:jobname**  
**jobname**

Gives a name to the batch job and the log file. This name will appear in the page heading of the log file. The name will also appear in the display from the **SHOW QUEUE (QUE /LI)** command. If you do not specify a name, the name of the first file submitted will be used.

**/[NO]LOGFILE**  
**/[NO]LO**

Suppresses the creation of a log file. The default is **/LOGFILE (/LO)**.

**/[NO]PRINT:printqueue**  
**/[NO]PRIN:printqueue**

The **/PRINT (/PRIN)** qualifier causes the log file to be queued to a line printer, printed, and deleted. The temporary file is created in **SP:[1,7]**. The **/NOPRINT (/NOPRIN)** qualifier causes a copy of the file to be placed in your directory instead of being printed. The default is **/PRINT**.

Although the **PRINT** queue is the default, you may send the log file to any print queue by specifying a different queue name with the **/PRINT** qualifier, in the following format: **/PRINT:PRINTQUEUE**. The default is **/PRINT:PRINT (/PRIN:PRINT)**.

#### **Note**

If you use the **/NOPRINT** qualifier and the first batch file on the command line is on a private disk, you must also use the **/NOTRANSFER** file qualifier. This syntax is required to ensure that the private disk has been mounted and is able to receive the batch log file when the batch job runs. In addition, you should ensure that you have write privileges for the log file directory.

#### **Examples**

```
DCL>SUBMIT JOAN.BAT
```

```
MCR>SUB JOAN.BAT
```

Submits the batch file **JOAN.BAT** for processing. The name of the job is **JOAN**. The log file **JOAN.LOG** will be printed and then deleted.



```
DCL>SUBMIT/NOPRINT JANE.BAT
```

```
MCR>SUB /NOPRIN=JANE.BAT
```

Submits the batch file JANE.BAT for processing. A permanent log file will be created in your directory. After the batch job completes, you may examine the log file JANE.LOG at your terminal or print it.

```
DCL>SUBMIT/NOLOGFILE JOAN.BAT
```

```
MCR>SUB /NOLO=JOAN.BAT
```

Submits a batch file and specifies that no log file be produced.

```
DCL>SUBMIT BABY.BAT,BOY.BAT,MAN.BAT,OLD.BAT
```

```
MCR>SUB BABY.BAT,BOY.BAT,MAN.BAT,OLD.BAT
```

Submits four batch files, which will be processed in the order submitted without interruption. The name of the batch job is BABY and the name of the log file is BABY.LOG. The four files constitute a single batch job, and the log file contains a record of each of file.

```
DCL>SUBMIT/QUEUE:BAT2/AFTER:27-JUN-89 LATE.BAT
```

```
MCR>SUB BAT2:/AF:::27-JUN-89=LATE.BAT
```

Submits the batch file LATE.BAT to the queue BAT2. It will not become eligible for processing until after midnight, June 27, 1989. Note that two extra colons are needed after the MCR switch /AF to take the place of the hours and minutes fields.

```
DCL>SUBMIT/NAME:FRETS GUITAR.BAT,BANJO.BAT/DELETE,UKE.BAT
```

```
MCR>SUB FRETS=GUITAR.BAT,BANJO.BAT/DEL,UKE.BAT/NODEL
```

Submits three batch files and assigns the name FRETS to the batch job. The batch file BANJO.BAT is deleted after it is processed. Notice that in the MCR command line, it is necessary to use the switch /NODEL on the last file. Without /NODEL, UKE.BAT would be deleted with BANJO.BAT.

### 3.8 The Batch Log File

A batch log file is a record of the virtual terminal's activity in executing a batch job. Unless you specify otherwise, the log file is printed on your system's line printer and then deleted after batch processing is completed.

The log file includes the following: the commands in the batch file or files and a notation of the time they were executed; output from these commands; any error messages or warnings received and the action taken as a result; comments; and if you have chosen to include them, the data in the file and login messages. Maps and files that have been printed by the batch job are appended to the log file when it is printed and have the same print characteristics as the log file. For further information on the print characteristics of files appended to the log file, see Section 2.4.1.

If you use the /NOLOG log file qualifier, no log file is created. If you use the /NOPRINT qualifier, the log file is not printed but does appear in your directory. However, if you have used the PRINT command within the batch file in order to print a file, or run a task that produces an output file, these files are still printed as separate print jobs.

The batch log file is divided into three general fields of information, as follows:

1. The log file heading, which contains the following:
  - Title "QMG Batch Job"
  - QMG batch job name
  - Version number of the batch processor
  - Day, month, year, and time execution of the batch job began
  - Batch log page number
  - Task name of the batch processor
2. The job flag, which appears directly after the job command and displays:
  - Job name
  - Device name of the virtual terminal the batch processor used to execute the job
  - UIC under which the batch job was logged in

This information is followed by system login messages.
3. The remaining portion of the log file, which documents the activity of the batch job and includes the following:
  - Commands, preceded by a notation of the time they were executed
  - Output from commands and tasks (normally written to your terminal)
  - Data for commands and tasks
  - Lines not executed as a result of a GOTO statement
  - Exit status messages
  - Comments

There are three line identifiers that appear in the space between the time stamps and the command lines in the log file. These identifiers are TERM, DATA, and SKIP:

TERM Identifies the line as output from a task to the virtual terminal

DATA Identifies the line as input data to a task

SKIP Indicates that the line was not executed because of a GOTO statement

Periods that appear below an identifier indicate a continuation of that particular line identifier.

The batch log file may also contain a message indicating that the severity level in a previous ON command was met or exceeded when a task exited. The message displays the exit status returned by the task.

Example 3-1 shows a user batch job, and Example 3-2 shows its log file.

### Example 3-1: Batch Job Oriole

```
$JOB ORIOLE [7,22]
$ON WARNING THEN GOTO PUFFIN
$CREATE ORIOLE.TXT
$DATA
These lines of text are data. They will be inserted into the
file ORIOLE.TXT by the batch processor.
$PRINT ORIOLE.TXT
$PUFFIN:$RUN HI
John James Audubon
$EOD
$EOJ
```

### Example 3-2: Log File for Batch Job Oriole

```
QMG Batch Job - EXAMPLE      BPR V04.11    29-SEP-87  14:48      Page 1
Processor BAPO
14:48:56      $JOB ORIOLE [7,306]
```

```
=====
User Job - ORIOLE      Terminal VT2:
                UIC = [7,306]
=====

TERM
.   RSX-11M-PLUS V4.0 BL40  [4,54] System      RAVEN
.
.   *****
.   * *
.   *   Welcome to RSX-11M-PLUS batch   *
.   *   Version 4.0 Base level 40   *
.   * *
.   *   This is file LB:[1,2]BATCH.TXT *
.   * *
.   *****
.
.   $ @LB:[1,2]SYSLOGIN.CMD
14:49:02      $ON WARNING THEN GOTO PUFFIN
14:49:02      $CREATE ORIOLE.TXT
DATA  These lines of text are data. They will be inserted into the
.   file ORIOLE.TXT by the batch processor.
14:49:03      $PRINT ORIOLE.TXT
TERM  PRI - Job 905, name "ORIOLE  ", submitted to queue "PRINT "
14:49:05      $PUFFIN:$RUN HI
TERM  This is the HI program
.   What is your name? John James Audubon
.   Hi there! John James Audubon
14:49:07      $EOD
14:49:07      $EOJ
TERM  Connect time:   0 hrs  0 mins 11 secs
.   CPU time used:   0 hrs  0 mins  4 secs
.   Task total:     15
```

In Example 3-3, PIP requires input from the virtual terminal TI: to create the file GREET.FTN. Therefore, the program itself is identified as DATA. Example 3-4 shows the output file for the user batch job in Example 3-3.

