

GC33-5403-1

**Systems**

**DOS/VS  
POWER/VS  
Installation and Operations**

**IBM**

Second Edition (September 1974)

This publication obsoletes the *DOS/VS Planning Guide to POWER/VS*, GC33-5403-0

Requests for copies of IBM publications should be made to your local IBM representative or to the IBM branch office serving your locality.

© Copyright International Business Machines Corporation 1974

## THIS MANUAL . . .

contains the information necessary to install and use POWER/VS. A good knowledge of DOS/VS is required.

The following parts make up this manual:

- **Introduction.** An overview of the purpose and features of POWER/VS.
- **POWER/VS Concepts and Facilities.** Includes the concepts and defines the terminology of POWER/VS. It also lists the hardware devices supported.  
*The terms defined in this section, especially those under the heading "Organization of the POWER/VS Data and Queue Files", should be studied carefully. They must be understood to use this manual effectively.*
- **POWER/VS Generation.** Describes how to tailor POWER/VS using the generation macros. Also described is the supplied, pre-assembled version of POWER/VS that will make generation unnecessary for many users.
- **POWER/VS Start-Up.** Contains the POWER/VS start-up procedures and describes the AUTOSTART facility.
- **Job Entry Control Language (JECL).** Describes the JECL statements, which are used to specify job requirements to POWER/VS.
- **POWER/VS Operator Command Language (POCL).** Describes the commands used by the central operator to communicate with POWER/VS.
- **Remote Operator Command Language (ROCL).** Describes the commands used by the remote terminal operator to communicate with POWER/VS.
- **POWER/VS Operator Messages.** Contains all the POWER/VS messages that might be printed at SYSLOG, SYSLST, or a terminal.
- **Appendixes.** Contains tables referenced in the body of the manual or information of interest to only certain users of POWER/VS. Refer to the table of contents.
- **Index.**

## NOTATION CONVENTIONS

- Uppercase letters, numbers, and punctuation marks must be coded exactly as shown. Operator commands may be entered in uppercase or lowercase letters.
- Lowercase letters and words represent information that must be supplied by the programmer or operator.
- Options contained in braces `{}`  represent alternatives, one of which must be chosen.
- Brackets `[]`  indicate that the element they enclose is optional. The brackets should not be coded.
- Stacked options contained within brackets `[AB]` represent alternatives, one and only one of which may be chosen.
- Default values are underlined.
- No comma is used before the first parameter specified.
- Keyword parameters consist of a keyword, followed by one or more values.
- Positional parameters must appear in a specified location, relative to other parameters. If a parameter is omitted, its absence is indicated by a comma. Trailing commas are not required.

# CONTENTS

<b>Introduction</b>	1.1
<b>POWER/VS Features</b>	1.3
Design	1.3
Job Management	1.3
I/O Management	1.4
Operator Facilities	1.4
Remote Job Entry (RJE) Facilities	1.5
<b>POWER/VS Concepts and Facilities</b>	2.1
POWER/VS Tasks	2.1
POWER/VS Direct Access Files	2.3
Organization of the POWER/VS Data and Queue Files	2.5
Job Attributes	2.9
POWER/VS Control Language	2.10
POWER/VS Operator Commands	2.10
Inter-partition Device Assignment	2.10
Dummy Printers and Punches	2.11
Supported Hardware Devices	2.11
Printer Support	2.12
3540 Support	2.12
<b>POWER/VS Generation</b>	3.1
Generation Macros	3.1
POWER Macro	3.1
PLINE Macro	3.5
PRMT Macro	3.6
Pre-assembled POWER/VS Program	3.8
POWER/VS File Labels	3.8
DOS/VS Supervisor Generation Requirements	3.9
Main Storage Requirements	3.10
Direct Access Storage Requirements	3.13
DOS/VS System Libraries	3.13
POWER/VS Files	3.13
<b>POWER/VS Start-up</b>	4.1
Start-up, Without AUTOSTART	4.1
Start-up, With AUTOSTART	4.3
Procedure	4.5
FORMAT Statement	4.6
Address Specification Statements	4.6
Start-up Considerations	4.7
<b>Job Entry Control Language (JECL)</b>	5.1
Functions of JECL Statements	5.1
Formats of JECL Statements	5.2
CTL Statement	5.3
JOB Statement	5.3
EOJ Statement	5.5
RDR Statement	5.5
LST or PRT Statement	5.7
PUN Statement	5.11
SLI Statement	5.14
SL Update Statement	5.15
DATA Statement	5.15
<b>POWER/VS Operator Command Language (POCL)</b>	6.1
Task Management Commands	6.2
PSTART Command	6.2
PSTOP Command	6.5
PGO Command	6.6
PEND Command	6.6
PCANCEL Command	6.7
PFLUSH Command	6.7
PRESTART Command	6.8

Queue Management Commands . . . . .	6.9
PDISPLAY Command . . . . .	6.9
PALTER Command . . . . .	6.11
PDELETE Command . . . . .	6.13
PRELEASE Command . . . . .	6.13
Miscellaneous Commands . . . . .	6.14
PINQUIRE Command . . . . .	6.14
PBRDCST Command . . . . .	6.15
PACCOUNT Command . . . . .	6.15
PSETUP Command . . . . .	6.16
<b>POWER/VS Remote Operator Command Language (ROCL)</b> . . . . .	<b>7.1</b>
Operational Characteristics . . . . .	7.2
Terminal Commands . . . . .	7.3
SIGNON Command . . . . .	7.3
SIGNOFF Command . . . . .	7.3
Task Management Commands . . . . .	7.4
START Command . . . . .	7.4
STOP Command . . . . .	7.4
GO Command . . . . .	7.5
FLUSH Command . . . . .	7.5
RESTART Command . . . . .	7.6
Queue Management Commands . . . . .	7.6
DISPLAY Command . . . . .	7.6
ALTER Command . . . . .	7.9
DELETE Command . . . . .	7.10
RELEASE Command . . . . .	7.11
Miscellaneous Commands . . . . .	7.12
BRDCST Command . . . . .	7.12
SETUP Command . . . . .	7.12
<b>Appendix A - Account Records</b> . . . . .	<b>8.1</b>
Read Account Record . . . . .	8.1
List Account Record . . . . .	8.2
Punch Account Record . . . . .	8.3
Execution Account Record . . . . .	8.3
Line Account Record . . . . .	8.5
<b>Appendix B - POWER/VS Cancel Codes</b> . . . . .	<b>9.1</b>
<b>Appendix C - RJE SIGNOFF Codes</b> . . . . .	<b>10.1</b>
<b>Appendix D - Status Report Format</b> . . . . .	<b>11.1</b>
File Statistics . . . . .	11.1
<b>Appendix E - Comparison of DOS/VS POWER and POWER/VS</b> . . . . .	<b>12.1</b>
General Differences . . . . .	12.1
Local Operation . . . . .	12.2
Remote Operation . . . . .	12.3
<b>Appendix F - User Accounting Routine (\$JOBACCT)</b> . . . . .	<b>13.1</b>
<b>Appendix G - RDREXIT Routine</b> . . . . .	<b>14.1</b>
<b>Appendix H - Sample Job Streams</b> . . . . .	<b>15.1</b>
<b>Appendix I - Jobs for Suggested AUTOSTART Procedure</b> . . . . .	<b>16.1</b>
<b>Appendix J - POWER/VS Generation Examples</b> . . . . .	<b>17.1</b>
<b>Appendix K - POWER/VS Operation Messages</b> . . . . .	<b>18.1</b>
<b>Index</b> . . . . .	<b>19.1</b>

## LIST OF FIGURES

Figure 1.	Processing Jobs A - E With and Without POWER/VS . . . . .	1.2
Figure 2.	POWER/VS Program Elements and Data Flow . . . . .	2.2
Figure 3.	POWER/VS Functional Tasks . . . . .	2.4
Figure 4.	Queue File and Data File Organization . . . . .	2.6
Figure 5.	Entries and Entry Segments . . . . .	2.7
Figure 6.	POWER/VS Partition Layout . . . . .	3.11
Figure 7.	Console Log for POWER/VS Start-up . . . . .	4.2
Figure 8.	Console Log for POWER/VS AUTOSTART . . . . .	4.4
Figure 9.	POWER/VS AUTOSTART Deck . . . . .	4.4
Figure 10.	Format of JECL Statements . . . . .	5.2
Figure 11.	Format of RJE Commands . . . . .	7.2

## INTRODUCTION

In all computing systems, electronic CPUs are very fast, whereas the largely mechanical unit-record input/output devices (card readers, card punches, and printers) are relatively slow. Because of this fact, the overall duration of most programs is largely dependent on the speed at which unit-record I/O can read and write data.

To reduce CPU dependency on unit-record devices, the technique known as spooling (Simultaneous Peripheral Operation On Line) was developed. *Spooling* separates the unit record input/output operations for a program from the execution of that program. Under control of a spooling program, such as POWER/VS, card input is first read and placed on intermediate disk storage. Then during execution of the problem program, data is read from and written to intermediate storage; program execution does not have to wait for unit record I/O operations. After execution, printing and punching are done while other problem programs are executing.

POWER/VS (Priority Output Writers, Execution processors, and input Readers/Virtual Storage) is a spooling program that also includes job scheduling. Besides reducing CPU dependency on the relatively slow speeds of card readers, card punches, and printers, POWER/VS executes each job according to its user-assigned priority and class.

Job processing under POWER/VS can be controlled at various stages by the operator. For example, jobs entered into the system can be held on disk and not executed until released by an operator command. Similarly, the operator can defer (hold) large-volume output queued for printing or punching to prevent the tie-up of printers and punches when the system is busy.

The time and resources that can be saved processing a job stream with POWER/VS is illustrated by Figure 1.

The upper part of the figure shows the processing of a DOS/VS job stream in a partition *without* POWER/VS spooling facilities. Job B, which prints a great deal, slows down throughput. Notice that *without* POWER/VS, the reading and printing devices are logically committed to the partition for the entire duration of the job stream.

The lower part of the figure shows the same job stream processed (in a single user-partition) *with* POWER/VS. Processing is shown divided into reading time, CPU time (execution), and printing time. The vertical broken lines are used to connect the reading, execution, and printing stages of each job. The horizontal broken lines represent time spent by the jobs in the respective queues awaiting execution or printing. Print (or punch) queue wait time can be reduced by segmenting output (see "Attributes" in "POWER/VS Concepts and Facilities"). Punching and punch queuing is omitted from the example for the sake of clarity.

Figure 1 shows that POWER/VS improves overall system performance in several ways:

- Total elapsed time is reduced.
- Resources are made available as indicated in the lower part of the figure:
  - The reader is free to read more jobs as soon as job E is entered into the input queue.
  - The partition is available before job A and after job E are executed.
  - The printer has some spare capacity that can be used to print output from other job streams.

WITHOUT POWER/VS



WITH POWER/VS

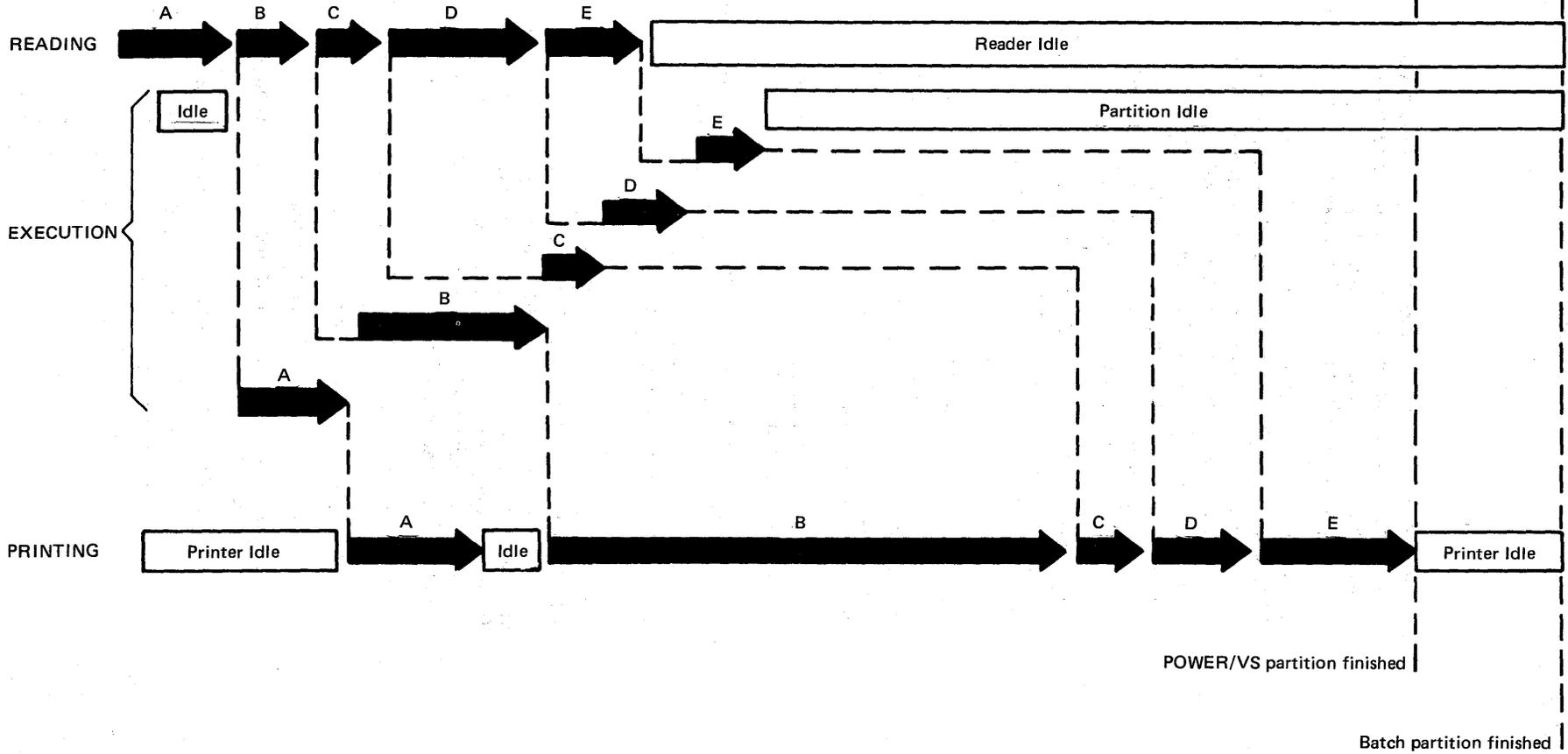


Figure 1. Processing Jobs A - E With and Without POWER/VS

## POWER/VS FEATURES

### Design

- POWER/VS runs in virtual mode, which reduces its real storage requirement.
- A large portion of POWER/VS code is re-entrant, which reduces page-out activity.
- The POWER/VS generation procedure is simple. If the IBM-supplied, pre-assembled program fits installation requirements, generation is not necessary.
- Writer-only operation is specified by partition at POWER/VS start-up or during operation. It is not necessary nor possible to generate a writer-only POWER/VS program.
- Up to four partitions can be controlled by POWER/VS. Each partition is separately initiated. The user does not specify these partitions at POWER/VS generation time.
- Automatic start-up of POWER/VS is provided through control cards.
- Device error recovery is provided. Only the affected POWER/VS task is canceled on an unrecoverable I/O error.
- POWER/VS code is modular for easier maintenance.
- Programs can be executed under POWER/VS in either real or virtual mode.
- Up to five disk volumes are supported for POWER/VS intermediate storage.
- Tape intermediate storage can be used for output.
- Up to eight printers and eight punches, whether physically present or not, can be defined for each partition.
- POWER/VS can be shut down and restarted without losing jobs queued for execution or output queued for printing or punching.
- POWER/VS job accounting records contain spooling activity information, the complete DOS/VS job accounting table for each job executed, and summary sign-on information for RJE lines.

### Job Management

- POWER/VS job queues are partition independent; regardless of the number of partitions controlled, POWER/VS maintains only one read queue, one list queue, and one punch queue, giving greater scheduling flexibility.
- Priority scheduling support is provided for up to ten priorities.
- Jobs can be assigned input and output classes so that jobs requiring the same resources for execution or output can be grouped together.
- The default job class, which is assigned to a POWER/VS job when no class is explicitly assigned, can be changed at any time.
- A job entry (execution or output) can be flushed without deleting it from the POWER/VS queues.
- Jobs that are to be executed repeatedly can be kept in the input queue after execution.
- Source library inclusion is supported, allowing DOS/VS job control and data to be called from the source statement library (system or private) by POWER/VS job entry control statements.
- Keyword parameters in POWER/VS job entry control statements reduce errors and also provide documentation of processing requirements.

## I/O Management

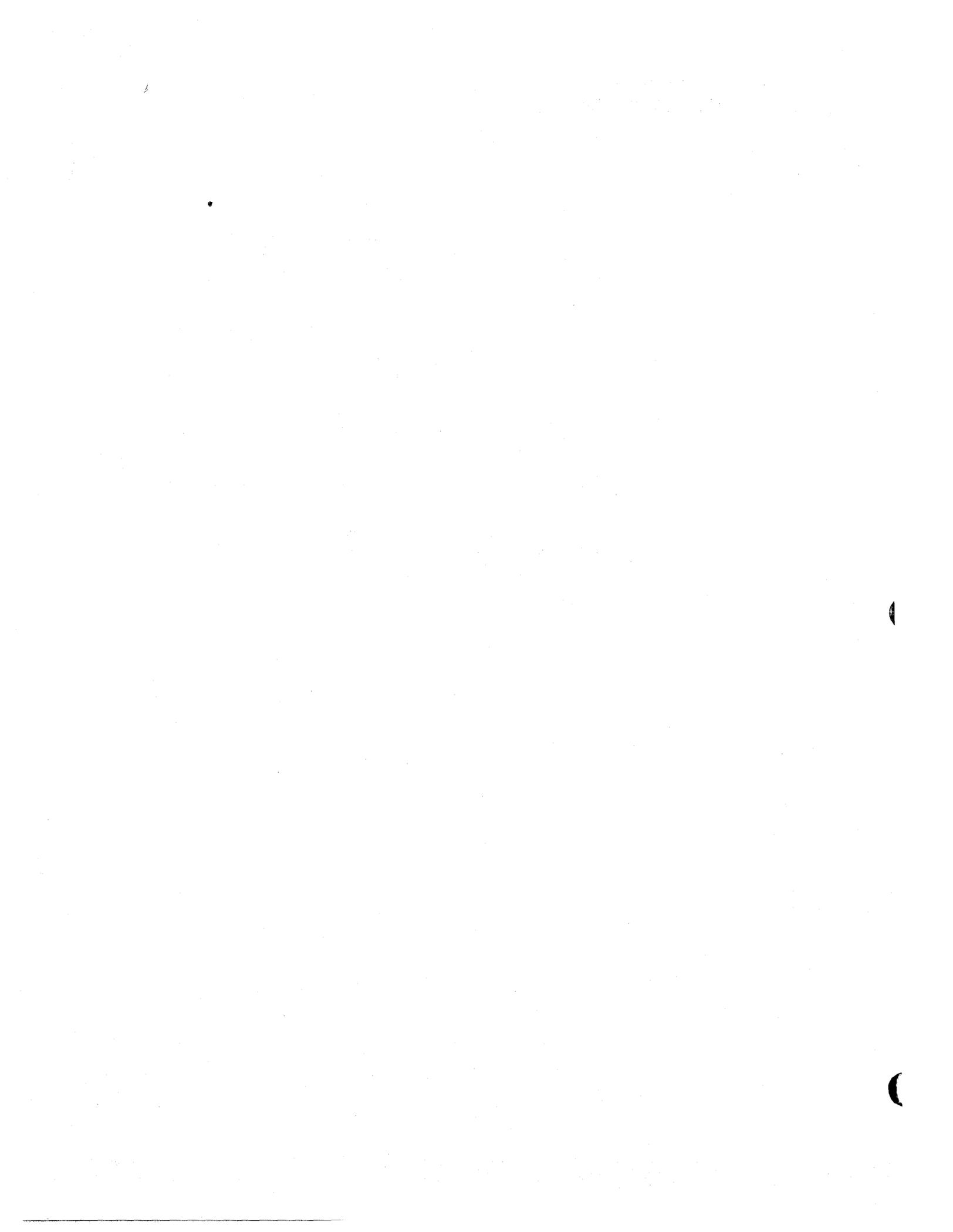
- Multiple print and punch classes are supported.
- Restart capabilities are provided for printed and punched output.
- The listing or punching of output for a job can be started before all list or punch output for that job has been spooled (output segmentation).
- The user can defer the printing or punching of output.
- Output, after it is processed, can be retained in the queue so that additional copies can be obtained.
- The punched output of a job (e.g., an assembly) can be returned directly to the input queue from the output queue.
- Output from jobs entered at a local card reader can be routed to a remote user.
- Output from jobs entered by a remote user can be routed to the central site or to any remote user.
- Special card stock can be specified and changed during punching of output.
- Special forms can be specified and changed during printing of output.
- Printer output can conform to special printer carriage control tapes.
- FCB and UCS buffer loading is supported using DOS/VS JCL and/or POWER/VS JECL.
- A limit can be placed on the number of list or punch output records that POWER/VS accepts from a job.
- A job separator facility is provided for printed and punched output for both local and remote jobs.
- Job streams placed in the reader during execution of POWER/VS are read immediately.
- Output can be printed/punched directly, without being spooled.

## Operator Facilities

- The central operator controls POWER/VS activity. He can display job status, release jobs from hold status, alter job priorities, delete jobs from queues, terminate jobs in progress, and can start and stop readers, writers, and RJE system components.
- Partitions, tasks, or jobs can be initiated at any time.
- The central operator can direct output to a local output device or to a specific remote user.
- Job disposition can be obtained or changed, and jobs can be deleted from the system by both the central operator and the remote terminal operator.
- Output can be discontinued, restarted, or canceled by either the central operator or the remote terminal operator.
- The status of any job or group of jobs can be displayed or altered.
- Printer set-up facilities are provided to allow proper form alignment.
- Messages can be broadcast to all RJE users.

## Remote Job Entry (RJE) Facilities

- The RJE commands are a subset of the central operator commands.
- System security is provided by password and user identification.
- The central operator has inquiry facilities to determine the status of any or all RJE lines.
- The idle time allowed a terminal (before it is automatically signed off) is specified by line during POWER/VS generation.
- "Hot reader" support allows a terminal to *send*, without operator intervention, as soon as the terminal is ready and the line is available.
- "Hot writer" support allows a terminal to *receive*, without operator intervention, as soon as output is queued at the central processor.
- RJE write tasks can be started by the remote operator, acting independently of the central operator.
- Remote job output can be requested only by the user submitting the job. However, the user can direct output to the central system for output processing or optionally retrieve a portion of the output, inspect it, and then direct it to the central site.
- The remote terminal user can send messages to and receive messages from the central site or another remote terminal.
- Up to 25 RJE lines can be active concurrently.
- ASCII support is provided.
- Data can be transmitted to the central site in either transparent or non-transparent mode.
- Hardware support is provided for horizontal tab, space compression/expansion, buffer expansion, additional buffer expansion, and the multiple record feature.



## POWER/VS CONCEPTS AND FACILITIES

POWER/VS is a reader/writer program that provides automatic staging of unit record input and output, class scheduling, and priority scheduling for all DOS/VS programs. User programs and DOS/VS job control statements can be processed by POWER/VS without modification.

Figure 2 shows the program elements and the data flow of POWER/VS:

- The job streams (POWER/VS control statements, DOS/VS job control statements, programs, and data cards) are read and stored in the read queue and POWER/VS data file.
- The jobs are scheduled and executed from the read queue and data file.
- Output is stored in the list and punch queues and POWER/VS data file before it is finally printed and/or punched.
- Readers, writers, and partitions are started and stopped, and job processing is controlled through operator commands entered from the console.

POWER/VS resides in one partition of a DOS/VS system and controls from one to four other partitions, each having a lower dispatching priority than the POWER/VS partition. POWER/VS is executed in virtual mode to take advantage of the virtual storage environment provided by DOS/VS. User programs execute under POWER/VS control in either virtual or real mode.

The control statements and input data required for program execution are read by POWER/VS from the appropriate unit-record reader, and written to intermediate direct access storage, where they remain until requested by job control or the user program. During program execution, each program request for unit-record input addressed to the device specified at start-up in the READER statement is intercepted by POWER/VS, which simulates a read operation by providing a data record from intermediate storage. Therefore, unless input is read directly by the user program, stacker selection of input data cards is not possible; POWER/VS ignores such requests. Similarly all unit-record list and punch output directed to device addresses specified at start-up in the PRINTERS and PUNCHES statements is intercepted by POWER/VS and written to intermediate storage (disk or tape). Such output records remain in intermediate storage until POWER/VS writes them to the appropriate unit record list or punch device.

### POWER/VS TASKS

POWER/VS performs four basic functions. They are:

- **Read.** A user job stream; composed of DOS/VS job control statements, optional POWER/VS control statements, and program input data; is read from a peripheral device, normally a card reader, and written to intermediate direct access storage.
- **Execute.** Each user job submitted is executed under the control of POWER/VS.
- **List.** The list output generated by the user program is written from intermediate direct access storage or tape to a printer.
- **Punch.** The punch output generated by the user program is written from intermediate direct access storage or tape to a card punch.

In order to perform these functions concurrently, POWER/VS is structured into a series of asynchronously executed tasks. All necessary task support is provided within POWER/VS. Multitasking support within the DOS/VS supervisor is *not* required.

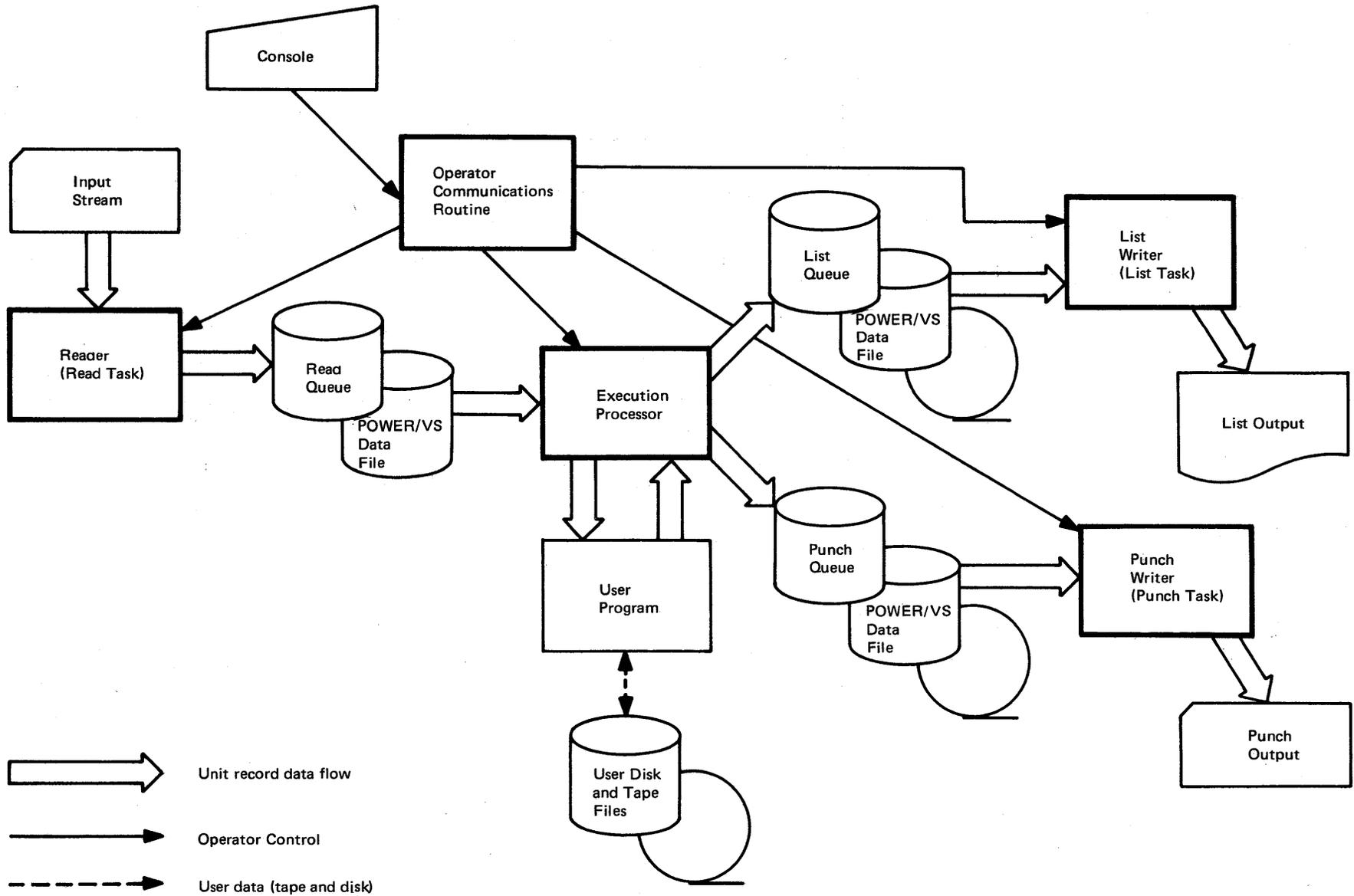


Figure 2. POWER/VS Program Elements and Data Flow

The POWER/VS tasks are:

- **Main Task.** It initiates and terminates POWER/VS execution.
- **Line Manager.** It controls all teleprocessing line activity.
- **Command Task.** It handles all commands submitted by the system operator, and initiates and terminates other POWER/VS tasks.
- **Wait Task.** It transfers the POWER/VS partition to and from the wait state.

*Note. The main task, line manager, command task, and wait task are internal "control tasks". The following are "functional tasks".*

- **Read Tasks.** They perform the read function. The operator can call for concurrent execution of as many read tasks as he has physical readers available.
- **List Tasks.** They perform the list function. The operator can call for concurrent execution of as many list tasks as he has physical printers available.
- **Punch Tasks.** They perform the punch function. The operator can call for concurrent execution of as many punch tasks as he has physical punches available.

*Note. Read, list, and punch tasks are collectively referred to as "read/write tasks."*

- **RJE Tasks.** They consist of read, list, and punch tasks. One read, one list, and one punch task per line can be active concurrently.
- **Execution Read Tasks.** They provide simulated reader support for the execute function. There is one execution read task active for each partition under POWER/VS control.
- **Execution List Tasks.** They provide simulated printer support for the execute function. An execution list task is automatically initiated when required by a user job.
- **Execution Punch Tasks.** They provide simulated punch support for the execute function. An execution punch task is automatically initiated when required by a user job.

*Note. Execution read tasks, execution list tasks, and execution punch tasks are collectively referred to as the "execution processor".*

Each read task is divided into two parts: *physical read* which performs the actual reading of the data from the card reader, and *logical read*, which saves the input data in the POWER/VS data file and creates a new entry in the queue file.

Each list task is divided into physical list and logical list. *Logical list* retrieves data from the list queue and passes it to *physical list*, which performs printer device dependent functions. Similarly, each punch task is divided into *logical punch* and *physical punch*. Figure 3 is a simplified illustration of the sequence in which functional POWER/VS tasks are executed to process a job.

## POWER/VS DIRECT ACCESS FILES

The intermediate disk storage used by POWER/VS is divided into three files: *queue file*, *data file*, and *account file*.

Data file records contain the spooled input and output data. Queue file records describe the processing requirements of each job and "point to" the job's associated data in the data file. The account file contains accounting information, and is optional.

The following defines the *logical* organization of POWER/VS direct access storage. Refer also to Figure 4.

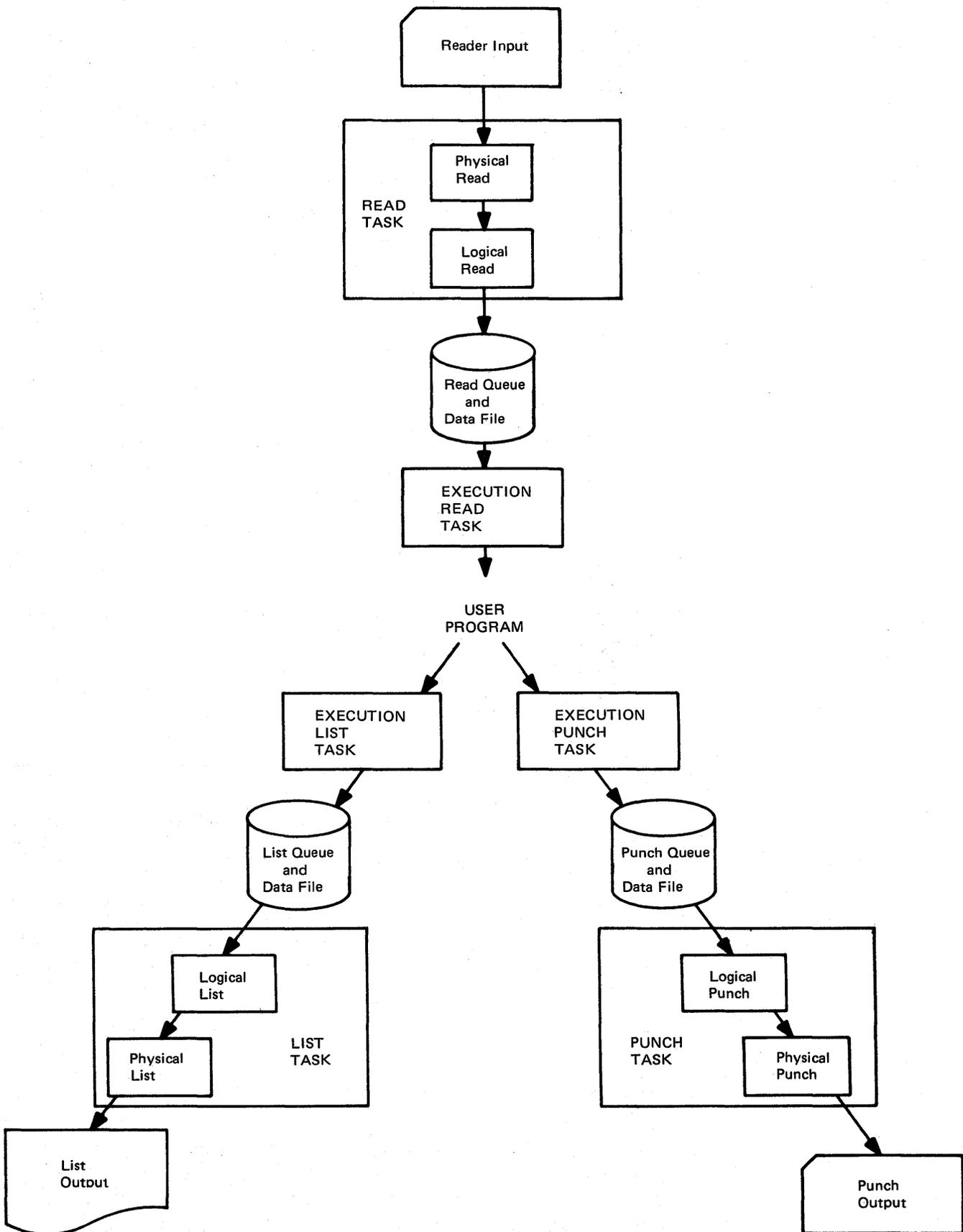


Figure 3. POWER/VS Functional Tasks

## Organization of the POWER/VS Data and Queue Files

POWER/VS breaks the input stream up into a series of discrete jobs, and assigns each a sequence number. Each job is described by one or more *read queue records* in the queue file to form what is called a *read queue set*. A read queue set together with its corresponding data in the data file form a *read entry*. File organization for list and punch output is similar to that for input; that is, POWER/VS generates *list queue sets*, *punch queue sets*, *list entries*, and *punch entries*. See Figure 4.

A single POWER/VS job may create multiple list and punch entries because of one or more of the following conditions:

- Multiple LST/PUN statements occur within the POWER/VS job.
- Multiple printers or punches are used by the DOS/VS job.
- A user program(s) issues an LFCB (load forms control buffer) macro instruction(s).
- Count-driven segmentation is used.

List and punch entries are further divided into *list entry segments* and *punch entry segments* if the segmented output feature is used. Figure 5 shows, through a job stream example, how list entries and list entry segments are created.

The three types of queue sets are logically assembled into three corresponding queues: the *read queue*, the *list queue*, and the *punch queue*. These three queues service all of the partitions running under POWER/VS control.

The read queue is also referred to as the *input queue*. The term *output queue* is used to refer collectively to the list queue and the punch queue. *Job entry* is a general term referring to what could be a read, a list, or a punch entry.

A description of all the POWER/VS files is maintained in a master record which is the first record of the queue file. In the event of a failure of the system, "warm start" information is recovered from the direct access files through the master record.

During POWER/VS execution, a copy of the master record is maintained in the queue control block in main storage. As the contents of the queue control block change, the master record is updated.

### Queue File

The master record and the queue records make up the queue file, a single extent of direct access storage. The logical unit SYS001 is assigned in the POWER/VS partition for this file. The file is assigned the standard file name IJQFILE. The user must supply // DLBL and // EXTENT statements for the POWER/VS queue file. Split cylinders are not supported. Each record is 152 bytes long.

### Data File

The problem program input and output records are contained in the POWER/VS data file. It is assigned the standard file name IJDFILE. The user must supply // DLBL and // EXTENT statements for this file. One to five separate extents can be assigned, using logical units SYS002 to SYS006 within the POWER/VS partition. Each extent must start on a cylinder boundary and must contain an integral number of cylinders; split cylinders are not supported. All data file extents must be on separate volumes but assigned to the same type of direct access storage device.