### Example 3-3: Batch Job Friend

```
$JOB FRIEND [301/31]
$!
$! Program development example - CLI is MCR.
$!
$PIP GREET.FTN=TI:
    PROGRAM GREET
    CHARACTER NAME(15)
    WRITE (5,90)
90    FORMAT (' What is your name, please?')
    READ (5,91) NAME
91    FORMAT (15A1)
    WRITE (5,92) NAME
92    FORMAT (' Hello there, ',15A1)
    CALL EXIT
    END
$FOR GREET=GREET
$TKB GREET=GREET, LB:[1,1]FOROTS/LB
$RUN GREET
Napoleon B.
$PIP GREET.*;*/DE
$EOJ
```

### Example 3-4: Log File for Batch Job Friend

QMG Batch Job - EXAMPLE            BPR V04.11            9-JUL-87 15:32            Page 1  
Processor BAPO  
15:32:14            \$JOB FRIEND [301/31]

```
=====
User Job - FRIEND            Terminal VT2:
                              UIC = [301,31]
=====
TERM
                              RSX-11M-PLUS V04 BL40    [1,54] System        RAVEN
15:32:18            $!
15:32:18            $! Program development example - CLI is MCR.
15:32:18            $!
15:32:19            $PIP GREET.FTN=TI:
                              PROGRAM GREET
                              CHARACTER NAME(15)
                              WRITE (5,90)
                              90        FORMAT (' What is your name, please?')
                              READ (5,91) NAME
                              91        FORMAT (15A1)
                              WRITE (5,92) NAME
                              92        FORMAT (' Hello there, ',15A1)
                              CALL EXIT
                              END
15:32:24            $FOR GREET=GREET
                              TERM    GREET
15:32:29            $TKB GREET=GREET, LB: [1,1]FOROTS/LB
15:33:18            $RUN GREET
                              TERM    What is your name, please?
                              DATA    Napoleon B.
                              TERM    Hello there, Napoleon B.
15:33:19            $PIP GREET.*;*/DE
15:33:20            $EOJ
                              TERM    Connect time: 1        mins.
                              .        CPU time used: 14     secs.
                              .        Task total: 15
```

## 3.9 Preparing Batch and Print Jobs from Card Decks

Batch jobs may also be submitted from card decks. When you place a deck of cards in the hopper of a spooled card reader and press the RESET button on the reader, the Card Reader Processor (CRP) reads the deck, copies the cards to a file, and submits the file to the Queue Manager. The entire deck of cards constitutes a single batch job. The Queue Manager then sends the file to the batch processor for execution.

The Queue Manager may also send the file to a line printer for printing. Printing the information that is on the cards as a file facilitates error checking. A spooled card reader is "owned" by the CRP, and other tasks may not access it.

### 3.9.1 Card Deck Format for Batch and Print Jobs

The format of a batch job card deck is similar to the format of a batch file on disk. However, a few additional control cards are necessary in order for the card deck to be read by the Card Reader Processor (CRP). The card deck includes the following cards:

1. \$SUBMIT card or \$PRINT card
2. \$PASSWORD card
3. \$JOB card
4. Cards bearing information to be batch processed or printed
5. \$EOJ card
6. End-of-file card

The \$SUBMIT card is the first card in a card deck that is being submitted for batch processing, and the \$PRINT card is the first card in a card deck that is being submitted for printing. The formats of the \$SUBMIT and \$PRINT cards are similar to those of the DCL commands SUBMIT and PRINT. The two cards are described in Sections 3.9.1.1 and 3.9.1.2.

The \$PASSWORD card bears the user's password.

The next cards in the deck constitute the batch job or print job.

The end-of-file card is the last card in the deck. It consists of the punches 12-11-0-1-6-7-8-9 in column 1. This card is not copied to the file created by the CRP.

Example 3-5 shows a user batch job that is to be submitted by the Card Reader Processor.

#### Example 3-5: A Batch Job to Be Submitted by the Card Reader Processor

```
$SUBMIT/QUEUE:BAT2/PRINT:LP1/USER:KENT KRYPTON
$PASSWORD CLARK
$JOB XRAY [20,20]
$RUN PLANET/EST=yes
$DATA
J. Olson
L. Lane
P. White
$EOD
$EOJ
end-of-file card
```

#### 3.9.1.1 The \$SUBMIT Card

The format of the \$SUBMIT card is similar to that of the DCL command SUBMIT.

##### Format

```
$SUBMIT[/qualifier[s]]/USER:userid [jobname]
```

##### Command Qualifiers

```
/AFTER:(dd-mmm-yy hh:mm)
/[NO]HOLD
/NAME:jobname
/[NO]SEQUENCE
```

`/PRIORITY:n`  
`/QUEUE:queuname`  
`/[NO]RESTART`

#### **Batch Log File Qualifiers**

`/[NO]LOG_FILE`  
`/[NO]PRINT`

These qualifiers are the same as those entered with the DCL command SUBMIT at a terminal. For descriptions, see Section 3.7. However, the following exceptions exist:

**`/USER:userid`**

The `/USER:userid` qualifier must be specified. The `userid` argument may be either your last name or your UIC in the form `[g,m]`.

**`/NAME:jobname`**

The batch job name may be specified with either the `/NAME` qualifier or the optional `jobname` parameter at the end of the command line. You may not specify both. If you do not specify a job name, the CRP assigns your job the default job name INPBATCH (unless your system manager has set up a different default job name). The Queue Manager assigns your UIC to the batch or print job.

**`/[NO]SEQUENCE`**

If `/SEQUENCE` is specified, columns 73 to 80 of each card are interpreted as a sequence field. Sequence fields are a means by which the CRP can check the cards to ensure that they are in the correct order and have not been accidentally rearranged. Each sequence field may be assigned a number that ascends in order from one card to the next, though not necessarily in any fixed amount. MYJOB335, MYJOB336, MYJOB447 is a valid sequence, for example. A sequence field may consist of any eight alphanumeric characters. The control cards, \$SUBMIT, \$PRINT, \$PASSWORD, and end-of-file, cannot include sequence fields.

The default is `/[NO]SEQUENCE`, and any information in columns 73 to 80 is included in the batch or print file. Therefore, if your cards include sequence fields, you must use the `/SEQUENCE` qualifier, so that the contents of the sequence fields will be deleted before the cards are copied to the file.

### **3.9.1.2 The \$PRINT Card**

The format of the \$PRINT card is similar to that of the DCL command PRINT.

#### **Format**

`$PRINT[/qualifier[s]]/USER:userid [jobname]`

#### **Command Qualifiers**

`/AFTER:(dd-mmm-yy hh:mm)`  
`/COPIES:n`  
`/DEVICE:queuname`  
`/[NO]FLAG_PAGE`  
`/FORMS:n`  
`/[NO]HOLD`

/[NO]JOB\_PAGE  
/LENGTH:n  
/LOWERCASE  
/NAME:jobname  
/PAGE\_COUNT:n  
/PRIORITY:n  
/QUEUE:queue  
/[NO]RESTART  
/UPPERCASE  
/USER:userid

The /NAME:jobname and /USER:userid qualifiers used on the \$PRINT card are the same as those used on the \$SUBMIT card. For descriptions of these two qualifiers, see Section 3.9.1.1. All of the other qualifiers are the same as those used with the DCL command PRINT. For descriptions, see Section 2.2.

### 3.9.1.3 The \$PASSWORD Card

The \$PASSWORD card has strict syntax. The word PASSWORD may not be abbreviated, and exactly one space must appear before the password field.

#### Format

\$PASSWORD password

### 3.9.2 Operating the Card Reader

When the Card Reader Processor (CRP) is initialized, a default batch queue and a default print queue are selected for \$SUBMIT and \$PRINT, respectively. These may be different from the queues BATCH and PRINT, or they may be the same. These defaults can be overridden by the /QUEUE qualifier.

Each spooled card reader displays messages on a terminal about the state of the card reader and the status of jobs read. The terminal may be the console terminal (CO:) or some other terminal.

To use the card reader, place the punched cards in the hopper of the card reader and press the button marked RESET. If there are already cards in the hopper being read, there is no need to wait until the hopper is empty; simply press the button marked STOP, place your cards on top of those in the hopper, and press RESET. It is wise to place an end-of-file card at the beginning of your deck as well as at the end. This ensures that your deck will not be read as part of the preceding job even if the end-of-file card has been omitted from the previous deck.

If the hopper runs out of cards before the Card Reader Processor has encountered an end-of-file card, the CRP will wait until more cards are put in the hopper and the RESET button is pressed again. This allows the CRP to read a deck of cards that is too big to fit in the hopper at one time.

The following are common errors that the CRP may detect while reading a deck of cards:

- Syntax error in a \$SUBMIT, \$PRINT, or \$PASSWORD card
- Missing \$PASSWORD card



- Invalid user account information
- Card out of sequence when the /SEQUENCE qualifier is specified

If one of these errors occurs, the CRP sends an error message to the terminal and continues reading cards until an end-of-file card is encountered. In this situation, however, the cards are not copied to a disk file; the CRP is simply searching for the end of the card deck. This is referred to as flushing. Once a job has been flushed, it is as if the job had not been read at all. If an end-of-file card is encountered before the hopper empties, the CRP starts processing the next group of cards as a new job.

If the CRP detects that a card is out of sequence, you can replace the \$SUBMIT card with a \$PRINT card to print the deck, and then use the listing to check the sequence numbers.

You may want to consult the *RSX-11M-PLUS and Micro/RSX I/O Drivers Reference Manual* for more detailed information on the operation of the card reader. Especially useful is the table that lists the American Standard Code for Information Interchange (ASCII) characters and their representation in two different card codes, DEC029 and DEC026. Some characters may not appear on the keyboard of your keypunch. For these, you will need to look up the character in the table and use multipunches (more than one keystroke in one column) to represent the desired ASCII character.



## Chapter 4

---

### At-a-Glance Guide

This chapter consists of summaries of command information from the previous chapters. These summaries do not include full information on the commands, but just enough to remind you of how the commands work.