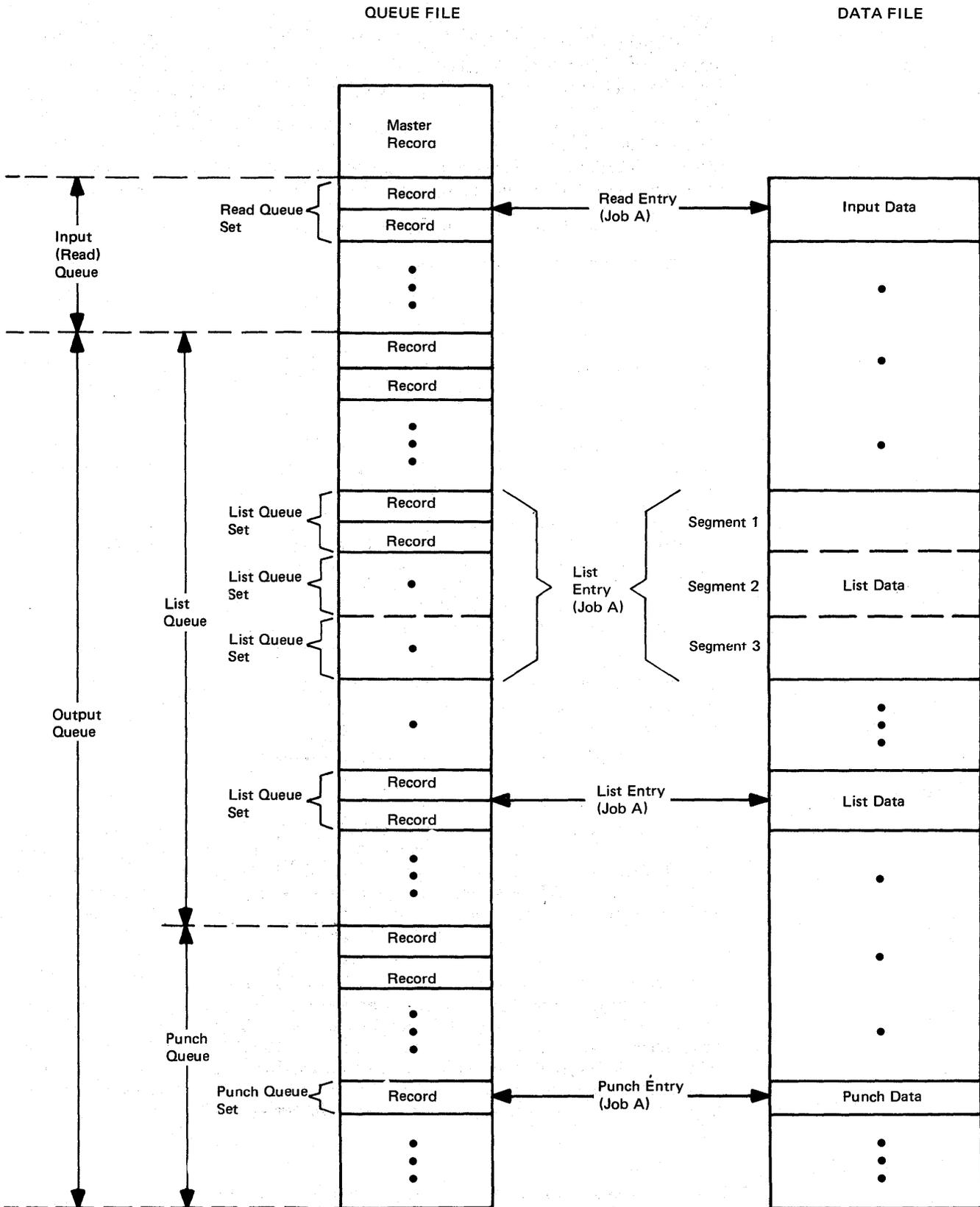


Figure 4. Queue File and Data File Organization. This example is a logical representation of the queue and data file entries generated by a POWER/VS job called Job A. One list entry shown is segmented.

The space assigned to the data file is managed by dividing it into units called *track groups*. The track group is the basic unit of space allocation used by POWER/VS. Each track group contains an integral number of tracks. The track group size can vary from a single track to an entire cylinder of direct access storage. When a cold start is performed, an appropriate track group size is chosen by the system based on the amount of disk storage provided. The user *can* specify an explicit track group size during POWER/VS generation. However, if the queue file is not large enough to describe the entire data file with this track group size, POWER/VS will determine a new track group and inform the user with a message.

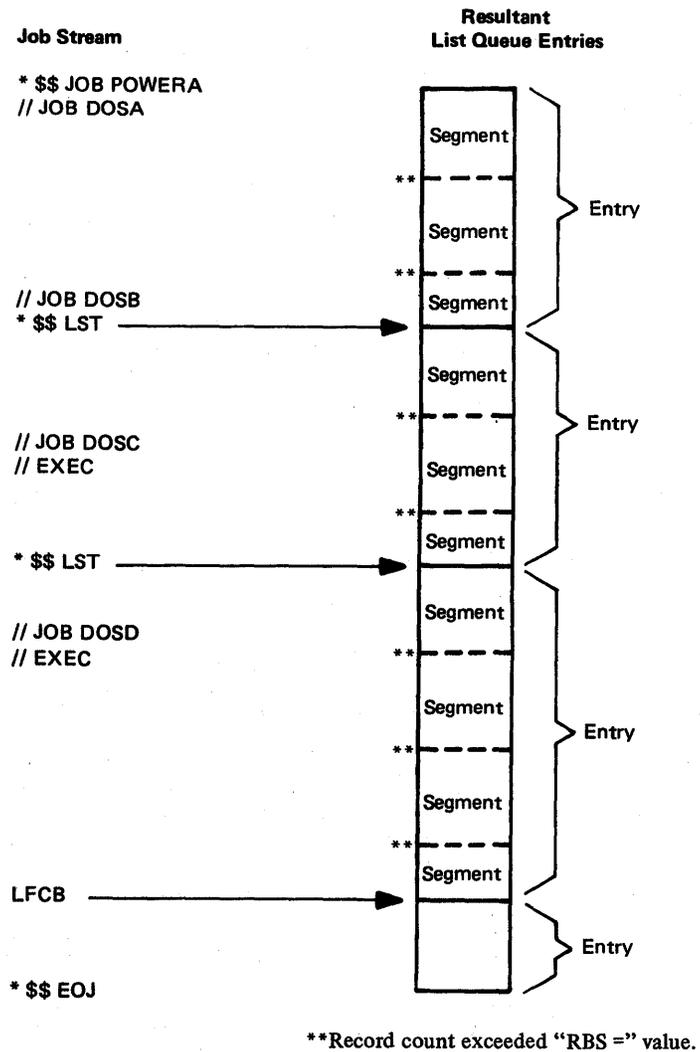


Figure 5. Entries and Entry Segments. POWER/VS creates a new list entry for each LST statement processed or LFCB macro executed. A list entry segment is created each time record count exceeds the "RBS=" value.

Each track group in the system has a corresponding queue record. Therefore, the total number of queue records must be equal to or larger than the total number of track groups. If not, part of the data file cannot be used.

The data of each job entry type (read, list, or punch) occupies only as many track groups as are required to contain it. These track groups are not necessarily contiguous.

The physical length of the records written to the data file is called the *data block size*. When a cold start is performed, an appropriate data block size is inserted by POWER/VS. The user can, however, specify an explicit value during POWER/VS generation.

## Account File

Accounting information is collected both from the POWER/VS partition and from the DOS/VS job accounting interface (JAI) and is placed in the account file together with full identifying information. The user does *not* have to provide his own job accounting routine. However, a routine can be written to include additional information in account records. See Appendix F.

To use POWER/VS accounting support:

- Specify JA=YES or JA=(n1,n2,n3,n4,n5) in the FOPT macro during DOS/VS supervisor generation.
- Specify ACCOUNT=YES in the POWER macro.
- Provide // DLBL and // EXTENT statements for the account file, using logical unit SYS000 within the POWER/VS partition. The standard file name is IJAFILE.

The POWER/VS account file can contain six types of records. Their contents are detailed in Appendix A.

- **Read account record.** Created for each entry written on the read queue.
- **List account record.** Created for each list entry processed by a list task.
- **Punch account record.** Created for each punch entry processed by a punch task.
- **Execution account record.** Created during program execution on the basis of one record per job step.
- **Line account record.** Created for each RJE user session. When the session is terminated, the record is written to the account file by the SIGNOFF processor.
- **Stop record.** A special line account record written when the teleprocessing line is stopped by the central operator.

Records are written to the account file in chronological order. After a warm start of the account file, new account records are written after the last record of the previous session. The user is responsible for sorting records and summarizing them to meet his own particular accounting requirements. The account file is a standard DOS/VS sequential disk file with variable length unblocked records, which can be processed by DOS/VS SORT/VS.

The POWER/VS account file can be processed directly (after termination of POWER/VS), or the file can be saved on another medium via the PACCOUNT command.

## JOB ATTRIBUTES

The attributes of a job entry determine how it will be processed. The major classes of attributes are: disposition, class, priority, output segmentation, output limitation, output destination.

- **Disposition.** Disposition determines how POWER/VS will route and schedule the associated input or output job entry. The possible disposition attributes are:
  - D** Process and delete. The job entry is automatically scheduled for processing by POWER/VS in accordance with its class and priority attributes. After processing, the entry is deleted and associated data space is freed.
  - H** Hold. The job entry remains in the queue; it is not executed or written to a unit record device by POWER/VS until the operator releases it using the PRELEASE command or until he changes the disposition attribute to D or K by means of the PALTER command. When the PRELEASE command is used, the job entry, after it is processed, is deleted from the queue and associated data space is freed.
  - K** Process and keep. The job entry is automatically scheduled for processing by POWER/VS in accordance with its class and priority attributes. On completion, the job entry is *not* deleted from the queue, and the disposition of the entry is changed to L.
  - L** Leave in queue. The job entry remains in the queue; it will not be processed by POWER/VS until the operator releases it using the PRELEASE command or until he changes the disposition attribute to D or K by means of the PALTER command. When the PRELEASE command is used, the job entry, after it is processed, returns to the leave state.

Three additional dispositions apply only to output:

- I** Return output to input queue. This option should be used only for jobs producing punch output in executable format; that is, including JCL.
- N** Output without spooling. Output is not intercepted but is immediately printed or punched.
- T** Spool to tape. Tape intermediate storage is used.

D,H,K,I and L are valid only when output is spooled to disk. N and T are invalid when output is to be printed at a terminal.

- **Class.** Class is a designation given to each job in a group of jobs that use a common set of system resources. These common resources might include: a partition size requirement; a high partition dispatching priority; or special printer forms, character set buffer, or card stock.

Each job has a class attribute for execution and another for output. Input class is specified by an alphabetic character, A through Z, or by a number, 0 through 4. When specified as a number, input class is partition dependent; 0 to 4 correspond with partitions BG to F4, respectively. Output class is specified by an alphabetic character, A through Z. Input and output classes are completely independent of each other.

As compared to DOS/VS POWER, which has partition dependent queues, POWER/VS has only one read queue, one list queue, and one punch queue, which service all partitions. A single read queue, makes it possible for a job to be executed in any one of four partitions, which helps balance the work load among the partitions. Also, having partition-independent queues means that the operator spends less time starting and stopping readers and writers.

When a partition is started, one to four classes are selected for execution in that partition. Classes are prioritized in the same order they are specified.

ESERVE

- **Priority.** Besides the priority determined by its class, each job is also assigned a scheduling priority within its class. Priority is specified as a digit from 0 to 9. Nine is the highest priority. If the priority is not specified, a default is assigned as defined in the PRI parameter during POWER/VS generation. Within each input or output queue, job entries are selected for processing on a first-in, first-out basis within priority, within class.
- **Output segmentation.** List or punch output from user programs can be broken into segments. Printing or punching can then begin before execution of the program is completed; that is, after the first segment has been spooled. Segmentation is done by record count, which is specified through the POWER generation macro or through the LST or PUN statement.
- **Output limitation.** A limit can be placed on the number of list or punch records (STDLIN or STDCARD parameter in the POWER macro or RBM parameter in the LST or PUN statement) that POWER/VS accepts from a specific job. When this limit is reached, (for example, 1000 lines have been printed) a warning message is given to the operator.
- **Output destination.** List and punch output can be routed to any terminal or to the central location by using the remote-id in the LST and PUN statements.

## POWER/VS CONTROL LANGUAGE

POWER/VS job entry control language (JECL) is used to delimit POWER/VS jobs and to specify special job requirements. Specifically, JECL can be used to control the attributes of job entries, to load forms control buffer and Universal Character Set images, and to insert source statement library data into the input stream. JECL supplements the DOS/VS job control language (JCL); the JCL statements required for normal DOS/VS operation are also required when POWER/VS is used.

## POWER/VS OPERATOR COMMANDS

The POWER/VS operator command language (POCL) allows the central operator to communicate with the system. A separate but very similar remote operator command language (ROCL) is provided for the remote terminal operator. The types of commands are:

- **Task management commands.** Allow the operator to initiate and terminate POWER/VS tasks.
- **Queue management commands.** Allow the operator to display and modify the contents of POWER/VS queues.
- **Miscellaneous commands.** Allow the operator to perform such operations as forms set-up and saving of the account file.
- **Terminal control commands.** Allow the remote operator to initiate and terminate POWER/VS RJE tasks and to display and modify POWER/VS queues.

## INTER-PARTITION DEVICE ASSIGNMENT

POWER/VS, as opposed to DOS/VS POWER, allows different partitions to make assignments to the same unit record device. Consequently, POWER/VS does not require that dummy (non-existent) devices be generated in the DOS/VS supervisor in order to allow each partition to "pretend" that it has its own reader, printer, and punch.

## DUMMY PRINTERS AND PUNCHES

With POWER/VS the dummy device concept still exists, but serves a different purpose. It allows each partition under its control to operate as if it had up to 8 printers and 8 punches. This means, for example, that as many as 8 printer files can be created by one job step without the need to assign these files to intermediate tape or disk for printing by subsequent jobs.

For a discussion of dummy printers and punches, refer to the LST and PUN parameters of the \* \$\$ LST and \* \$\$ PUN statements in "Job Entry Control Language" and to the PRINTERS and PUNCHES statements in "POWER/VS Start-Up".

## SUPPORTED HARDWARE DEVICES

POWER/VS can execute in any model of System/370 that is supported by DOS/VS and has at least 96K of real storage.

With the exception of those *models* not supported by DOS/VS, the following *devices* are supported:

Readers	Printers	Punches	Spooling Devices	Terminals
1442	1403	1442	2400 Series	2770
2501	1443	2520	3400 Series	2780
2520	3203	2540	2314/2319	3780
2540	3211	2560	3330/3333	
2560	5203	3525	3340	
3504		5425		
3505				
3525				
3540				
5425				

1. The IBM 2495 Tape Cartridge Reader cannot be used as a spooling device.
2. The IBM 2400 and 3400 series magnetic tape units cannot be used to spool input
3. The IBM 3540 Diskette Input/Output Unit is not supported by this component release of POWER/VS.
4. 3505, 1442, 2520, 2560, 3525, and 5425 are treated as sequential files only (not as combined files).

The card I/O functions supported by POWER/VS are indicated in the following table. Each row represents a valid combination of functions.

DEVICE	FUNCTION					
	READ	PUNCH	STACKER	SELECT	INTERPRET	PRINT
1442	X	X		X		
2501	X					
2520	X	X		X		
2540	X	X		X		
2560	X	X X		X X X	X	X X
3504	X					
3505	X					
3525	X	X X		X X X	X	X X
3540	X					
5425	X	X X		X X X	X	X X

The following I/O devices are supported by RJE:

- IBM 545 Punch (Model 3 or 4) on the 2770
- IBM 2213 Printer (Model 1 or 2) on the 2770
- IBM 2502 Card Reader (Model A1 or A2) on the 2770
- IBM 2203 Printer (Model A1 or A2) on the 2770
- IBM 3781 Punch on the 3780

Teleprocessing control units supported by RJE are:

- IBM 2701 Data Adapter Unit
- IBM 2703 Transmission Control Unit
- IBM 3704 Communications Controller in 2703 emulation mode
- IBM 3705 Communications Controller in 2703 emulation mode
- Integrated Communications Adapter for Models 115, 125, and 135

Teleprocessing connections must be point-to-point on switched or non-switched (leased) lines. Multipoint connections are not supported.

### Printer Support

The Universal Character Set buffer (UCB) and forms control buffer (FCB) features are supported by POWER/VS. The execution processor accepts UCB and FCB load requests from the various supported partitions for later action at list time. Upon encountering an FCB load command, the execution processor updates the internal buffer representation (line count table) to reflect the FCB to be used at print time. UCB and FCB specifications can be made either in the user program or in the POWER/VS LST or PRT cards. (See "Job Entry Control Language".)

### 3540 Support

*Note. The following description of support for the IBM 3540 Diskette Input/Output Unit is provided for planning purposes only. Support is not included in this component release. It is planned for availability on March 1975.*

POWER/VS is able to combine input from a card reader and a 3540 into a single sequential input stream. It is also possible to read the entire job stream from a 3540 file. Multivolume diskette files are supported.

The size of the records read from SYSIN files is obtained from the HDR1 label on the file. Records must be 80 or 81 bytes. Only the first 80 bytes are copied to the data file. These records are scanned for control statements. The SYSIN records can be read only by a user program that SYSIN assigned to a card reader. Because DTFDU cannot be assigned to a card reader, DTFDU cannot be used to access these files.

For data files, a read task reads records of any length between 1 to 128 characters. These records are not examined for control statements and are written on the spool file exactly as read. The data records cannot be read by programs accessing the SYSIN device. They can only be read by the programmer logical unit specified in the JECL RDR statement. This logical unit must be assigned to a 3540.

## POWER/VS GENERATION

POWER/VS generation involves:

- Selection of macro options
- Assembly of the macros to create the POWER/VS generation table

If the generated version supplied by IBM is adequate, POWER/VS generation is not required. The supplied program is described later in this chapter.

After generation, POWER/VS is initialized as described in "POWER/VS Start-Up".

## GENERATION MACROS

POWER/VS generation macros consist of the macro "POWER" and two RJE macros, "PLINE" and "PRMT".

## POWER Macro

The POWER macro is the only *required* POWER/VS generation macro.

The "name" field of the POWER macro becomes the phase name of the PHASE card (first card of the POWER/VS object deck). "name" is one to eight alphameric characters. The name "POWER" is generated if a name is not specified.

*Name    Operation    Operands*

<b>POWER</b>	POWER	DBLK= { $\frac{0}{n}$ }	*
name		,TRACKGP= { $\frac{0}{n}$ }	*
		,LTAB= { $\frac{(10,00,05,10,15,20,25,30,35,40,45,50,56)}{(d0,d1,d2,d3,d4,d5,d6,d7,d8,d9,d10,d11,d12)}$ }	*
		,PRI= { $\frac{3}{n}$ }	*
		,SUBLIB= { $\frac{NO}{s}$ }	*
		,ACCOUNT= { $\frac{NO}{YES}$ }	*
		,STDLINE= { $\frac{(0,0)}{(n,m)}$ }	*
		,STDCARD= { $\frac{(0,0)}{(n,m)}$ }	*
		,JLOG= { $\frac{YES}{NO}$ }	*
		,JSEP= { $\frac{(0,0)}{(n,m)}$ }	*
		,RBS= { $\frac{(0,0)}{(n,m)}$ }	*
		,RDREXIT= { $\frac{NO}{name}$ }	

PAUSE = NO  
          YES

DBLK= $\left\{ \begin{array}{c} 0 \\ n \end{array} \right\}$

Specifies the block size (in bytes) of data file records and, consequently, the size of each data buffer. (Data buffers are used as I/O areas by execution processors, readers/writers, and RJE routines.) "n" must be an integer from 516 to 2008.

If DBLK is omitted or set to 0 and if a cold start is performed, the system assumes a block size appropriate to the device type used for the data file:

Device Type	Data Block Size
2314/2319	920
3330/3333	952
3340	808

*Note. The increased disk arm activity produced by DBLK specifications smaller than the default values might affect POWER/VS performance.*

TRACKGP= $\left\{ \begin{array}{c} 0 \\ n \end{array} \right\}$

Specifies the number of tracks in a track group. A track group is the amount of disk space that is made available in the data file upon a request for space by a POWER/VS task. "n" can be any integer from 1 to the number of tracks on a cylinder of the device assigned to the data file. 3990 = 19 MAX # of TRKS/cyl.

Track groups do not cross cylinder boundaries. Remaining tracks are added to the last track group on each cylinder.

If the TRACKGP parameter is omitted or set equal to 0 (or conflicts are detected at initialization time), and a cold start is performed, POWER/VS calculates a suitable track group size. When a warm start is performed, the track group size established at the previous cold start is used.

LTAB= $\left\{ \begin{array}{c} (10,00,05,10,15,20,25,30,35,40,45,50,56) \\ (d0,d1,d2,d3,d4,d5,d6,d7,d8,d9,d10,d11,d12) \end{array} \right\}$

Defines the system's standard forms control tape or forms control buffer for its printers. If LTAB is specified, all entries in the sublist (d0 through d12) must be present. Each entry is specified as two digits.

LTAB specifies the distance (in number of print lines) from the first carriage control tape channel punch to the other channel punches, and to the end of the page. d0 specifies the distance from the channel 12 punch to the first punch; d1 specifies the distance between the first punch and the channel 1 punch; d2 specifies the distance between the first punch and the channel 2 punch; and so on. If channel 1 is the first punch, the distance d1 is expressed as 00. If a channel is not punched, 00 should be specified for that channel. An LTAB of all zeros can be specified to avoid a channel 12 indication.

If LTAB is not specified, the default is:

(10,00,05,10,15,20,25,30,35,40,45,50,56)

The default defines a 66 line carriage control tape (buffer), where the distance between channels 1 and 12 is 56 lines, the distance between channels 12 and 1 is 10 lines, the distance between channels 11 and 12 is 6 lines, and the distance between all other channels is 5 lines.

PRI= $\left\{ \begin{array}{c} 3 \\ n \end{array} \right\}$

Specifies the default priority (1-9) if priority is not specified in the \* \$\$ JOB statement. Nine is the highest priority.

If PRI is not specified, the default value is 3. (Priority is discussed in "POWER/VS Concepts and Facilities").

SUBLIB=  $\left\{ \begin{array}{c} \text{NO} \\ \text{s} \end{array} \right\}$

Specifies the sublibrary of the source statement library to be searched when the sublibrary is not specified in the \* \$\$ SLI statement. When an SLI statement is encountered in the job stream, POWER/VS searches the private source statement library (if one has been assigned in the POWER/VS partition) and the system source statement library for the requested book, and inserts it into the job stream. If NO is specified or if this operand is omitted, the SLI statement is not supported; it is treated as data.

"s" may be A through Z, 0 through 9, \$, #, or @. (Note that sublibraries A, C, D, E, F, and Z already have special purposes in DOS/VS.)

If SUBLIB is omitted, NO is assumed, and sublibrary inclusion is not supported.

ACCOUNT=  $\left\{ \begin{array}{c} \text{NO} \\ \text{YES} \end{array} \right\}$

Specifies whether or not POWER/VS is to maintain accounting information. If NO is specified, no information is maintained (apart from that provided in the status report; see "Appendix D"). If YES is specified, accounting information is gathered, merged with DOS/VS accounting information, and stored in the POWER/VS account file (IJAFILE).

// DLBL and // EXTENT information for the POWER/VS account file must be provided when POWER/VS is initialized. The logical unit specified in the EXTENT card must be SYS000. Job accounting must be specified in the DOS/VS supervisor.

STDLINE=  $\left\{ \begin{array}{c} (0,0) \\ (n,m) \end{array} \right\}$

✕

"n" is an arbitrary limit on the number of lines that can be generated for a job entry before warning message 1Q52I is issued to the operator stating that the limit has been exceeded. "m" specifies the additional number of lines that can be generated before the same message is reissued. Both n and m can be up to 6 digits.

A specification of 0 means that no warning will be given. If one of the entries is omitted, the value provided is used for both entries.

If STDLINE is omitted, (0,0) is assumed.

STDCARD=  $\left\{ \begin{array}{c} (0,0) \\ (n,m) \end{array} \right\}$

"n" is an arbitrary limit on the number of card images that each job entry can generate before warning message 1Q52I is issued to the operator stating that the limit has been exceeded. "m" specifies the additional number of records that can be generated before the same message is reissued. Both n and m can be up to 6 digits.

A specification of 0 means that no warning will be given. If one of the entries is omitted, the value provided is used for both entries. If STDCARD is omitted, (0,0) is assumed.

JLOG=  $\left\{ \begin{array}{l} \text{YES} \\ \text{NO} \end{array} \right\}$

Specifies whether POWER/VS is to display each job name together with the job number whenever an \* \$\$ JOB statement is read. If YES is specified, all such statements are logged on the system console (SYSLOG) at execution time. If NO is specified, no such logging takes place.

JSEP=  $\left\{ \begin{array}{l} (0,0) \\ (n,m) \end{array} \right\}$

Specifies the number of special pages or cards to be included in the output to separate job entries. "n" specifies the number of separator pages. "m" specifies the number of separator cards. "n" and "m" can be any digit from 0 to 9. If either is omitted, 0 is assumed.

The information in the separator pages or cards pertains to the output that follows. A separator page is printed with 10 lines, each of which contains the POWER/VS job name and number, user information from the \* \$\$ JOB card, and the date and time. separator cards contain only the job name.

*Note: Stacker selection is ignored for punch output with card separators.*

RBS=  $\left\{ \begin{array}{l} (0,0) \\ (n,m) \end{array} \right\}$

Specifies the number of list pages (n) and/or the number of punch records (m) that can be processed before the output is segmented. This parameter is used only when spooling to disk intermediate storage. "n" and "m" can be from one to six digits. A specification of 0 means that no segmentation will take place. If n or m is omitted, 0 is assumed.

*Note. Skip to channel 1 delimits a page; that is, if there are no skips to channel 1, page count is not incremented.*

RDREXIT=  $\left\{ \begin{array}{l} \text{NO} \\ \text{name} \end{array} \right\}$

Specifies whether a user-written POWER/VS read task exit is supported. Upon initialization of POWER/VS, the phase with the name specified in the RDREXIT parameter is loaded into the POWER/VS partition. See *Appendix G. - RDREXIT Routines.*

CUST-MAY WANT  
TO CATALOG VARIOUS  
VERSIONS OF POWER/VS  
W/DIFFERENT PARAMETERS  
OR OPTIONS

## PLINE Macro

REQD FOR RJE

The PLINE macro defines the hardware characteristics of an RJE *line* (the integrated communications adapter) or the transmission control unit at the central processor. One macro must be specified per line. Up to 25 RJE lines are supported by POWER/VS. However, with an integrated communications adapter (ICA) only four BSC lines are supported on a System/370 Model 115, only six BSC lines on a System/370 Model 125, and only eight BSC lines on the System/370 Model 135.

The POWER macro must precede the PLINE macro.

```
PLINE      ADDR=X'cuu'          *
           [TIMEOUT= { NO }    *
             n                  *
           [PSWRD=password ]   *
           [TRNSP= { YES }     *
             NO                 *
           [CODE= { EBCDIC }   *
             ASCII              *
```

**ADDR=X'cuu'** Specifies the line address (channel and unit number) in hexadecimal. This is the same address used to identify the control unit to the DOS/VS supervisor.

This parameter is required.

**TIMEOUT= { NO }  
n** Specifies the number of minutes a terminal is allowed to remain idle before SIGNOFF is forced. "n" may be any number from 0 to 255. If NO is specified or this parameter is omitted, the terminal is allowed to remain idle with the line open until manually disconnected.

**PSWRD=password** Specifies the security password that must be submitted by the remote operator to connect his terminal to the central system. The password can be up to 8 characters in length. It can be overridden (until the next PSTOP command) by the central operator via the PSTART command. If this parameter is omitted, no password processing will be done.

**TRNSP= { YES }  
NO** Specifies the transparency feature on the integrated communications adapter or the transmission control unit. This feature is needed to transmit object decks.

NO can be specified only if no object decks are to be transmitted over the line and if all other data is transmitted in non-transparent mode. In non-transparent mode, all characters transmitted to the terminal that are between X'00' and X'40' are converted to zeros.

**CODE= { EBCDIC }  
ASCII** Specifies the transmission code, which can be either EBCDIC or ASCII. The six-bit transmission code is not supported.

X

## PRMT Macro

The PRMT macro defines the hardware characteristics of an RJE *terminal*, identifies the user, and describes where his output is to be routed. At least one PRMT macro must be supplied for each user. If any feature is specified for a terminal on which that feature is not supported, the parameter is ignored.

All PRMT macros must be supplied after the PLINE macro(s), and must be specified in consecutive ascending sequence, starting with one, according to their remote-ids. (See the REMOTE parameter below.)

Two formats of the PRMT macro can be used. The format that includes the REF parameter can be used to define a user of a terminal that has the same features as those described by a previous PRMT macro.

```

PRMT          REMOTE=nnn,          *
              TYPE= { 2770 }        *
                  { 2780 }          *
                  { 3780 }          *
              [ ,LSTROUT= { nnn }   *
                  { remid } ]      *
              [ ,PUNROUT= { nnn }   *
                  { remid } ]      *
              [ ,TRNSP= { YES }     *
                  { NO } ]         *
              [ ,BE= { YES }        *
                  { NO } ]         *
              [ ,ABE= { NO }        *
                  { YES } ]        *
              [ ,HFC= { NO }        *
                  { YES } ]        *
              [ ,SCE= { NO }        *
                  { YES } ]        *
              [ ,MRF= { NO }        *
                  { YES } ]        *
              [ ,LIST= { 120 }       *
                  { 80 }            *
                  { 132 }           *
                  { 144 }           *
              ]                    *
              [ ,TRACE= { NO }      *
                  { YES } ]        *
              [ ,CS= { NO }         *
                  { YES } ]

```

or

```

PRMT          REMOTE=nnn,          *
              REF=mmm,            *
              [ ,LSTROUT= { nnn }   *
                  { remid } ]      *
              [ ,PUNROUT= { nnn }   *
                  { remid } ]      *
              [ ,TRACE= { NO }      *
                  { YES } ]

```

REMOTE=nnn

Specifies the remote-id (which identifies the user of an RJE terminal) to which this PRMT macro applies. "nnn" can be any integer from 1 to 200. If REMOTE=1, reference must not be made to another macro (the REF parameter cannot be used).

The remote-id defines a user rather than a terminal. Multiple users of the same terminal may have different remote-ids. This prevents output for one user to be mistakenly sent to another, and it allows for separate accounting. In a situation where, for example, two users of the same terminal have different remote-id's, two PRMT macros must be generated, but instead of specifying the terminal characteristics twice, the second macro may simply refer to the first (REF=mmm). The REF option can also be used when different terminals are used with identical hardware characteristics.

Remote-id must be assigned and generated in consecutive ascending sequence. Reference can only be made to another macro with a smaller number. The first PRMT must not refer to another macro and must specify REMOTE=1.

TYPE=  $\left\{ \begin{array}{l} 2770 \\ 2780 \\ 3780 \end{array} \right\}$

Specifies the type of terminal, which may be a 2770, 2780, or 3780.

LSTROUT=  $\left\{ \begin{array}{l} \text{nnn} \\ \text{remid} \end{array} \right\}$

Specifies where the list output from jobs submitted by this remote-id is to be routed by default. If this parameter is omitted, the same nnn specified in the REMOTE parameter is assumed.

PUNROUT=  $\left\{ \begin{array}{l} \text{nnn} \\ \text{remid} \end{array} \right\}$

Specifies where the punch output from jobs submitted by this remote-id is to be routed by default. If this parameter is omitted, the same nnn specified in the REMOTE parameter is assumed.

TRNSP=  $\left\{ \begin{array}{l} \text{YES} \\ \text{NO} \end{array} \right\}$

Specifies the transparency feature on the terminal. This feature is required for the transmission of object decks.

If NO is specified, all characters between X'00' and X'40' are converted to zeros. YES is assumed if this parameter is omitted.

BE=  $\left\{ \begin{array}{l} \text{YES} \\ \text{NO} \end{array} \right\} \left\{ \begin{array}{l} \text{NO} \\ \text{YES} \end{array} \right\}$

Specifies buffer expansion, which applies to the 2770 only. If NO is specified, the size of the buffer is 128 bytes. If YES is specified or if this parameter is omitted the size of the buffer is 256 bytes.

ABE=  $\left\{ \begin{array}{l} \text{NO} \\ \text{YES} \end{array} \right\}$

Specifies additional buffer expansion for the 2770. The size of the buffer is then 512 bytes. If NO is specified or if this parameter is omitted, buffer size is determined by the BE parameter.

HFC=  $\left\{ \begin{array}{l} \text{NO} \\ \text{YES} \end{array} \right\}$

Specifies horizontal format control on the 2770, 2780 or 3780.

*Note. If HFC and SCE are both specified, HFC is ignored.*

SCE=  $\left\{ \begin{array}{l} \text{NO} \\ \text{YES} \end{array} \right\}$

Specifies space compression/expansion, which applies to the 2770 and 3780 only.

MRF=  $\left\{ \begin{array}{l} \text{NO} \\ \text{YES} \end{array} \right\}$

Specifies the multiple record feature for the 2780.

LIST=  $\left\{ \begin{array}{l} 120 \\ 80 \\ 132 \\ 144 \end{array} \right\}$

Specifies the printer width (the number of print positions in a line). The numbers 80, 120, 132 and 144 are valid entries. When this parameter is omitted, 120 is assumed.

TRACE= { NO / YES }

Specifies that all I/O interrupts are to be traced for this terminal as soon as this remote-id is signed on.

CS= { NO / YES }

Specifies support for the component selection feature on the 3780. This is required for the 3781 punch.

REF=mmm

Specifies the remote-id of another PRMT macro that describes the terminal characteristics of this PRMT macro. "mmm" can be any integer lower than the nnn used in the REMOTE parameter. "mmm" can therefore be in the range 1 to 199.

### PRE-ASSEMBLED POWER/VS PROGRAM

The pre-assembled (generated) POWER/VS program supplied by IBM specifies the following parameters:

POWER	POWER	DBLK=0,	*
		TRACKGP=0,	*
		LTAB=(10,00,05,10,15,20,25,30,35,40,45,50,56),	*
		PRI=3,	*
		SUBLIB=NO,	*
		ACCOUNT=NO,	*
		STDLINE=(0,0),	*
		STDCARD=(0,0),	*
		JLOG=YES,	*
		JSEP=(0,0),	*
		RBS=(0,0),	*
		RDREXIT=NO	*

Note. All of the above parameters have taken default values.

### POWER/VS FILE LABELS

The following example job stream catalogs and then executes a procedure to create standard labels for POWER/VS files and DOS/VS system files that reside on two 3340 disks.

One extent is defined on the SYSRES pack for the queue file. It can contain 624 152-byte queue records and thus can describe 618 track groups.

The data file resides on two volumes. Its size is 103 cylinders, which is equivalent to 618 track groups, 2 tracks per track group.

There is also a standard label shown for the POWER/VS account file. If the account file is to be emptied and copied to a user disk file by a PACCOUNT command, standard labels for this user file can also be included.

```

// JOB CATALOG 'STDLABEL' PROCEDURE FOR 3340 POWER/VS
// EXEC MAINT
// CATALP STDLABEL
* STANDARD LABELS FOR 3340 'POWER/VS' SYSTEM, 09/30/74
// OPTION STDLABEL
// DLBL IJSYSRS, 'DOS/VS+POWER/VS.SYSRES.FILE', 99/365, SD
// EXTENT SYSRES, SYSRES, 1, 0, 1, 7199      600 CYLS, 0 - 599, ON SYSRES
// DLBL IJSYSCN, 'DOS/VS.HARDCOPY.FILE', 99/365, SD
// EXTENT SYSREG, SYSRES, 1, 0, 7200, 60      5 CYLS, 600 - 604, ON SYSRES
* EXTENT SYSVIS, SYSRES, 1, 1, 7260, 168     14 CYLS, 605 - 618, ON SYSRES
* POWER/VS AUTOSTART FILES, 7448, 12       1 CYL , 619 - 619, ON SYSRES
// DLBL IJQFILE, 'POWER/VS.QUEUE.FILE', 99/365, DA
// EXTENT SYS001, , 1, 0, 7440, 24          2 CYLS, 620 - 621, ANY DRIVE
// DLBL IJDFILE, 'POWER/VS.DATA.FILE', 99/365, DA
// EXTENT SYS002, , 1, 0, 7464, 804        67 CYLS, 622 - 688, ANY DRIVE
// EXTENT SYS003, , 1, 1, 3480, 432       36 CYLS, 290 - 325, NOT SYSRES
// DLBL IJAFILE, 'POWER/VS.ACCOUNTING.FILE', 99/365, DA
// EXTENT SYS000, , 1, 0, 8268, 60        5 CYLS, 689 - 693, ANY DRIVE
// DLBL IJSYSRC, 'DOS/VS.RECORORDER.FILE', 99/365, SD
// EXTENT SYSREC, SYSRES, 1, 0, 8328, 12   1 CYL , 694 - 694, ON SYSRES
* EXTENT VTOC                               1 CYL , 695 - 695, ON SYSRES
// DLBL IJSYSLN, 'DOS/VS.SYSLNK.FILE', 0, SD
// EXTENT SYSLNK, , 1, 0, 12, 108         9 CYLS, 1 - 9, ANY DRIVE
// DLBL IJSYS01, 'DOS/VS.WORK-FILE.1', 0, SD
// EXTENT SYS001, , 8, 0, 120, 1000, 3    1/3 SPLIT 10 - 259, ANY DRIVE
// DLBL IJSYS02, 'DOS/VS.WORK-FILE.2', 0, SD
// EXTENT SYS002, , 8, 0, 124, 1000, 7    1/3 SPLIT 10 - 259, ANY DRIVE
// DLBL IJSYS03, 'DOS/VS.WORK-FILE.3', 0, SD
// EXTENT SYS003, , 8, 0, 128, 1000, 11   1/3 SPLIT 10 - 259, ANY DRIVE
// DLBL IJSYS04, 'DOS/VS.WORK-FILE.4', 0, SD
// EXTENT SYS004, , 1, 0, 3120, 360      30 CYLS, 260 - 289, ANY DRIVE
* DOS/VS 3340 STANDARD LABELS CREATED *
/*
/*
/*
// EXEC PROC=STDLABEL
/£

```

## DOS/VS SUPERVISOR GENERATION REQUIREMENTS

POWER/VS requires a DOS/VS supervisor generated with POWER=YES in the SUPVR macro. This generation option forces the additional options PHO=YES, PFI=YES, ECPREAL=YES, and NPARTS=3. (NPARTS specifications 2, 4, and 5 are also valid.)