The commands covered in this chapter are listed below:

#### Batch-Specific Commands

PRINT (PRI)

SUBMIT (SUB)

SHOW QUEUE (QUE /LI)

SHOW PROCESSOR (QUE /LI:DEV)

SET QUEUE (QUE /MOD)

HOLD (QUE /HO) and RELEASE (QUE /REL)

DELETE (QUE /DEL)

## 4.1 PRINT (PRI) Command

PRINT queues files for printing on a line printer. It may also be used to queue jobs for other output devices.

### Format

```
DCL> PRINT/commandqualifier[s] file[s]/filequalifier[s]
MCR> PRI [queuename:][jobname][jobswitch][=]file[s][/fileswitch[es]]
```

#### DCL Command Qualifiers

```
/[NO]ADJACENT
/JOBCOUNT:n
/QUEUE:queuename
/LOWERCASE
/UPPERCASE
/[NO]HOLD
/PAGE_COUNT:n
/NAME:jobname
/PRIORITY:n
/FORMS:n
/LENGTH:n
/[NO]RESTART
/[NO]FLAG_PAGE
/AFTER:(dd-mmm-yy hh:mm)
/DEVICE:ddnn:
/NOJOBPAGE
```

#### DCL File Qualifiers

```
/[NO]DELETE
/COPIES:n
/[NO]TRANSFER
```

#### MCR Job Switches

```
/[NO]AD
/CO:n
queuename:
/LOW
/NOLOW
/[NO]HO
/PA:n
jobname
/PRIO:n
/FO:n
/LE:n
/[NO]RES
/[NO]FL
/AF:hh:mm:dd-mmm-yy
queuename:
/NOJO
```

#### MCR File Switches

```
/[NO]DEL
/CO:n
/[NO]TR
```

### Notes

1. DCL file qualifiers override settings of command qualifiers.
2. The default file type for PRINT is LST.
3. If you want your job restarted, you must use the /RESTART (/RES) qualifier.
4. For /AFTER: (/AF:), omit the calendar field to default to today's date.

## 4.2 Batch-Specific Command Summary

The following sections summarize the batch-specific commands and their functions.

## 4.2.1 Opening and Closing Batch Jobs

### JOB

JOB marks the beginning of the batch job and must appear first in the file.

### Format

`$JOB [/TIME:(hh:mm)] [loglabel] [[uic]]`

No password is required or accepted.

The `loglabel` field of JOB affects only the batch log.

The `/TIME:(hh:mm)` qualifier limits the job to hours (hh) and minutes (mm) of CPU time.

### EOJ

EOJ marks the logical end of a batch job.

### Format

`$EOJ`

## 4.2.2 Data in Batch Jobs

### DATA

DATA marks the beginning of a data block included in the batch job. The data block includes any required user input, other than DCL, MCR, or user-written CLI commands, for the batch job.

### Format

`$DATA [qualifier[s]]`

You do not need DATA unless you need its qualifiers. (Any line without a dollar sign (\$) in the first position will be interpreted as data.)

### Command Qualifiers

`/DOLLARS[:"string"]`

`/NOCOPY`

The argument string for the `/DOLLARS` qualifier allows the creation of a batch command file within a batch job. `/DOLLARS` without an argument string is equivalent to `/DOLLARS="$EOD"`.

The `/NOCOPY` qualifier prevents reprinting of the data blocks in the log file. The default is `/COPY`.

### EOD

EOD marks the end of a data block. If you use `/DOLLARS` with DATA, the data block will be closed by default upon encountering an EOD. If you used an argument string with `/DOLLARS`, EOD is negated.

### 4.2.3 Sequence Control Commands

These commands respond to the exit-with-status codes from tasks. Status can be SUCCESS, WARNING, ERROR, or SEVEREERROR. You cannot respond to SUCCESS.

#### STOP

STOP stops the batch job. It can be used alone or with ON or IF.

#### Format

\$STOP

\$ON statuscode THEN STOP

\$IF statuscode THEN STOP

STOP without a qualifier is a batch-specific command.

STOP followed by any command line element will be passed to the current CLI.

#### CONTINUE

CONTINUE is a no-operation. It can be used alone or with ON or IF.

#### Format

\$CONTINUE

\$ON statuscode THEN CONTINUE

\$IF statuscode THEN CONTINUE

#### GOTO

GOTO directs the batch processor to skip directly to a line with a specified label and continue processing from there. It can be used alone or with ON or IF.

#### Format

\$GOTO label

The label can be any six alphanumeric characters. There is no colon on the GOTO line.

#### Format

\$label:[command]

The label must be terminated by a colon (:).

GOTO may reference forwards or backwards.

#### ON

ON checks throughout the job for the return of a status code. If that code or a more severe code is returned, ON specifies an action.

### Format

\$ON statuscode THEN { STOP  
CONTINUE }  
GOTO label }

ON WARNING THEN STOP is the default setting. If you do not override it, any WARNING will stop the batch job.

You cannot check for a return of SUCCESS.

### SET

SET with special batch parameters is used to override ON (or the default) or to reinstate it.

### Format

\$SET NO ON or \$SET NOON

\$SET ON

SET NO ON has no effect on any IF your job may include.

### IF

IF checks for a status code following execution of a given command in the batch job.

### Format

\$IF statuscode THEN { STOP  
CONTINUE }  
GOTO label }

The IF command will override a previous ON command if the status code returned matches the severity specified in the IF command.

You cannot check for a return of SUCCESS.

## 4.3 SUBMIT (SUB) Command

### Format

DCL> SUBMIT [/qualifier] filespec[/filequalifier/logfilequalifier]

MCR> SUB [queueName:]jobname/jobswitch/logfileswitch]=filespec[/fileswitch]

#### DCL Command Qualifiers

/AFTER:(mm/dd/yy hh:mm)  
(dd-mmm-yy hh:mm)

/[NO]HOLD

/NAME:jobname

/[NO]RESTART

/PRIORITY:n

/QUEUE:queueName

#### MCR Job Switches

/AF:hh:mm:dd-mmm-yy

/[NO]HO

jobname

/[NO]RE

/PRIO:n

queueName:

#### DCL File Qualifiers

/[NO]DELETE

/[NO]TRANSFER

#### MCR File Switches

/[NO]DEL

/[NO]TR

DCL Batch Logfile Qualifiers	MCR Batch Logfile Switches
/NAME:jobname	jobname
/[NO]LOGFILE	/[NO]LO
/[NO]PRINT	/[NO]PRIN
/PRINT:printqueue	/PRIN:printqueue

### Notes

1. The default file type for SUBMIT is CMD.
2. If you want your job restarted, you must use the /RESTART (/RES) qualifier. Note that the /RESTART qualifier and the /DELETE (/DEL) file qualifier cannot be used together, as the system deletes the file before it can be restarted.
3. For /AFTER: (/AF:), omit the calendar field to default to today's date and time.

## 4.4 SHOW QUEUE (QUE /LI) Command

SHOW QUEUE (QUE /LI) displays information about print jobs and batch jobs in queues.

### Format

```
DCL> SHOW QUEUE [queuename]/qualifier[s]
MCR> QUE [queuename:][[uic]][jobname]/switch[es]
```

DCL Qualifiers	MCR Switches
/FULL	/FU
/FILES	/LI
/BRIEF	/BR
/DEVICE	/LI:P
/ENTRY:nnn	/EN:nnn
/FORMS[:n]	/FO[:n]
/NAME:jobname	jobname
/OWNER_UIC:uic	[uic]
/PRINT	/LI:P
/BATCH	/LI:B

### Note

There are no prompts from this command.

## 4.5 SHOW PROCESSOR (QUE /LI:DEV) Command

SHOW PROCESSOR (QUE /LI:DEV) displays information about the initialized characteristics of batch processors, printers, and other devices under control of the Queue Manager.



## Format

```
DCL> SHOW PROCESSOR/qualifier
MCR> QUE [processorname:]/switch
```

DCL Qualifiers	MCR Switches
processorname[:]	/LI:DEV
/BATCH	/LI:DEV:B
/PRINT or /DEVICE	/LI:DEV:P
/INPUT or /CARD_READER	/LI:DEV:I

## 4.6 SET QUEUE (QUE /MOD) Command

SET QUEUE (QUE /MOD) modifies attributes given to either print jobs, batch jobs, or files that compose jobs in queues. Such jobs and files have been entered in queues by either the PRINT command or SUBMIT command.