If POWER/VS accounting support is required, job accounting must also be generated in the supervisor.

Since RJE uses its own teleprocessing access method, it does not require any teleprocessing support to be generated in the supervisor.

Both the ALLOC and the ALLOCR macro are used to define the POWER/VS partition. Refer to "Storage Requirements" to determine the appropriate real and virtual storage sizes.

If dummy printers and punches are to be supported, they should be generated, using the DVCGEN macro.

The number of LUB entries for programmer logical units in the POWER/VS partition must be large enough to accommodate all read and write tasks that may be running concurrently. If the number of programmer logical units in the POWER/VS partition is insufficient, (i.e., too many tasks are started concurrently) a message is issued. Programmer logical units SYS000 through SYS006 are reserved for POWER/VS files. The logical units used for POWER/VS files must be assigned by the user. The other logical units are automatically assigned and unassigned during POWER/VS execution.

Typically, at least 10 LUB entries are required: 7 for POWER/VS disk files and 3 for the read, list, and punch tasks. Additional LUB entries are needed for RJE tasks: one entry per task, per line. LUB entries are also required for tape files used to either spool output or save the contents of the account file.

## MAIN STORAGE REQUIREMENTS

Both a virtual partition and a corresponding real partition must be allocated for POWER/VS.

The virtual partition contains the permanent area, fixable area, and pageable area.

- **Permanent Area.** Consists of three pages (6K) that are fixed at POWER/VS initialization and remain fixed until POWER/VS is terminated. These pages contain the POWER/VS nucleus and control tables, which do not tolerate paging activity.
- **Fixable Area.** Consists of those pages that are fixed when a task is started and freed when the task becomes inoperative. The size of the fixable area determines the number of data buffers available to the system and hence the number of POWER/VS tasks that can execute concurrently. Besides data buffers, this area contains dynamic control blocks.

Page fixing and freeing is controlled by the storage management service of POWER/VS. Free pages within the fixable area form part of the DOS/VS page pool and are therefore available for use by other partitions.

- **Pageable Area.** Consists of pages that can be freed when other partitions require additional real storage. These pages contain the major part of the POWER/VS code, which is re-entrant and therefore never needs to be paged out (saved). The virtual address space required by this area is 128K.

All of the permanent area resides in the *real* partition space allocated to the POWER/VS partition by the ALLOC statement. The fixable area is the remainder of the real storage allocated. The permanent and fixable areas together cannot exceed 128K. An ALLOC greater than 128K, however, is valid.

If a user RDREXIT routine is to be used, add its size to the virtual address space requirement for the POWER/VS partition; that is, increase the pageable area size beyond 128K. See Figure 6.

Pages within the fixable area are PFIXED and PFREED as required throughout POWER/VS execution. A request for storage is satisfied immediately if space is available in the fixable area. If not available, a message is issued and the task waits until storage is freed.

Use the following table to determine the maximum real storage requirement for the permanent and fixable areas.

Task or Line	Real Storage (bytes)
Permanent area	6144
Dynamic control blocks	2048
Each read/write task (one for each active unit record device)	1024 + N
Each execution read task (one for each active card reader, real or dummy, referenced in a POWER/VS controlled partition)	488 + N + X
Each execution write task (one for each active printer/punch, real or dummy, referenced in a POWER/VS controlled partition)	488 + N
Each inactive RJE line	288
Each active RJE line	768 + P + N

Where:

$$N = 32 * n$$

$$n = \frac{DBLK + 39}{32}$$

DBLK represents the value specified in the DBLK parameter of the POWER macro. DBLK can vary from 516 to 2008.

P = the terminal buffer size (128, 256, or 512).

X=0 if the SLI facility is *not* used by the execution read task. X=288 when the SLI facility *is* used by an execution read task.

To determine the total virtual address space required by the POWER/VS partition (ALLOC), add the real storage requirement to the 128K required by the pageable area.

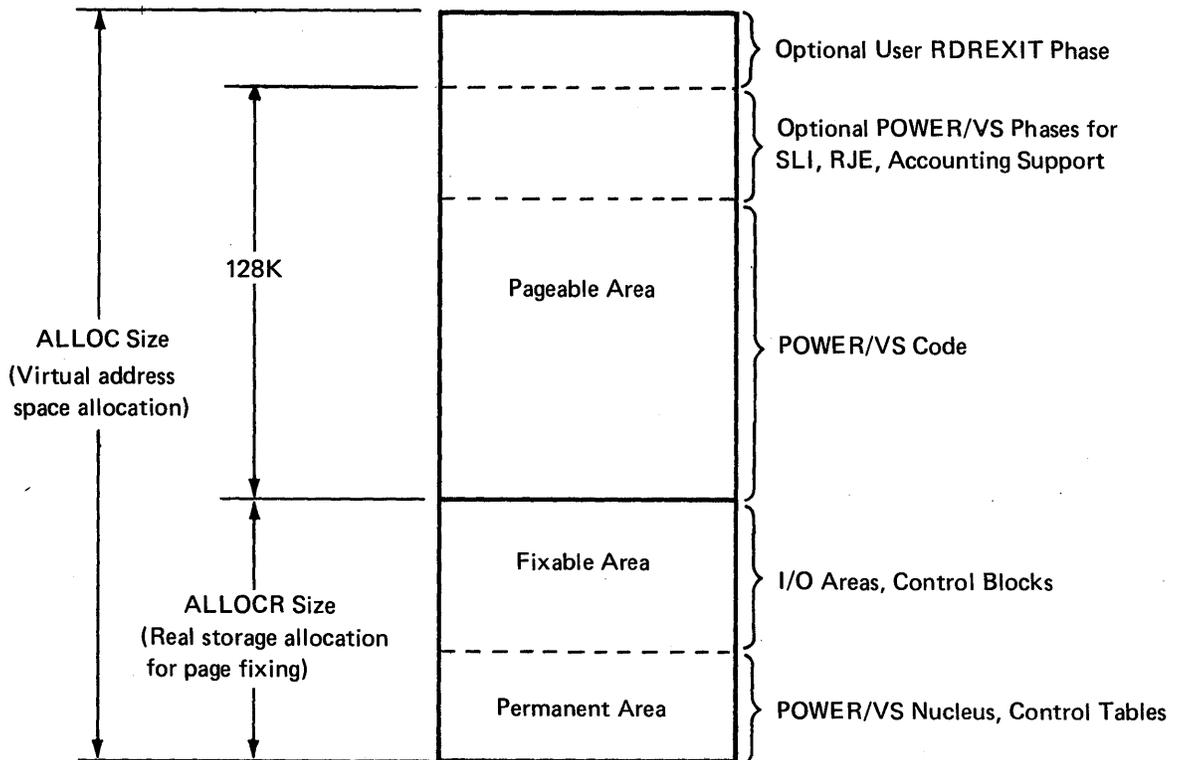


Figure 6. POWER/VS Partition Layout

*Examples:*

In each of the following examples, ALLOC and ALLOCR are calculated for the POWER/VS partition (assumed to be F1).

**Entry System Example**

- 1 card reader, 1 card punch, 1 printer. Cards are only punched occasionally (2 read/write tasks).
- 1 controlled partition, referencing a reader, printer, and punch (3 execution processors).
- Minimum DBLK of 516 bytes, to reduce POWER/VS real storage requirement.
- SLI facility.

Permanent area			6144
Dynamic control blocks			2048
Read/write tasks	$2*(1024 + 32*17)$	=	3136
Execution read tasks	$1*(448 + 32*17 + 288)$	=	1280
Execution write tasks	$2*(448 + 32*17)$	=	1984
			<u>14592</u> bytes

Hence, the following allocations would be required to run POWER/VS in F1:

ALLOCR F1R=16K  
ALLOC F1=144K (128K + 16K)

**Typical System Example**

- 1 card reader, 1 card punch, 1 printer (3 read/write tasks).
- 2 controlled partitions. Normally, one partition references 1 reader and 2 printers, the other a reader, printer, and punch (6 execution processors).
- DBLK default value of 808 bytes for 3340 disk.
- SLI facility used in both partitions.

Permanent area			6144
Dynamic control blocks			2048
Read/write tasks	$3*(1024 + 32*26)$	=	5568
Execution read tasks	$2*(448 + 32*26 + 288)$	=	3136
Execution write tasks	$4*(448 + 32*26)$	=	5120
			<u>22016</u> bytes

Hence: ALLOCR F1R=22K  
ALLOC F1=150K

**Large System Example**

- 2 card readers, 1 card punch, 3 printers (6 read/write tasks).
- 8 RJE lines. Normally 5 active.
- 4 controlled partitions. Normally:
  - One partition references 1 reader and 1 printer.
  - Two partitions each reference 1 reader, 1 punch, and 1 printer.
  - One partition references 1 reader, 1 punch, and 4 printers.(Total of 4 execution read tasks and 10 execution write tasks.)
- DBLK maximum of 2008 bytes for best POWER/VS performance.
- Terminal buffer size of 256 bytes.
- SLI facility used in all 4 partitions.
- 4K RDREXIT routine included.

Permanent area		6144
Dynamic control blocks		2048
Read/write tasks	$6*(1024 + 32*63)$	= 18240
Execution read tasks	$4*(448 + 32*63 + 288)$	= 11008
Execution write tasks	$10*(448 + 32*63)$	= 24640
Inactive RJE lines	$3*288$	= 864
Active RJE lines	$5*(768 + 256 + 32*63)$	= 15200
User RDREXIT routine		<u>4096</u>
		82240 bytes

Hence: ALLOCR F1R=82K  
 ALLOC F1=210K

An ALLOCR value larger than calculated can be specified. This would create a larger fixable area and thus might increase paging activity in the system.

An ALLOCR value smaller than calculated can be specified. This would reduce the size of the fixable area. As a result, POWER/VS tasks may occasionally have to wait for real storage space to be freed.

## DIRECT ACCESS STORAGE REQUIREMENTS

### DOS/VS System Libraries

POWER/VS is distributed as source macros and core-image phases, including a version of POWER/VS that was generated using the default values of the POWER macro parameters. (Refer to 'Pre-assembled POWER/VS Program'.)

### POWER/VS Files

The POWER/VS data file can occupy up to five extents of direct access storage. The queue file and account file occupy one extent each.

The following information will assist in establishing preliminary POWER/VS file sizes.

#### *Data File*

For the data file extents, estimate separately the total number of input/output card images and the total number of line images spooled to disk in a typical 8 hour shift. Choose a file size large enough to hold half this amount of data. This should prevent POWER/VS from running out of file space. File extents can be respecified if they later prove to be too large or too small.

The size of the DBLK parameter in the POWER macro determines how many records will fit into a data block and how many data blocks will fit on a track. Also note that POWER/VS omits trailing blanks from all records placed on the data file.

To quickly determine a reasonable data file size, use one of the following as a guideline.

- 2314      100 cylinders hold 90,000 line images or 135,00 card images
- 3330      100 cylinders hold 160,00 line images or 240,000 card images
- 3340      100 cylinders hold 60,000 line images or 90,000 card images

#### *Queue File*

The queue file should be large enough to support the entire data file; that is, there should be one queue record for each track group in the data file.

The number of track groups depends on the size of the data file and on the number of tracks per track group (specified in the TRACKGP parameter of the POWER macro).

If the minimum TRACKGP value of 1 is used, then one queue record is needed for each track of the data file. This increases queue management overhead and disk activity, but best utilizes the disk space available in the data file. Conversely, if the track group size is large, queue management overhead is reduced, and because only one job entry can use a track group, disk space is wasted whenever that entry does not fill a track group. A track group of 2 is often a good compromise.

The following table shows queue records per track/cylinder for each disk type.

Device	Queue Records per Track	Tracks per Cylinder	Queue Records per Cylinder
2314	28	20	560
3330	45	19	855
3340	26	12	312

*Note. Up to 6 queue records are reserved for POWER/VS internal use. The remainder of the records in the file are used to describe track groups.*

If a track group size is not specified explicitly, POWER/VS, at initialization, calculates the number of tracks within the extents provided by the data file, and determines the number of 152-byte records it can write within the queue file. From these two figures it determines the smallest track group size that allows all of the data file to be used. Moreover, if the user-specified track group is too small and the entire data file cannot be used, POWER/VS changes the track group value and informs the operator. If there are more cylinders of data file than there are queue records, the extra data file space is not used.

The queue file extents in the example under 'POWER/VS File Labels' define two cylinders of 3340 disk. This equates to 624 queue records, which can describe 618 (i.e., 624-6) track groups. This, in turn, allows for 51 cylinders of 3340 data file space if TRACKGP=1, 103 cylinders if TRACKGP=2, 154 cylinders if TRACKGP=3, etc.

### Account File

The following list shows approximately how many DOS/VS jobs can be handled by one cylinder of the account file.

- 2314      110 jobs
- 3330      170 jobs
- 3340      60 jobs

These estimates are based on an average of 5 account records per job.

When the account file becomes 80% full, a warning message is issued. The file should then be saved or deleted using the PACCOUNT command.

*Note. If any POWER/VS file fills completely, the operator is notified, and any task requiring space in the file waits until space becomes available.*

## POWER/VS START-UP

Start-up of POWER/VS can be done exclusively through the operator console, or in many cases, the procedure can be simplified for the operator by using the AUTOSTART facility, which allows most of the commands and specifications to be entered in the form of card images. Examples of both approaches to POWER/VS start-up follow.

'Start-Up Considerations', at the end of this section, contains information which can help to determine which approach is better for a particular environment.

### START-UP, WITHOUT AUTOSTART

The following steps whose numbers are suffixed by 'A' relate to the sample console log, Figure 7. In the example, the POWER/VS partition is F1, and BG is the only controlled partition.

*Note. Before starting POWER/VS, be certain that all devices to be spooled (dummy as well as real) are defined in the DOS/VS supervisor.*

- 1 Mount the disk pack(s) used for the POWER/VS files on the disk drive(s) and ready them.
- 2A Make the reader, printer, and punch available for POWER/VS use by *unassigning* these devices in the partition that 'owns' them.
- 3A Stop BG.
- 4A Start the POWER/VS partition (F1).
- 5A If the current ALLOCR and/or ALLOC are not suitable, allocate sufficient real and virtual storage to the partition in which POWER/VS will execute. Refer to 'Storage Requirements' in 'POWER/VS Generation' to estimate the required real partition size. A value of 22K should be adequate for most systems. The required virtual size must be at least as large as the sum of 128K and the real partition size.
- 6 Optionally assign SYSRDR to the unit-record device from which job control statements will be read to initialize POWER/VS. If SYSRDR is not assigned in the POWER/VS partition, all job control statements in the following steps must be entered via SYSLOG.

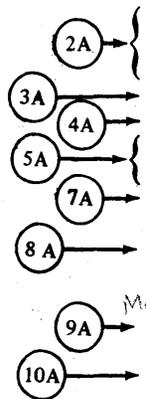
Assign SYSIPT (or SYSIN) if the AUTOSTART facility is to be used. See 'AUTOSTART' in this chapter.

Neither SYSRDR nor SYSIPT is assigned in this example.

- 7A If required, assign SYSLST so that:
  - The POWER/VS status report plus a list of all entries in the POWER/VS queue is printed after a warm start. See Appendix D.
  - A dump of the POWER/VS partition will occur if POWER/VS abnormally terminates during initialization (step 13A).
  - Not assigning SYSLST allows start-up and operation of POWER/VS without an available printer (i.e., there is no printer attached, printer is 'down', or printer is assigned to a non-spooled partition).
- 8A Enter a DOS/VS JOB statement, which properly begins the new job and identifies the POWER/VS partition on the console log.
- 9A Assign SYS000 to disk. If this logical unit is not assigned, account file logging is ignored, even if the DOS/VS supervisor and POWER/VS have accounting support.

F1 shows 28K

USE FOR  
GUIDE  
TO BRING  
UP PWR



```

BG * POWER/V5 START-UP ENTIRELY FROM CONSOLE *
BG *
BG ASSGN SYSLST,UA
BG ASSGN SYSPCH,UA
BG ASSGN SYSIN,UA
BG STOP
AR START F1
F1 ALLOC R F1=22K,F2R=0K,BGR=22K
F1 ALLOC F1=150K
F1 ASSGN SYSLST,PRINTER
F1 IT20I SYSLST HAS BEEN ASSIGNED TO X'00E'
F1 // JOB POWER/V5
F1 // JOB POWER/V5
DATE 08/06/74,CLOCK 00/14/40
F1 ASSGN SYS000,SYSRES
F1 IT20I SYS000 HAS BEEN ASSIGNED TO X'160'
F1 ASSGN SYS001,SYSRES
F1 IT20I SYS001 HAS BEEN ASSIGNED TO X'160'
F1 ASSGN SYS002,SYSRES
F1 IT20I SYS002 HAS BEEN ASSIGNED TO X'160'
F1 ASSGN SYS003,DISK,VOL=111111,SHR
F1 IT20I SYS003 HAS BEEN ASSIGNED TO X'161'
F1 EXEC POWERVS
F1 IQ11D FORMAT QUEUES=Q,D,A
*F1 4733A EQUAL FILE ID IN VTDC          IJQFILE
  SYS001=160 SYSRES
  POWER/V5.QUEUE.FILE
F1 DELETE
*F1 4733A EQUAL FILE ID IN VTDC          IJDFILE
  SYS002=160 SYSRES
  POWER/V5.DATA.FILE
F1 DELETE
*F1 4733A EQUAL FILE ID IN VTDC          IJDFILE
  SYS003=161 111111
  POWER/V5.DATA.FILE
F1 DELETE
*F1 4733A EQUAL FILE ID IN VTDC          IJAFILE
  SYS000=160 SYSRES
  POWER/V5.ACCOUNTING.FILE
F1 DELETE
F1 IQ12I POWER/V5 INITIALIZATION COMPLETED
AR PSTART RDR,00C
*F1 0P08A INTERV REQ SYS069=00C
AR PSTART PUN,00D,AXP
F1 IQ34I PUN WAITING FOR WORK ON 00D
AR PSTART LST,00E
F1 IQ34I LST WAITING FOR WORK ON 00E
AR PSTART LST,04E,HBCA
F1 IQ34I LST WAITING FOR WORK ON 04E
AR PSTART BG,00B
*F1 1R86D PLEASE SPECIFY DEVICES TO BE SPOOLED
F1 1R50D BG READER=00C
F1 1R50D BG PRINTERS=00E,01E,02E
F1 1R50D BG PUNCHES=00D,04D
BG ASSGN SYSLST,PRINTER
BG IT20I SYSLST HAS BEEN ASSIGNED TO X'00E'
BG ASSGN SYSPCH,PUNCH
BG IT20I SYSPCH HAS BEEN ASSIGNED TO X'04D'
BG ASSGN SYSIN,READER
BG IT20I SYSIN HAS BEEN ASSIGNED TO X'00C'
BG
F1 IQ34I BG WAITING FOR WORK
AR C A
F1 1R48I PUN,00D,AXP
F1 1R48I LST,00E,A
F1 1R48I LST,04E,HBCA
F1 1R48I BG,00C,00B
F1 1R48I RDR,00C,A
  
```

DO A MAP CMD - VERIFY FIREAL  
PART R30 = 26K  
SIZE R31 = 30K  
(16K)  
(18K)  
WITH BG

START F1 PART  
DO MAP  
EXECUTE PROC = POWER VS1  
FOR 125  
EXECUTE PROC = PWRVS  
DOES ASSIGN

D9A FORMATS DATA  
& QUEUE FILES

EXECUTE POWER VS B  
IF YOU DO NOT HAVE DATA

ON 125  
RDR,00D,A  
VM,00C,A

D RDR, ALL (DISPLAYS READER Q.)  
LOOK AT JECL, ETC. SEE WHAT  
IS WRONG THEN DO ALTER  
COMMAND TO CORRECT

A. RDR, JOB ON C  
S LST, 00E, A  
(S PUN, 00D, A)

Figure 7. Console Log for POWER/V5 Start-up

See P. 6.1

RDR = 00D

- 10A Assign SYS001 to disk for the POWER/VS queue file.
- 11A Assign SYS002 to disk for the POWER/VS data file. If the data file is to consist of multiple extents, SYS003 to SYS006 can also be assigned.
- 12 Provide DLBL and EXTENT information for the POWER/VS disk files specified above. This can be done in one of four ways: as part of the job stream, within a start-up procedure, in the POWER/VS partition standard labels, or in standard labels (see 'POWER/VS file Labels' in 'POWER/VS Generation').
- 13A To initialize POWER/VS, specify the name of the default or user-created POWER/VS generation table in an EXEC statement.
- 14A With initialization, all required direct access files are opened. In addition, the prompting message 1Q11D FORMAT QUEUES= is issued, which allows direct access files to be formatted. Possible replies to the message are:
- |                   |                                                  |
|-------------------|--------------------------------------------------|
| No or (END/ENTER) | - No formatting (warm start)                     |
| A                 | - Format account file                            |
| Q(A)              | - Format queue file (and account file)           |
| D(A)              | - Format queue and data files (and account file) |

**Note:** When the data file is specified, the queue file is also formatted.

Reformatting an existing file causes message 4733A to be logged. The operator should then delete the existing file. If the operator decides not to format the files, he can cancel POWER/VS by pressing ENTER or keying CANCEL.

- 15A Start POWER/VS tasks and user partitions. Commands must be entered through the console if SYSIPT was not previously assigned. If SYSIPT was assigned during initialization, the POWER/VS AUTOSTART facility can be used to initiate tasks and partitions. Commands can be entered in any order and at any time.
- Any number of partitions can be started. They must be inactive DOS/VS partitions with lower priority than the POWER/VS partition.
- 16A For each partition started under POWER/VS, specify the reader, printer, and punch device addresses to be spooled. If a partition is started with an operator command, these addresses are provided as replies to prompting questions on SYSLOG. They are provided as parameters in READER, PRINTERS, and PUNCHES cards if the AUTOSTART facility is used.
- 17A After starting a partition under control of POWER/VS; assign SYSIN, SYSLST, and SYSPCH for that partition.

PEND  
 F9 STOP F1  
 BG  
 S BG A  
 EXEC IPW \$\$\$ DD  
 (see sec 6 of PLM)  
 FOR WRITE UP  
 DISPLAY QUEUE & DATA FILE  
 2067

ON VM PROC = POWER VS 2  
 ON 125 PROC = BGPWR

**START-UP, WITH AUTOSTART**

AUTOSTART allows any number of POWER/VS PSTART commands and the addresses of spooled unit-record devices to be prepared on cards (or on tape or disk) and submitted as batch data to POWER/VS at start-up. This simplifies start-up by reducing operator involvement.

AUTOSTART is particularly suited to the frequent or regular initiation of a POWER/VS environment where the device addresses, tasks, spooled partitions, and RJE lines remain the same.

```

BG *  P O W E R / V S  S T A R T - U P  D E C K  *
*BG 1C10A PLEASE ASSIGN SYSRDR.
BG STOP
AR START F1
F1 ASSGN SYSIN,READER
F1 1T20I  SYSIN  HAS BEEN ASSIGNED TO X'00C'
F1
F1  ALLOCR F1R=22K,F2R=0K,BGR=20K
F1  ALLOC  F1=150K
F1 1T20I  SYSLST HAS BEEN ASSIGNED TO X'00E'
F1 // JOB POWER/VS
    DATE 08/06/74,CLOCK 00/29/34
F1 1T20I  SYS000 HAS BEEN ASSIGNED TO X'160'
F1 1T20I  SYS001 HAS BEEN ASSIGNED TO X'160'
F1 1T20I  SYS002 HAS BEEN ASSIGNED TO X'160'
F1 1T20I  SYS003 HAS BEEN ASSIGNED TO X'161'
F1 1Q20I  AUTOSTART IN PROGRESS
F1 1R75I  F2 AUTOSTARTED
F1 1Q34I  LST WAITING FOR WORK ON 00E
F1 1R75I  BG AUTOSTARTED
F1 1Q34I  LST WAITING FOR WORK ON 04E
BG ASSGN SYSIN,SYSRES
F1 1Q12I  POWER/VS INITIALIZATION COMPLETED
F2 ASSGN SYSIN,SYSRES
F1 1Q34I  PUN WAITING FOR WORK ON 00D
BG 1T20I  SYSIN  HAS BEEN ASSIGNED TO X'160'
*F1 1Q35A  EOF ON 00C
BG
F2 1T20I  SYSIN  HAS BEEN ASSIGNED TO X'160'
BG 1T20I  SYSLST HAS BEEN ASSIGNED TO X'00E'
F2
BG 1T20I  SYSPCH HAS BEEN ASSIGNED TO X'04D'
F2 1T20I  SYSLST HAS BEEN ASSIGNED TO X'00E'

```

Diagram annotations: 3B points to 'BG STOP', 4B points to 'AR START F1', 6B points to 'F1 ASSGN SYSIN,READER' and 'F1 1T20I SYSIN HAS BEEN ASSIGNED TO X'00C'', 8B points to '\*F1 1Q35A EOF ON 00C'.

Figure 8. Console Log for POWER/VS AUTOSTART

```

*  P O W E R / V S  S T A R T - U P  D E C K  *
ASSGN SYSPCH,UA
ASSGN SYSLST,UA
ASSGN SYSIN,UA
ALLOCR F1R=22K,F2R=0K,BGR=20K
ALLOC  F1=150K
ASSGN SYSLST,PRINTER
// JOB POWER/VS
ASSGN SYS000,SYSRES  POWER/VS ACCOUNTING FILE
ASSGN SYS001,SYSRES  POWER/VS QUEUE FILE
ASSGN SYS002,SYSRES  POWER/VS DATA FILE
ASSGN SYS003,DISK,VOL=111111,SHR  OPTIONAL 2ND EXTENT TO DATA FILE
// EXEC POWERVS
FORMAT=NQ
PSTART RDR,COC
PSTART LST,00E
PSTART LST,04E,HBCA
PSTART PUN,00D,AXP
PSTART F2,H2CA
READER=00C
PRINTERS=00E,01E,02E
PUNCHES=00D,04D
PSTART BG,0ABI
READER=00C
PRINTERS=00E,01E,02E
PUNCHES=00D,04D
/*

```

Diagram annotations: 2C points to 'ASSGN SYSPCH,UA', 'ASSGN SYSLST,UA', and 'ASSGN SYSIN,UA'; 5C points to 'ALLOCR F1R=22K,F2R=0K,BGR=20K' and 'ALLOC F1=150K'; 7C points to 'PSTART RDR,COC', 'PSTART LST,00E', 'PSTART LST,04E,HBCA', 'PSTART PUN,00D,AXP', and 'PSTART F2,H2CA'.

Figure 9. POWER/VS AUTOSTART Deck

## PROCEDURE

The following steps whose numbers are suffixed by 'B' relate to the sample console log, Figure 8, and those steps suffixed by 'C' relate to the sample card listing, Figure 9. In the example, the POWER/VS partition is F1, and BG and F2 are the controlled partitions.

**Note.** Before starting POWER/VS, be certain that all devices to be spooled (dummy as well as real) are defined in the DOS/VS supervisor.

- 1 Mount the disk pack(s) used for the POWER/VS files on the disk drive(s) and ready them.
- 2C Make the reader, printer, and punch available for POWER/VS use by *unassigning* these devices in the partition that 'owns' them.
- 3B Stop BG.
- 4B Start the POWER/VS partition (F1).
- 5C If the current ALLOCR and/or ALLOC are not suitable, allocate sufficient real and virtual storage to the partition in which POWER/VS will execute. Refer to 'Storage Requirements' in 'POWER/VS Generation' to estimate the required real partition size. A value of 22K should be adequate for most systems. The required virtual size must be at least as large as the sum of 128K and the real partition size.
- 6B Assign SYSIN (SYSRDR plus SYSIPT) to a card reader, to tape, or to sequential disk.  
After assigning SYSIN, press END (ENTER on the System/370 Models 115 and 125) to begin reading from SYSRDR.
- 7C Job control statements as described in steps 7A-16A are then read, and POWER/VS is loaded.  
POWER/VS reads from SYSIPT until it finds a /\* statement. All cards between the EXEC statement and /\* must be valid AUTOSTART commands; that is, a FORMAT statement, PSTART commands, and address specification statements. All these commands must start in column one.  
At this point SYSRDR, SYSPCH, and SYSLST have been assigned in the POWER/VS partition, and tasks and partitions specified in PSTART commands have been started.  
User job streams to be read by POWER/VS can be placed immediately after the /\* in the AUTOSTART deck. These jobs are automatically read after POWER/VS is initiated.

**Note.** RJE lines can be started through AUTOSTART.

- 8B Job control in each partition started gives the operator control through SYSLOG. This provides the opportunity to make or change assignments, execute procedures, or enter job streams.  
For each partition, press END (or ENTER) when all job control commands, if any, have been entered.

## FORMAT Statement

FORMAT is normally the first statement of the AUTOSTART deck. It is used to format POWER/VS files.

FORMAT=	{	NO	- No formatting (warm start).
		A	- Format account file.
		D [A]	- Format data and queue files (and account file).
		Q [A]	- Format queue file (and account file).

If FORMAT is not the first statement encountered, or is invalid, the operator is prompted with the following message:

1Q11D FORMAT QUEUES=

The following are valid replies:

{	NO	or END/ENTER
	A	
	D [A]	
	Q [A]	

*Note. When the data file is specified, the queue file is also formatted.*

## Address Specification Statements

For each partition started, the input/output device addresses to be intercepted and spooled by POWER/VS must be specified by the following address specification statements. These statements must appear in the order shown here:

READER=	{	NO	}
		uraddr	}
PRINTERS=	{	NO	}
		uraddr,uraddr2 ... uraddr8	}
PUNCHES=	{	NO	}
		uraddr,uraddr2 ... uraddr8	}

**uraddr.** Address of the unit-record device whose input/output is to be spooled.

**uraddr2, ... uraddr8.** Address(es) of additional printers or punches to be spooled.

**NO.** Specifies no spooling. READER=NO provides writer-only support for a partition. Specifying NO for both PRINTERS and PUNCHES defines a reader-only partition. NO cannot be specified for all three.

If there are errors in address specification statements or if any are missing, the operator is prompted with the following messages:

1R86D PLEASE SPECIFY DEVICES TO BE SPOOLED  
1R50D READER=  
1R50D PRINTERS=  
1R50D PUNCHES=

## START-UP CONSIDERATIONS

POWER/VS can run in any partition, including BG, and can support all partitions with a lower dispatching priority. Although start-up is simpler when POWER/VS runs in BG, there are the following restrictions:

- Certain DOS/VS system maintenance functions (MAINT) only execute in BG.
- CE on-line diagnostic programs such as OLTEP must run in BG. If POWER/VS were run in a foreground partition, CE diagnostics could run concurrently.

Once it is established which partition POWER/VS will execute in and what the required real partition size must be, start-up and system operation can be simplified by including standard defaults in the supervisor, using the following macros:

ALLOC	- POWER/VS virtual partition size.
ALLOCR	- POWER/VS real partition size.
PRTY	- POWER/VS partition priority, and priorities for spooled partitions.
DVCGEN	- Spool device specifications. One for each device including dummy printers/punches.
ASSGN	- POWER/VS logical unit assignments for account, queue, and data file. - SYSRDR/IPT/PCH/LST assignments in each POWER/VS supported partition or SYSRDR assignments to disk for each foreground partition.

SYS000, SYS001, and SYS002 can be assigned to different device types. However, SYS003 through SYS006 must be assigned to the same device type as SYS002.

Unit-record assignments for POWER/VS supported partitions can be made in the supervisor to simplify start-up and eliminate the need to enter partition-dependent statements.

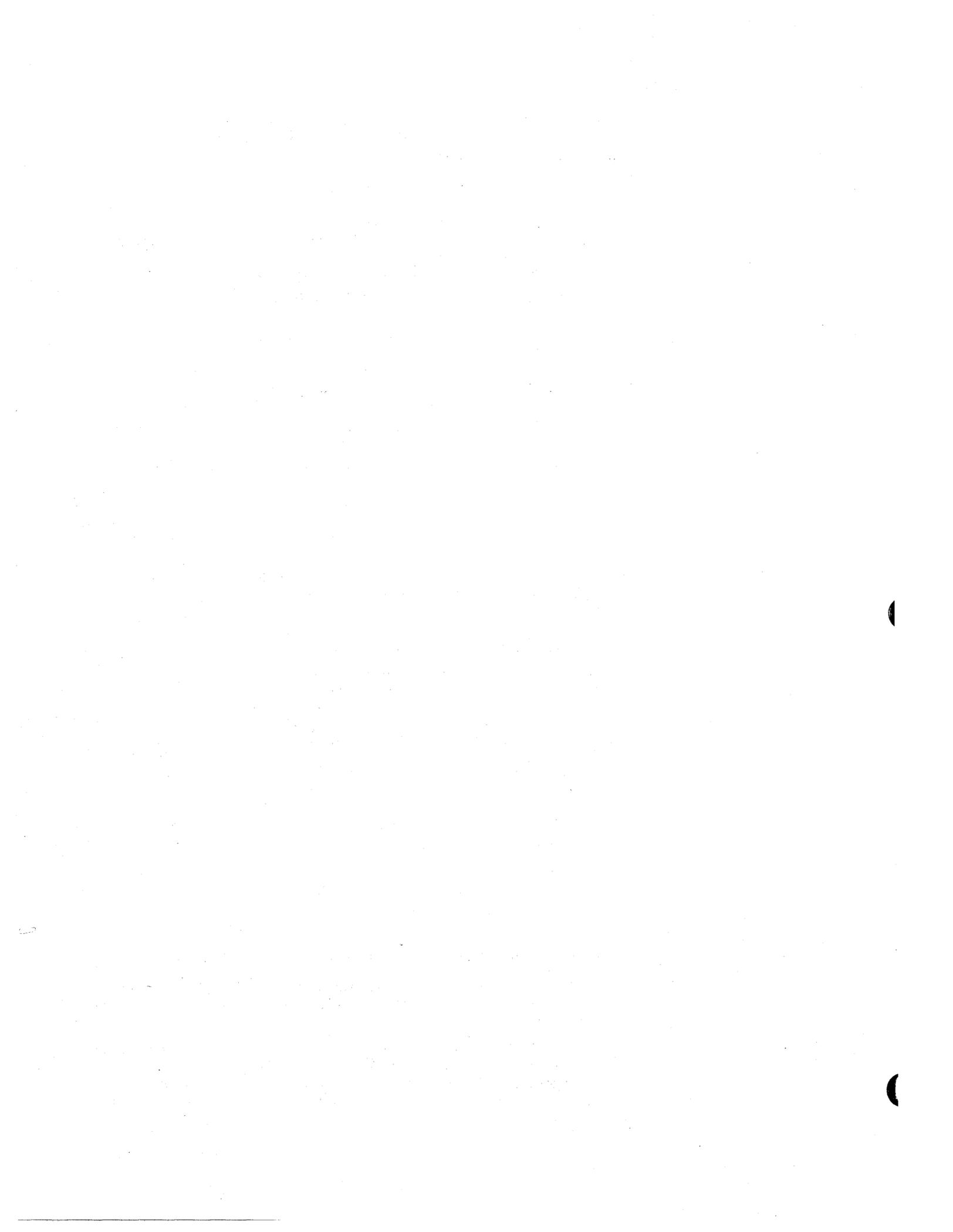
One approach is to use separate dummy device addresses for each partition. This, however, can create problems when a partition is started without POWER/VS. An alternative is to assign SYSRDR in each POWER/VS supported partition to a disk extent containing appropriate DOS/VS JCL assignments. This allows several partitions to be pre-assigned in the supervisor, minimizes the requirement for dummy devices, reduces operator input, and allows the user to create or change partition-dependent start-up job streams without altering the supervisor. The program and job stream to create such files are shown in "Appendix I".

The DOS/VS PRTY operator command can be executed at any time, as long as it does not cause POWER/VS to have lower priority than any of its supported partitions.

The set of classes handled by a partition can be changed by stopping the partition and then starting it again with a different set of classes. Also, the same classes could be specified in a different order of priority.

Special consideration should be given to *interactive* data communications systems:

- If POWER/VS is terminated for any reason, all spooled partitions are also terminated. Therefore, a program such as CICS/VS running under POWER/VS would be abnormally terminated.
- Under POWER/VS, the performance of interactive teleprocessing programs might be affected. For example, best response time is obtained by CICS/VS when it has the highest priority. Under control of POWER/VS, it could not have the highest priority. Performance would be more adversely affected on the smaller System/370 models.



# JOB ENTRY CONTROL LANGUAGE (JECL)

\* \$ \$ CTL

JECL is used to specify POWER/VS job-processing options. JECL statements do *not* override or replace DOS/VS JCL statements in any way.

## FUNCTIONS OF JECL STATEMENTS

The following table shows the JECL statements and their functions:

JECL Statement	Function
* \$ \$ CTL	Specifies a default input class
* \$ \$ JOB	Indicates the beginning of a POWER/VS job and provides handling information
* \$ \$ EOJ	Indicates the end of a POWER/VS job
* \$ \$ RDR (Note 1)	Inserts a diskette file into the input stream
* \$ \$ LST	Provides handling information for printed output
* \$ \$ PRT	Provides handling information for punched output
* \$ \$ PUN	Provides handling information for punched output
* \$ \$ SLI	Inserts data from a sublibrary into the job stream
* \$ \$ /*	Used in a source statement library book to indicate the end of a DOS/VS job step (used for the SLI statement only)
* \$ \$ /&	Used in a source statement library book to indicate the end of a DOS/VS job (used for the SLI statement only)
* \$ \$ DATA	Inserts data into a book retrieved from a source statement library.

Note 1. Not supported in the component release of POWER/VS.

When POWER/VS is in operation, JECL may or may not be used to delimit jobs. If JECL is used to delimit jobs, job entries are contained between the POWER/VS \* \$ \$ JOB and \* \$ \$ EOJ cards. An \* \$ \$ JOB card encountered in the input stream without a preceding \* \$ \$ EOJ card is used as a delimiter for the previous POWER/VS job. If JECL is *not* used to delimit jobs, job entries are contained between the DOS/VS // JOB and /& cards. (If /& is missing, POWER/VS creates one when the next // JOB card is encountered). The POWER/VS job name is then copied from the DOS/VS job name. Other JECL statements can still be included.

Since JECL statements begin with an asterisk-blank, job control treats them as comments when POWER/VS is not in control.

The CTL, JOB, and EOJ statements should be presented at DOS/VS job control time. When in the middle of a user data stream, they are treated as POWER/VS control statements. The RDR, LST, PUN, SLI, and DATA statements can be included anywhere in the input stream. More than one LST or PUN statement is allowed per POWER/VS job.

JECL must always be used in a writer-only partition, but is used with these restrictions:

- RDR, CTL, SLI, and DATA statements are treated as comments.
- LST and PUN statements are recognized at job control time, or when user generated and written to SYSLOG.
- A number of JOB statement parameters are not meaningful and are ignored.

JECL statements will be considered data if read by a writer-only partition. This allows source statement books containing JECL to be catalogued.

## FORMATS OF JECL STATEMENTS

Each JECL statement consists of the following fields. Refer to Figure 10.

1. **Identification field.** Contains the characters \* \$\$ (asterisk-blank-dollar-dollar) in columns 1 through 4.
2. **Operation field.** Specifies the JECL operation. It can directly follow the second dollar sign or be separated from the second dollar sign by one or more blanks. At least one blank must be placed between the operation field and the operand field.
3. **Operand field.** Contains one or more keyword or positional parameters, separated by commas. Keyword and positional parameters *cannot* be mixed within one statement.

**Note.** *Some POWER/VS features are supported only through keyword notation.*

Positional parameters must be specified in a fixed order. If a positional parameter is omitted, its delimiting comma must be coded. If trailing parameters are omitted, trailing commas can also be omitted. If no parameters are specified, the entire operand field can be omitted. In the case where a comment field is present and the entire operand field is blank, the omitted operand field is indicated by a comma, which is preceded and followed by at least one blank. If the operand field cannot be contained on one card, a continuation card can be produced according to the following rules:

- The operand field can be broken after a comma that separates two parameters, and can be continued on the next card starting anywhere between columns 6 and 16 (inclusive). Column 72 of the first card must contain a non-blank character. This process can be repeated for as many cards as are necessary.
- Columns 1 to 4 of each continuation statement must contain \* \$\$ (asterisk-blank-dollar-dollar).
- Column 72 of a non-continued (single) statement or a last continuation statement must be blank.

**Note.** *In a writer-only partition, continuation of a \* \$\$ JOB card is not recognized.*

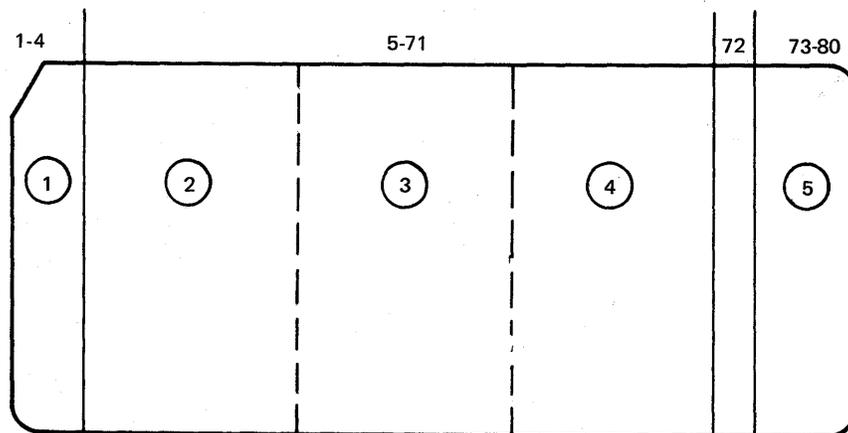


Figure 10. Format of JECL statements

4. **Comments field.** Can contain any information considered helpful by the user. The comment field can be broken anywhere and continued on the next statement, in any column, subject to the same continuation rules given for the operand field.
5. **Sequence field.** Contains up to eight characters of optional information used for control statement identification. If present, the sequence field is positionally dependent and must be coded starting in column 73.

Underlined and capitalized parameter names and values that appear in the individual JECL statement formats indicate defaults that apply when the parameters are not specified.

#### CTL Statement

The CTL statement establishes the default *input* class that is assigned to all jobs whose input class was not assigned in a JOB statement. A CTL specification overrides an input class specification that may have been made through a PSTART command.

The CTL statement can be placed at any POWER/VS job boundary in the input stream, and becomes effective immediately. It remains in effect until the next CTL statement is read or until the read task is stopped with a PSTOP command.

```
* $$ CTL      CLASS= { A }           (keyword form only)
                   class }
```

**class.** Specifies the input class to which any subsequent jobs are assigned if they do not contain a class specification in their JECL JOB statement. Class can be specified as an alphameric character from A-Z or 0-4. For a more detailed discussion of class, refer to the "POWER/VS Concepts and Facilities".

#### Examples:

- \* \$\$ CTL CLASS=3            Jobs without a class specification in the JECL JOB card are executed in partition F3 (partition-dependent).
- \* \$\$ CTL CLASS=B           Jobs without a class specification in the JECL JOB card are executed in any partition that has class B assigned to it (partition-independent).

#### JOB Statement

The \* \$\$ JOB statement can be placed anywhere in the DOS/VS job stream to specify POWER/VS job attributes and to begin a POWER/VS job. The \* \$\$ EOJ statement is normally used to end a POWER/VS job. However, if the EOJ statement is not used, the \* \$\$ JOB statement of the next job delimits this job. If the JOB statement is omitted, POWER/VS jobs are delimited by the DOS/VS // JOB and /& statements, and the POWER/VS job name is copied from the DOS/VS job name.

**Note.** *It is allowable to have multiple DOS/VS jobs within a POWER/VS job, and vice versa.*

```
* $$ JOB      [ JNM= { AUTONAME } ]           (keyword form)
                jobname }
                [ ,DISP= { D } ]
                disposition }
                [ ,PRI=priority ]
                [ ,CLASS=class ]
                [ ,USER=user-information ]
```

SEE  
PAGE 29

```
* $$ JOB [AUTONAME] (positional form)
          [jobname]
          , [D]
          [disposition]
          , [priority]
          , [class ]
```

**Note.** User information cannot be specified with the positional form.

#### Operands

**jobname.** Specifies the name by which the job is known to POWER/VS. If the job contains several DOS/VS jobs, they are all logged in POWER/VS by the job name in the JECL JOB statement. The default name, AUTONAME, is assigned if jobname is not specified.

A JECL jobname specification can be from 1 to 8 alphameric characters, beginning with an alphabetic character. However, use of the job names ALL, HOLD, FREE, RJE, LOCAL, or one-character job names should be avoided as they may conflict with some POWER/VS operator commands.

**disposition.** Specifies how POWER/VS will route and schedule the associated entry in the read queue.

A read entry must have one of the following disposition attributes:

- D Dispatch and delete. The job is automatically scheduled by POWER/VS in accordance with the class and priority attributes. On completion of job execution, the read entry is deleted from the read queue and data file space is released. D is the default value if no disposition is specified.
- H Hold job. The entry remains in the read queue; it is not dispatched by POWER/VS until the disposition attribute is changed to D or K through the PALTER or PRELEASE command.
- K Dispatch and keep. The entry is automatically scheduled by POWER/VS in accordance with the class and priority attributes. On completion of job execution, the read entry is *not* deleted from the read queue, and the disposition of the entry becomes L.
- L Leave in queue. The entry remains in the read queue; it is not executed by POWER/VS until the operator specifically changes the disposition attribute to D or K by using the PALTER command.

Input specifications for disposition, priority, and class are ignored in writer-only partitions.

**priority.** Specifies the priority assigned to this job in both input and output queues. It is specified as a single digit from 0 to 9. Nine is the highest priority. If the priority parameter is not specified, POWER/VS assigns the default priority defined in the PRI parameter of the POWER macro. Jobs are dispatched on a first-in, first-out basis within priority, within class.

**class.** Assigns a class to the job entry in the input queue. It can be specified as an alphameric character from A-Z or 0-4. This parameter can be used to group jobs that require the same resources for execution.

Classes 0 through 4 have a specific meaning and are partition dependent classes; jobs with class 0 can only be selected for execution in BG, jobs with class 1 can only be selected for execution in F1, and so on. Classes A through Z are partition-independent classes; such jobs can be executed in any partition with a matching class. When the JOB statement is coded in its positional form, class 0 may be coded as BG, and class 1 to class 4 may be coded as F1 to F4.

If class is not specified, the input class specified in the CTL statement applies; if no CTL statement is in effect, the class is established by the PSTART command. If no class is specified at all, default class A is assigned.

**user-information.** Up to 16 bytes of user information can be specified. This information is printed on SYSLOG when the execution read task encounters an \* \$\$ JOB statement (if JLOG=YES). It also appears in the POWER/VS account records and in separator pages. User information must be specified within quotes.

*Examples:*

- |                                                                                 |                                                                                                                                                                                  |
|---------------------------------------------------------------------------------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| * \$\$ JOB JNM=PAYROLL,PRI=0<br><i>or</i><br>* \$\$ JOB PAYROLL,,0              | Job named PAYROLL is given the lowest priority in its class. Disposition defaults to D. Class defaults to the CTL statement or PSTART command specification.                     |
| * \$\$ JOB JNM=TEST,DISP=H<br><i>or</i><br>* \$\$ JOB TEST,H                    | Job named TEST is placed in the hold state. Priority is that specified at POWER/VS generation. Class defaults to the CTL statement to the PSTART command specification, or to A. |
| * \$\$ JOB JNM=ASSEMBLY,PRI=2,CLASS=2<br><i>or</i><br>* \$\$ JOB ASSEMBLY,,2,F2 | Job named ASSEMBLY is given a priority of 2 within input class 2 (partition F2). Disposition defaults to D.                                                                      |
| * \$\$ JOB , comment                                                            | Job takes default name, AUTONAME. Note that a "blank-comma-blank" preceeds the comment field.                                                                                    |

**EOJ Statement**

The EOJ statement marks the end of the POWER/VS job and, if used, is normally placed at the end of a DOS/VS job. However, if this statement is presented at any other time than DOS/VS job control time, it is still acted upon by POWER/VS.

\* \$\$ EOJ

**RDR Statement**

The RDR statement is used to insert a 3540 diskette file into input being read from a card reader. It is also possible to read both data and JCL from the 3540 and then to spool the input, or to immediately process the input in a writer-only partition.

**Note.** *The 3540 is not supported in the component release of POWER/VS.*

**Case 1: Reading Exclusively From The 3540**

To read exclusively from the 3540, issue the following format of the PSTART command (see "POWER/VS Operator Command Language"):

PSTART RDR,3540addr,'filename'[,maxno] [,S]

JCL and data are both on one 3540 SYSIN file. Note that a RDR statement is *not* used.

## Case 2: Merging 3540 Data File With Card Reader Input

To merge a 3540 data file with input from a card reader, issue the following format of the PSTART command:

```
PSTART RDR,uraddr,3540addr
```

This initiates the read task; input is entered from the card reader. When a RDR statement is encountered (which might be the first card), input is continued from the 3540. At the end of the 3540 file, input is again accepted from the card reader.

```
* $$ RDR      [LUN=logical-unit-number]      (keyword form)
               [,FID='file-id']
               [,NOD= { 1
                       number-of-diskettes } ]
               [,VSC=S]
```

```
* $$ RDR      [logical-unit-number]          (positional form)
               , [file-id']
               , [ 1
                 number-of-diskettes ]
               , [S]
```

**logical-unit-number.** Specifies the programmer logical unit number (SYSnnn) used by the user program to read the input data file. SYSnnn must *not* be specified for SYSIN files.

**'file-id'.** Specifies the file name (in quotes) that appears in the HDR1 label on the 3540 diskette. One to eight alphanumeric characters are specified. If this parameter is omitted, POWER/VS reads the file specified by the first non-secured HDR1 label found on the diskette that is currently mounted in the 3540. Record size is acquired from the HDR1 label.

**number-of-diskettes.** Specifies the maximum number of diskettes to be read. One to three digits can be specified. Allowed values are 1 to 255. The default value is 1. Reading stops after the specified number of diskettes are processed, or after reading a diskette whose HDR1 label indicated that it was the last volume of the file.

**S.** Specifies volume sequence checking. If omitted, no sequence checking takes place. Sequence numbers must start with 1 and must be incremented by 1.

### Example:

With JCL in the card reader, and data on the 3540, the operator enters a PSTART command for card reader X'00C':

```
PSTART RDR,00C,00B
```

This command tells POWER/VS to start a read task at address X'00C' with the ability to read also from a 3540 device at address X'00B'. Both devices belong to the read task and cannot be used by any other partition or POWER/VS task until the read task terminates.

The Following cards are in X'00C':

```
* $$ JOB JNM=EX1,PRI=9
// JOB EX1
// ASSGN SYS008,X'00B'
// DLBL FILE,'FILE-ID',,DU
// EXTENT SYS008
// EXEC PROG
* $$ RDR SYS008,'FILE-ID',2
/*
/&
* $$ EOJ
```

**Note.** The SYS008 specification in the \* \$\$ RDR statement would not be required if the symbolic unit was assembled into the DTFDU.

The RDR statement causes the read task to suspend card reading and to read a maximum of 2 diskettes of the data file named FILE-ID. Records on the 3540 can be from 1 to 128 bytes long and are not examined for JCL or JECL control statements by either the read task or the execution processor. When the end of the FILE-ID file is reached, card reading resumes.

During execution, the user program might not read all of the FILE-ID records spooled by the read task. To prevent the remainder of the records from being passed to job control, any request to the SYSIN device causes POWER/VS to skip records until the end of the FILE-ID file.

#### **LST or PRT Statement**

The LST (or PRT) statement defines the attributes of the intercepted list output of a POWER/VS job.

Multiple LST statements can occur within a POWER/VS job. They can define multiple printers, real and dummy, and they can define and redefine the attributes of the output of the job. When LST statements appear within the spooled input data for a user program they, in effect, "segment" the output for that program.

The specifications of a particular list statement remain in effect until a subsequent LST statement with the same LST parameter (same printer) is encountered, or until the end of the POWER/VS job. Each \* \$\$ LST statement creates a separate list entry.

If a job does not include a LST statement, default values established during POWER/VS generation are assumed. LST statements do *not* require that the \* \$\$ JOB and \* \$\$ EOJ statements also be used.

LST and PRT are equivalent operation codes. LST is used throughout this manual.

\* \$\$ {LST } [ DISP= { D } ] (keyword form)  
 {PRT } [ ,CLASS= { A } ]  
 [ ,REMOTE = remid ]  
 [ ,FNO= { } ]  
 [ ,JSEP=sep ]  
 [ ,COPY= { 1 } ]  
 [ ,RBM=(norbm1,norbm2) ]  
 [ ,LTAB=linetab ]  
 [ ,RBS=norbs ]  
 [ ,UCS=(phasename [,option]) ]  
 [ ,FCB=phasename ]  
 [ ,LST=listaddr ]

\* \$\$ {LST } [ D ] [ A ] (positional form)  
 {PRT } [ disposition [ class ] ]  
 [ forms-number ]  
 [ 1 ]  
 [ number-of-copies ]  
 [ (tapeaddr [,X'ss']) ]  
 [,norbm1 ]  
 [,linetab ]

### Operands

**disposition.** Specifies the disposition of the printed output after execution of the job. The disposition specification options are listed below:

- D Process and delete. The list entry is automatically written to a printer in accordance with its class and priority attributes.
- H Hold. The list entry remains in the list queue; it is not written to a printer by POWER/VS until the disposition attribute is changed to D or K by means of the PALTER or PRELEASE command.
- K Process and keep. The list entry is automatically written to a printer in accordance with its class and priority attributes. When printing is completed, the list entry is *not* deleted from the list queue, and the disposition of the entry becomes L.
- L Leave in queue. The list entry remains in the list queue; it is not written to a printer until the operator changes the disposition attribute to D or K using the PALTER or PRELEASE command.
- N Output without spooling. If N is specified, print requests are not intercepted. Output is written directly, without spooling.
- T Spool to tape. The output is spooled to tape intermediate storage. PSTART is used to initiate printing.

If the disposition specification is omitted, D is assumed.

If the N option is specified and the device is not available at execution time, a message is issued and list output is spooled.

D, H, K, I and L are only valid when the output is spooled to disk.

When the output is to be printed at a terminal, N and T are invalid specifications.

**Note.** *There is no comma between disposition and class in the positional form of the LST statement.*

**class.** Specifies the class of the job's printed output. It can be specified as any alphabetic character. The partition-dependent input classes, 0 through 4 cannot be specified for output.

Class can be used in conjunction with "forms-number" to group like types of printed output. A list task can be started by class to process the output. Class is only meaningful when the output is spooled to disk. If class is specified and the positional form is used, disposition must also be specified.

If the class operand is omitted, class A is assigned.

**remid.** Specifies the "to" remote-id to which list output is to be routed. Each list entry also has a "from" remote-id, which identifies the user who submitted the job initially.

"remid" can be a number from 0 to 200. Numbers 1 to 200 represent specific remote users. If remid is 0, list output is directed to the central installation.

If this parameter is omitted or if the positional form of the \* \$\$ LST statement (which does not support this parameter) is used, list output is routed to the central location if the job was submitted locally, or is routed to the LSTROUT specification in the PRMT macro if the job was submitted remotely.

**forms-number.** Specifies the type of forms to be used. "forms-number" is one to four alphanumeric characters long. If forms-number is not specified, it is assumed to be four blanks, representing the installation's standard forms. When a list task prepares to process output from a list entry, it determines whether this forms-number matches that of the previous list entry. If the numbers match, the list task processes the output. If the numbers do *not* match, message 1Q40A is issued.

**sep.** Specifies the number of job separator pages (0 to 9). If not specified, the value defined during POWER/VS generation is assumed.

**number-of-copies.** Specifies the number of copies to be printed.

"number-of-copies" is one or two digits long. If not specified, 1 is assumed. If 0 or 00 is specified, one copy is produced. This parameter is ignored when the output is not spooled to disk. If additional copies are required when disposition is T, the tape must be reprinted.

**tapeaddr.** Specifies the hardware address of the tape drive to be used for spooling when the job is being executed. Its format is one of the following:

```
cuu
X'cuu'
cuu,X'ss'
X'cuu',X'ss'
```

X'ss' specifies the density of the tape, where "ss" is a standard DOS/VS code. (The X'ss' parameter can only be specified in the keyword form of LST).

The tapeaddr parameter forces a disposition of T. If (DISP=T) is specified and tapeaddr is omitted, the operator will be asked to specify the tape address at job execution time.

Multiple files can be written to a single tape. Multivolume files are also supported. No label checking is performed.

**norbm1.** Specifies the number of list records that must be intercepted before message 1Q52I is sent to the operator. This parameter is one to six digits long. If norbm1 is omitted, the value defined during POWER/VS generation is assumed.

**norbm1.** Specifies the number of list records that must be intercepted before message 1Q52I is sent to the operator. This parameter is one to six digits long. If norbm1 is omitted, the value defined during POWER/VS generation is assumed.

**norbm2.** Specifies the additional number of list records intercepted each time before message 1Q52I is sent out again. The number is one to six digits long. If norbm2 is omitted, the preceding comma as well as the parentheses can be omitted, and the value defined during POWER/VS generation is assumed. This value is also assumed when the positional form of the LST statement is used.

**linetab.** Specifies the carriage control tape format to be used when print requests are intercepted. This parameter is 26 digits long; each subparameter is two digits. If linetab is not specified, the LTAB parameter defined during POWER/VS generation is used. The format of linetab is:

d0,d1,d2, ... d12

d0 is the distance between the channel 12 and the first punch; d1 is the distance between the *first* punch and the channel 1 punch; d2 is the distance between the *first* punch and the channel 2 punch; and so on. If channel 1 is the first punch, distance d1 is expressed as 00. If a channel is not punched, 00 must be specified for that channel. See "Examples".

Any linetab specification is ignored if an FCB phase name is also present in the \* \$\$ LST statement. An LFCB macro issued in the problem program updates the LTAB format.

**Note.** *It is advisable to use the forms number parameter in conjunction with the linetab parameter. This causes the list task to pause, thus allowing the operator to mount the correct carriage control tape on the printer.*

**norbs.** Specifies the number of list pages intercepted before the output is segmented. This number overrides the POWER/VS generation specification of the RBS parameter of the POWER macro.

This parameter can be one to six digits long. A zero means that no segmentation will take place.

Each LST statement, besides possibly specifying a different norbs value, resets record count to zero; that is, a new segment is started. A new count is also started when an LFCB macro instruction is issued. "norbs" cannot be specified in the positional form of LST, and is valid only when spooling to disk.

**phasename.** Specifies the name of the phase under which an FCB or UCS buffer (UCB) image is cataloged in the core image library. An optional subparameter can be specified along with the UCB phasename. The format of the parameter then becomes (phasename,option), where "option" can be specified as F, C, CF, or FC.

F Signifies that the UCB is to be loaded with the folding operation code to permit printing of uppercase for lowercase bit configurations.

C Prevents data checks from being generated because of print line mismatches with the UCB.

The specified core image library phase is loaded into the FCB or UCB when a list task starts processing the output related to the LST statement.

When a LST statement with an FCB phase name specification is encountered by the execution processor, the specified buffer image is read to update the internal LTAB representation to reflect the new buffer. Any LTAB specification in the LST statement is therefore ignored if FCB is also specified. The FCB specification for a non-FCB printer is accepted, but is only used to update the LTAB.

This parameter is not supported by the positional form of LST.

**listaddr.** Specifies either the logical or physical address of the output printer for the list entry being created. The address can be either in the form SYSxxx, where xxx is LST or any valid programmer logical unit; or in the form cuu (or X'cuu'), which is the channel and unit number of an actual or a dummy printer. (For a discussion of dummy printers, refer to "Dummy Printers and Punches" in the "POWER/VS Concepts and Facilities").

If this parameter is omitted, the default is the first printer address specified at partition start-up.

This parameter is needed when multiple printers are used to print multiple reports produced by a single program. This parameter is not supported by the positional form of the LST statement.

*Examples:*

In this job, printed output is spooled to disk and requires the installation's special form number 32A. One copy is requested, and the STDLINE value is used. The linetab parameter defines a carriage control tape for a page of 66 lines. Channel 3 is punched four lines below channel 1; no channels other than 1, 3, and 12 are used.

\* \$\$ LST FNO=32A,LTAB=100000040000000000000000056

*or*

\* \$\$ LST ,32A,,,100000040000000000000000056

\* \$\$ LST DISP=N

*or*

\* \$\$ LST N

Print requests are not intercepted, but are sent directly to the physical device.

\* \$\$ LST DISP=T,TADDR=283

*or*

\* \$\$ LST T,,283

Printed output is spooled to tape unit 283.

\* \$\$ LST CLASS=A,FNO=32A,RBS=100

Printed output is designated class A. Special form number 32A is required. Printed output is segmented after every 100 pages.

## PUN Statement

The PUN statement defines the attributes of the intercepted punch output of a POWER/VS job.

Multiple PUN statements can occur within a POWER/VS job. They can define multiple punches, dummy and real, and they can define and redefine the attributes of the output of the job. When PUN statements appear within the spooled input data for a user program they, in effect, "segment" the output for that program.

The specifications of a particular PUN statement remain in effect either until a subsequent PUN statement for the same punch is encountered or until the end of the POWER/VS job. *Each \* \$\$ PUN statement creates a separate punch entry.* If a job does not include a PUN statement, default values established during POWER/VS generation are assumed.

PUN statements can appear anywhere in the POWER/VS job, including in user program spooled input data. PUN statements do not require that \* \$\$ JOB and \* \$\$ EOJ statements also be used.

```

* $$ PUN [ DISP= { D } ] (keyword form)
           [ disposition ]
           [ ,CLASS= { A } ]
           [ class ]
           [,REMOTE=remid]
           [ ,FNO= { bbbb } ]
           [ forms-number ]
           [,JSEP=sep]
           [ ,COPY= { 1 } ]
           [ number-of-copies ]
           [,TADDR=(tapeaddr[,X'ss'])]
           [,RBM=(norbm1,norbm2)]
           [,PUN=punaddr]
           [,RBS=norbs]

```

```

* $$ PUN [ D ] [ A ] (positional form)
          disposition [ class ]
          [ bbbb ]
          forms-number
          [ 1 ]
          (number-of-copies)
          (tapeaddr[,X'ss'])
          [,norbm1]

```

### Operands

**disposition.** Specifies the disposition of the punched output after execution of the job. The disposition specification options are listed below:

- D Process and delete. The punch entry is automatically written to a punch in accordance with its class and priority attributes.
- H Hold. The punch entry remains in the punch queue; it is not written to a punch by POWER/VS until the disposition attribute is changed to D or K by means of the PALTER and PRELEASE command.
- I Return output to input queue. In some situations it may be useful to return the punch output of a POWER/VS job directly to the input queue. For example, in situations where the assembler normally stores its output (including JCL) on tape for subsequent input to a catalog run, it can, using POWER/VS, store output directly into the input queue. The input entry created has the same jobname, remote-id, and priority as the entry producing the output. Class becomes that specified in the PUN statement; disposition becomes D.

Specification of the I option suppresses count-driven output segmentation. Also, the multiple copies specification is ignored.

This facility should only be used for jobs producing punch output in executable format.

- K Process and keep. The output is automatically written to a punch in accordance with its class and priority attributes. When punching is completed, the punch entry is *not* deleted from the punch queue, and the disposition of the entry becomes L.
- L Leave in queue. The job remains in the punch queue; it is not written to a punch until the operator changes the disposition attribute to D or K by means of the PALTER command.

**N** No spooling. If N is specified, punch requests are not intercepted. It allows direct punch writer capabilities without spooling.

**T** Tape spooling. The output is spooled to tape.

If the disposition specification is omitted, D is assumed.

If the N option is specified, and the device is not available at execution time, message 1Q46I is issued and punch output is spooled.

D, H, K and L are only valid when the output is spooled to disk.

When the output is to be punched at a terminal, N and T are invalid specifications.

**Note.** *There is no comma between disposition and class in the positional form of the PUN statement.*

**class.** Specifies the class of the punch output of the job. It can be specified as any alphabetic character. The partition-dependent input classes, 0 through 4, cannot be specified for output. Class can be used in conjunction with forms-number to group like types of punch output. Punch tasks can be started by class to process the output. Class is only meaningful when the output is spooled to disk. If class is specified and the positional form is used, disposition must also be specified. If the class operand is omitted, class A is assigned.

**remid.** Specifies the remote-id to which punch output is to be routed ("to" remote-id). Each punch entry also has a "from" remote-id, which identifies the user who submitted the job.

"remid" can be a number from 0 to 200. Numbers 1 to 200 are specific remote users. If remid is 0, punch output is directed to the central installation.

If this parameter is omitted or if the positional form of the \* \$\$ PUN statement is used, punch output is routed to the central location for locally submitted job entries or is routed according to the PUNROUT specification in the PRMT macro for job entries submitted remotely.

**forms-number.** Specifies the type of card forms to be used. "forms-number" is one to four alphameric characters long. If forms-number is not specified, it is assumed to be four blanks, representing the installation's standard card forms. When a punch task prepares to process output from a punch entry, it determines whether this forms-number matches that of the previous punch entry. If the numbers match, the punch task processes the output. If the numbers do *not* match, message 1Q40A is issued.

**sep.** Specifies the number of job separator cards (0 to 9). If not specified, the value defined during POWER/VS generation is assumed. The positional form of the PUN statement does not include this parameter.

**number-of-copies.** Specifies the number of copies to be punched from the punch entry. "number-of-copies" is one or two digits long. If not specified, 1 is assumed. If 0 or 00 is specified, one copy is produced. This parameter is ignored when the output is not spooled to disk. If additional copies are required when disposition is T, the tape must be repunched.

**tapeaddr.** Specifies the hardware address of the tape drive to be used for spooling when the job is executed. Its format is one of the following:

```
cuu
X'cuu'
cuu,X'ss'
X'cuu',X'ss'
```

X'ss' specifies the density of the tape, where "ss" is a standard DOS/VS code. (The X'ss' parameter can only be specified in the keyword form of PUN.)

The tapeaddr parameter forces T if invalidly specified. If DISP=T and tapeaddr is omitted, the operator is asked to specify the tape address at job execution time.

Multiple files can be written to a single tape. Multivolume files are also supported. No label checking is performed.

**norbm1.** Specifies the number of punch records that must be intercepted before message 1Q52I is sent to the operator. This parameter is one to six digits long. If norbm1 is omitted, the value defined during POWER/VS generation is assumed.

**norbm2.** Specifies the additional number of punch records intercepted each time before message 1Q52I is sent out again. The number is one to six digits long. If norbm2 is omitted, the preceding comma as well as the parentheses can be omitted, and the value defined during POWER/VS generation is assumed. This value is also assumed if the positional form of the PUN statement is used.

**norbs.** Specifies the number of punch records intercepted before the output is segmented. This number overrides the POWER/VS generation specification in the RBS parameter of the POWER macro.

This parameter can be one to six digits long. A zero means that no segmentation will take place.

Each PUN statement, besides possibly specifying a different norbs value, resets record count to zero; that is, a new segment is started.

"norbs" cannot be specified in the positional form of PUN.

**punaddr.** Specifies either the logical or physical address of the output punch for the punch entry being created. The address can be either in the form SYSxxx where xxx is PCH or any valid programmer logical unit; or in the form cuu (or X'cuu'), which is the channel and unit number of an actual or a dummy punch. (For a discussion of dummy punches, refer to "Dummy Printers and Punches" in the "POWER/VS Concepts and Facilities"). If this parameter is omitted, the default is the first punch address specified at partition start-up.

This parameter is needed when multiple punches are used to punch multiple card files produced by a single program.

**Note.** *The address 01D should not be used for a dummy punch on the System/370 Models 115 and 125.*

*Examples:*

* \$\$ PUN FNO=6509 or	Punch output requires the special card forms "6509". One copy is requested.
* \$\$ PUN ,6509	
* \$\$ PUN DISP=N or	Punch requests are not intercepted, but are sent directly to the punch unit.
* \$\$ PUN N	
* \$\$ PUN DISP=T,TADDR=285 or	Punch output is spooled to tape unit 285.
* \$\$ PUN T,,285	

## SLI Statement

At execution time, the \* \$\$ SLI (source library inclusion) statement causes the private and system source statement libraries to be searched for a book, which, when found, is inserted into the job stream. The private library, if one is assigned to the POWER/VS partition, is searched first.

An SLI statement does not require that \* \$\$ JOB and \* \$\$ EOJ statements also be used.

\* \$\$ SLI [sublib.]bookname

**sublib.** Specifies the name of the sublibrary to be searched. Allowed sublibrary names are A through Z, 0 through 9, \$, #, or @. Sublibraries A, C, D, E, F, and Z are already defined as special purpose libraries in DOS/VS. This parameter is optional. If omitted, the SUBLIB specification at POWER/VS generation is used.

**bookname.** Specifies the book name to be searched for and inserted into the job stream. Books can be cataloged in the source statement library (SSL) by using the DOS/VS program "MAINT". To include /\* and /& statements in SSL books, the statements must be replaced by \* \$\$/\* and \* \$\$/& delimiter statements. See the examples that follow the heading "DATA Statement".

### SL Update Statement

In conjunction with *SLI* statements, *SL update* statements are used to modify source library book images placed in the input stream. (Note that these books are not modified in the library.) SL (source library) update statements follow an \* \$\$ SLI statement (not necessary immediately), and are identified by \$SLI in columns 73 - 76. Specifically, they can be used to add, delete, and replace statements in the book referenced by the accompanying SLI statement. Columns 78 - 80 contain a sequence number. Column 77 contains one of the following codes:

- A        Insert *after* book statement with same sequence number.
- B        Insert *before* book statement with same sequence number.
- D        Delete the book statement with same sequence number.
- other    Replace the book statement that has same sequence number, or insert this statement.

Columns 1 - 72 contain the statement being added. (For a deletion, columns 1 - 72 are ignored.)

SL update statements and \* \$\$ DATA statements, when one or both are used, must follow \* \$\$ SLI in the same sequence as the statements in the book that they correspond to. See the examples that follow the heading "DATA Statement".

### DATA Statement

The \* \$\$ DATA statement allows the user to insert a deck of cards from the card reader into a book that is retrieved from the source statement library by an SLI statement. At execution time, the DATA statement in the job stream is matched by name to a DATA statement in the source book. At that point in the source book, the cards (read queue data records) that follow the DATA statement, up to and including /\* or /&, are inserted. If names do not match, or if no DATA statement is present in the read queue, the DATA statement in the source book is ignored.

DATA statements must be preceded by an SLI statement; otherwise they are treated as comments. SL update statements (\$SLI in columns 73 - 76) and DATA statements can be used together. However, they must appear in the job stream in the same order as the corresponding images appear in the source book.

The DATA statement is ignored in a writer-only partition.

\* \$\$ DATA name

**name.** Specifies the name of the corresponding DATA statement in the source statement library book where data is to be inserted. "name" can be from one to eight alphameric characters. The first character must be alphabetic.

*Examples:*

1. Job to insert data from the card reader into the book "ASSM".

Job from card reader	Source book "ASSM"
* \$\$ JOB INSERT	BKEND ASSM
* \$\$ SLI ASSM	// JOB INSERT
* \$\$ DATA INPUTA	// EXEC ASSEMBLY
POWER ACCOUNT=YES	* \$\$ DATA INPUTA
END	* \$\$/&
/*	BKEND
* \$\$ EOJ	

**Resulting job stream**

// JOB INSERT	(From book "ASSM")
// EXEC ASSEMBLY	(From book "ASSM")
POWER ACCOUNT=YES	(From card reader)
END	(From card reader)
/*	(From card reader)
/&	(From book "ASSM")

2. Assume that a private source statement library is assigned to the POWER/VS partition. A job (read in from the card reader) references the book "XMPL" in that library (sublibrary B) and contains SL update statements as well as DATA statements. The book statements replace the SLI statement. However, statements XMPL0020, XMPL0100, and XMPL0120 are overridden by the corresponding SL update statements in the input stream. In addition, statement \$SLIA100 is inserted. Also, the cards between the DATA and the first /\* statement in the input job stream replace the DATA statement in XMPL. The resulting job stream is placed in the read queue.