### Job Format

```
DCL> SET QUEUE queuename:jobname/qualifier[/qualifier[s]]
DCL> SET QUEUE /ENTRY:nnn/qualifier[/qualifier[s]]
MCR> QUE queuename:jobname/MOD/switch[/switch[es]]
MCR> QUE /EN:nnn/MOD/switch[/switch[es]]
```

DCL Qualifiers	MCR Switches
/AFTER:(dd-mmm-yy hh:mm)	/AF:hh:mm:dd-mmm-yy
/JOBCOUNT:n	/CO:n
/FORMS:n	/FO:n
/LENGTH:n	/LE:N
/LOWERCASE	/LO
/PAGE_COUNT:n	/PA:n
/PRIORITY:n	/PRIO:n
/[NO]RESTART	/[NO]RE
/UPPERCASE	/NOLO

### File Format

```
DCL> SET QUEUE/ENTRY:nnn/FILE_POSITION:n/qualifier[/qualifier[s]]
DCL> SET QUEUE queuename jobname/FILE_POSITION:n/qualifier[/qualifier[s]]
MCR> QUE /EN:nnn/MOD/Fl:n/switch[/switch[es]]
MCR> QUE queuename:jobname/MOD/Fl:n/switch[/switch[es]]
```

DCL Qualifiers	MCR Switches
/COPIES:n	/CO:n
/[NO]DELETE	/[NO]DEL

## Notes

1. When you issue the PRINT or SUBMIT command, you specify “attributes” of the QMG job. SET QUEUE changes the matching attribute. You cannot change the attributes of an active job.
2. You can change some attributes of files that make up a QMG print job by specifying the FILE\_POSITION:n (FI:n) qualifier in the SET QUEUE command line.

## 4.7 HOLD (QUE /HO) and RELEASE (QUE /REL) Commands

You can specify that a job be held when you issue your PRINT or SUBMIT command. You can also hold jobs with the HOLD command and release such jobs with the RELEASE command.

### 4.7.1 HOLD (QUE /HO)

HOLD (QUE /HO) blocks a job in its queue until it is explicitly released.

#### Format

```
DCL> HOLD/JOB queueName jobName
DCL> HOLD/ENTRY:nnn
MCR> QUE queueName:jobName/HO
MCR> QUE /EN:nnn/HO
```

### 4.7.2 RELEASE (QUE /REL)

RELEASE (QUE /REL) unblocks a job that has been held in queue.

#### Format

```
DCL> RELEASE/JOB queueName jobName
DCL> RELEASE/ENTRY:nnn
MCR> QUE queueName:jobName/REL
MCR> QUE /EN:nnn/REL
```

## 4.8 DELETE (QUE /DEL) Command

DELETE deletes queues or QMG jobs by name or by the job’s unique entry number.

#### Format

```
DCL> DELETE/JOB queueName jobName[FILE_POSITION:n]
DCL> DELETE/ENTRY:nnn[/FILE_POSITION:n]
MCR> QUE queueName:jobName/FILE_POSITION:n/DEL
MCR> QUE /EN:nnn/FILE_POSITION:n/DEL
```

## Notes

1. Use these commands to delete a job in a queue by specifying either the job's entry number or job name. You can also delete a single file in a job by specifying the `/FILE_POSITION:n (/FI:n)` qualifier.
2. Nonprivileged users can delete only their own jobs; privileged users can delete any job.
3. There can be more than one job in a queue with the same name. The `DELETE` command will delete the first job of a given name in the queue. You can distinguish between jobs with the same name by using the unique entry number instead.



# Appendix A

---

## Error Messages

### A.1 Batch Processing Error Messages

Commands in batch jobs can send error messages that appear in your batch log. Batch processors (BPR) also send error messages. Some of these messages result from system problems and are returned to the operator's console. Others reflect difficulties in processing your batch job and are returned to your batch log.

#### A.1.1 Error Messages in Batch Logs

**BPR—Batch file already open**

*Explanation:* QMG or BPR attempted to open the specified file when it was already open.

*User Action:* System problem. See your operator or system manager.

**BPR—Batch file close failure**

*Explanation:* The batch processor failed to close the specified file.

*User Action:* The batch job should not be affected. Check the UFD to see if the file is locked. If so, issue the DCL command UNLOCK or the MCR command PIP/UN.

**BPR—Batch file deletion failure**

*Explanation:* The /DELETE qualifier to the SUBMIT command failed to execute.

*User Action:* Check the directory. The file may not be in the directory, or the batch job lacked delete access to the file. Delete the file with the DCL command DELETE.

**BPR—CPU time limit exceeded, user job terminated**

*Explanation:* The batch job ran longer than the /TIME qualifier to the JOB command permitted.

*User Action:* Retry the batch job and increase the amount of CPU time using the /TIME qualifier.

**BPR—I/O error**

*Explanation:* The batch processor failed to read from or write to the virtual terminal, or the batch processor was unable to open the batch file.

*User Action:* Retry the batch job. This may indicate a system error. See your operator or system manager.

**BPR—Virtual terminal I/O was aborted**

*Explanation:* The task sending I/O to the virtual terminal was aborted.

*User Action:* None.

**BPR—Label undefined**

*Explanation:* The batch job specified a label that was not in the job.

*User Action:* Check the job to see that the label is present in proper form. The label must begin with a dollar sign (\$) and end with a colon (:).

**BPR—Log file directory not found - aborted batch job**

*Explanation:* The SUBMIT command included the /NOPRINT qualifier, but the batch job cannot find the directory for the batch log.

*User Action:* See your system manager.

**BPR—Logon privilege violation**

*Explanation:* The \$JOB command gave an incorrect UIC.

*User Action:* Check the command for proper syntax and retry the job. For nonprivileged users, the UIC in \$JOB must match the login UIC in effect when the SUBMIT command was entered.

**BPR—Output request from incorrect virtual terminal**

*Explanation:* System error.

*User Action:* See your operator or system manager.

**BPR—Spawn failure**

*Explanation:* System error.

*User Action:* See your operator or system manager.

**BPR—Specified maximum CPU time too large**

*Explanation:* The largest amount of CPU time that may be specified is 65535 minutes (1092 hours, 15 minutes).

*User Action:* Specify less CPU time.

**BPR—Syntax error**

*Explanation:* A command line in the batch job does not start with a dollar sign (\$), or special batch commands (Table 3-1) have improper syntax.

*User Action:* Check the batch file for proper syntax and retry the job.

**BPR—Syntax error—\$JOB does not appear first**

*Explanation:* \$JOB logs in the user batch job on the virtual terminal and must appear first in the batch file.

*User Action:* Edit the batch file so that \$JOB appears first.

**BPR—Virtual terminal output too long for buffer**

*Explanation:* A stream of characters from the virtual terminal was too long to fit in a storage buffer.

*User Action:* Make sure that the task outputs return and line-feed characters when performing terminal I/O.

## A.1.2 Error Messages to the Operator's Console

**BPR—Batch file input error**

*Explanation:* The batch processor could not read from the batch file.

*User Action:* Check the batch file status. The file may not be in the directory.

**BPR—Batch job jobname still in progress**

*Explanation:* System error. QMG attempted to start the batch job while another job was in progress.

*User Action:* Inform the system manager. System managers may wish to consult the *RSX-11M-PLUS and Micro/RSX System Management Guide* for information on running the Queue Manager.

**BPR—Error during send to QMG**

*Explanation:* System error.

*User Action:* Inform the system manager. System managers may wish to consult the *RSX-11M-PLUS and Micro/RSX System Management Guide* for information on running the Queue Manager.

**BPR—Illegal error - severity code n**

*Explanation:* System error.

*User Action:* Inform the system manager. A Software Performance Report (SPR) should be submitted including the console output and any other relevant information. If possible, reproduce the error.

**BPR—Incorrect Emit Status Block (ESB) address returned by spawned task**

*Explanation:* System error.

*User Action:* Inform the system manager.

**BPR—Log file close failure**

*Explanation:* BPR failed to close the log file.

*User Action:* Check the log file status. The device may not be available or the file may not be in the directory.

**BPR—Log file open error**

*Explanation:* BPR failed to open the log file.

*User Action:* Check the log file destination. The device may be write-locked.

**BPR—Log file output error**

*Explanation:* BPR failed to write to the log file.

*User Action:* Check the log file status. The device may not be available or the file may not be in the directory.

**BPR—Output request from incorrect virtual terminal**

*Explanation:* System error.

*User Action:* Inform the system manager.

**BPR—Unable to create virtual terminal**

*Explanation:* The system does not support virtual terminals.

*User Action:* A new system generation is required.

### **A.1.3 Error Messages from the Card Reader Processor**

The user action suggested for these errors is directed to both privileged and nonprivileged users. In most cases where the suggestion to see your operator or system manager is included, the condition can only be corrected by a privileged user.

**CRn—Reader spooled**

*Explanation:* The card reader has been initialized and is ready to read cards.

*User Action:* None.

**CRn—Reader unspooled**

*Explanation:* The card reader processor has been removed. Jobs can no longer be read.

*User Action:* None.



**CRn—Job (jobname) was entered in queue (queuename)**

*Explanation:* A job has been successfully read and entered in a queue.

*User Action:* None.

**CRn—Illegal \$SUBMIT or \$PRINT card - job flushing  
Invalid card was " "**

*Explanation:* A syntax error was encountered.

*User Action:* Repunch the control card and resubmit your card deck.

**CRn—\$PASSWORD card illegal or missing - job (jobname) flushing**

*Explanation:* A syntax error was encountered.