**Job from card reader**

* \$\$ JOB ASSEMSLI	
* \$\$ SLI B.XMPL	
// ASSGN SYSPCH,X'182'	Col. 73-80
* \$\$ DATA INPUTA	\$SLI0020
DEFAULT POWER ACCOUNT=YES	
END	
/*	
// ASSGN SYSIPT,X'182'	\$SLI0100
// ASSGN SYS010,X'183'	\$SLIA100
// MTC RUN,X'182'	\$SLI0120
* \$\$ EOJ	

Book "XMPL" in source statement library

Col. 73-80

```

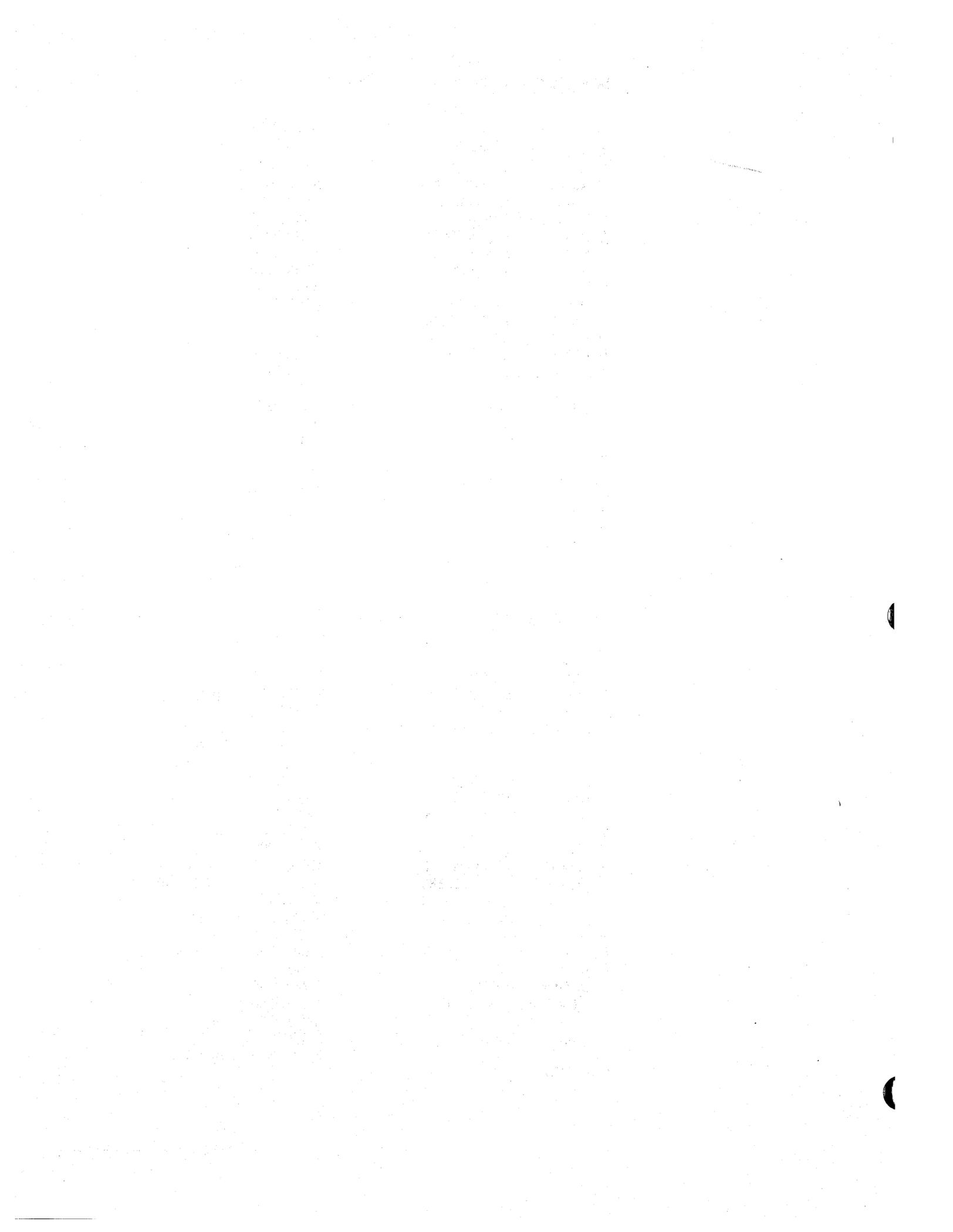
BKEND      B.XMPL
// JOB     ASSEMBLE           XMPL0010
// ASSGN   SYSPCH,X'180'      XMPL0020
// EXEC    ASSEMBLY           XMPL0030
* $$ DATA INPUTA            XMPL0040
// MTC     WTM,SYSPCH         XMPL0050
// MTC     WTM,SYSPCH         XMPL0060
// MTC     REW,SYSPCH         XMPL0070
* $$/&                                           XMPL0080
// JOB     CATALOG            XMPL0090
// ASSGN   SYSIPT,X'180'      XMPL0100
// EXEC    MAINT              XMPL0110
// MTC     RUN,X'180'         XMPL0120
* $$/*                                           XMPL0130
* $$/&                                           XMPL0140
// JOB     LINK               XMPL0150
// OPTION  CATAL              XMPL0160
           INCLUDE DEFAULT      XMPL0180
* $$/*                                           XMPL0200
// EXEC    LNKEDT            XMPL0210
* $$/&                                           XMPL0220
BKEND
    
```

Resulting job stream placed in read queue

Col. 73-80

```

* $$ JOB   ASSEMSLI
// JOB     ASSEMBLE           XMPL0010
// ASSGN   SYSPCH,X'182'      $SLI0020   updated
// EXEC    ASSEMBLY           XMPL0030
DEFAULT POWER ACCOUNT=YES    inserted
END                          inserted
/*                             inserted
// MTC     WTM,SYSPCH         XMPL0050
// MTC     WTM,SYSPCH         XMPL0060
// MTC     REW,SYSPCH         XMPL0070
/&                                           XMPL0080   modified
// JOB     CATALOG            XMPL0090
// ASSGN   SYSIPT,X'182'      $SLI0100   updated
// ASSGN   SYS010,X'183'      $SLIA100   inserted after
// EXEC    MAINT              XMPL0110
// MTC     RUN,X'182'         $SLI0120   updated
/*                             XMPL0130   modified
/&                                           XMPL0140   modified
// JOB     LINK               XMPL0150
// OPTION  CATAL              XMPL0160
           INCLUDE DEFAULT      XMPL0180
/*                             XMPL0200   modified
// EXEC    LNKEDT            XMPL0210
/&                                           XMPL0220   modified
* $$ EOJ
    
```



## POWER/VS OPERATOR COMMAND LANGUAGE (POCL)

The operator communicates with POWER/VS through the attention routine (AR). When the request key is pressed, the following message is displayed on SYSLOG:

AR 1160A READY FOR COMMUNICATIONS

At this point, the operator can enter POWER/VS commands. (For the display operator console on a System/370 Model 115 or Model 125, the request key does not have to be pressed to give the attention routine control; the console is normally ready to accept attention routine commands.)

POWER/VS operator commands include:

- **Task Management commands.** Used to control read/write tasks and execution processors.
- **Queue Management commands.** Used to control the various input/output queues.
- **Miscellaneous commands.** Enable the operator to, for example, align printer forms or save the POWER/VS account file.

The operator commands consist of two fields, the operation field and the operand field. The operand field contains one or more parameters, separated by commas, or contains no parameters at all. The operator commands can be entered in either uppercase or lowercase.

POWER/VS supports abbreviated as well as extended operation codes. All command options (parameters) are valid for both formats.

The following table shows the abbreviated and the extended command codes:

Type	Extended format	Abbreviated format	Function
Task management	PSTART	S	start a task or partition
	PSTOP	P	stop a task or partition
	PGO	G	activate a task or partition
	PEND*		end POWER/VS execution
	PCANCEL	C	cancel a POWER/VS status report
	PFLUSH	F	flush an active job entry
Queue management	PRESTART	T	restart a write task
	PDISPLAY	D	display a job status
	PALTER	A	alter attributes
	PDELETE	L	delete a job entry or a message
Miscellaneous	PRELEASE	R	release a job entry
	PBRDCST	B	transmit a message
	PINQUIRE	I	check terminal status
	PACCOUNT	J	process account file
	PSETUP		print page layout

\* The one-character operation code for PEND, (E) is not supported, since the operator might inadvertently end the execution of POWER/VS.

Both the abbreviated and the extended commands are presented in this manual in the following format:

<i>Operation</i>	<i>Operand</i>
{ com1 }	parm1,parm2,.....parmn
{ com2 }	

"com1" is the extended command code, and "com2" the equivalent single-character code. If an optional parameter is omitted, its delimiting comma must be coded. Trailing commas for omitted parameters are not required, however.

The following parameters are frequently used in commands:

**jobname.** Specifies the name by which the job entry is known to POWER/VS. "jobname" may be two to eight alphanumeric characters. (A one-character jobname would be treated as a class specification when used in some POWER/VS operator commands.) Each job entry is identified by a jobname and a jobnumber, assigned to it by the POWER/VS system. If the possibility of duplicate jobnames exists, the jobnumber parameter should also be specified. If a duplicate jobname exists and jobnumber is not specified, the first job entry found with the specified jobname is affected by the command.

**jobnumber.** Specifies the number assigned to all input and output entries associated with a POWER/VS job. The jobnumber can be specified by one to five digits. The maximum number is 65535. The display command (PDISPLAY) can be used to obtain the correct jobnumber.

**Note.** *POWER/VS assigns internal job subnumbers to segmented output job entries. Any reference to a "chain" of segmented entries affects all entries in the "chain".*

**uraddr.** Specifies the unit record device address associated with a read or write task. Its format is:

cuu or X'cuu'

Where: c = channel address  
uu = unit address

**partition.** Specifies the partition for which the command is to be executed. The possible entries are:

BG, F4, F3, F2, or F1.

**priority.** Specifies the priority of the job entry. Priority is specified as a single digit from 0 to 9. Nine is the highest priority.

**queue.** Specifies the queue for which the command is to be executed; that is:

LST = the list queue  
PUN = the punch queue  
RDR = the read queue

Both LST and PRT are valid as list queue identifiers. LST will be used throughout this manual.

**remid.** Specifies the remote-id to which the command applies. Remote users may be identified by numbers from 1 to 200. A specification of 0 indicates the central location.

**lineaddr.** Specifies the line address associated with an RJE task. Its format is:

cuu or X'cuu'

Where: c = channel address  
uu = unit address

## TASK MANAGEMENT COMMANDS

### PSTART Command

This command starts a POWER/VS task or brings a partition under control of POWER/VS.

The *task* and *RJE* formats are used to start read/write tasks and RJE tasks, respectively. The *partition* format is used to place a partition under POWER/VS control.

POWER/VS requires a programmer logical unit for each unit record device, each spool tape, or each RJE line specified in a PSTART command. If no free LUB is available in the POWER/VS partition, a message is displayed and the command is ignored.

{ PSTART } S	{	task,uraddr,[class] task,uraddr,tapeaddr partition,[class] RJE,lineaddr[,password] RDR,3540addr,'filename'[,maxno] [,S] RDR,uraddr,3540addr	}
-----------------	---	------------------------------------------------------------------------------------------------------------------------------------------------------------	---

*Note. The RDR format is only for the 3540 Diskette Input/Output Unit.*

#### Operands

**task.** Specifies the task to be started as RDR, LST, or PUN.

RDR = read task  
LST = list task  
PUN = punch task

When a read task is started, input entries are built from the input data. When a write task is started, output entries are processed in priority sequence within class.

**uraddr.** Specifies the unit record device address associated with the read/write task. Its format is either:

cuu or X'cuu'

**lineaddr.** Specifies the line address associated with the RJE task. Its format is either:

cuu or X'cuu'

**tapeaddr.** Specifies the address of the tape unit on which a spool tape is mounted. Its format is:

X'cuu'

Only list and punch tasks can be started from a tape. If the tape is not recognized as a spool tape, a message is issued and the command is ignored.

**password.** When an RJE line is started, a password can be set for the line by the central operator. This password can be any combination of up to eight alphameric characters. A default password is defined during POWER/VS generation, and is used if no password is specified by the central operator. If the password is specified in the PSTART command, this password is in effect until a PSTOP command is issued. If a password is set, a remote operator must use it to sign on.

**class.** The meaning of this parameter depends on the type of task to be started.

For a write task, "class" defines the output class(es) upon which the task operates. Up to four classes can be designated by specifying one to four alphabetic characters from A through Z. The order specified is the order in which the classes will be processed. If no class parameter is specified, only class A is selected.

For a read task, "class" defines the input class that is assigned to all jobs without a class specification in their \* \$\$ JOB cards, when no CTL statement is in effect. It may be specified as an alphameric character from A through Z or from 0 through 4. If no class parameter is specified, class defaults to A.

For a partition, "class" defines the input class(es) that can be executed in this partition. Up to four classes can be designated by specifying one to four alphameric characters from A through Z or from 0 through 4. The order specified is the order in which the classes will be executed. If no class is specified, only job entries with matching partition-type (0-4) input class are selected.

**Note.** Only one of the classes specified in the "partition" form of a *PSTART* command can be a partition-type class (i.e., 0-4), and the partition-type class specified must match the partition being started. For example: *PSTART F2,AC2*.

For a more detailed discussion of class, refer to "POWER/VS Concepts and Facilities".

**partition.** Specifies the partition to be brought under POWER/VS control as:

BG, F4, F3, F2 or F1

If the partition specified is already controlled by POWER/VS, the *PSTART* command acts as a normal *AR START* command. If the partition is not under POWER/VS control, the *PSTART* command first brings it under control of POWER/VS and then performs the *AR START* function.

If the partition to be placed under POWER/VS control is still active, or not supported by the DOS/VS system, or if it does not have lower priority than the POWER/VS partition, a message is issued and the command is ignored.

**3540addr.** Specifies the address of the 3540 Diskette Input/Output Unit associated with the read task. Its format is:

cuu or X'cuu'

'filename'. Specifies the name, in quotes, of the file to be read, which must be the same as the name in the HDR1 label on the 3540 diskette. This name is the file identifier obtained from the DLBL statement when the file was created.

**maxno.** Specifies the maximum number of 3540 diskettes to be read. The largest possible value is 255; default is 1. The RDR task terminates when the specified number of diskettes have been read or after reading a diskette whose HDR1 label indicates it is the last diskette in the file.

**S.** Specifies volume sequence checking. If specified, the sequence number of the first volume must be 1, and succeeding volumes 2, 3, etc.

**uraddr,3540addr.** Specifies a card reader and 3540 Diskette Input/Output Unit, respectively. A data file from the 3540 is to be inserted into the input from the card reader to create a single sequential input stream. This parameter has the format:

cuu,cuu or X'cuu',X'cuu'

See the "RDR Statement" under "POWER/VS Job Entry Control Language" for a discussion of the complete procedure to merge card reader input and 3540 files.

*Examples:*

<i>PSTART RDR,00C,2</i>	Start read task on X'00C'. Assign class 2 to all jobs without specific class assignment.
<i>PSTART LST,00E,AB</i>	Start list task for class A and B.
<i>S PUN,00D,X'285'</i>	Start tape punch task on X'00D' from tape mounted on 285.
<i>PSTART BG</i>	Bring BG under control of POWER/VS. Select input class 0 only.
<i>PSTART F1,AB</i>	Bring F1 under control of POWER/VS. Select input classes A and B only.
<i>PSTART RJE,030,TORONTO</i>	Start RJE task on line 030. TORONTO is the security password.
<i>PSTART RDR,00B,'FILE-ID',31</i>	Start a read task on a 3540 on X'00B'. Read up to 31 diskettes on the file called 'FILE-ID'. Reading stops after 31 diskettes or after reading a diskette with a last diskette indicator in its label.
<i>PSTART RDR,00C,00B</i>	Start read task on X'00C'. Reading can also be done from a 3540 on X'00B'.

## PSTOP Command

PSTOP terminates a POWER/VS task or releases a partition from POWER/VS control.

When a read task is stopped, further input processing is terminated, and the current read entry is *not* placed in the input queue.

When a write task is stopped, further output processing is terminated, but the current job entry segment is *not* deleted from the output queue. When a write task is restarted, processing starts either at the first record or next record of the same job entry segment, or starts by processing the next job entry, depending on the parameters specified in the PSTOP command.

For read/write or RJE tasks, POWER/VS acknowledges the PSTOP command with a message at both the central location and at the terminal.

$$\left. \begin{array}{l} \{ \text{PSTOP} \} \\ \{ \text{P} \} \end{array} \right\} \left( \begin{array}{l} \text{uraddr} [ , \text{EOJ} \\ \quad \quad \quad \text{RESTART} ] \\ \text{partition} \\ \text{lineaddr} [ , \text{EOJ} ] \end{array} \right)$$

### Operands

**uraddr.** Specifies the unit record device address associated with the read/write task being stopped as:

cuu or X'cuu'

**EOJ.** Command to stop the task when it has completed processing the current job entry. This parameter should always be specified for RJE tasks, unless an emergency occurs.

**RESTART.** Specifies that when the task is started again, processing of the stopped job entry will begin at the record following the last one processed before the PSTOP command was issued. RESTART applies only to output processing.

**Note.** If RESTART is not specified along with uraddr, processing will begin at the first record of the current job entry when this task is started again (via PSTART).

**partition.** Specifies the POWER/VS controlled partition to be stopped as:

BG, F4, F3, F2, or F1

When a partition is stopped by a PSTOP command, it is also released by POWER/VS.

This form of the PSTOP command is not effective until processing of the current read entry is completed.

After the partition is released by POWER/VS, SYSRDR and all assignments to POWER/VS controlled devices for this partition are released. (It is the user's responsibility to re-assign these devices when the partition is restarted.) The partition can now be started as a normal batch partition. To restart it as a POWER/VS controlled partition, the DOS/VS STOP or UNBATCH command should be issued, followed by a PSTART command.

**lineaddr.** Specifies the line address associated with the RJE task. It has the format:

cuu or X'cuu'

### Examples:

PSTOP 00E,EOJ

Stop write task operating on 00E after completion of the current list entry.

PSTOP F3

Release F3 from control of POWER/VS after completion of the current read entry.

**PSTOP 00D,RESTART** Stop write task operating on 00D. When the task is restarted, processing of the stopped entry continues at the point of interruption.

**PSTOP 030** Stop the RJE task operating on line 030, immediately. If this task is reading, the input records already processed are lost. If the task is writing, output starts from the first record of the list entry when the task is restarted.

## PGO Command

PGO reactivates a POWER/VS task that is waiting for an operator response. Write tasks are reactivated by using the `uraddr` parameter. Execution processors are reactivated by the `partition, cuu` parameter.

The PGO command is normally used after an action-type message is issued to the operator from POWER/VS. It is *not* used to restart partitions or tasks stopped by a PSTOP command.

**Note.** *Read tasks, operating on card readers that give device end interrupt when they are readied, are automatically reactivated when the device end occurs. They do not need a PGO command.*

$$\left\{ \begin{array}{l} \text{PGO} \\ \text{G} \end{array} \right\} \left\{ \begin{array}{l} \text{uraddr} \\ \text{partition, cuu} \end{array} \right\}$$

### Operands

**uraddr.** Specifies the unit record device to be reactivated. It has the format:

`cuu or X'cuu'`

**partition, cuu.** Specifies the partition and unit record device address whose output is being spooled to tape. "partition" can be:

BG, F4, F3, F2, or F1

### Examples:

PGO 00E                      Reactivate list task after, for example, aligning printer forms.

G BG,00E                     Reactivate BG after mounting a new spool tape for printer output.

## PEND Command

The PEND command is used to terminate the POWER/VS program. Unless KILL is specified, all active tasks are allowed to continue until the job entry they are currently processing is completed. POWER/VS controlled partitions are released as soon as the job corresponding to the current input entry is terminated. After all supported partitions are released and all readers and writers have stopped, the POWER/VS partition is released, and the system is restored for normal DOS/VS operation. If KILL is specified, POWER/VS is cancelled immediately.

POWER/VS acknowledges the PEND command with a message.

$$\text{PEND} \quad \left[ \begin{array}{l} \text{uraddr} \\ \text{KILL[,uraddr]} \end{array} \right]$$

### Operands

**uraddr.** Specifies the unit record device address of the printer on which the status report is to be printed. A dump is printed if KILL is specified. If `uraddr` is not specified, no status report or dump is produced. "uraddr" can be entered in either of two formats:

`cuu or X'cuu'`



**HOLD.** Specifies that the corresponding job entry is *not* to be deleted, but put in the hold state; that is, the input entry must not be deleted when a partition is flushed, or the output entry must not be deleted when a write task is flushed. If this parameter is specified when flushing a read task, it is ignored.

The HOLD option is useful when a job needs more devices or other resources than are currently available; the job can then be rescheduled when the resources become available. Jobs requiring the same resources can be grouped together in input or output classes to prevent such inconvenience.

**partition.** Specifies the partition in which the job to be flushed is executing as:

BG, F4, F3, F2, or F1

*Examples:*

PFLUSH 00E	Delete list entry currently being printed. Continue with next entry.
PFLUSH BG	Terminate processing of current read entry for BG. Delete this entry. Continue with next entry.
PFLUSH BG,HOLD	Same as preceding example, except that the read entry is preserved and put in the hold state.

## PRESTART Command

The PRESTART command directs the specified list/punch task to stop and to resume printing/punching either from the beginning of the list/punch entry, or up to 9999 pages (or cards) forward or backward from the point of interruption.

The number of duplicate pages or cards written because of a PRESTART command is separately counted and stored in the list/punch account record. A page is defined as a skip-to-channel-one.

If too large a backward count is specified, or if the operand is omitted, printing/punching resumes from the first page/card. If too large a forward count is specified, message 1Q42I is issued.

```
{ PRESTART } uraddr[,n]
{ T }
```

### *Operands*

**uraddr.** Specifies the unit record device address associated with the write task. Its format is:

cuu or X'cuu'

**n.** Specifies a signed or unsigned value from 0 to 9999. A plus sign indicates a forward count, a minus sign indicates a backward count from the last interrupted page/card. No sign indicates a count from the beginning.

*Example:*

PRESTART 00E,-10	Reprint the previous ten pages and continue to end of output.
------------------	---------------------------------------------------------------

## QUEUE MANAGEMENT COMMANDS

### PDISPLAY Command

The PDISPLAY command provides the operator with a status report on SYSLOG of the job entry or entries specified by the operand. Each job entry is represented by one line containing the following items:

Items	Characters
jobname	8
jobnumber	5
priority	1
disposition	1
class	1
"from" remote-id	3
"to" remote-id	3
number of records/pages	6
number of copies	2
forms-number	4

The last four items are displayed for output only. For punch output, the number of records is displayed whereas for list output the number of pages is displayed.

Each line of the status report starts with the message number 1R46I. If the status report is made up of entries from different queues, the name of the queue is printed before the first entry is displayed.

The disposition of a job entry may appear as H for hold, K for keep, L for leave, D for delete (after processing), or as an \* if an execution processor or a write task is currently processing the entry.

The class can appear as an alphameric character from A to Z, or from 0 to 4.

Remote-id is displayed as a number from 000 to 200. The remote-id of the central location is 000.

If a write task is processing a job entry when the PDISPLAY command is entered, "number of copies" is the number of copies *left* to be produced, including the one in progress. The "number of records or pages" is the number of records or pages left to be produced. The operator can use this information to determine whether to stop a write task or let it continue until the current job entry is completed.

{ PDISPLAY } { D	queue.jobname[.jobnumber]
	queue[,ALL]
	queue,HOLD
	queue,FREE
	queue,RJE[,remid]
	queue,LOCAL
	queue,*abc
	queue,class
	ALL[,listaddr]
	HOLD
	FREE
	RJE[,remid]
	LOCAL
	*abc
	MSG
	A
	M
	Q

## *Operands*

**queue.** Specifies the queue for which the PDISPLAY command is to be executed:

LST = the list queue  
PUN = the punch queue  
RDR = the read queue.

**jobname.** Specifies the name by which the job entry is known to POWER/VS. Jobname can be two to eight alphabetic characters.

**jobnumber.** Specifies the number assigned to the job entry by POWER/VS. The jobnumber can be two to five digits.

**ALL.** Specifies a request for the status of all job entries in the specified queue. If the queue parameter is not specified, status information on all entries in all queues is displayed on SYSLOG.

**listaddr.** Specifies the printer to which the status report is to be sent. Its format is:  
cuu or X'cuu'

**HOLD.** Specifies a request for the status of all entries in the specified queue that are *not* available for processing (i.e., disposition is H or L). If the queue parameter is *not* specified, status information is displayed on all such job entries.

**FREE.** Specifies a request for the status of all entries in the specified queue that *are* available for processing. (i.e., disposition D or K). If the queue parameter is *not* specified status information is displayed on all such job entries.

**RJE.** Specifies a request for the status of all RJE-type job entries in the specified queue. If the queue parameter is *not* specified, status information is displayed on all such job entries.

**RJE,remid.** Specifies a request for the status of all job entries in the specified queue that are submitted by or routed to a specific remote user. If the queue parameter is *not* specified, status information is displayed on all such job entries.

**LOCAL.** Specifies a request for the status of all job entries in the specified queue that are submitted from or routed to the central location. If the queue parameter is *not* specified, status information is displayed on all such job entries.

**class.** Specifies a request for the status of all job entries in the queue with the specified class. Class can be specified as a character from A through Z, or from 0 to 4 (input class only).

**\*abc.** Specifies a request for the status of all job entries having the first n characters of their jobnames in common. "abc" represents any combination of from one to seven alphabetic characters. If the queue parameter is *not* specified, status information is displayed for all queues. The asterisk (\*) must be keyed.

**MSG.** Specifies a request to display all ALLUSERS-type messages that have been entered, together with their originators. The number displayed with each message can be used to delete that message.

**A.** Specifies a request for a list of all active read/write tasks together with the job entries they are currently processing

**M.** Specifies a request to re-display a list of all POWER/VS messages for which an operator reply is still required.

**Q.** Specifies a request for the number of free queue records, free track groups, and free POWER/VS account records (if applicable).

Examples:

Display the status of all entries in all queues.

```
AR PDISPLAY ALL
F1 1R46I  READER QUEUE  P D C          *****
F1 1R46I  DISKDUMP 00C68 9 L A          *****
F1 1R46I  PAUSE    00070 9 L A          *****
F1 1R46I  DITTO    00071 3 L A          *****
F1 1R46I  PRINTLOG 00069 1 L A          *****
F1 1R46I  LIST QUEUE  P D C  PAGES CC FORM *****
F1 1R46I  PAUSE    00070 9 D A          1 1 *****
F1 1R46I  PAYINPUT 00063 3 D A          3 1 *****
F1 1R46I  PAYEDIT  00064 3 D A          75 1 *****
F1 1R46I  PAYSORT  00065 3 D A          5 1 *****
F1 1R46I  PAYCARDS 00066 3 D A          7 1 CHEX *****
F1 1R46I  PAYLIST  00067 3 D A          44 1 *****
F1 1R46I  PRINTLOG 00069 1 D A          5 1 *****
F1 1R46I  PUNCH QUEUE  P D C  CARDS CC FORM *****
F1 1R46I  PAYCARDS 00066 3 D A          233 1 *****
```

Display the status of all jobs that begin with PAY.

```
AR PCISPLAY *PAY
F1 1R46I  READER QUEUE NOTHING TO DISPLAY *****
F1 1R46I  LIST QUEUE  P D C  PAGES CC FCRM *****
F1 1R46I  PAYINPUT 00063 3 D A          3 1 *****
F1 1R46I  PAYEDIT  00064 3 D A          75 1 *****
F1 1R46I  PAYSORT  00065 3 D A          5 1 *****
F1 1R46I  PAYCARDS 00066 3 D A          7 1 CHEX *****
F1 1R46I  PAYLIST  00067 3 D A          44 1 *****
F1 1R46I  PUNCH QUEUE  P D C  CARDS CC FORM *****
F1 1R46I  PAYCARDS 00066 3 D A          233 1 *****
AR PRELEASE RDR,PAUSE
F1 1R88I  OK
F1 1Q47I  BG PAUSE    00070 FROM 000
BG // JOB PAUSEJOB
```

The lines of the status report are *not* chained, and can, therefore, be interspersed with other DOS/VS messages.

## PALTER Command

The PALTER command is used to change the various parameters of a job entry that is already queued. The operands are the same as those specified in the JOB, LST, and PUN statements, and they have the same format. Any number of parameters can be modified by one command.

If an input entry (RDR) is specified and an execution processor is currently processing this entry, message 1Q88I is issued.

The priority attribute of a job entry in the input queue is identical to that of the corresponding entries in the output queues. On the other hand, the disposition and class attributes may be totally different in corresponding input and output job entries, since input disposition and class are defined in the JOB statement, whereas output disposition and class are defined in the LST and PUN statements.

The COPY and REMOTE parameters apply to output only. When applied to an input entry, they are ignored. If an output entry is specified in the PALTER command and a write task is currently processing the entry, only the number-of-copies can be altered. (When output is segmented and the first segment has been deleted from the queue, "number-of-copies" cannot be altered.) If any other parameter is specified, message 1Q88I is issued and the command is ignored. The same message is issued if the output is segmented and the first segment is already deleted from the output queue.

PDISPLAY can be used to check the results of the PALTER command.

```

{ PALTER }
  A
  queue, { jobname [jobnumber] }
           { ALL
             *abc
           }
           [,PRI=priority]
           [,DISP=disposition]
           [,CLASS=class]
           [,COPY=number-of-copies]
           [,REMOTE=remid]

```

*Operands*

**queue.** Specifies the queue for which the command is to be executed:

LST = the list queue  
 PUN = the punch queue  
 RDR = the read queue

**jobname.** Specifies the name by which the job entry is known to POWER/VS. Jobname can be two to eight alphanumeric characters.

**jobnumber.** Specifies the number assigned to the job entry by POWER/VS. The jobnumber may be one to five digits long. The PDISPLAY command may be used to obtain the correct specification for the jobnumber. This number can be used to identify different jobs with the same jobname.

**ALL.** Requests that ALL entries in the specified queue are to be altered.

**\*abc.** Requests to alter all job entries that have the first "n" characters of their jobnames in common. "abc" represents any combination of from one to seven alphanumeric characters. The asterisk (\*) must be keyed.

**priority.** Specifies the priority of the job entry. Priority is specified as a single digit from 0 to 9. Nine is the highest priority.

**disposition.** Can be specified as H for hold, K for keep, L for leave, or D for delete after processing.

**class.** Can be specified as any alphabetic character from A through Z. Also, specifications from 0 through 4 are allowed for input classes.

**remid.** Can be specified as any number from 0 to 200. "0" indicates the central location.

Only the "to" remote-id can be modified with a PALTER command, the "from" remote-id cannot be modified.

**number-of-copies.** Specifies the number of copies to be produced after the command is issued, including the one in progress.

*Examples:*

PALTER LST,PAYROLL,DISP=H,CLASS=B

Alter the disposition of the output of job PAYROLL to H, and alter the class to B.

A PUN,\*PAYR,DISP=D

Alter the dispositions of all jobs that begin with PAYR to D.

## PDELETE Command

The PDELETE command is used to remove one or more job entries from the specified queue. Job entries being processed by an execution processor or write task are not affected by this command.

$$\left. \begin{array}{l} \text{PDELETE} \\ \text{L} \end{array} \right\} \left\{ \begin{array}{l} \text{queue,jobname[,jobnumber]} \\ \text{queue,ALL} \\ \text{queue,class} \\ \text{queue,*abc} \\ \text{MSG[,n]} \end{array} \right\}$$

### Operands

**queue.** Specifies the queue for which the command is to be executed:

LST = the list queue  
PUN = the punch queue  
RDR = the read queue

**jobname.** Specifies the name by which the job entry is known to POWER/VS. Jobname can be two to eight alphanumeric characters.

**jobnumber.** Specifies the number assigned to the job entry by POWER/VS. The jobnumber is from one to five digits long. PDISPLAY can be used to obtain the jobnumber.

**ALL.** All job entries are to be deleted from the specified queue.

**class.** All job entries of a certain class are to be deleted from the specified queue.

**\*abc.** All job entries with the same first n characters are to be deleted. "abc" represents any combination of up to seven alphanumeric characters.

**MSG,n.** ALLUSERS-type message number n is to be deleted. The actual message number can be obtained with a PDISPLAY MSG command. If n is omitted, all ALLUSERS-type messages issued by the central operator are deleted.

### Examples:

PDELETE RDR,ALL	Delete the entire read queue.
PDELETE LST,B	Delete all entries in the list queue with a class of B.
PDELETE MSG	Delete all ALLUSERS-type messages issued by the central operator.

## PRELEASE Command

The PRELEASE command is used to take one or more job entries out of the hold or leave state and make them available for processing. After processing, a job entry originally in the leave state returns to the leave state; a job entry originally in the hold state is deleted from the queue.

If a PRELEASE command is issued for a job entry that is not in the hold state or leave state, message 1Q88I is issued and the command is ignored. The same message is issued when the command is entered for a job entry that is to be printed or punched at a terminal. Such a job entry cannot be released by the central operator.

$$\left. \begin{array}{l} \text{PRELEASE} \\ \text{R} \end{array} \right\} \left\{ \begin{array}{l} \text{queue,jobname[,jobnumber]} \\ \text{queue[,ALL]} \\ \text{queue,class} \\ \text{queue,*abc} \end{array} \right\}$$

### Operands

**queue.** Specifies the queue for which the command is to be executed:

LST = the list queue  
PUN = the punch queue  
RDR = the read queue

**jobname.** Specifies the name by which the job entry is known to POWER/VS. Jobname may be two to eight alphanumeric characters.

**jobnumber.** Specifies the number assigned to the job entry by POWER/VS. The jobnumber is one to five digits long. The PDISPLAY can be used to obtain the number.

**ALL.** All job entries are to be released from the specified queue.

**class.** All job entries of a certain class are to be released from the specified queue.

**\*abc.** All job entries with the same first n characters are to be released. "abc" represents any combination of up to seven alphanumeric characters.

### Examples:

PRELEASE LST,PAYROLL Release the entry PAYROLL from the list queue.  
PRELEASE RDR,ALL Release the entire read queue.  
PRELEASE LST,\*PAY Release all entries in the list queue whose names begin with P-A-Y.

## MISCELLANEOUS COMMANDS

### PINQUIRE Command

The PINQUIRE command provides the central operator with a status report for a specific RJE line or for all lines. Line status can be: processing, inactive, not initiated, or not supported.

- **Processing.** The remote user has entered a valid SIGNON command. In this case, the remote-id is displayed together with the line status.
- **Inactive.** The line has been started by the central operator with a PSTART command, but no user is currently signed on.
- **Not initiated.** The line has not been started by the central operator.
- **Not supported.** During POWER/VS generation, no PLINE macro was specified for this line.

```
{ PINQUIRE }      { lineaddr }  
{ I          }      { ALL    }
```

### Operands

**lineaddr.** Specifies the line address associated with the RJE task. Its format is:  
cuu or X'cuu'

**ALL.** The status of all supported lines are to be displayed.

### Example:

```
PINQUIRE ALL Display the status of all supported lines.  
030 PROCESSING 022  
031 PROCESSING 150  
032 PROCESSING 004  
033 NOT INITIATED  
034 INACTIVE
```

## PBRDCST Command

The PBRDCST command enables the central operator to communicate with remote users. Messages can be submitted for one specific user or for all users.

Messages generated by the PBRDCST command are stored in virtual storage and displayed only on request of the remote user. Up to 14 messages can be queued.

Messages of the ALLUSERS-type are queued and given a message sequence number. The PDISPLAY command can be used to obtain the actual number.

$$\left\{ \begin{array}{l} \text{PBRDCST} \\ \text{B} \end{array} \right\} \text{ remid, 'text'}$$

### Operands

**remid.** Specifies the remote-id to which the command applies. Remote users can be identified by numbers from 1 to 200. A specification of 0 indicates the central location, and a specification of ALLUSERS indicated all users.

**'text'.** Specifies the text of the message. The message can consist of from 1 to 40 characters enclosed in single quotation marks. A single quotation mark within the message must be written as two quotation marks.

### Examples:

PBRDCST 150,'RJE will be shut down in 30 min.'      Broadcast the message to the user whose remote-id is 150.

PBRDCST ALLUSERS,'Service resumes at 8.00 a.m.'      Broadcast the message to all users in the system.

## PACCOUNT Command

The PACCOUNT command can be issued at any time to save the accumulated account file records. The operator is notified by a warning message when the account file becomes 80% full.

If any POWER/VS task requires an account record and there are no free records, a message is issued, and the task requesting the account record is placed in the wait state until the account file is emptied by the PACCOUNT command. After a message is issued stating "ACCOUNT FILE SAVED", the entire DASD space allocated to the file is available for more account records.

It is the user's responsibility to process the accounting information.

$$\left\{ \begin{array}{l} \text{PACCOUNT} \\ \text{J} \end{array} \right\} \left[ \begin{array}{l} \text{tapeaddr[,X'ss']} [,filename] \\ \text{DISK,filename} \\ \text{DEL} \end{array} \right]$$

### Operands

If no operand is specified in the PACCOUNT command, the account file is spooled to disk. It is stored with priority 1 in output class P with DISP=H in the punch queue. In order to get the account information punched, the operator can start a punch write task on output class P.

**tapeaddr.** Specifies the tape on which the file is to be written. Its format is:

*cuu or X'cuu'*

Optionally, the density and mode of the output tape can be specified in the format *X'ss'*. Before any record is written to the specified tape, label checking is performed.

**filename.** If filename is specified along with tapeaddr, the file created will be a standard labeled tape file. "filename" is one to eight alphameric characters long. A standard label for the tape file must be present on the label cylinder. If a filename is *not* specified, the file will be a standard *unlabeled* DOS/VS tape file containing variable length records.

**DISK,filename.** Specifies that the POWER/VS account file is to be written to disk. "filename" is one to eight alphameric characters long.

It is the user's responsibility to store label information for a sequential disk output file on the label cylinder. The disk file created is a copy of the IJAFILE. The device type on which the receiving file resides must be the same as the device type on which IJAFILE resides.

**DEL.** Specifies that the account file records are to be deleted.

*Examples:*

PACCOUNT 280                      Save the account file on an unlabelled tape mounted on tape unit X'280'.

PACCOUNT DEL                      Delete all account file records.

PACCOUNT                              Spool account file records to punch queue.

## **PSETUP Command**

The PSETUP command can be entered only after message 1Q40A, requesting a special forms mount, is displayed. It causes *n* pages of spooled list data to be printed. All printable characters of the page are replaced by the character X. During the printing of the *n* pages, the operator can manually align the printer forms and can reissue PSETUP as many times as is necessary.

The number of pages printed by PSETUP will be counted and stored separately in the list account record.

When the PGO command is issued to resume printing, the pages are printed again with their original contents.

PSETUP uraddr[*n*]

*Operands*

**uraddr.** Specifies the unit record device associated with the write task. Its format is:

*cuu or X'cuu'*

**n.** One or two digits that specify the number of pages to be printed. If *n* is omitted, only one page is printed.

Example:  
PSETUP 00E,2  
then,  
PGO 00E

Print two pages to allow forms alignment.

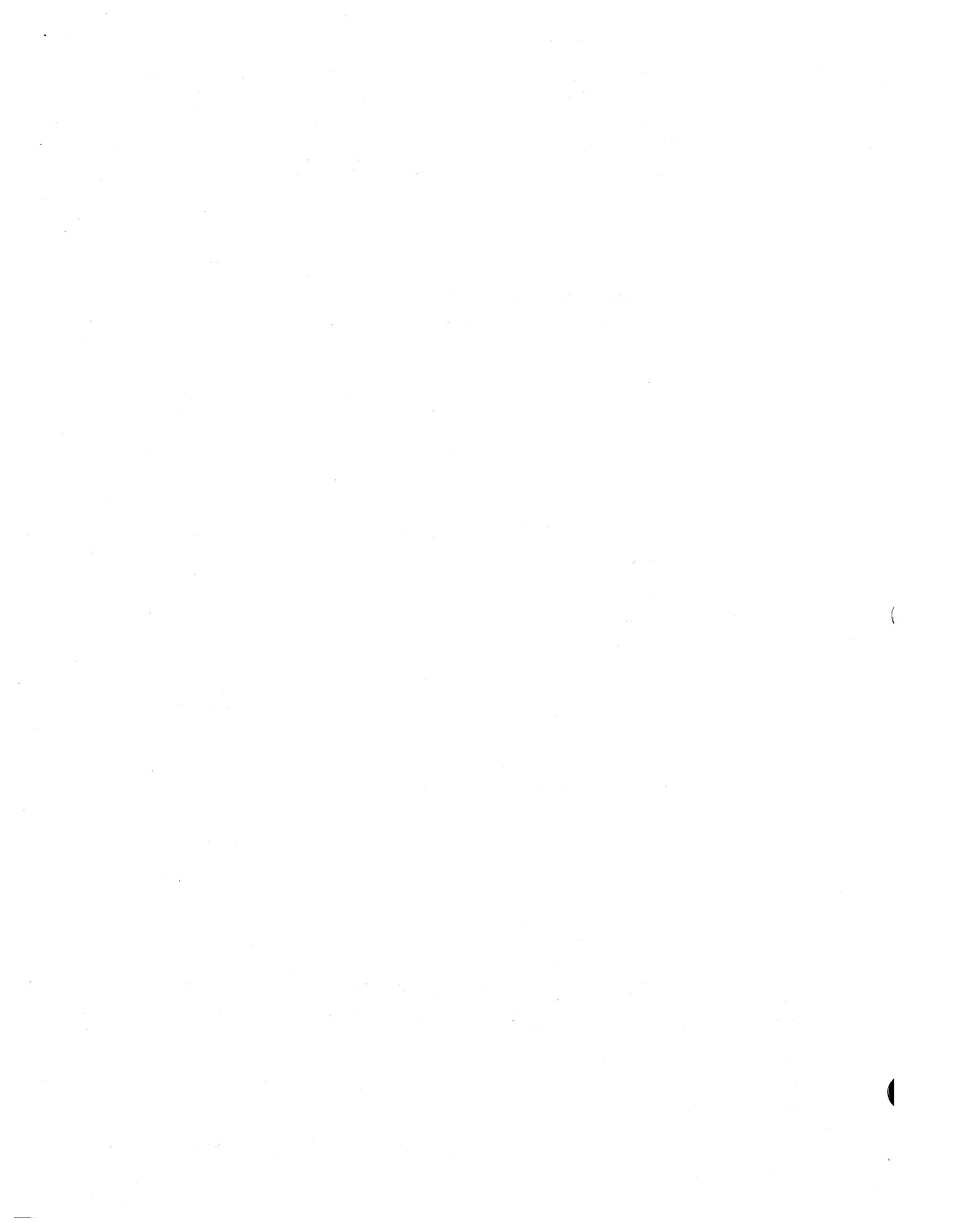
Pages are printed again with their original contents. Printing continues.

1st Page {  
XXXX XXX XX  
XXX XX XXX XXXXX XXXXXXXXXXXXXXXXXXXX XXXXXXXX  
XXX XXXXXXXX XXXXXXXXXXXXXXXX XXX XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX XXXXXXXX XXXXXXXX  
X XXXX

2nd Page {  
XXXX XXX XX  
XXX XX XXX XXXXX XXXXXXXXXXXXXXXXXXXXXXXX XXXXXXXX  
XXXX XXXXXXXX XXXXXXXXXXXXXXXX XXX XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX XXXXXXXX XXXXXXXX  
X XXXX

1st Page {  
SEPT 30, 74  
PAY TO THE ORDER OF.....ONE EMPLOYEE \*\*\*123.45  
ONE HUNDRED TWENTY-THREE AND ----- 34/100 DOLLARS  
# 2001

2nd Page {  
SEPT 30, 74  
PAY TO THE CRDER OF.....ANOTHER EMPLOYEE \*\*\*543.21  
FIVE HUNDRED FORTY-THREE AND ----- 21/100 DOLLARS  
# 2002



## POWER/VS REMOTE OPERATOR COMMAND LANGUAGE (ROCL)

Remote job entry (RJE) is an extension of POWER/VS and allows the user to submit jobs from remote locations via IBM 2770, 2780 or 3780 terminals for processing under DOS/VS at a central installation.

The central installation and the terminal can communicate through messages and commands. POWER/VS messages and responses to POWER/VS terminal commands are sent to the remote operator as list output. They are stored in virtual storage and automatically printed (and subsequently deleted) whenever the line is available for print operations.

Broadcast messages are stored in virtual storage. They are displayed only upon request.

The POWER/VS Remote Operator Command Language (ROCL) is used to control jobs that are entered at a terminal for processing at a central installation. There are four types of commands:

- **Terminal control commands.** Start and stop user sessions.
- **Task management commands.** Apply to RJE write tasks. (The RJE read task is started by the central operator when he brings up the line, and its operation is controlled by the system, as are execution processors.)
- **Queue management commands.** Apply to jobs that are submitted by or routed to the same remote-id as the one issuing the command.
- **Miscellaneous commands.** Print page layouts or transmit messages.

The following table includes all valid terminal commands.

Type	Command	Function
Terminal control	* .. SIGNON	start a user session
	* .. SIGNOFF	terminate a user session
Task management	* .. START	start a writer or start message generation
	* .. STOP	stop a writer or stop message generation
	* .. GO	re-activate a write task
	* .. FLUSH	flush an active job entry
	* .. RESTART	restart a writer
Queue management	* .. DISPLAY	display job status
	* .. ALTER	alter job attributes
	* .. DELETE	delete a job or a message
	* .. RELEASE	release a job
Miscellaneous	* .. BRDCST	transmit a message
	* .. SETUP	print page layout

POWER/VS RJE terminal commands are entered on punched cards through the reader at the terminal. (See Figure 11.) They must be submitted outside POWER/VS job boundaries, otherwise they are treated as user data.

Each RJE command consists of the following fields:

- 1 **Identification field.** Contains an \* in column 1, blank in column 2, and .. (periods) in columns 3 and 4. Column 5 must be blank.
- 2 **Operation field.** Specifies the RJE command. At least one blank must separate this field from the following field.
- 3 **Operand field.** Contains one or more parameters, separated by commas. At least one blank must separate the operand field and the comments field.

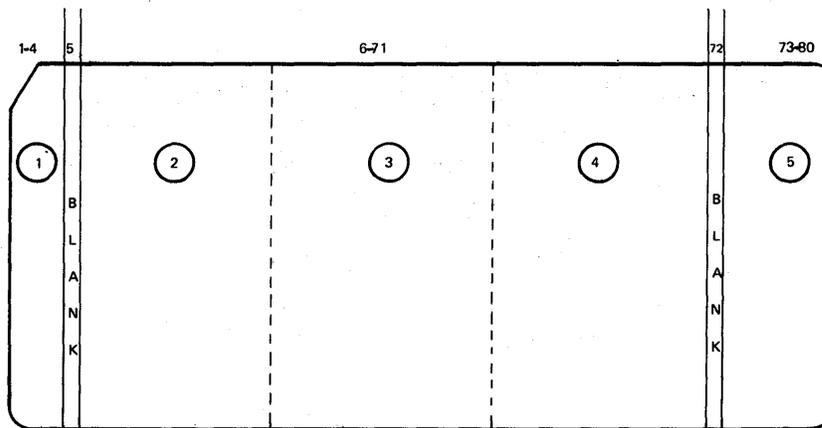


Figure 11. Format of RJE Commands

- 4 **Comments field.** May contain any information considered helpful by the user. Continuation of the comments field is *not* allowed.  
The operation field, operand field, and comments field must be contained in columns 6 through 71. Column 72 must be blank.
- 5 **Sequence field.** Sequence numbers are useful when a number of commands with the same operation code are submitted from a terminal. Columns 73 - 80 are returned in diagnostic messages.

## OPERATIONAL CHARACTERISTICS

When the central operator starts a line, he starts, in fact, an RJE read task (reader).

The first card submitted from the terminal must be a SIGNON command. This command identifies the terminal and attaches it logically to the system.

After the SIGNON command has been successfully processed, the remote operator can submit any command or input data through the reader. Input data must start with a JECL or DOS/VS JOB card.

Once a reader is started, it continues reading until EOF (end of file). Input records are grouped in blocks before they are transmitted over the line.

Commands are removed from the input stream, and processed. Output that may result from these commands, like diagnostic messages and status reports, cannot immediately be returned to the terminal, since the line is still in read mode. Therefore, this type of output is temporarily queued.

When EOF is encountered on the reader, the line is automatically turned around, and any POWER/VS messages are displayed on the printer. This is the only type of output that is produced spontaneously by the system. For normal job output a write task must explicitly be started by the terminal operator.

After a writer is started, it normally prints or punches all available output of the selected class. The writer is then put in a dormant state, and the line is turned around, after which the reader is started again. After all input is read and the messages are printed, the writer can resume operation (if output has been stored in the meantime). If the operator does not want this *hot writer* facility, he may deactivate the writer with a STOP command. For example, a hot punch writer could be undesirable when the same hopper is used for both read and punch operations.

When no more input or output is left to be processed, or when operator intervention is required on the reader, printer, or punch, the audible alarm sounds on the terminal, and POWER/VS expects either a ready interrupt from the reader or for more output to be stored in the output queues (hot writer). The terminal is signed off by POWER/VS after *n* minutes of idle time, where *n* is the value defined during POWER/VS generation through the TIMEOUT generation parameter. (If TIMEOUT=NO, sign off is not forced).

When a writer is interrupted (I/O error, for example), the operator can enter a FLUSH, a RESTART, or a STOP command through the reader.

*Note.* If a punch writer and a list writer are active at the same time, list output receives priority over punch output.

## TERMINAL COMMANDS

### SIGNON Command

The SIGNON command identifies the terminal, and makes the logical connection between the terminal and the system. It must be the first card submitted from the terminal after the line is started by the central operator. It can also be submitted during the course of the session, in which case it terminates the current session and initiates the next session.

The operator must use the remote-id assigned to him during POWER/VS generation, since the terminal characteristics are defined in relation to this remote-id. If the same remote-id is used to sign onto another terminal with different hardware characteristics, causing unpredictable results.

The security password must match the password set for the line by the central operator. If the central operator did not specify a password when the line was started, it must match the password set for the line during POWER/VS generation. If no password is set for the line, the password parameter can be omitted.

\* .. SIGNON remotid[,password] user information

#### *Operands*

**remotid.** Specifies the remote-id assigned to the terminal during POWER/VS generation. Remote users are identified by numbers from 1 to 200.

**password.** Specifies the password set for the line by the central operator, or set during POWER/VS generation. It can be any combination of up to eight alphameric characters.

**user information.** Consists of up to sixteen bytes of user information, which are moved to the line account record. This information is punched in columns 56 to 71.

*Note.* An invalid SIGNON command is interpreted as a SIGNOFF command. The terminal is disconnected if it is on a switched line.

#### *Example:*

\* .. SIGNON 002,SECRET Sign on at a terminal with remote-id 002 and password SECRET.

### SIGNOFF Command

The SIGNOFF command informs the system that the remote operator will terminate the session at the next end-of-file condition on the reader. POWER/VS will disconnect the terminal from the system, stop any active writers, and prepare the line for other terminals to sign on.

*Note.* A terminal is forced to sign off when it has been idle for a number of minutes, as defined during POWER/VS generation.

\* .. SIGNOFF (no operand)

#### *Example:*

\* .. SIGNOFF Terminate a session.

## TASK MANAGEMENT COMMANDS

### START Command

The **START** command is used to initiate an RJE write task (writer). The RJE read task (reader), is controlled only by the central operator.

The remote operator must enter a **START** command in order to receive his output. **POWER/VS** does not send unsolicited output to any terminal.

A **START** command is invalid when applied to an active task.

The **START** command is also used to cause broadcast messages to be accepted. (When an RJE task is started, the terminal accepts broadcast messages until **MSG** is specified in the **STOP** command.)

```
* .. START      { task
                  { task,class }
                  { MSG }
```

#### *Operands*

**task.** Specifies the task to be started as **LST** or **PUN**. Write entries are processed on a first-in, first-out basis within priority, within class.

**class.** Defines the output class(es) upon which the task is to operate. Up to four classes can be designated by specifying one to four alphabetic characters from **A** through **Z**. The order specified is the order in which the classes will be processed. If no class parameter is specified, only class **A** is selected.

**MSG.** Specifies that all subsequent broadcast messages directed to this terminal are to be accepted (printed).

#### *Examples:*

```
* .. START PUN           Start punch task for class A.
* .. START LST,AB       Start list task for classes A and B.
```

### STOP Command

The **STOP** command is used to terminate an RJE write task. The terminal operator cannot control the operation of an RJE read task.

A **STOP** command is invalid when applied to an inactive task.

The **STOP** command is also used to cause broadcast messages to be ignored (not accepted).

```
* .. STOP      { task
                  { task,EOJ }
                  { task,RESTART }
                  { MSG }
```

### *Operands*

**task.** Specifies the task to be stopped. LST or PUN are the valid entries.

**EOJ.** Specifies that the task will not stop until it has completed processing the current entry.

**RESTART.** Specifies that when the task is started again, processing will begin at the record following the last one processed before the STOP command was issued.

**MSG.** Specifies that all subsequent broadcast messages directed to this terminal are to be ignored. In addition, existing messages in the queue are deleted immediately.

**Note.** If either EOJ or RESTART is not specified along with "task", processing will begin at the first record of the current job entry when this task is started again (via START).

### *Examples:*

- \* .. STOP PUN,RESTART      Stop the punch task. When the task is restarted, processing of the entry continues at the point of interruption.
- \* .. STOP LST,EOJ          Stop the list task after completion of the current list entry.
- \* .. STOP LST              Stop the list task immediately. When the task is restarted, processing will start at the beginning of the interrupted list entry.
- \* .. STOP MSG              Ignore broadcast messages.

## **GO Command**

The GO command is used to re-activate a write task after receiving the "mount forms" message.

- \* .. GO task

### *Operands*

**task.** Specifies the task to be re-activated as LST or PUN.

### *Example:*

- \* .. GO LST                  Re-activate the list task. The proper forms have been mounted.

## **FLUSH Command**

The FLUSH command is used to discontinue the job entry currently being processed by an RJE write task.

When the write task is flushed, further output processing is terminated for this POWER/VS job, and the write entry is deleted, unless HOLD is specified.

FLUSH is invalid when applied to an inactive task. Also, FLUSH can only be issued for a task that has been interrupted by the remote or central operator or by an action-type POWER/VS message.

- \* .. FLUSH      { task  
                  } task,HOLD }

### *Operands*

**task.** Specifies the task to be deleted as LST or PUN.

**HOLD.** Specifies that the corresponding write entry is not to be deleted, but put in the hold state.

### *Examples:*

- \* .. FLUSH PUN                      Delete the punch entry currently being punched.  
                                         Continue with the next entry.
- \* .. FLUSH LST,HOLD                Skip the list entry currently being printed,  
                                         but do not delete it from the queue.  
                                         Continue with the next entry.

## **RESTART Command**

The RESTART command directs the RJE write task to discontinue printing or punching immediately and to resume output from a specified point. A list writer can resume printing from either the beginning of the list entry, or up to 9999 pages forward or backward from the interrupted page. A punch writer can resume punching from either the beginning of the punch entry, or up to 9999 cards forward or backward.

The number of duplicated pages printed or cards punched because of a RESTART command is separately counted and stored in the list account record (if created).

RESTART is invalid when applied to an inactive task. Also, RESTART is only applicable to a task that has been interrupted by the operator or by an action-type POWER/VS message.

```
* .. RESTART     { task }  
                  { task,n }
```

### *Operands*

**task.** Specifies the task to be restarted as LST or PUN.

**n.** Specifies a signed or unsigned value from 0 to 9999. A plus sign indicates the number of pages or cards forward and a minus sign indicates the number of pages or cards backward from the last interrupted page or card. No sign indicates a specified page or card count from the beginning.

If the operand is omitted or is too large a backward count, output resumes from the first page or card. If too large a forward count is specified, message 1Q42I is issued and the command is ignored.

### *Example:*

- \* .. RESTART LST,-10                Reprint the previous ten pages and continue  
                                         to the end of output.

## **QUEUE MANAGEMENT COMMANDS**

### **DISPLAY Command**

The DISPLAY command provides the remote operator with a status report of the specified job entry or entries. The remote operator can only display the status of jobs that he submitted himself or that have been routed to him.

Each job entry is represented by one line containing the following items:

Items	Characters
jobname	8
jobnumber	5
priority	1
disposition	1
class	1
FROM remote-id	3
TO remote-id	3
number of records (pages)	6
number of copies	2
forms-number	4

The last four items are displayed for output only. For punch output, the number of records is displayed, whereas for list output the number of pages is displayed.

Each line of the status report starts with the message number 1R46I. If the status report is made up of entries from different queues, the name of the queue is printed before the first job entry is displayed.

The disposition of a job entry may appear as H for hold, K for keep, L for leave, D for delete (after processing) or as an \* if an execution processor or a write task is currently processing the entry.

The class can appear as an alphameric character from A to Z, or from 0 to 4. Remote-id is displayed as a number from 001 to 200 if the job is entered from a terminal or if the output is routed to a terminal. The remote-id of the central location (000) is *not* printed in the status report.

If a write task is processing a entry when the DISPLAY command is entered, "number of copies" is the number of copies *left* to be produced, including the one in progress. The "number of records or pages" is the number of records or pages left to be produced. The operator can use this information to determine whether to stop a write task or let it continue until the current entry is completed.

Output from the DISPLAY command is stored in virtual storage, and is transmitted as soon as the line becomes available for write operations.

```
* .. DISPLAY { queue,jobname[,jobnumber]
               queue[,ALL]
               queue,HOLD
               queue,FREE
               queue,*abc
               queue,class
               ALL
               HOLD
               FREE
               *abc
               MSG }
```

### Operands

**queue.** Specifies the queue for which the DISPLAY command is to be executed:

- LST = the list queue
- PUN = the punch queue
- RDR = the read queue

**jobname.** Specifies the name by which the job entry is known to POWER/VS. Jobname can be two to eight alphameric characters.

**jobnumber.** Specifies the number assigned to the job entry by POWER/VS. The jobnumber is one to five digits long.

**ALL.** Requests the status of all entries in the specified queue. If the queue parameter is not specified, status information on all entries in all queues is displayed.

**HOLD.** Requests the status of all entries in the specified queue that are *not* available for processing (that is, in hold state or leave state). If the queue parameter is *not* specified, status information is displayed on all such entries.

**FREE.** Requests the status of all entries in the specified queue that are available for processing (that is, in keep state or dispatchable state). If the queue parameter is *not* specified, status information is displayed on all such entries.

**class.** Requests the status of all job entries in the queue with the specified class. Class can be specified as an alphameric character from A through Z, or from 0 to 4.

For a more detailed discussion of class, refer to "POWER/VS Concepts and Facilities".

**\*abc.** Requests the status of all entries having the first n characters of their jobnames in common. "abc" represents any combination of from one to seven characters. If the queue parameter is *not* specified, status information is displayed for all queues.

**MSG.** Requests the display of all ALLUSERS-type messages that have been entered, together with their originators and all operator messages routed specifically to this remote-id. Operator messages are automatically deleted when they are displayed.

*Examples:*

\* .. DISPLAY RDR,\*PAY                    Display the status of all entries for remote-id 015 that are in the read queue and that begin with PAY.

```

F1 1R46I  READER QUEUE NOTHING TO DISPLAY      *****
F1 1R46I  LIST QUEUE  P D C  PAGES CC FCRM     *****
F1 1R46I  PAYINPUT 00063 3 D A      3 1        *****
F1 1R46I  PAYEDIT  00064 3 D A     75 1        *****
F1 1R46I  PAYSORT  00065 3 D A      5 1        *****
F1 1R46I  PAYCARDS 00066 3 D A      7 1 CHEX    *****
F1 1R46I  PAYLIST  00067 3 D A     44 1        *****
F1 1R46I  PUNCH QUEUE P D C  CARDS CC FORM     *****
F1 1R46I  PAYCARDS 00066 3 D A    233 1        *****

```

\* .. DISPLAY ALL                        Display the status of all entries in all queues.

```

F1 1R46I  READER QUEUE  P D C                  *****
F1 1R46I  DISK DUMP 00068 9 L A                 *****
F1 1R46I  PAUSE     00070 9 L A                 *****
F1 1R46I  DITTO     00071 3 L A                 *****
F1 1R46I  PRINTLOG 00069 1 L A                 *****
F1 1R46I  LIST QUEUE P D C  PAGES CC FORM     *****
F1 1R46I  PAUSE     00070 9 D A      1 1        *****
F1 1R46I  PAYINPUT 00063 3 D A      3 1        *****
F1 1R46I  PAYEDIT  00064 3 D A     75 1        *****
F1 1R46I  PAYSORT  00065 3 D A      5 1        *****
F1 1R46I  PAYCARDS 00066 3 D A      7 1 CHEX    *****
F1 1R46I  PAYLIST  00067 3 D A     44 1        *****
F1 1R46I  PRINTLOG 00069 1 D A      5 1        *****
F1 1R46I  PUNCH QUEUE P D C  CARDS CC FORM     *****
F1 1R46I  PAYCARDS 00066 3 D A    233 1        *****

```

## ALTER Command

The ALTER command is used to change the various parameters of a job entry after it has been logged in POWER/VS. The operands are the same as those specified in the JOB, LST, and PUN statements and have the same format. Any number of parameters can be modified by one command. However, the remote user can alter attributes of only those jobs that were submitted by himself or were routed to him.

If an input queue entry is specified and an execution processor is currently processing this entry, message 1R88I is issued.

The priority attribute of an entry in the read queue is identical to that of the corresponding entries in the write queues. On the other hand, the disposition and class attributes may be totally different in corresponding read and list/punch entries, since input disposition and class are defined in the JOB statement, whereas output disposition and class are defined in the LST and PUN statements.

The COPY and REMOTE parameters apply to the output only. When applied to a read entry, they are ignored. If a write entry is specified in the ALTER command and a write task is currently processing the entry, only the number-of-copies can be altered. ("number-of-copies" cannot be altered if an entry is segmented and the first segment has been deleted from the queue.) If any other parameter is specified, message 1R31I is issued and the command is ignored. The same message is issued if the output is segmented and the first segment is already deleted from the output queue.

No confirmative message is issued after the ALTER command. DISPLAY can be used for verification.

```
* .. ALTER      queue { jobname [jobnumber] }
                  ALL
                  *abc

                  [,PRI=priority]
                  [,DISP=disposition]
                  [,CLASS=class]
                  [,COPY=number-of-copies]
                  [,REMOTE=remid]
```

### Operands

**queue.** Specifies the queue for which the command is to be executed:

LST = the list queue  
PUN = the punch queue  
RDR = the read queue

**jobname.** Specifies the name by which the job entry is known to POWER/VS. Jobname can be two to eight alphanumeric characters.

**jobnumber.** Specifies the number assigned to the job entry by POWER/VS. The jobnumber may be one to five digits long. The DISPLAY command may be used to obtain the jobnumber.

**priority.** Specifies the priority of the job entry. Priority is specified as a single digit from 0 to 9. Nine is the highest priority.

**disposition.** Can be specified as H for hold, K for keep, L for leave, or D for delete (after processing).

**class.** Can be specified as any alphabetic character from A through Z. Also, specifications from 0 through 4 are allowed for input classes.

**remid.** Can be specified as any number from 0 to 200. "0" indicates the central location.

Only the "to" remote-id can be modified with an ALTER command; the "from" remote-id cannot be modified.

**number-of-copies.** Specifies the number of copies to be produced after the command is issued, including the one in progress.

*Example:*

```
* .. ALTER LST,PAYROLL,DISP=D,CLASS=B
```

Alter the disposition of the output of job PAYROLL to D, and alter the class to B.

## DELETE Command

The DELETE command is used to remove one or more job entries from the specified queue. Job entries being processed by an execution processor or write task are not affected by this command. The remote user can delete only jobs or messages that were submitted by himself or were routed to him.

```
* .. DELETE { queue,jobname[,jobnumber]
              { queue,ALL
              { queue,class
              { queue,*abc
              { MSG[,n]
```

### Operands

**queue.** Specifies the queue for which the command is to be executed:

LST = the list queue  
PUN = the punch queue  
RDR = the read queue

**jobname.** Specifies the name by which the job entry is known to POWER/VS. Jobname can be two to eight alphameric characters.

**jobnumber.** Specifies the number assigned to the job entry by POWER/VS. The jobnumber is from one to five digits long. The DISPLAY command can be used to obtain the jobnumber.

**ALL.** All job entries are to be deleted from the specified queue.

**class.** All job entries of a certain class are to be deleted from the specified queue.

**\*abc.** All job entries with the same first n characters are to be deleted. "abc" represents any combination of up to seven alphameric characters.

**MSG,n.** ALLUSERS-type message is to be deleted, where "n" is the message number. The message number can be obtained with a DISPLAY MSG command. If n is omitted, all ALLUSERS-type messages that were entered from this remote-id are deleted.

*Examples:*

```
* .. DELETE RDR,ALL      Delete the entire read queue.
* .. DELETE LST,B        Delete all entries in the list queue with a
                           class of B.
* .. DELETE MSG          Delete all ALLUSERS-type messages entered by
                           this remote-id.
```

## RELEASE Command

The RELEASE command is used to take one or more job entries out of the hold or leave state and make them available for processing. After processing, a job entry originally in the leave state returns to the leave state; a job in the hold state is deleted from the queue.

If a RELEASE command is issued for a job entry that is not in the hold state or leave state, a message is issued and the command is ignored.

The remote user can release only those jobs that were submitted by himself or were routed to him.

```
* .. RELEASE      { queue,jobname[,jobnumber]
                   { queue[,ALL]
                   { queue,class
                   { queue,*abc }
```

### Operands

**queue.** Specifies the queue for which the command is to be executed:

LST = the list queue  
PUN = the punch queue  
RDR = the read queue

**jobname.** Specifies the name by which the job entry is known to POWER/VS. Jobname can be two to eight alphanumeric characters.

**jobnumber.** Specifies the number assigned to the job entry by POWER/VS. The jobnumber is one to five digits long. The DISPLAY command can be used to obtain the number.

**All.** All job entries are to be released from the specified queue.

**class.** All job entries of a certain class are to be released from the specified queue.

**\*abc.** All job entries with the same first n characters are to be released. "abc" represents any combination of up to seven alphanumeric characters.

### Examples:

```
* .. RELEASE LST,PAYROLL  Release the entry "PAYROLL" from list queue.
* .. RELEASE RDR,ALL      Release the entire read queue.
* .. RELEASE LST,*PAY     Release all entries in the list queue that begin
                           with PAY.
```

## MISCELLANEOUS COMMANDS

### BRDCST Command

The BRDCST command enables the remote operator to communicate with the central operator and other remote users. Messages can be submitted for one specific user or for all users.

Messages generated by the BRDCST command for specific remote-ids are stored in the list queue and displayed only on request.

Messages of the ALLUSERS-type are queued and given a message sequence number. The DISPLAY MSG command can be used to obtain the actual number.

```
* .. BRDCST    remotid,'text'
```

#### *Operands*

**remotid.** Specifies the remote-id to which the command applies. Remote users can be identified by numbers from 1 to 200. A specification of 0 indicates the central location, and a specification of ALLUSERS indicates all users.

**'text'.** Specifies the text of the message. The message can consist of from 1 to 40 characters enclosed in single quotation marks. A single quotation mark within the message must be written as two quotation marks.

#### *Examples:*

```
* .. BRDCST 150,'Payroll will be completed in 30 min.'
```

Broadcast the message to the user whose  
remote-id is 150.

```
* .. BRDCST 0,'Inventory is finished'
```

Broadcast the message to the central operator.

### SETUP Command

The SETUP command can be entered only after a message requesting a special forms mount is displayed. It causes n pages of spooled list data to be printed. All printable characters of the page are replaced by the character X. During the printing of the n pages, the operator can manually align the printer forms.

The number of pages printed by SETUP are counted and stored separately in the list account record.

When the RESTART command is issued to resume printing, the pages are printed again with their original contents.

```
* .. SETUP    [n]
```

**n.** One or two digits that specify the number of pages to be printed. If n is omitted, only one page is printed.

Example:

\* .. SETUP 2

Print two pages to allow forms alignment.

1st Page {

XXXX XXX XX

XXX XX XXX XXXXX XXXXXXXXXXXXXXXXXXXX XXXXXXXX XXXXXXXXXXXX

XXX XXXXXXXX XXXXXXXXXXXXXXXX XXX XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX XXXXXXXX XXXXXXXX

X XXXX

2nd Page {

XXXX XXX XX

XXX XX XXX XXXXX XXXXXXXXXXXXXXXXXXXX XXXXXXXX XXXXXXXXXXXX

XXXX XXXXXXXX XXXXXXXXXXXXXXXX XXX XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX XXXXXXXX XXXXXXXX

X XXXX

\* ...GO LST

Pages are printed again with their original contents. Printing continues.

1st Page {

SEPT 30, 74

PAY TO THE ORDER OF.....ONE EMPLOYEE \*\*\*123.45

ONE HUNDRED TWENTY-THREE AND ----- 34/100 DOLLARS

# 2001

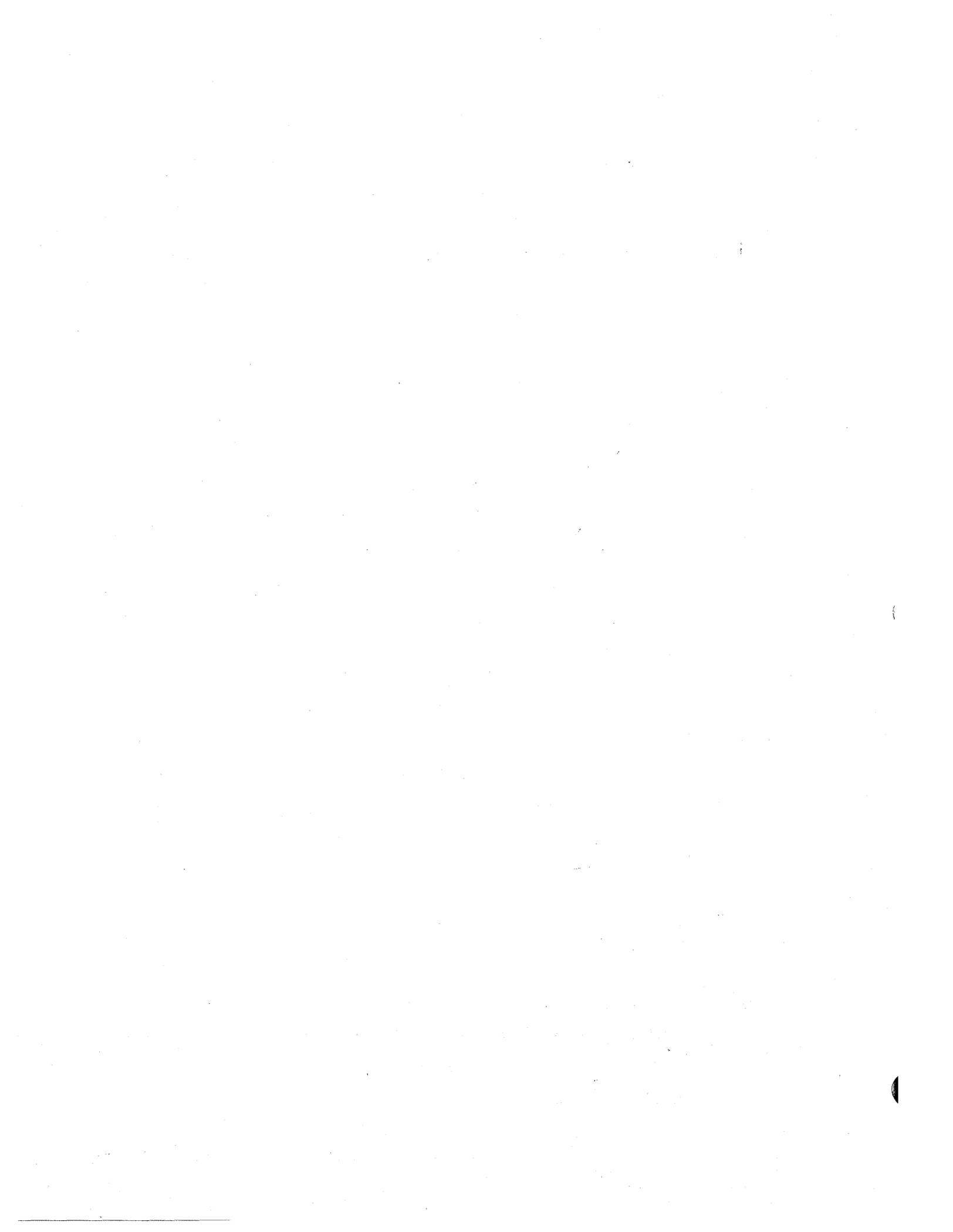
2nd Page {

SEPT 30, 74

PAY TO THE CRDER OF.....ANOTHER EMPLOYEE \*\*\*543.21

FIVE HUNCRED FORTY-THREE AND ----- 21/100 DOLLARS

# 2002



## APPENDIX A—ACCOUNT RECORDS

The following tables describe the contents of the record types that make up the account file. (See "Account File" in "POWER/VS Concepts and Facilities".) The last column, column F, indicates the format of each field in the record:

- a = alphanumeric
- b = binary
- p = packed decimal

If JECL is *not* used to delimit POWER/VS jobs, the jobname and user information placed in the account records are taken from the // JOB statement.

### READ ACCOUNT RECORD

A read account record is created for each read entry. Whether or not the entry has actually been placed in the queue file is indicated by the POWER/VS cancel code.

The read account record is written to the POWER/VS account file after the corresponding read entry is processed by the read task.

Read account records are not created for a writer-only partition.

Bytes	Name	Description	F
00-07	ARDY	Date, in format specified in DOS/VS SYSGEN (mm/dd/yy or dd/mm/yy)	a
08-11	ARST	Start time of read (OHHMMSSF; F=sign)	p
12-15	ARET	Stop time of read (OHHMMSSF; F=sign)	p
16-31	ARUI	16 bytes of user information from * \$\$ JOB or // JOB card	a
32-39	ARNM	POWER/VS jobname from * \$\$ JOB or // JOB card	a
40-41	ARNO	Jobnumber assigned by POWER/VS	b
42	ARRI	Record identifier (R)	a
43	ARCN	POWER/VS cancel code (See Appendix B)	b
44		Reserved	
45-47	ARCU	Reader device address	b
48	ARFJ	"From" remote-id	b
49	ARTJ	"To" remote-id (note 1)	b
50	ARCL	Input class	a
51	ARPY	Input priority number	a
52-55	ARNR	Number of records read (note 2)	b
56-57	ARNT	Number of tracks for input storage	b

#### Notes

1. The "to" remote-id is a dummy entry in the read account record. It is copied from the "from" remote-id.
2. The number of records read includes the number of records added or deleted through the RDREXIT routine.

## LIST ACCOUNT RECORD

A list account record is created for each list entry created by the execution list task.

A list account record is written to the POWER/VS account file after the list entry is printed.

Bytes	Name	Description	F
00-07	ALDY	Date, in format specified in DOS/VS SYSGEN (mm/dd/yy or dd/mm/yy)	a
08-11	ALST	Start time of list (0HHMMSSF; F=sign)	p
12-15	ALET	Stop time of list (0HHMMSSF; F=sign)	p
16-31	ALUI	16 bytes of user information from * \$\$ JOB or // JOB card	a
32-39	ALNM	POWER/VS jobname from * \$\$ JOB card or // JOB card	a
40-41	ALNO	Jobnumber assigned by POWER/VS	b
42	ALRI	Record identifier (L)	a
43	ALCN	POWER/VS cancel code (See Appendix B)	b
44		Reserved	
45-47	ALCU	Printer device address	b
48	ALFJ	"From" remote-id	b
49	ALTJ	"To" remote-id	b
50	ALCL	Printed output class	a
51	ALPY	Printed output priority number	a
52-55	ALNR	Number of lines printed	b
56-57	ALNT	Number of tracks for output storage (note 2)	b
58	ALSN	Job subnumber assigned by POWER/VS (note 3)	b
59	ALNC	Number of printed copies (note 4)	b
60-63	ALFI	Printed forms identification	a
64-67	ALNA	Number of extra records printed due to PRESTART, PSETUP, or JSEP	b
68-69	ALNP	Number of pages printed (skips to channel 1)	b
70-71	ALNE	Number of extra pages printed due to PRESTART, PSETUP, or JSEP	b

### Notes.

1. If present, the user information is copied into each list entry generated for the job.
2. Only for spooling to disk; when spooling to tape, this field is zero.
3. For each list entry within a POWER/VS job, a separate list account record is created with a unique job subnumber.
4. Total for all copies of that segment.

## PUNCH ACCOUNT RECORD

A punch account record is created for each punch entry created by the execution punch task.

A punch account record is written to the POWER/VS account file after the punch entry is punched.