*User Action:* Repunch the control card and resubmit your card deck.

**CRn—Invalid account was given - job (jobname) flushing**

*Explanation:* The account information taken from the /USER:userid qualifier and the \$PASSWORD card do not match any entries in the account file.

*User Action:* Inspect your control cards and see if your password or userid was mispunched.

**CRn—Sequence number check - job (jobname) flushing  
Invalid card was " "**

*Explanation:* The CRP read a card whose sequence number in columns 72 to 80 was not larger than the sequence number of the previous card.

*User Action:* Replace the \$SUBMIT card with a \$PRINT card to get a listing of your card deck. Determine where your cards got out of sequence.

**CRn—Job (jobname) flushing due to operator request**

*Explanation:* An operator issued a command that aborted your job while it was being read.

*User Action:* See your operator or system manager.

**CRn—Queue (queuename) does not exist - job (jobname) flushing**

*Explanation:* The queue specified for your job (either by default or explicitly with the /QUEUE:queuename qualifier) does not exist.

*User Action:* Repunch the \$SUBMIT or \$PRINT card to specify an existing queue.

**CRn—Queue (queuename) marked for delete - job (jobname) flushing**

*Explanation:* The queue specified for your job is marked for deletion and no new jobs may be entered into it.

*User Action:* See your operator or system manager.

**CRn—No queue file space available - job (jobname) flushing**

*Explanation:* The queue file QUEUE.SYS is full and no more jobs can be entered in any queues.

*User Action:* Wait until the queue file has emptied or been reinitialized. Resubmit your card deck.

**CRn—Card reader not ready**

*Explanation:* Usually, this message means that the hopper was emptied while a job was being read. This message may also be given if a pick check, read check, or stack check occurred.

*User Action:* Check the card reader. Put more cards in the hopper and press the RESET button to start reading again.

**CRn—Illegal error message number = nn - parameter address = nnnnnn**

*Explanation:* System error.

*User Action:* See your operator or system manager.

**CRn—Account file open error**

*Explanation:* System error. The account file could not be opened.

*User Action:* See your operator or system manager.

**CRn—Account file close error**

*Explanation:* System error. The account file could not be closed.

*User Action:* See your operator or system manager.

**CRn—Account record locked**

*Explanation:* System error. A record in the account file could not be read.

*User Action:* See your operator or system manager.

**CRn—Couldn't find physical LB**

*Explanation:* System error.

*User Action:* See your operator or system manager.

**CRn—Account file read error nn**

*Explanation:* System error. An error occurred in reading the account file.

*User Action:* See your operator or system manager.

**CRn—Card reader read error nn**

*Explanation:* An error occurred in reading a card.

*User Action:* See your operator or system manager.

**CRn—Output file write error nn**

*Explanation:* System error. An error occurred in copying the contents of a card to the card image file.

*User Action:* See your operator or system manager.

**CRn—Output file open error nn**

*Explanation:* The card image file could not be opened.

*User Action:* See your operator or system manager.

**CRn—Output file close error nn**

*Explanation:* The card image file could not be closed.

*User Action:* See your operator or system manager.

**CRn—Error code nn returned by QMG - function code of packet sent = nnn**

*Explanation:* System error.

*User Action:* See your operator or system manager.

**CRn—Send directive failed - DSW=nn**

*Explanation:* System error.

*User Action:* See your operator or system manager.

**CRn—Exiting due to fatal error**

*Explanation:* An unrecoverable error condition occurred, causing CRP to exit immediately.

*User Action:* See your operator or system manager.

## A.2 Output Despooler Error Messages

All of these messages indicate a FATAL exit status for the job. These error messages from the output despooler task appear on the operator's console and in the listing on the output device. When these errors occur, the despooler task prints the job flag pages first and then the error messages on the first page of the listing.

The messages are preceded by the name of the output despooler (desp, in the following list). This name is derived from the name of the output device owned by the despooler, such as, LP0: and TT2:.

I/O error code numbers are explained in the *RSX-11M-PLUS and Micro/RSX I/O Operations Reference Manual*.

**desp—Attempt to space nn pages beyond eof on- filespec**

*Explanation:* A privileged user's START/PROCESSOR command attempted to start beyond the end of the specified file.

*User Action:* Correct the command and retry.

**desp—I/O Error nn on file- filespec**

*Explanation:* The task encountered an I/O error nn, which is the I/O error code number. These codes are explained in the *RSX-11M-PLUS and Micro/RSX I/O Operations Reference Manual*.

*User Action:* Look up the error and take corrective action.

**desp—Job limit of nn pages exceeded on file- filespec**

*Explanation:* The PRINT command included the /PAGES:nn qualifier. The job exceeded this limit.

*User Action:* Shorten the file or issue the PRINT command with a higher limit.

**desp—Open error nn on file- filespec**

*Explanation:* The task could not open the file named in the error message for output. The value nn is the I/O error code number. These codes are explained in the *RSX-11M-PLUS and Micro/RSX I/O Operations Reference Manual*.

*User Action:* Check the directory listing for the state of the file. Look up the error.

**desp—Print error nn on file- filespec**

*Explanation:* The task could not print the file named in the error message. The value nn is the I/O error code number. These codes are explained in the *RSX-11M-PLUS and Micro/RSX I/O Operations Reference Manual*.

*User Action:* Check the directory listing for the state of the file. Look up the error.

## A.3 DCL and MCR Error Messages from Queue Manager Commands

These error messages appear on your terminal as you issue DCL or MCR commands.

The error message is preceded by the first three letters of the erroneous command. For instance, the message

```
-- JOB DOES NOT EXIST
```

might result from several commands, including HOLD or DELETE. Thus, you might see either of the following error messages:

```
HOL -- JOB DOES NOT EXIST
```

```
DEL -- JOB DOES NOT EXIST
```

If your CLI is MCR, the error message is also preceded by the first three letters of the command.

```
QUE -- JOB DOES NOT EXIST
```

The messages are listed in alphabetical order.

The user action suggested for these errors is directed to both privileged and nonprivileged users. In most cases where the suggestion to see your operator or system manager is included, the condition can only be corrected by a privileged user.

**com—Bad queue name**

*Explanation:* The queue name included illegal or too many characters.

*User Action:* Try new queue name. Limit: six Radix-50 characters.

**com—QMG not active**

*Explanation:* The Queue Manager task is not active. It may be stopped, aborted, not running, not installed, or the like.

*User Action:* Wait and retry the command. If QMG does not return, see your operator or system manager.

**com—Processor does not exist**

*Explanation:* A command named a device that is not set spooled.

*User Action:* Retry the command with the proper device name.

**com—Directive failure**

*Explanation:* System error.

*User Action:* See your operator or system manager.

**com—Illegal argument value**

*Explanation:* A command specified an illegal argument for a qualifier (switch).

*User Action:* Check for proper value range and retype the command.

**com—Bad processor name**

*Explanation:* The processor name includes invalid or too many characters.

*User Action:* Retry the command with the proper processor name. Names are limited to six Radix-50 characters. Print processors must be named after the device they control, such as, LP2: controlled by LP2. Batch processors must have names in the form BAPn. Applications processors can have any legal name.

**com—I/O Error**

*Explanation:* System error.

*User Action:* Try again. See your operator or system manager if the error persists.

**com—Job does not exist**

*Explanation:* A command named a job that does not exist in the form the command recognized or under the name the command recognized. The job name includes UIC.

*User Action:* Issue SHOW QUEUE (QUE /LI). Determine the name and status of the job. Check for the proper command syntax and retype the command.

**com—Operation inconsistent with job state**

*Explanation:* A command attempted action inconsistent with the job state.

*User Action:* Issue SHOW QUEUE (QUE /LI). Determine the status of the job. Check for the proper syntax and retype the command.

**com—Queue does not exist**

*Explanation:* A command named a queue that does not exist in the current system.

*User Action:* Issue SHOW QUEUE (QUE /LI). Retry the command with the name of an existing queue. See your operator or system manager.

**com—Queue file full**

*Explanation:* The queue file [1,7]QUEUE.SYS on SP for RSX-11M-PLUS is full.

*User Action:* Wait and try again. If the error message persists, see your operator or system manager.

**com—Queue marked for delete**

*Explanation:* New jobs cannot be added to a queue marked for deletion.

*User Action:* Send the job to another queue.

**com—Redundant operation**

*Explanation:* A command requested an operation that has been done.

*User Action:* None.

# Glossary

---

This glossary contains brief definitions of terms used in the *RSX-11M-PLUS Batch and Queue Operations Manual*.

## **ACTIVE JOB**

An active job is a job currently being processed.

## **APPLICATIONS PROCESSOR**

An output print processor written by the user for a special purpose or specific job. See your system manager for information on any applications processors in your installation. Applications processors are treated like print processors in most cases.

## **BATCH COMMAND**

A batch command is a CLI command in a batch job. All batch commands must be preceded by a dollar sign (\$) in the first position of the command line. Compare with batch-specific command.