Bytes	Name	Description	F
00-07	APDY	Date, in format specified in DOS/VS SYSGEN (mm/dd/yy or dd/mm/yy)	a
08-11	APST	Start time of punch (0HHMMSSF; F=sign)	p
12-15	APET	Stop time of punch, (0HHMMSSF; F=sign)	p
16-31	APUI	16 bytes of user information from * \$\$ JOB or // JOB card	a
32-39	APNM	POWER/VS jobname from * \$\$ JOB card or // JOB card	a
40-41	APNO	Jobnumber assigned by POWER/VS	b
42	APRI	Record identifier (P)	a
43	APCN	POWER/VS cancel code (See Appendix B)	b
44	APRJ	Reserved	b
45-47	APCU	Punch device address	b
48	APFJ	"From" remote-id	b
49	APTJ	"To" remote-id	b
50	APCL	Punched output class	a
51	APPY	Punched output priority number	a
52-55	APNR	Number of records punched	b
56-57	APNT	Number of tracks for output storage (note 2)	b
58	APSN	Job subnumber assigned by POWER/VS (note 3)	b
59	APNC	Number of punched copies (note 4)	b
60-63	APFI	Punch forms identification	a
64-67	APNA	Number of extra cards punched due to PRESTART or JSEP	b

### Notes.

1. If present, the user information is copied into each punch entry generated for the job.
2. Only for spooling to disk; when spooling to tape this field is zero.
3. For each punch entry within a POWER/VS job, a separate punch account record is created with a unique job subnumber.
4. Total for all copies of that segment.

## EXECUTION ACCOUNT RECORD

One execution account record per DOS/VS job step is constructed. It contains information passed to the account file by the DOS/VS job accounting interface, plus information produced by the POWER/VS accounting routine. If the job or job step is canceled before completion, statistics reflect processing up to that time.

Bytes	Name	Description	F
00-07	AEDY	Date, in format specified in DOS/VS SYSGEN	a
08-11	AEST	Start time of job step (0HHMMSSF; F=sign)	p
12-15	AEET	Stop time of job step (0HHMMSSF; F=sign)	p
16-31	AEUI	16 bytes of user information from * \$\$ JOB card	a
32-39	AENM	POWER/VS jobname or "AUTONAME"	a
40-41	AENO	Jobnumber assigned by POWER/VS	b
42	AERI	Record identifier (E)	a
43	AECN	POWER/VS cancel code (See Appendix B)	b
44		Reserved	
45-47		Reserved	
48	AEFJ	"From" remote-id	b
49	AETJ	"To" remote-id (note 1)	b
50	AECL	Class	a
51	AEPY	Priority	b
52-55		Number of lines spooled	b
56-59		Number of cards spooled	b
60-61		Number of pages spooled	b
62-63		Length of DOS/VS SIO accounting table	b
64-65		Length of total account record	b
66-71		Reserved	
72-79		DOS/VS jobname from // JOB card	a
80-95		16 bytes of user information from // JOB card	a
96-97		Partition-id in EBCDIC format	a
98		DOS/VS cancel code	b
99		Type of record; S = job step, L = last step	a
100-103		Reserved	
104-111		Phase name, taken from execute card	a
112-115		End address of active program phase, from COMREG	
116-119		CPU time elapsed in a job step, counted in 300ths of a second	b
120-123		Overhead time (in 300ths of a second); system overhead time divided among running partitions	b
124-127		All-bound time (in 300ths of a second); system wait state time divided among running partitions	b
128-		SIO tables; 6 bytes for each device specified at system generation, as follows: 2 bytes for device address (Ocuu), 4 bytes for count of SIOs in current job step (note 2)	b
		Overflow byte; normally X'20', but X'30' if more devices are used within a partition than specified at DOS/VS system generation	b
		User accounting information (note 3)	

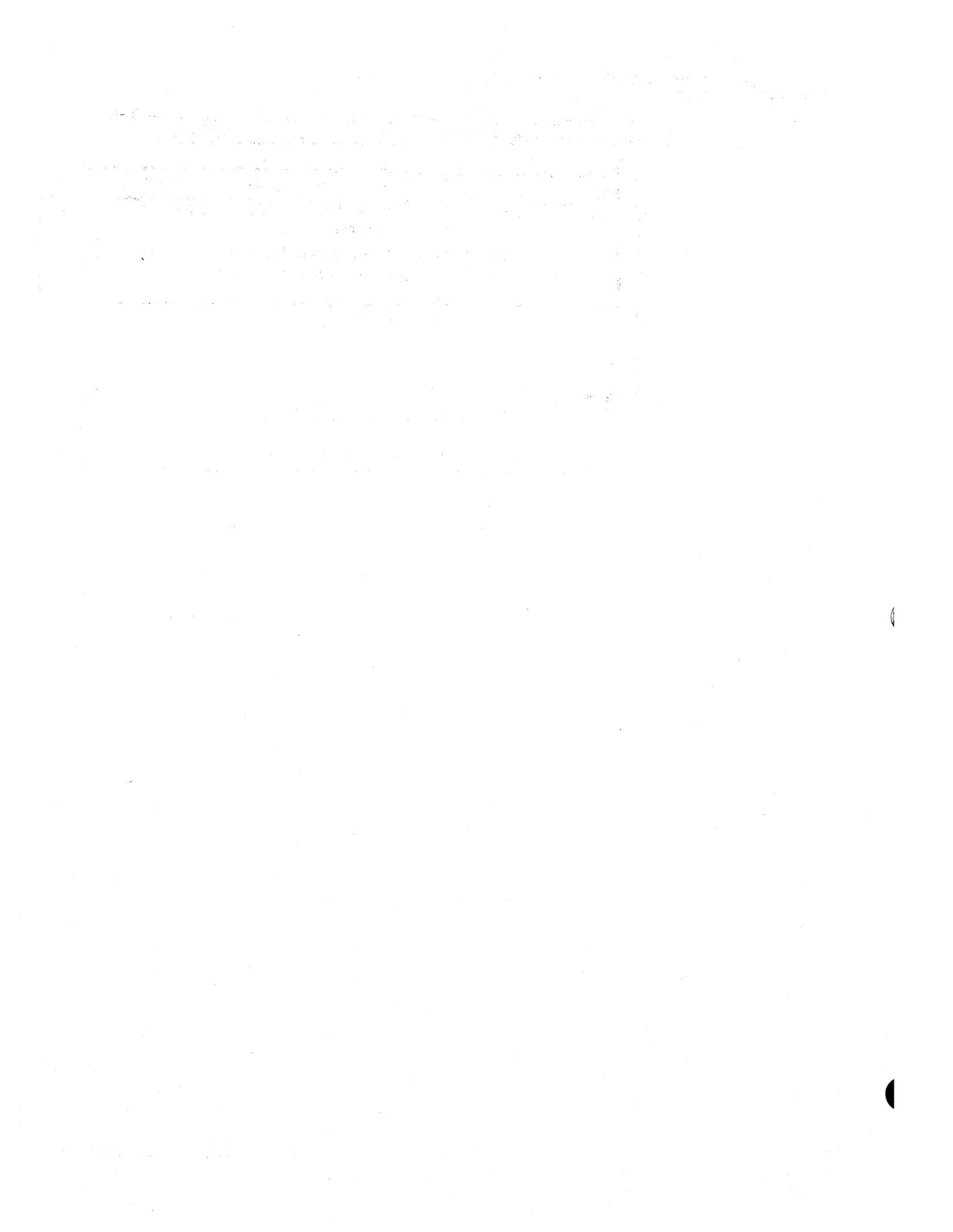
#### Notes.

1. The "to" remote-id is a dummy entry in the execution account record; it is copied from the "from" remote-id.
2. POWER/VS updates the SIO tables in the execution account record with the number of start I/Os it intercepted.
3. Appendix F explains how to add user account information to an execution account record through a user-written routine (\$JOBACCT).

## LINE ACCOUNT RECORD

A line account record is written when each RJE user session is terminated. The SIGNOFF code indicates whether the session was normally or abnormally terminated.

Bytes	Name	Description	F
00-07	ATDY	Date, in format specified in DOS/VS SYSGEN (mm/dd/yy) or (dd/mm/yy)	a
08-11	ATST	SIGNON time (0HHMMSSF; F=sign)	p
12-15	ATET	SIGNOFF time (0HHMMSSF; F=sign)	p
16-31	ATUI	16 bytes of user information from the signon command	a
32-39	ATNM	Line password	a
40-41		Reserved	
42	ATRI	Record identifier (T)	a
43	ATCN	SIGNOFF code (See Appendix C)	b
44	ATEJ	Remote identifier	b
45-47	ATCU	Line address	b

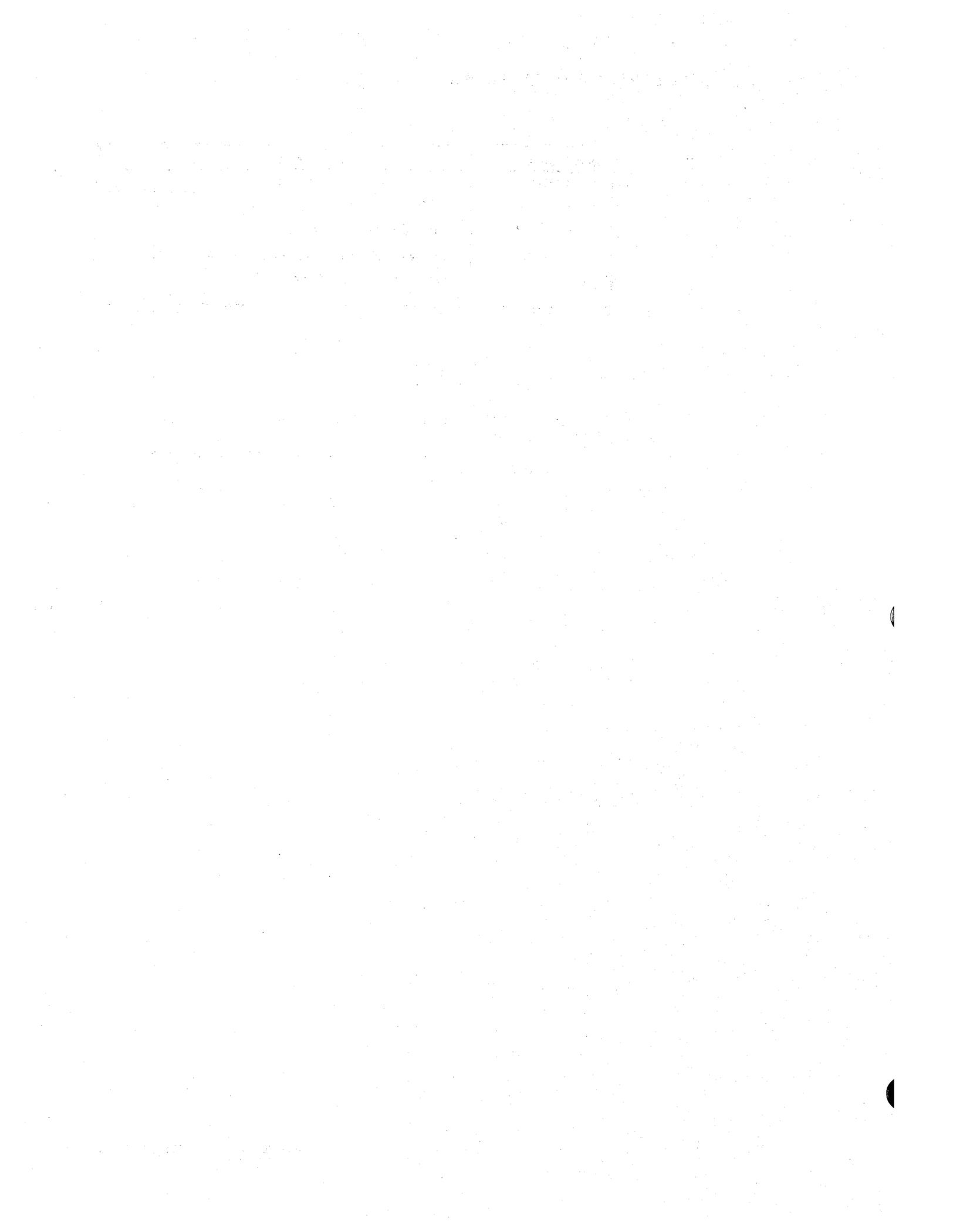


## APPENDIX B—POWER/VS CANCEL CODES

Cancel Code	Condition
X'10'	Normal end of POWER/VS job or task (note 1)
X'30'	PSTOP has been issued (note 2)
X'40'	PFLUSH has been issued
X'60'	POWER/VS job has been flushed via RDREXIT
X'70'	cancelled due to I/O error

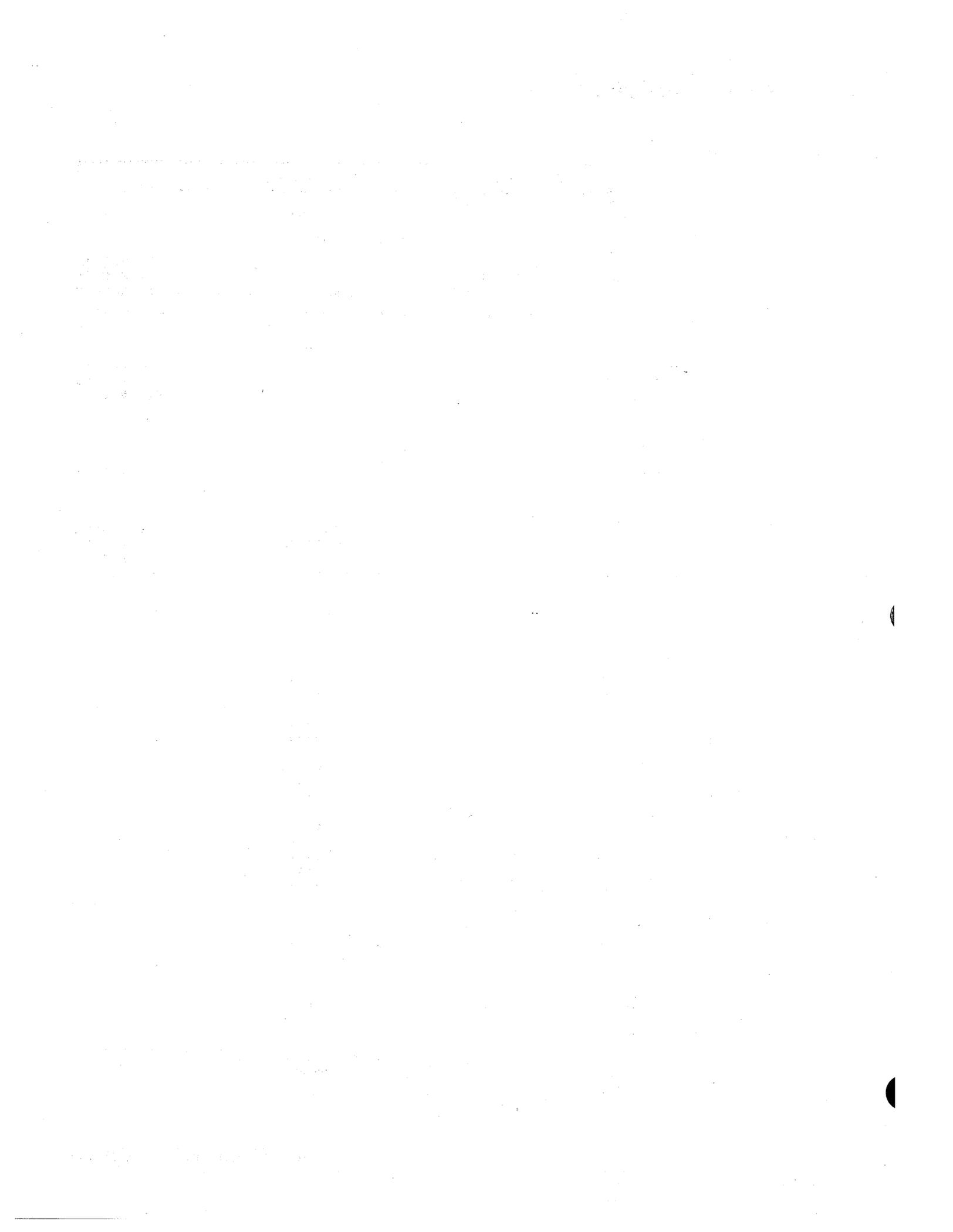
### Notes

1. Although *no* abnormal POWER/VS termination occurred, the DOS/VS jobs associated with the job entry *could have* been canceled via DOS/VS.
2. The PSTOP cancel code is not stored in an account record if the EOJ option was specified in the PSTOP command.



## APPENDIX C—RJE SIGNOFF CODES

SIGNOFF Code	Condition
X'01'	SIGNON/SIGNOFF card read
X'02'	Line stopped by central operator
X'04'	SIGNOFF forced because idle time limit exceeded
X'08'	SIGNOFF forced because of irrecoverable I/O error



# APPENDIX D—STATUS REPORT FORMAT

## FILE STATISTICS

POWER/VS STATUS REPORT

08/06/74

01/17/43

QUEUE FILE IJQFILE

TOTAL NUMBER OF QUEUE RECORDS 618 RECORDS  
NUMBER OF FREE QUEUE RECORDS 604 RECORDS  
MAXIMUM NUMBER OF QUEUE RECORDS USED IN PRESENT SESSION 14 RECORDS

DATA FILE IJDFILE

TOTAL NUMBER OF TRACKS 1236 TRACKS  
TRACK GROUP SIZE 2 TRACKS  
DATA BLOCK SIZE 516 BYTES

ACCOUNT FILE IJAFILE

TOTAL NUMBER OF TRACKS 60 TRACKS  
PERCENTAGE OF FILE THAT IS FILLED 4 %

REAL STORAGE ALLOCATED TO POWER/VS PARTITION 11 PAGES  
NUMBER OF TIMES TASKS WERE WAITING FOR REAL STORAGE 0 TIMES  
MAXIMUM NUMBER OF REAL PAGES FIXED 5 PAGES  
MAXIMUM NUMBER OF TASKS ACTIVE AT ONE POINT IN TIME 2 TASKS

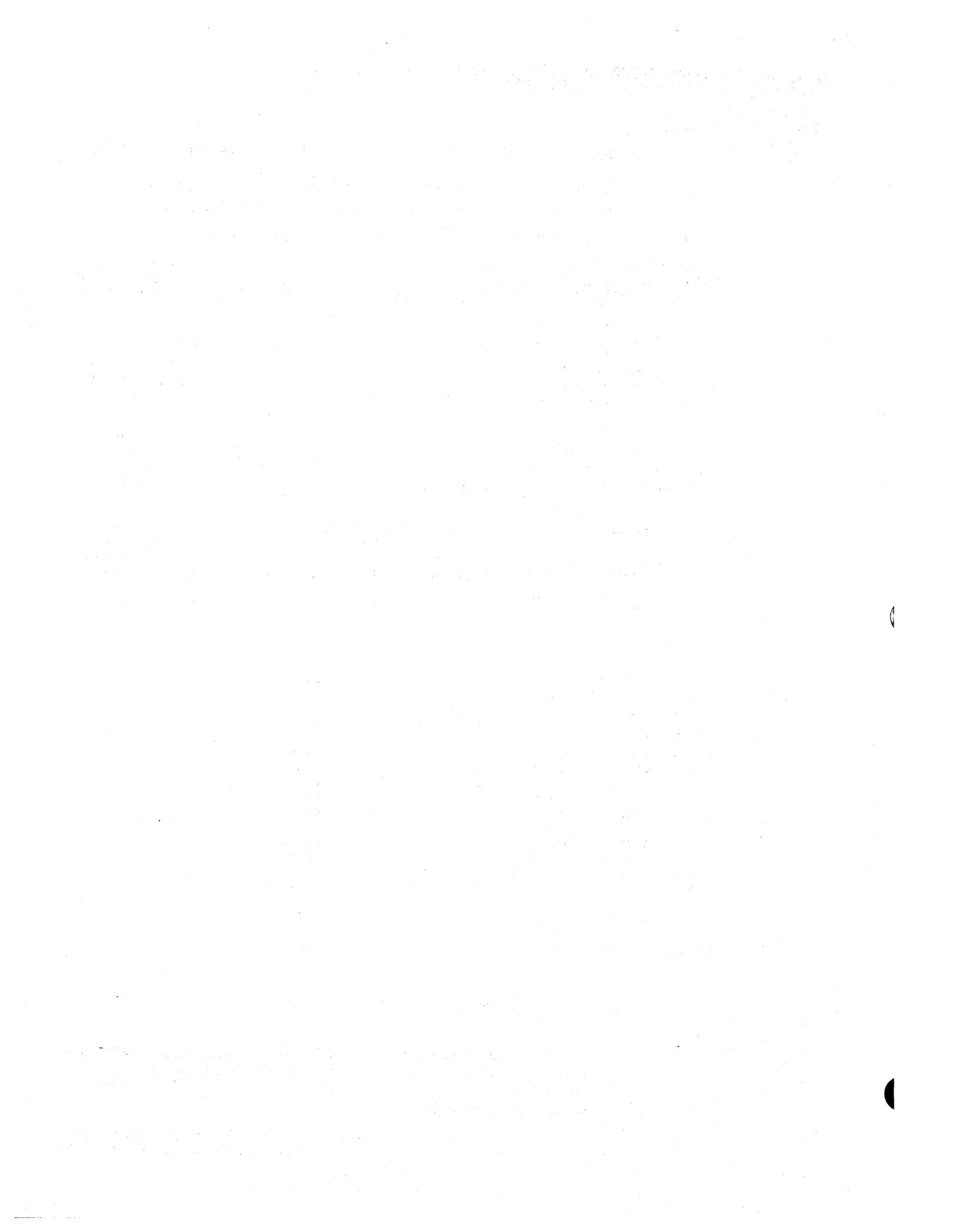
```
1R46I READER QUEUE P D C *****  
1R46I DISKCUMP 00030 9 L A *****  
1R46I PAUSE 00032 9 K A *****  
1R46I ONE 00020 3 D A *****  
1R46I TWO 00021 3 D A *****  
1R46I THREE 00022 3 D A *****  
1R46I SIX 00025 3 D A *****  
1R46I SEVEN 00026 3 D A *****  
1R46I EIGHT 00027 3 D A *****  
1R46I NINE 00028 3 D A *****  
1R46I TEN 00029 3 D A *****  
1R46I DITTO 00033 3 L A *****  
1R46I PRINTLOG 00031 1 K A *****  
1R46I FOUR 00023 3 D B *****  
1R46I FIVE 00024 3 D B *****
```

1R46I LIST QUEUE NOTHING TO DISPLAY \*\*\*\*\*

1R46I PUNCH QUEUE NOTHING TO DISPLAY \*\*\*\*\*

If no POWER/VS account support is provided (either ACCOUNT=NO in POWER macro and/or JA=NO in DOS/VIS FOPT macro), the account file statistics are replaced by the message:

NO ACCOUNT SUPPORT



## APPENDIX E—COMPARISON OF DOS/VS POWER AND POWER/VS

### GENERAL DIFFERENCES

The following general differences exist between POWER/VS and DOS/VS POWER:

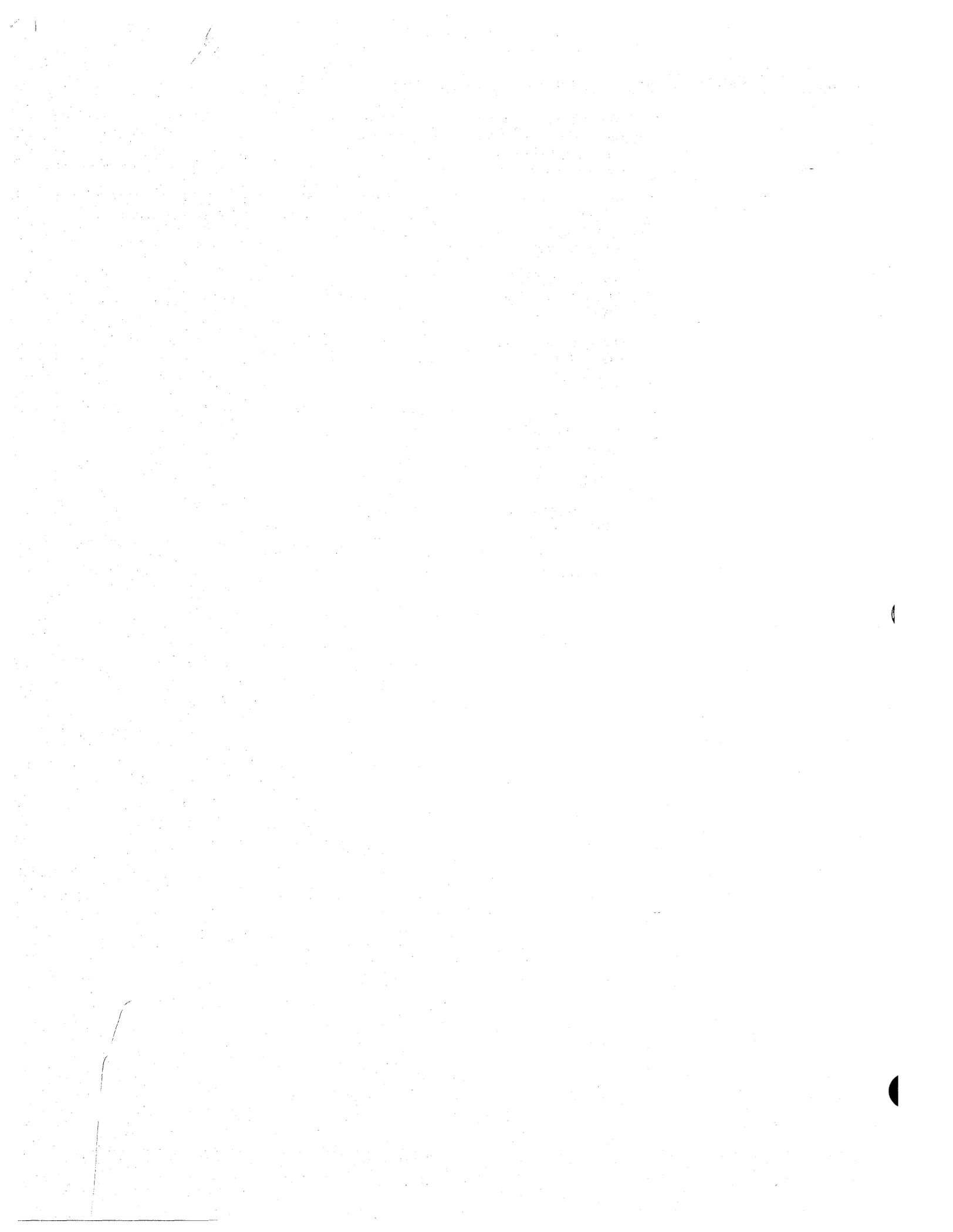
- A new set of POWER/VS messages has been created.
- POWER/VS direct access files are not compatible with DOS/VS POWER disk files.
- One read queue, one list queue, and one punch queue are maintained for the whole system.
- POWER/VS uses its own access methods. Therefore, programs that depend on any access method characteristic of earlier versions of POWER should be carefully reviewed before they are used under POWER/VS.
- DOS/VS POWER JECL statements are fully supported by POWER/VS.
- A new set of operator commands has been created for POWER/VS. The corresponding DOS/VS POWER commands are still valid, with the exception of the H, E, M, O, and Z commands.
- A completely new set of remote commands has been defined for POWER/VS. The new commands are incompatible with DOS/VS POWER remote commands.
- The DOS/VS POWER AUTOSTART feature has been replaced by a new AUTOSTART feature.
- Remote users are defined by remote-id (number) not user-id (name).
- The IBM 2311 Disk Storage Drive is not supported as a spooling device.
- Unit-record I/O from subtasks in a multitasking environment is fully supported.

**LOCAL OPERATION**

	<b>DOS/VS POWER</b>	<b>POWER/VS</b>
<b>Run mode</b>	Real	Virtual
<b>Input classes</b>	Partition-dependent	Both partition-dependent and partition-independent supported.
<b>Output classes</b>	Partition-dependent (Each partition has its own list and punch queues)	Partition-independent (One list queue and one punch queue for all partitions)
<b>Output segmentation</b>	No	Yes
<b>Generation</b>	Assembly required	Short or no assembly required
<b>Initiation</b>	Automatic start-up of one input reader, one list writer, and one punch writer	Automatic start-up of readers/writers and supported partitions
<b>Job accounting</b>	Spools DOS/VS and accounting data separately	DOS/VS and POWER/VS accounting data merged
<b>Printer forms change</b>	Start of job, only	At start or within job
<b>Permanent I/O error</b>	POWER may be canceled	Only the POWER/VS task affected is canceled
<b>Flush with hold</b>	No	Yes
<b>FCB support</b>	Partial	Yes
<b>Forms setup assistance</b>	Partial	Yes
<b>Queue management commands</b>	Yes	Yes
<b>Hot local reader</b>	Yes	Yes
<b>Warm start</b>	Yes	Yes
<b>Page forward/backward</b>	Yes	Yes
<b>Output page/card limit check</b>	Yes	Yes
<b>Job separator pages</b>	SYSGEN option, only	Yes; can be specified in a job stream
<b>Dummy devices</b>	Required for support of multiple partitions	Not required for support of multiple partitions, but allows 8 dummy printers and 8 dummy punches to be used in each partition.
<b>Multiple Printer/Punch Support</b>	No	YES; up to 8 printers and 8 punches

**REMOTE OPERATION**

	<b>DOS/VS POWER</b>	<b>POWER/VS</b>
<b>No. of RJE lines</b>	Up to 5	Up to 25 ✓
<b>RJE TP access method</b>	BTAM	RTAM
<b>Hot RJE reader/writer</b>	No	Yes
<b>RJE accounting</b>	No	Yes
<b>Leased/switched line support</b>	Yes	Yes
<b>ASCII RJE support</b>	No	Yes
<b>Transparency feature</b>	Mandatory if punching	Needed only if transmitting object decks
<b>RJE output routing</b>	Partial	Yes
<b>Timeout</b>	Yes	Yes; no maximum time limit
<b>RJE writer classes</b>	No	Yes
<b>Print separators</b>	No	Yes



## APPENDIX F—USER ACCOUNTING ROUTINE (\$JOBACCT)

\$JOBACCT is an optional user-written routine through which user accounting information can be added to the DOS/VS accounting table in the supervisor. Subsequently, this is merged with POWER/VS accounting information in the POWER/VS partition to form a complete execution account record.

\$JOBACCT is called by \$JOBCTLN upon reading either a /& statement or the second and subsequent // EXEC statements of a user job containing more than one job step; that is, \$JOBACCT is called at the end of each job or job step. Register 14 contains the address of the return point.

The user can, in \$JOBACCT, test byte 164, bit 0 of the communications region. If the bit is *on*, the partition in which the job step executed is under control of POWER/VS, and POWER/VS accounting is supported.

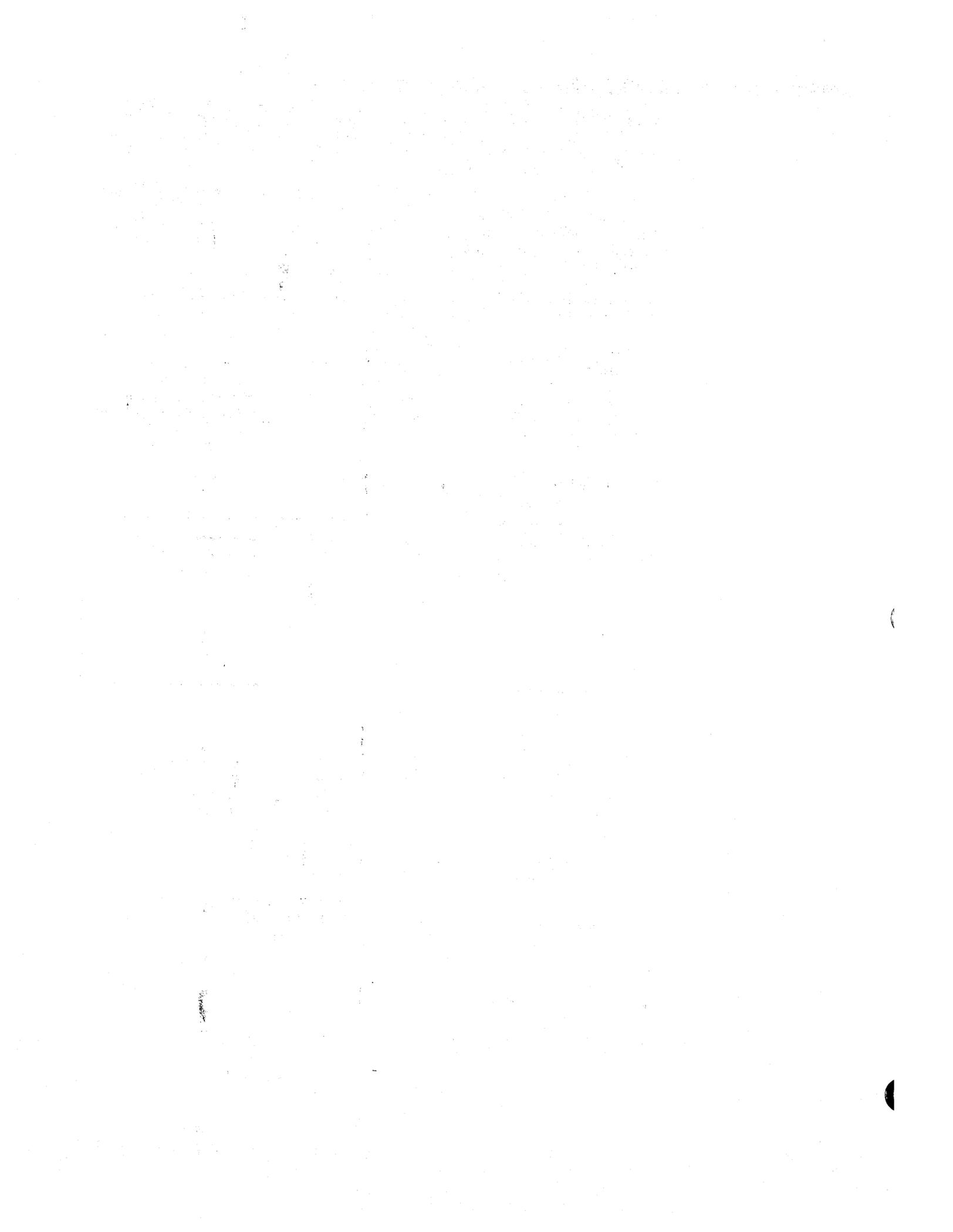
To add additional information to the standard execution account record, the PUTACCT macro (supplied with POWER/VS) can be used. It has the format:

PUTACCT (r1),(r2)

Register r1 must point to the area in \$JOBACCT containing the user account information, and register r2 must contain the length of the area. The total length of the execution account record cannot exceed 2000 bytes. PUTACCT acts as a NOP if the *partition* is not supported by POWER/VS or if *accounting* is not supported by POWER/VS.

After re-entering \$JOBCTLN from \$JOBACCT, the execution account record is written to the POWER/VS file.

Refer to the *DOS/VS System Management Guide*, GC33-5371 for further information about DOS/VS \$JOBACCT routines.



## APPENDIX G—RDREXIT ROUTINE

The RDREXIT routine permits each DOS/VS JCL statement or POWER/VS JECL statement to be examined as it is read by POWER/VS. Any statement can be changed or deleted, and other statements can be inserted. These changes do not permanently alter the JCL or JECL.

Some of the ways the RDREXIT facility can be applied are: changing the disposition of a certain job class to *hold* during prime shift; changing the disposition of output to *spool* during the day or to *write direct* (i.e., no spooling) at night; causing a job to run *real* in one instance and *virtual* at other times; establishing a SIZE parameter on EXEC statements; gathering job accounting information.

When POWER/VS is started, the phase (routine) specified in the RDREXIT parameter of the POWER macro is loaded into the POWER/VS partition. A POWER/VS read task gives control to this routine via a BALR 14,15 instruction after reading a DOS/VS JCL statement or POWER/VS JECL statement. The address of the statement is passed in register 0 and the length in register 1. To exit from the routine, control must be returned to the read task via a BR 14 instruction. Between entry and exit from the routine, no operation may be performed by the RDREXIT that might cause a wait condition for the POWER/VS partition. The load address of the phase is not fixed, and therefore must be self-relocating or the relocating loader option must have been specified during DOS/VS system generation.