## **BATCH-SPECIFIC COMMAND**

Batch-specific commands are commands that are independent of the CLI and designed exclusively for batch processing. These commands can be used for logging the batch job on and off the system, for including data, and for sequence control in the batch job when errors are returned. They must be preceded by a dollar sign (\$).

## **BATCH JOB**

One or more files containing commands and data that are submitted to the Queue Manager for execution by the batch processor.

## **BATCH LOG FILE**

Each QMG batch job produces a log file that contains a record of the virtual terminal's activity in executing a batch job. You can save a permanent copy of the log file. You can also include comments in the log file. You can print the log file.

Regardless of the number of batch jobs submitted with a single SUBMIT (SUB) command, only one log file will be created. You can differentiate between jobs within the log by using the log label field of the JOB command, which logs in each batch job.

**BATCH PROCESSING**

A means of automatically passing commands and data for processing. Batch-specific commands, CLI commands, and data are placed in a file and submitted to the system for execution.

You do not have to be present when your batch job is run, nor do batch jobs require a physical terminal to run. Batch processing uses virtual terminals and thus does not tie up interactive terminals.

**BATCH PROCESSOR**

A batch processor is a task that passes commands and data to CLIs and other tasks using a software called a virtual terminal.

**BATCH QUEUE**

BATCH queue refers to the default queue for batch jobs. A batch (lowercase) queue refers to any queue that is used to maintain a waiting line of batch jobs. (A queue may be assigned to more than one processor.)

**CLI**

An acronym standing for Command Line Interpreter. CLIs act as interpreters between people and computer operating systems. See MCR and DCL.

**COMMAND**

A command is an instruction or request for the operating system to perform a particular action. See CLI.

**COMMAND LINE INTERPRETER**

See CLI.

**COMMAND QUALIFIER**

A command qualifier (DCL) or job switch (MCR) is a command element preceded by a slash (/) that alters the action of a command. See FILE QUALIFIER.

**DATA**

Any information that is required for the execution of a batch job. This is a special definition used in batch processing on RSX-11M-PLUS systems. The term *data* includes not only data in the conventional sense of information to be processed by a program but also responses to prompts from CLI commands or from indirect command files. In fact, anything—except commands—that would have to be typed by the user in an interactive session is called data.

**DCL**

DIGITAL Command Language. An RSX-11M-PLUS-supported CLI designed for ease of use. DCL uses English language words, and prompts you for necessary parameters. Compare with MCR.



**DEFAULT**

A default is a value or condition that is assumed by the operating system and performed automatically.

For example: The default queue for the PRINT command is PRINT. There are other queues available on the system, but, to place your job in one of these other queues, you must specify the queue in the PRINT command. If you do not specify one of these other queues, your print job will default to the queue called PRINT.

**DESPOOLER**

A print processor may also be called a despooler.

**DEVICE**

Any peripheral hardware or software connected to the processor and capable of receiving, storing, or transmitting data. Line printers and terminals are examples of record-oriented devices. Magnetic tapes and disk devices are examples of mass-storage devices. Terminal line interfaces and interprocessor links are examples of communication devices.

**DEVICE-SPECIFIC QUEUE**

Some print queues have names derived from the names of the output devices on the system, such as LP0, named after LP0. These queues are needed for certain system tasks that send print jobs to line printers directly. These queues will usually send their output to the printer they are named after, but they need not do so.

**DIGITAL COMMAND LANGUAGE**

See DCL.

**DISK**

A disk is a high-speed, random access mass-storage device. RSX-11M-PLUS systems use disks as their major form of mass-storage device.

**ERROR**

An exit status code found on batch logs. ERROR indicates that some action you have requested has failed.

**ERROR MESSAGE**

Error messages are sent by the system when some action you have requested fails.

Each error message identifies the particular part of the system that detected the error.

**EXIT STATUS CODE**

All system tasks on the RSX-11M-PLUS systems, DCL and MCR commands, and utilities return a status code when they have completed their operations. The codes are:

SUCCESS	Results should be as expected
WARNING	Task succeeded but irregularities are possible
ERROR	Results unlikely to be as expected
SEVERE ERROR	One or more fatal errors, or ABORT encountered

As a general rule, you can expect that any error message you receive on your terminal will reflect an exit status less than success. Any error message tagged `_*FATAL_*` will probably represent a status return of ERROR or SEVEREERROR, and any error message tagged `_*DIAG_*` will probably represent a status return of WARNING. These rules are general, however, and may not apply to the behavior of every system task.

You can use the ON and IF commands inside your batch job to program responses to exit status codes that may be returned by tasks in your job.

#### **FIELD**

A field is a portion of a command or a command element.

#### **FILE QUALIFIER**

A file qualifier (DCL) or switch (MCR) is a command element preceded by a slash (/) that alters or modifies the action of a command as it affects that particular file. Depending upon the position of the file qualifier (switch), all or only certain files may be affected. See COMMAND QUALIFIER.

#### **FORMS**

All standard-length forms use form feeds, which are set on the hardware. All nonstandard forms use simulated form feeds, which are set through the software by replacing the form-feed character with the appropriate number of line feeds.

Whether a form length is standard or not depends on whether it can be set using hardware facilities on the line printer. If it can be set on the line printer, then it is a standard form length.

Your system manager can tell you what /FORM:n option to specify in your PRINT command for any particular form. In this way, your print jobs will be queued for the proper printer and held until the proper forms are placed on the printer.

#### **FORM FEED**

A form feed is a character that directs the line printer to move to the top of the next continuous-feed form page. The line printer moves past the perforation in the paper to the top of a new form.

#### **GENERAL QUEUE**

Queues with names not derived from hardware devices. Each hardware output device has a queue named after it. See DEVICE-SPECIFIC QUEUE. All other queues are general, including the default PRINT queue, other print queues, and all batch queues.

#### **HARDCOPY TERMINAL**

Terminals that print output on paper are called hardcopy terminals.

#### **INDIRECT COMMAND FILES**

A means of automatically passing commands for processing on an RSX-11M-PLUS operating system. Indirect command files remain active at the terminal from which they are invoked. Indirect command file directives permit symbol substitution, testing system functions, and such programming techniques as loops, counters, and subroutines.

**INSTALLATION**

The installation is the full computer system (hardware and software) at your location. This includes the operating system, the programming languages, and all applications tasks, as well as the computer and all its hardware devices.

**INTERACTIVE TERMINAL**

A standard hardware terminal with a keyboard from which you can type responses to prompts. An interactive terminal has two functions: (1) sending input to the operating system from a keyboard and displaying it and (2) receiving output from the operating system and displaying it. Compare with VIRTUAL TERMINAL.

**LINE PRINTER**

An output device that prints files a line at a time. Even if your system does not have an actual hardware device of the line-printer type, it will have some device designated as the system output device.

**LOG**

A log is a record of activity. In this manual, the term refers to a file produced by a batch processor. See BATCH LOG FILE.

**LOGICAL PAGE**

The line printer moves to the top of a new PHYSICAL PAGE each time a form-feed character (CTRL/L) is encountered. If you use the /LENGTH qualifier on the PRINT command, you define a LOGICAL PAGE length. If the line printer does not encounter a form-feed character within the length of the logical page, it will automatically supply one at the end of the logical page, moving the printer to the top of the next physical page.

**LOGIN**

The LOGIN command identifies you to the operating system and informs the system that you have certain privileges and are using a particular terminal.

**LOGOUT**

The LOGOUT command informs the operating system that you have finished using a particular terminal.

**MCR**

Monitor Console Routine. An RSX-11M-PLUS-supported CLI. Most MCR commands use mnemonics in strict syntax. See DCL.

**NONPRIVILEGED USER**

Nonprivileged users have access to most of the system's facilities but cannot alter the system.

**OPERATING SYSTEM**

An operating system is a set of tasks that collectively automate the management of computer resources to provide efficient computer operation.

An operating system is used for user communication with the computer, for program development, and for scheduling the use of the central processing unit and its peripherals for efficiency.

**OPERATOR**

The person who is in direct charge of the computer.

**OPERATOR'S CONSOLE**

A hardcopy terminal, usually located near the computer, used by the operator. This terminal has the logical name CO: and records certain system information.

**PERIPHERAL DEVICES**

Any unit, distinct from the CPU and memory, that can provide the system with input or accept output from it, is called a peripheral device, or a peripheral.

**PHYSICAL PAGE**

A physical page is the part of a printer form between one perforation and the next perforation. Standard line-printer paper has a physical page length of 66 lines. Each time the line printer encounters a form-feed character (CTRL/L), the line printer moves the paper to the top of the next physical page. Compare LOGICAL PAGE.

**PRINT COMMAND**

The PRINT command queues files for printing on a line printer. The PRINT command may also be used to enter jobs in queues for other output devices, such as plotters or punches. The PRINT command is to print jobs as the SUBMIT command is to batch jobs.

**PRINT PROCESSOR**

Each spooled output device has a print processor, a task that attaches the device and directs output to it. The Queue Manager handles the orderly printing of files through the print processor.

**PRINT QUEUE**

PRINT queue refers to the default queue for print jobs. A print (lowercase) queue refers to any queue that is used to contain a list of print jobs.

**PRIORITY**

Priority is a rank assigned to a task to determine its precedence in obtaining system resources when the task is run. The default priority for a task is 50. QMG jobs are also assigned a priority. The default priority for a job is 50.

**PRIVILEGED USER**

Privileged users can alter the operating system through the use of privileged commands.

**PRINT JOB**

A print job consists of one or more files to be printed at the same time. Jobs in a print job go to the same line printer in the order they were listed and without interruption. The print job has a name derived from the name of the first file in the job or from a name given in the PRINT command.

**QUALIFIER**

See COMMAND QUALIFIER and FILE QUALIFIER.

**QUEUE**

A queue is a waiting line of jobs that are scheduled or processed according to system or user priorities.

**QUEUE FILE**

A disk file used by the Queue Manager for storage of control information and entries for all batch and print processors, all jobs and files to be processed, and all queues. On RSX-11M-PLUS systems the file is SP0:[1,7]QUEUE.SYS. Since this information is retained on disk, your job will not be lost should the system stop operating.

**QUEUE MANAGER**

The Queue Manager (QMG) is a system program that provides for the orderly processing of print and batch jobs.

**SEQUENCE CONTROL COMMANDS**

IF and ON are special batch commands that control the batch job when errors are encountered. They provide a means of programming error response into the batch job when used with the commands STOP, CONTINUE, GOTO, and SET [NO]ON.

See the individual command descriptions for details.

**SEVEREERROR**

Exit status code. This is usually returned when one or more fatal errors are encountered or a task is aborted. When used in a sequence control command, this term should be typed as a single word: SEVEREERROR.

**SPOOL**

This is an acronym for Shared Peripheral Operations On Line. It refers to the process of sending output to a peripheral device, particularly a line printer, in an orderly fashion.

Many users and tasks may be directing output to the line printer at the same time. Rather than print three lines of one file and then two of another, as might happen if the output was passed directly to the line printer, the output is sent to a mass-storage device—usually a disk—to be saved in separate files.

The process of saving files on disk in this fashion is called spooling. The orderly transfer of these files from the disk to the output device is called despooling.

**SPOOLED DEVICE**

A spooled device is a line printer or other output device under control of the Queue Manager. Spooled devices are initialized with certain attributes by the system manager.

**SUBMIT COMMAND**

The SUBMIT command sends user batch jobs to the Queue Manager for execution by the batch processor.

**SUCCESS**

An exit status code not recognized by the batch processor and not shown in the batch log file.

**SYNTAX**

Syntax is the form that a command must follow. Misspelled words are the most common syntax errors, but misplaced delimiters are also common.

**SYSTEM MANAGER**

The person responsible for setting policies for using the RSX-11M-PLUS operating system.

**TRANSPARENT SPOOLING**

Transparent spooling is the substitution of the device name of a spooled output device for a file specification in any system task or user-written task that creates an output file.

**UFD**

All files on mass-storage devices are cataloged in User File Directories, or UFDs. The UFD is a file listing all the files included in the Directory. The UFD is a two-number code in the form [g,m] that is included in every file specification, either explicitly or by default, and that the file system uses to locate the file.

In most cases, the UFD will be the same as the User Identification Code, or UIC, under which you logged in. See UIC.

**UIC**

User Identification Code. Each RSX-11M-PLUS user has a two-number identification code enclosed in brackets that is used (with password) for logging in. The number is in the form [g,m,], g being the user's group number and m the user's member number.

**VIRTUAL TERMINAL**

A software terminal used by a batch processor to issue commands within the batch job. Compare with INTERACTIVE TERMINAL.

**WARNING**

Exit status code. The task succeeded, but irregularities are possible.

# Index

---

## A

---

/ADJACENT qualifier  
  PRINT command, 2-3  
/AD switch  
  PRI command, 2-3  
/AF:::TOMORROW switch, 2-3, 3-15  
/AF job switch  
  SUB command, 3-15  
/AF switch  
  PRI command, 2-3  
  QUE /MOD command, 2-18  
/AFTER:TOMORROW qualifier, 2-3, 3-15  
/AFTER qualifier  
  PRINT command, 2-3  
  SET QUEUE command, 2-19  
  SUBMIT command, 3-15  
ALLOCATE command  
  /TYPE qualifier, 3-13

## B

---

Batch file  
  example, 3-2  
Batch job  
  contents of, 3-1  
  defined, 3-1  
  queuing, 3-15  
Batch log file, 3-19  
  example, 3-21, 3-22  
  line identifiers  
    DATA, 3-20  
    SKIP, 3-20  
    TERM, 3-20  
/BATCH qualifier  
  SHOW QUEUE command, 2-14  
Batch-specific command  
  list, 3-4  
/BRIEF qualifier

/BRIEF qualifier (cont'd.)  
  SHOW QUEUE command, 2-14  
/BR switch  
  QUE /LI command, 2-14

## C

---

Card deck  
  batch jobs, 3-23  
  \$EOJ card, 3-24  
  example, 3-24  
  \$JOB card, 3-24  
  \$PASSWORD card, 3-24  
  \$PRINT card, 3-24  
  sequence field, 3-24  
  \$SUBMIT card, 3-24  
Card Reader Processor  
  See CRP  
Comment character in batch jobs  
  format, 3-12  
CONTINUE command  
  See also ON and IF commands  
  format, 3-9  
/COPIES file qualifier  
  PRINT command, 2-3  
  SET QUEUE command, 2-19  
/CO switch  
  file  
    PRI command, 2-3  
    QUE /MOD command, 2-19  
  job  
    PRI command, 2-3  
    QUE /MOD command, 2-19  
CRP, 3-23  
  error recovery, 3-27  
  operation, 3-26

## D

---

DATA command, 3-2  
  example, 3-6, 3-7  
  format, 3-6  
  qualifiers, 3-6  
DELETE command  
  deleting a job from a queue, 1-9  
  examples, 2-11  
  format, 2-10  
/DELETE file qualifier  
  PRINT command, 2-3  
  SET QUEUE command, 2-19  
  SUBMIT command, 3-15  
/DEL file switch  
  PRI command, 2-3  
  QUE /MOD command, 2-19  
  SUB command, 3-15  
Device  
  allocating from batch job, 3-13  
  deallocating, 3-13  
  private, 3-13  
/DEVICE qualifier  
  PRINT command, 2-3  
  SHOW QUEUE command, 2-14  
/DOLLARS qualifier  
  DATA command, 3-6

## E

---

/EN:nnn switch  
  QUE /DEL command, 2-10  
  QUE /LI command, 2-14  
/ENTRY:nnn qualifier  
  DELETE command, 2-10  
  SHOW QUEUE command, 2-14  
EOD command, 3-2  
  format, 3-8  
EOJ command  
  format, 3-6  
/EST=yes switch, 3-8  
Exit status code, 3-3

## F

---

/FI:n file switch  
  QUE /MOD command, 2-19  
/FI:n switch  
  QUE /DEL command, 2-10  
/FILE\_POSITION file qualifier  
  SET QUEUE command, 2-19  
/FILE\_POSITION qualifier  
  DELETE command, 2-10  
File specification

File specification (cont'd.)  
  multiple, 3-15  
/FILES qualifier  
  SHOW QUEUE command, 2-14  
File type  
  default, 3-15  
/FLAG\_PAGE qualifier  
  PRINT command, 2-3  
/FL switch  
  PRI command, 2-3  
/FO job switch  
  QUE /MOD command, 2-19  
/FORMS qualifier  
  PRINT command, 2-3  
  QUE /LI command, 2-14  
  SET QUEUE command, 2-19  
/FO switch  
  PRI command, 2-3  
  QUE /LI command, 2-14  
/FULL qualifier  
  SHOW QUEUE command, 2-14  
/FU switch  
  QUE /LI command, 2-14

## G

---

GOTO command, 3-9  
  See also ON and IF commands  
  example, 3-9  
  format  
    label, 3-9

## H

---

/HO job switch  
  SUB command, 3-15  
HOLD command  
  format  
    entry number, 2-22  
    job name, 2-22  
/HOLD qualifier  
  PRINT command, 2-3  
  SUBMIT command, 3-15  
/HO switch  
  PRI command, 2-3

## I

---

IF command  
  See also STOP, CONTINUE, and GOTO  
  commands  
  example, 3-12  
  format, 3-11



Indirect command file  
  example, 3-13  
  format, 3-13  
  in batch jobs, 3-13

## J

---

### Job

  holding, 2-1  
  modifying, 2-1  
  releasing, 2-1  
  removing from queue, 2-10  
/JOB\_PAGE qualifier  
  PRINT command, 2-3  
JOB command, 3-4  
  example, 3-5  
  format, 3-4  
  qualifiers, 3-4  
/JOBCOUNT:n qualifier  
  SET QUEUE command, 2-19  
/JOBCOUNT qualifier  
  PRINT command, 2-3  
Jobname job switch  
  SUB command, 3-15  
Jobname log file switch  
  SUB command, 3-15  
Jobname parameter  
  DELETE command, 2-10  
Jobname switch  
  PRI command, 2-3  
  QUE /LI command, 2-14  
/JO switch  
  PRI command, 2-3

## L

---

Label, 3-3  
/LE job switch  
  QUE /MOD command, 2-19  
/LENGTH job qualifier  
  SET QUEUE command, 2-19  
/LENGTH qualifier  
  PRINT command, 2-3  
/LE switch  
  PRI command, 2-3  
/LI:B switch  
  QUE /LI command, 2-14  
/LI:P switch  
  QUE /LI command, 2-14  
/LI switch  
  QUE /LI command, 2-14  
Log file, 3-1  
/LOGFILE qualifier

/LOGFILE qualifier (cont'd.)  
  SUBMIT command, 3-15  
Login messages  
  suppressing, 3-4  
Loglabel, 3-4  
/LO log file switch  
  SUB command, 3-15  
/LOWERCASE qualifier  
  PRINT command, 2-3  
  SET QUEUE command, 2-19  
/LOW job switch  
  QUE /MOD command, 2-19  
/LOW switch  
  PRI command, 2-3

## M

---

MOU command, 3-14  
MOUNT command, 3-14

## N

---

/NAME:jobname qualifier  
  PRINT command, 2-3  
/NAME log file qualifier  
  SUBMIT command, 3-15  
/NAME qualifier  
  \$PRINT card, 3-26  
  SHOW QUEUE command, 2-14  
  \$SUBMIT card, 3-24  
  SUBMIT command, 3-15  
/NOCOPY qualifier  
  DATA command, 3-6  
/NOLOW job switch  
  QUE /MOD command, 2-19  
/NOLOW switch  
  PRI command, 2-3

## O

---

ON command  
  See also STOP, CONTINUE, and GOTO  
  commands  
  default setting, 3-10  
  example, 3-10  
  format, 3-10  
/OWNER\_UIC qualifier  
  SHOW QUEUE command, 2-14

## P

---

/PAGE\_COUNT qualifier  
  PRINT command, 2-3  
  SET QUEUE command, 2-19

- /PA job switch
  - QUE /MOD command, 2-19
- \$PASSWORD card
  - format, 3-26
- /PA switch
  - PRI command, 2-3
- PRI command
  - file switches, 2-3
  - format, 2-3
  - job switches, 2-3
- \$PRINT card
  - command qualifiers, 3-25
  - format, 3-25
- PRINT command, 2-2
  - command qualifiers, 2-3
  - examples, 2-7
  - file qualifiers, 2-3
  - format, 2-3
- Print job
  - queued by system tasks, 2-12
    - DCL COPY command, 2-12
    - PIP, 2-12
    - PIP line printer, 2-12
  - queued by user tasks
    - opening file on LPn:, 2-14
    - .PRINT, 2-13
    - PRINT command, 2-13
  - submission notification, 2-2
- /PRINT log file qualifier
  - SUBMIT command, 3-15
- Print processor, 1-3
- /PRINT qualifier
  - SHOW QUEUE command, 2-14
- /PRIO job switch
  - QUE /MOD command, 2-19
  - SUB command, 3-15
- /PRIORITY qualifier
  - PRINT command, 2-3
  - SET QUEUE command, 2-19
  - SUBMIT command, 3-15
- /PRIO switch
  - PRI command, 2-3

## Q

---

- QUE /DEL, 1-9
- QUE /DEL command
  - format, 2-10
- QUE /HO command
  - format
    - entry number, 2-22
    - job name, 2-22

- QUE /LI:DEV command
  - format, 2-17
- QUE /LI command, 2-14
  - display, 2-14 to 2-17
  - format, 2-14
  - switches, 2-14
- QUE /MOD command
  - example, 2-19
  - file format
    - switches, 2-19
  - job format
    - switches, 2-19
- QUE /REL command
  - format
    - entry number, 2-22
    - job name, 2-22
- Queue
  - displaying information, 2-1
  - job
    - holding, 2-1
    - releasing, 2-1
    - status
      - altering, 2-1
- Queuename: switch
  - PRI command, 2-3
- Queuename job switch
  - SUB command, 3-15
- Queuename parameter
  - DELETE command, 2-10
- /QUEUE qualifier
  - PRINT command, 2-3
  - SUBMIT command, 3-15

## R

---

- RELEASE command
  - format
    - entry number, 2-22
    - job name, 2-22
- /RES job switch
  - QUE /MOD command, 2-19
  - SUB command, 3-15
- /RES switch
  - PRI command, 2-3
- /RESTART qualifier
  - PRINT command, 2-3
  - SET QUEUE command, 2-19
  - SUBMIT command, 3-15

## S

---

- /SEQUENCE qualifier
  - \$SUBMIT command, 3-24

SET command  
  format, 3-11  
SET QUEUE command  
  example, 2-19  
  file format  
    qualifiers, 2-19  
  job format  
    qualifiers, 2-19  
SHOW PROCESSOR command  
  format, 2-17  
SHOW QUEUE (QUE /LI)  
  display, 1-8  
  format, 1-7  
SHOW QUEUE command, 2-14  
  display, 2-14 to 2-17  
  format, 2-14  
  qualifiers, 2-14  
Spooled device, 2-1  
Spooling  
  defined, 2-1  
/STATUS:TASK qualifier, 3-8  
Status code, 3-11  
  ERROR, 3-10  
  SEVERERROR, 3-10  
  SUCCESS, 3-10  
  WARNING, 3-10  
STOP command  
  See also ON and IF commands  
  format, 3-8  
SUB command  
  example, 3-18  
  file switches, 3-15  
  log file switches, 3-15  
  MCR job switches, 3-15  
\$SUBMIT card  
  command qualifiers, 3-24  
  See SUBMIT command  
  format, 3-24  
SUBMIT command  
  DCL command qualifiers, 3-15  
  example, 3-18  
  file qualifiers, 3-15  
  format, 3-1, 3-15  
  log file qualifiers, 3-15  
  /NOPRINT qualifier, 3-1

## T

---

Task output  
  spooling, 2-12  
/TIME qualifier  
  JOB command, 3-4

/TRANSFER file qualifier  
  PRINT command, 2-9  
  SUBMIT command, 3-15  
Transparent spooling, 2-1  
  EDT WRITE command, 2-2  
/TR file switch  
  PRI command, 2-9  
  SUB command, 3-15

## U

---

UIC, 3-1  
  slash format, 3-4  
[uic] switch  
  QUE /LI command, 2-14  
/UPPERCASE qualifier  
  PRINT command, 2-3  
  SET QUEUE command, 2-18  
User Identification Code  
  See UIC  
/USER qualifier  
  \$PRINT card, 3-25  
  \$SUBMIT card, 3-24

## V

---

Volume  
  mounting from batch job, 3-13

## W

---

/WAIT qualifier  
  MOUNT command, 3-14



---

**READER'S  
COMMENTS**

Your comments and suggestions are welcome and will help us in our continuous effort to improve the quality and usefulness of our documentation and software.

Remember, the system includes information that you read on your terminal: help files, error messages, prompts, and so on. Please let us know if you have comments about this information, too.

Did you find this manual understandable, usable, and well organized? Please make suggestions for improvement.

---

---

---

---

---

---

Did you find errors in this manual? If so, specify the error and the page number.

---

---

---

---

---

---

What kind of user are you?     Programmer     Nonprogrammer

Years of experience as a computer programmer/user: \_\_\_\_\_

Name \_\_\_\_\_ Date \_\_\_\_\_

Organization \_\_\_\_\_

Street \_\_\_\_\_

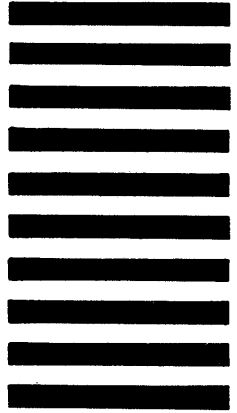
City \_\_\_\_\_ State \_\_\_\_\_ Zip Code \_\_\_\_\_  
or Country

--- Do Not Tear - Fold Here and Tape ---

**digital**<sup>TM</sup>



No Postage  
Necessary  
if Mailed  
in the  
United States



**BUSINESS REPLY MAIL**  
FIRST CLASS PERMIT NO. 33 MAYNARD MASS.

POSTAGE WILL BE PAID BY ADDRESSEE

DIGITAL EQUIPMENT CORPORATION  
Corporate User Publications—Spit Brook  
ZK01-3/J35  
110 SPIT BROOK ROAD  
NASHUA, NH 03062-9987



--- Do Not Tear - Fold Here ---