Before returning to the POWER/VS read task, a return code must be supplied in register 15. The return codes are:

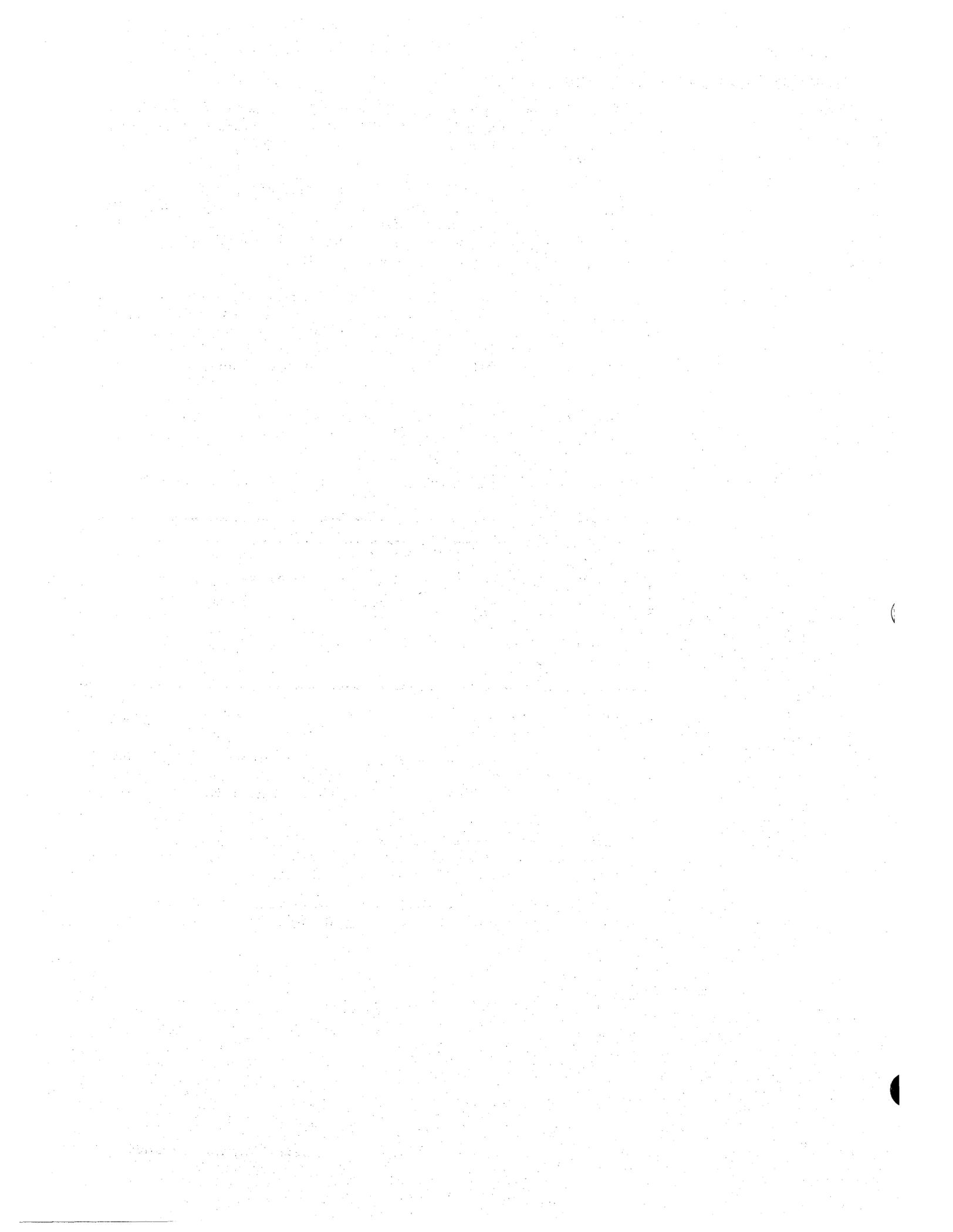
Return code	Meaning
X'00'	Normal; the current statement will be processed by POWER/VS.
X'04'	Delete; the current statement will be ignored by POWER/VS.
X'08'	Insert; the new statement provided will be processed by POWER/VS.
X'0C'	Flush the DOS/VS job.
X'10'	Flush the POWER/VS job.

When new statements are added (return code X'08'), the address of the statement must be passed in register 0, and its length passed in register 1.

Any number of statements can be inserted. The original statement is re-presented after each inserted statement has been processed. When all the insertions have been made, a return code of X'00' or X'04' is placed in register 15 to accept or delete the original statement.

The RDREXIT routine must *not* alter the contents of registers 10, 11, 12, or 13; these registers are reserved for POWER/VS internal use. Register 11 points to the task control block of the read task and can be used to identify the task.

If ACCOUNT=YES was specified, *number of records read* in the read account record will include records added or deleted through the RDREXIT routine.



## APPENDIX H – SAMPLE JOB STREAMS

```

// JOB PAYINPUT
*
// EXEC PAYSTEP1
*
/&
// JOB PAYECIT
*
// EXEC PAYSTEP2
*
/&
// JOB PAYS CRT
*
// EXEC PAYSTEP3
*
/&
* ** JOB PAYCARDS
// JOB PAYCARDS
*
// EXEC PAYSTEP4
*
* ** LST FNC=CHEX
// EXEC PAYSTEP5
*
/&
* ** EOJ
* ** JOB PAYLIST
* ** LST COPY=3,DISP=L,PRI=9
// JOB PAYLIST
*
// EXEC PAYSTEP6
*
/&
* ** EOJ
* ** JOB JNM=DISKDUMP,PRI=9,CLASS=A,DISP=L
// JOB DISK DUMP - DISPLAY POWER/VS FILE CONTENTS
// ASSGN SYS000,SYSRES POWER/VS ACCOUNTING FILE
// ASSGN SYS001,SYSRES POWER/VS QUEUE FILE
// ASSGN SYS002,SYSRES POWER/VS DATA FILE
// ASSGN SYS003,DISK,VOL=111111,SHR OPTIONAL 2ND EXTENT TC DATA FILE
* DUMP FUNCTION=Q/D/A/ALL/JOBNAME(,JOB-NUMBER)/EOJ
// EXEC IPW**CD
/&
* ** EOJ
* ** JOB JNM=PRINTLOG,PRI=1,DISP=K
// JOB PRINTLOG DISPLAY
// EXEC PRINTLOG
/&
* ** JOB JNM=PAUSE,DISP=K,PRI=9
// JOB PAUSEJOB
// PAUSE * YOU ARE NOW IN CONTROL; ENTER JCL *
/&
* ** JOB JNM=DITTO,DISP=L
// JOB DITTC
// EXEC DITTO
/&
* ** EOJ

```

POWER/VS Job Number	DOS/VS Job Stream	Comments
1	// JOB ONE * // EXEC JOBSTEPA * /E	DOS/VS Job, with <i>no</i> JCL changes.
2	// JOB TWO * // EXEC JOBSTEPP * * \$\$ PUN CLASS=X // EXEC JOBSTEP C * /E	No * \$\$ JOB/EOJ required for LST or PUN statements.
3	* \$\$ JOB THIRD // JOB THREE * // EXEC JOBSTEP D * /E	Optional POWER/VS JECL.
4	* \$\$ EOJ * \$\$ JOB FOURTH // JOB FOUR * // EXEC JOBSTEP E * // EXEC JOBSTEP F * /E	No * \$\$ EOJ required, if POWER/VS job is followed by * \$\$ JOB statement.
5	* \$\$ CTL CLASS=B * \$\$ JOB FIFTH // JOB FIVE * // EXEC JOBSTEP G * /E	Default CLASS changed to B.
6	* \$\$ JOB SIXTH // JOB SIX * // EXEC JOBSTEP H * /E	No * \$\$ EOJ required.
7	// JOB SEVEN * // EXEC JOBSTEP I * /E	Multiple DOS/VS jobs in one POWER/VS job. (* \$\$ JOB and * \$\$ EOJ are both required for this.)
8	* \$\$ JOB SEVENTH // JOB EIGHT * // EXEC JOBSTEP J * * \$\$ JOB EIGHTH * // EXEC JOBSTEP K * /E	Multiple POWER/VS jobs for one DOS/VS job.
9	* \$\$ EOJ * \$\$ CTL CLASS=A // JOB NINE * // EXEC JOBSTEP L * // JOB TEN * \$\$ LST FNC=8X11 * // EXEC JOBSTEP M * * \$\$ LST JSEP=2,RBS=100 // EXEC JOBSTEP N * /E	Default CLASS reset to A.
10		POWER/VS will generate the missing /E.
		Multiple LST outputs per job. (2nd report is segmented.)

APPENDIX I – JOBS FOR SUGGESTED AUTOSTART PROCEDURE

```

// JOB DISKLOAD PROGRAM CREATES POWER/V5 AUTO-START DISK FILES
// OPTION CATAL
// OPTION NCDECK,NOXREF,CATAL
// PHASE DISKLOAD,*
// EXEC ASSEMBLY
*
*           THIS PROGRAM PERFORMS A CARD-TO-DISK FUNCTION.
*
*           INPUT  = SYSIPT, UP TO /*
*
*           OUTPUT = SEQUENTIAL DISK FILE 'UOUT', ASSGND TO SYS005.
*                   RECORDS WRITTEN ARE DUPLICATES OF INPUT CARDS,
*                   FOLLOWED BY /*, /&
*
*
*           PRINT NOGEN
*           START 0
*           BALR 12,0
*           USING *,12
*           OPEN  CARD,UOUT
*           EQU   *
*           GET   CARD
*           PUT   UOUT
*           B     LOOP
*
*           ENTRY POINT
*           INITIALIZE BASE REGISTER
*           ESTABLISH ADDRESSABILITY
*           OPEN FILES
*
*           READ A CARD
*           AND COPY IT TO DISK
*           REPEAT FOR ALL INPUT CARDS UP TO /*
*
*
*           EQU   *
*           PUT   UOUT
*           MVC   RECORD,EOJ
*           PUT   UOUT
*           CLCSE CARD,UOUT
*           EOJ
*           COPY '/*' CARD
*           PUT '/&' IN I/C AREA
*           AND WRITE '/&' ON DISK
*           CLOSE FILES
*           EXIT
*
*
*           DISKREC DS 2F
*           RECORD DS CL80
*           EOJ     DC CL80'/&&'
*           IOCS COUNT FIELD
*           CARD INPUT/OUTPUT AREA
*           /& CARD IMAGE
*
*
*           CARD   DTFDI
*
*                   DEVADDR=SYSIPT,
*                   IOAREA1=RECORD,
*                   EOFADDR=END,
*                   RECSIZE=80
*
*
*           FOR 2314 OR 3330, CHANGE 'DEVICE' SPECIFICATIGN BELOW.
*
*           UOUT   DTFSO
*
*                   DEVADDR=SYS005,
*                   DEVICE=3340,
*                   BLKSIZE=88,
*                   IOAREA1=DISKREC,
*                   TYPEFLE=OUTPUT
*
*           ENC
*
*           /*
*           // EXEC LNKEDT
*           /&

```

X  
X  
X  
X  
X  
X  
X  
X

```

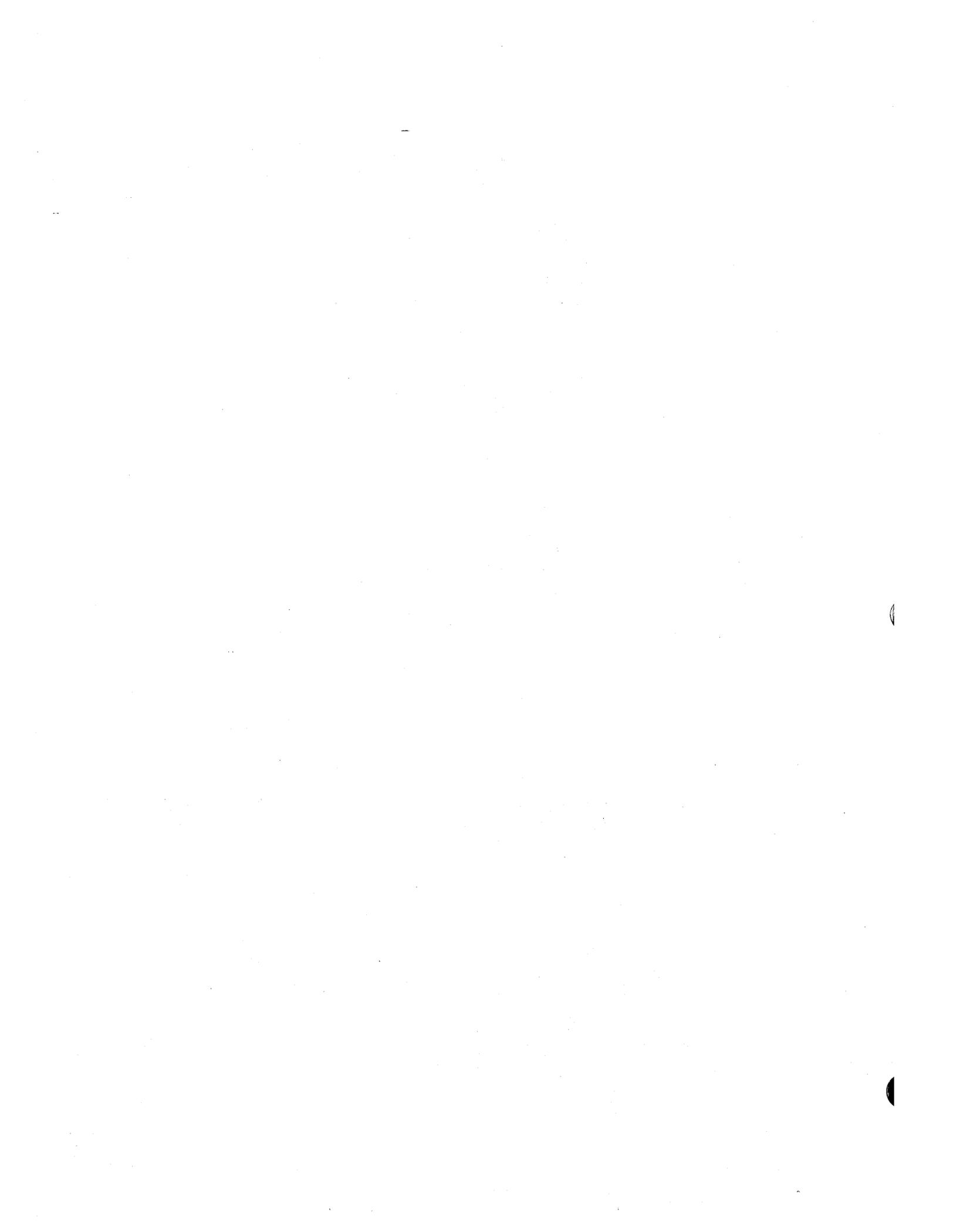
// JOB LCAC & FORMAT POWER/V5 AUTO-START DISK FILES
// ASSGN SYS005,SYSRES
// DLBL UOUT,'SYSIN.FILE.FOR.POWER/V5.START-UP.IN.F1',99/365,SD
// EXTENT SYSC05,SYSRES,1,0,7428,4      1/3 CYL, 619 - 619, CN SYSRES
// EXEC DISKLOAD
*   P O W E R / V S   A U T O - S T A R T   *
  CVCCN X'00C'
  CVCCN X'00D'
  CVCCN X'00E'
  CVCCN X'04E'
  CVCUP X'00C'
  CVCUP X'00D'
  CVCUP X'00E'
  CVCUP X'04E'
  ASSGN SYSLST,PRINTER
// JOB POWER/V5
  ASSGN SYS0C0,SYSRES POWER/V5 ACCOUNTING FILE
  ASSGN SYSC01,SYSRES POWER/V5 QUEUE FILE
  ASSGN SYS002,SYSRES POWER/V5 DATA FILE
  ASSGN SYS003,DISK,VOL=111111,SHR  OPTIIONAL 2ND EXTENT TC DATA FILE
// EXEC PCWERVS
FORMAT=NC
PSTART RCR,00C
PSTART LST,00E
PSTART LST,04E,HBCA
PSTART PUN,00D,AXP
PSTART F2,H2CA
READER=0CC
PRINTERS=COE,01E,02E
PUNCHES=00D,04D
PSTART F3,ABC3
READER=0CC
PRINTERS=0CE,01E,02E
PUNCHES=CCD,04D
PSTART BG,0ABI
READER=0CC
PRINTERS=00E,01E,02E
PUNCHES=COC,04D
/*
// DLBL UOUT,'SYSIN.FILE.FOR.F2.START-UP.UNDER.POWER/V5',99/365,SD
// EXTENT SYSC05,SYSRES,1,0,7432,2      1/6 CYL, 619 - 619, CN SYSRES
// EXEC DISKLOAD
  ASSGN SYSLST,PRINTER
  ASSGN SYSPCH,PUNCH
// EXEC PROC=//ASSGN
  CLOSE SYSIN,READER
/*
// DLBL UOUT,'SYSIN.FILE.FOR.F3.START-UP.UNDER.POWER/V5',99/365,SD
// EXTENT SYSC05,SYSRES,1,0,7434,2      1/6 CYL, 619 - 619, CN SYSRES
// EXEC DISKLOAD
  ASSGN SYSLST,PRINTER
  ASSGN SYSPCH,PUNCH
// EXEC PROC=//ASSGN
  CLOSE SYSIN,READER
/*
// DLBL LCUT,'SYSIN.FILE.FOR.F4.START-UP.UNDER.POWER/V5',99/365,SD
// EXTENT SYSC05,SYSRES,1,0,7436,2      1/6 CYL, 619 - 619, CN SYSRES
// EXEC DISKLOAD
  ASSGN SYSLST,PRINTER
  ASSGN SYSPCH,PUNCH
// EXEC PROC=//ASSGN
  CLOSE SYSIN,READER
/*
// DLBL UCUT,'SYSIN.FILE.FOR.BG.START-UP.UNDER.POWER/V5',99/365,SD
// EXTENT SYSC05,SYSRES,1,0,7438,2      1/6 CYL, 619 - 619, CN SYSRES
// EXEC DISKLOAD
  ASSGN SYSLST,PRINTER
  ASSGN SYSPCH,PUNCH
// EXEC PROC=//ASSGN
  CLOSE SYSIN,READER
/*
/&

```

```

// JOB CATALOG '$$ASSGN' PROCEDURES
// EXEC MAINT
CATALP $1ASSGN
ASSGN SYSLST,PRINTER
ASSGN SYSPCH,PUNCH
ASSGN SYS001,DISK,VOL=111111,SHR
ASSGN SYS002,SYS001
ASSGN SYS003,SYS001
ASSGN SYS004,SYS001
ASSGN SYSLNK,SYS001
/*
CATALP $2ASSGN
ASSGN SYSLST,PRINTER
ASSGN SYSPCH,PUNCH
ASSGN SYS001,DISK,VOL=222222,SHR
ASSGN SYS002,SYS001
ASSGN SYS003,SYS001
ASSGN SYS004,SYS001
ASSGN SYSLNK,SYS001
/*
CATALP $3ASSGN
ASSGN SYSLST,PRINTER
ASSGN SYSPCH,PUNCH
ASSGN SYS001,DISK,VOL=333333,SHR
ASSGN SYS002,SYS001
ASSGN SYS003,SYS001
ASSGN SYS004,SYS001
ASSGN SYSLNK,SYS001
/*
CATALP $4ASSGN
ASSGN SYSLST,PRINTER
ASSGN SYSPCH,PUNCH
ASSGN SYS001,DISK,VOL=333333,SHR
ASSGN SYS002,SYS001
ASSGN SYS003,SYS001
ASSGN SYS004,SYS001
ASSGN SYSLNK,SYS001
/*
CATALP $BASSGN
ASSGN SYSLST,PRINTER
ASSGN SYSPCH,PUNCH
ASSGN SYS001,DISK,VOL=111111,SHR
ASSGN SYS002,SYS001
ASSGN SYS003,SYS001
ASSGN SYS004,SYS001
ASSGN SYSLNK,SYS001
/*
/*
/*
// JOB CATALOG '$$PARSTD' POWER/VS START-UP PROCEDURES
// EXEC MAINT
CATALP $1PARSTD
// OPTION PARSTD
// DLBL IJSYSIN,'POWER/VS.START-UP.SYSIN.FILE',99/365,SD
// EXTENT SYSRDR,SYSRES,1,0,7428,4      1/3 CYL, 619 - 619, ON SYSRES
* 'F1' PARTITION STANDARD LABELS CREATED *
/*
/*
CATALP $2PARSTD
// OPTION PARSTD
// DLBL IJSYSIN,'SYSIN.FILE.FOR.F2.START-UP.UNDER.POWER/VS',99/365,SD
// EXTENT SYSRDR,SYSRES,1,0,7432,2      1/6 CYL, 619 - 619, ON SYSRES
* 'F2' PARTITION STANDARD LABELS CREATED *
/*
/*
CATALP $3PARSTD
// OPTION PARSTD
// DLBL IJSYSIN,'SYSIN.FILE.FOR.F3.START-UP.UNDER.POWER/VS',99/365,SD
// EXTENT SYSRDR,SYSRES,1,0,7434,2      1/6 CYL, 619 - 619, ON SYSRES
* 'F3' PARTITION STANDARD LABELS CREATED *
/*
/*
CATALP $4PARSTD
// OPTION PARSTD
// DLBL IJSYSIN,'SYSIN.FILE.FOR.F4.START-UP.UNDER.POWER/VS',99/365,SD
// EXTENT SYSRDR,SYSRES,1,0,7436,2      1/6 CYL, 619 - 619, ON SYSRES
* 'F4' PARTITION STANDARD LABELS CREATED *
/*
/*
CATALP $BPARSTD
// OPTION PARSTD
// DLBL IJSYSIN,'SYSIN.FILE.FOR.BG.START-UP.UNDER.POWER/VS',99/365,SD
// EXTENT SYSRDR,SYSRES,1,0,7438,2      1/6 CYL, 619 - 619, ON SYSRES
* 'BG' PARTITION STANDARD LABELS CREATED *
/*
/*
/*

```



# APPENDIX J – POWER/VIS GENERATION EXAMPLES

```

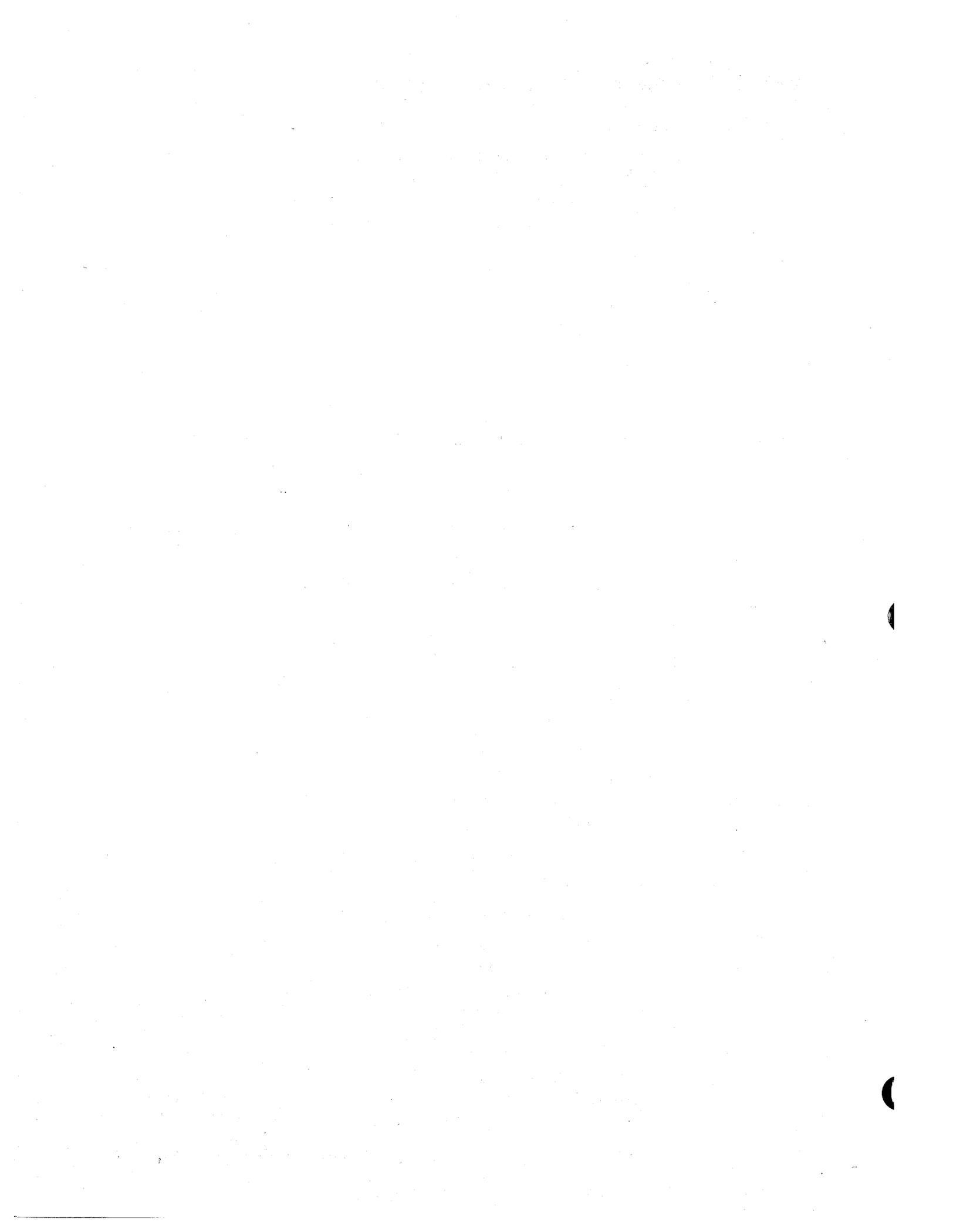
// JOB POWER/VIS COMPILE & CATALOG TAILORED NON-RJE VERSION OF POWER/VIS
// OPTION NODECK,NOXREF,CATAL
// EXEC ASSEMBLY
      PRINT NOGEN
POWERVS  POWER                                X
      DBLK=516,                                X
      TRACKGP=2,                                X
      SUBLIB=P,                                  X
      ACCOUNT=YES,                              X
      JSEP=(1,5),                                X
      RBS=(0,1000)
      END
/*
// EXEC LNKEDT
/&

```

```

// JOB CREATE EXAMPLE 'RJE' VERSION OF POWER/VIS
// OPTION NODECK,NOXREF,CATAL
// EXEC ASSEMBLY
      PRINT NOGEN
POWERTP  POWER                                X
      DBLK=516,                                X
      TRACKGP=2,                                X
      SUBLIB=P,                                  X
      ACCOUNT=YES,                              X
      JSEP=(1,5),                                X
      RBS=(0,1000)
      PLINE                                     X
      ADDR=X'030',                              X
      PSWRD=SECRET,                              X
      TIMEOUT=NO,                                X
      TRNSP=YES,                                  X
      CODE=EBCDIC
      PLINE                                     X
      ADDR=X'031',                              X
      PSWRD=MUSTKNOW,                            X
      CODE=ASCII
      PLINE                                     X
      ADDR=X'032'
      PRMT                                       X
      REMOTE=1,                                  X
      TYPE=2770
      PRMT                                       X
      REMOTE=2,                                  X
      TYPE=2780
      PRMT                                       X
      REMOTE=3,                                  X
      TYPE=3780
      PRMT                                       X
      REMOTE=4,                                  X
      TYPE=3780,                                  X
      LSTROUT=100,                                X
      PUNROUT=100,                                X
      TRNSP=YES,                                  X
      BE=YES,                                      X
      HFC=NO,                                      X
      SCE=NO,                                      X
      ABE=NO,                                      X
      CS=NO,                                       X
      MRF=NO,                                       X
      TRACE=NO,                                    X
      LIST=120
      PRMT                                       X
      REMOTE=5,                                  X
      REF=3
      END
/*
// EXEC LNKEDT
/&

```



## APPENDIX K—POWER/VS OPERATOR MESSAGES

### 1Qxx = POWER/VS MESSAGES

#### 1Q01I POWER/VS CANNOT RUN IN REAL MODE

**Cause:** POWER/VS can only be executed in virtual mode.

**System Action:** POWER/VS initialization is terminated.

**Operator Action:** Omit the REAL parameter from the // EXEC card that causes DOS/VS to load POWER/VS.

#### 1Q02I POWER/VS CANNOT RUN AS A SUBTASK

**Cause:** POWER/VS is attached as a subtask to a maintask. POWER/VS must run as a maintask.

**System Action:** POWER/VS initialization is terminated.

**Programmer Action:** Change the environment of POWER/VS so that it runs as a maintask in one of the five partitions.

#### 1Q03I INSUFFICIENT REAL STORAGE AVAILABLE. xxK REQUIRED

**Cause:** Not enough real storage is provided for the POWER/VS partition.

**System Action:** POWER/VS initialization is terminated.

**Operator Action:** Use the ALLOCR statement/command to increase the size of the real partition that corresponds to the POWER/VS partition. The real partition must be at least as large as indicated in the message.

#### 1Q04I QUEUE FILE MISMATCH

**Cause:** Either (1) warm start is tried with a version of POWER/VS different from the one which formatted the existing queue file, or (2) warm start was tried on the data file and cold start on the queue file (FORMAT=q) but the physical blocksize of the data file does not match the DBLK value specified during POWER/VS generation.

**System Action:** POWER/VS initialization is terminated.

**Operator Action:** Reformat the queue file by performing a cold start of POWER/VS. If the information in the queue file is required, you must restart using the original version.

#### 1Q05I PAGEABLE AREA nnnK TOO SMALL

**Cause:** The size of the pageable area in the POWER/VS partition is not large enough to contain the POWER/VS code.

**System Action:** POWER/VS initialization is terminated.

**Operator Action:** Enlarge the size of the pageable area by nnnK bytes. To do this, either use the ALLOC statement/command to increase the size of the POWER/VS partition by nnnK bytes, or use the ALLOCR statement/command to decrease the size of the corresponding real partition by nnnK bytes.

1Q06I SYSLOG NOT ASSIGNED TO CONSOLE

**Cause:** POWER/VS detects that SYSLOG is not assigned to a console device.

**System Action:** POWER/VS initialization is terminated.

**Operator Action:** Assign SYSLOG to the console and restart POWER/VS.

1Q07I INVALID LOGICAL UNIT filename

**Cause:** The indicated file could not be opened successfully.

**System Action:** POWER/VS initialization is terminated.

**Operator Action:** Check for errors in the DOS/VS job control statements or commands defining the file.

1Q08I SUPERVISOR WITHOUT POWER/VS SUPPORT

**Cause:** Self explanatory.

**System Action:** POWER/VS initialization is terminated.

**Programmer Action:** A different supervisor must be used with the generation option POWER=YES specified in the SUPVR macro.

1Q09I TRACKGROUP CHANGED TO nn

**Cause:** (1) The total number of track groups that has been calculated from the specified extent size and the TRACKGP parameter in the POWER macro is larger than the total number of queue records.(2) The TRACKGP specification exceeds the maximum number of tracks per cylinder of the disk assigned to the data file.

**System Action:** The TRACKGP specified during POWER/VS generation is ignored. POWER/VS calculates a new TRACKGP number (nn) and processing continues.

**Operator Action:** None.

1Q10I SUPERVISOR WITHOUT ACCOUNTING SUPPORT

**Cause:** ACCOUNT=YES has been specified during POWER/VS generation, but a DOS/VS supervisor is used without job accounting support.

**System Action:** POWER/VS continues processing without accounting support.

**Operator Action:** None.

1Q11D FORMAT QUEUES=

**Cause:** This message is issued when POWER/VS is initialized either without using the AUTOSTART procedure, or with the AUTOSTART procedure but the FORMAT= command was omitted or invalid.

**System Action:** POWER/VS waits for the operator's reply.

**Operator Action:** Reply with one of the following:

d [a]

q [a]

a

NO

END or ENTER only.

where:

d: specifies that the data file and the queue file should be formatted.

q: specifies that only the queue file should be formatted. Note that any data in the data file is effectively deleted, since pointers to the data file in the queue file records are cleared.

a: specifies that the account file should be formatted. Because the POWER/VS account file is composed of variable-length records, it is not pre-formatted in the same way as the queue file and the data file. Instead, pointers are set to position the account output at the beginning of the file.

NO or END or ENTER: specifies that no formatting is desired (warm start).

#### 1Q12I POWER/VS INITIALIZATION COMPLETED

**Cause:** POWER/VS has been initialized.

**System Action:** The system enters the wait state until the operator enters the next command, or the system continues by processing any commands entered via the AUTOSTART procedure.

**Operator Action:** Enter next command, if no AUTOSTART.

#### 1Q13I ERRONEOUS AUTOSTART CARD(S) READ

**Cause:** During the AUTOSTART procedure an invalid FORMAT= command was issued.

**System Action:** The system issues message 1Q11D FORMAT QUEUES= so that the operator can submit the correct information.

**Operator Action:** None.

#### 1Q14I NO MATCHING PUB FOR cuu

**Cause:** The RJE line indicated by its channel and unit number (cuu) is defined in POWER/VS but not in the DOS/VS supervisor.

**System Action:** The line is deleted from POWER/VS.

**Operator Action:** Re-IPL the system and ADD the line, or continue without using the line.

#### 1Q15I PHASE phasename NOT FOUND

**Cause:** This POWER/VS phase cannot be found in the private core image library that is mounted or in the system core image library.

**System Action:** POWER/VS initialization is terminated.

**Operator Action:** Mount the correct private core image library or catalog the phase into the core image library. Then restart POWER/VS.

1Q16I INVALID [LST/PUN] ROUTING FOR remind

**Cause:** A remote block is detected describing a user whose list or punch output is to be routed to a non-existing remote-id.

**System Action:** The default list or punch routing for this remote-id is reset to 0 (central location).

**Operator Action:** Notify the systems programmer.

**Programmer Action:** Correct the remote-id.

1Q17I QUEUE FILE TOO SMALL

**Cause:** There are not enough entries in the queue file to cover all track groups in the data file.

**System Action:** Part of the data file will not be used.

**Operator Action:** Notify the systems programmer.

**Programmer Action:** Increase the size of the queue file or decrease the size of the data file.

1Q18I TOO MANY EXTENTS IJDFILE

**Cause:** More than five extents were used for the data file.

**System Action:** POWER/VS initialization is terminated.

**Operator Action:** Change the EXTENT cards for the IJDFILE so that there are no more than five extents and restart POWER/VS.

1Q19I INVALID EXTENT IJDFILE

**Cause:** A submitted extent for the data file does not start on head 0 or does not end on a cylinder boundary.

**System Action:** POWER/VS initialization is terminated.

**Operator Action:** Correct the EXTENT information and restart POWER/VS.

1Q20I AUTOSTART IN PROGRESS

**Cause:** Self explanatory.

**System Action:** As indicated by the message.

**Operator Action:** None.

1Q21I POWER/VS HAS BEEN TERMINATED

**Cause:** The PEND command was issued and all POWER/VS tasks have been terminated.

**System Action:** The POWER/VS partition is restored for normal DOS/VS operation.

**Operator Action:** None.

1Q22I POWER/VS ALREADY ACTIVE

**Cause:** An attempt was made to initialize POWER/VS but POWER/VS is already active.

**System Action:** The initialization is terminated. POWER/VS continues processing.

**Operator Action:** None.

1Q30D ABNORMAL POWER/VS TERMINATION. PRINTER=

**Cause:** One of the POWER/VS tasks was terminated abnormally.

**System Action:** POWER/VS waits for the operator's reply. After acknowledging the operator's reply, the POWER/VS session is terminated.

**Operator Action:** If you want a dump, specify the channel and unit number (cuu) of a printer. If you do not want a dump, reply NO or press end-of-block.

1Q31I MORE THAN 80 % FULL ACCOUNT FILE (IJAFILE) \*\*\*\*\*

**Cause:** 80% of the POWER/VS account file is already filled.

**System Action:** Processing continues.

**Operator Action:** Issue a PACCOUNT command to empty the account file.

1Q32I NO MORE ACCOUNT FILE (IJAFILE) SPACE FOR ttt,cuu

**Cause:** The account file is full.

**System Action:** The task (identified by ttt and its accompanying cuu) issuing the message is put in the wait state. It is automatically reactivated when the account file is emptied.

**Operator Action:** Empty the account file by issuing the PACCOUNT command.

1Q33I STOPPED ttt,cuu

**Cause:** The PSTOP command was issued to terminate a particular POWER/VS task or to release a partition from the control by POWER/VS.

**System Action:** The reader/writer task (identified by ttt and its accompanying cuu) is detached from POWER/VS and the real storage it was using is released.

**Operator Action:** None.

1Q34I { ttt } WAITING FOR WORK [ ON cuu ]  
      { pp }

**Cause:** This message occurs when there is no data for an execution processor or reader/writer task to process.

**System Action:** The task (ttt) or partition (pp) remains active but its data buffers are released. When data is made available again for processing on cuu, the stopped task will automatically continue.

**Operator Action:** Place more input in the card reader, or make list and punch output available for processing. If there is no more work to be done, issue a PSTOP command to stop the task.

1Q35I EOF ON cuu

**Cause:** The last card read from cuu was not a job delimiter statement, either \* \$\$ EOJ or /&.

**System Action:** The reader task issuing the message is put in the wait state.

**Operator Action:** Put the missing card(s) in the card reader (cuu) and ready the device.

1Q36I ttt CANCELED DUE TO I/O ERROR ON cuu

**Cause:** The DOS/VS error recovery routines detected a cancel condition during an I/O operation requested by the task (ttt) on the specified unit cuu, or the operator replied CANCEL on a decision type message issued by the DOS/VS error recovery routines.

**System Action:** The task is canceled. In case of an error that disrupts operation, for instance, an error on the master queue record, the POWER/VS partition is canceled, thus terminating the system and restoring DOS/VS for normal operation.

**Operator Action:** Perform the normal action required for the related DOS/VS error recovery message that accompanies this message, before reactivation of the task or before reinitialization of POWER/VS.

1Q37I JECL CARD INCORRECT NEAR COLUMN n

**Cause:** There is an error in the format or the contents of an operand near column n in the JECL card.

**System Action:** The JECL statement in error is printed on the console together with the above message. The invalid operand is replaced by its system default value.

**Programmer Action:** If you do not want the default, correct the JECL statement and resubmit the job.

1Q38I NO DASD SPACE AVAILABLE FOR ttt,cuu

**Cause:** A reader task, RJE task or execution processor has requested a queue file record or track group, and none is available.

**System Action:** The task issuing the message is put in the wait state. When a queue file record or track group becomes available, the task is reactivated automatically.

**Operator Action:** To make DASD space available, either

- (1) Start a list or punch writer task to process queue entries in the output queue. After all output from a queue entry has been processed, the queue file record and the associated track groups are freed.
- (2) Delete a queue entry from the queue file. After deletion of a queue entry, the queue file record and the associated track groups are freed.

1Q39I JOB jobname FLUSHED BY THE OPERATOR

**Cause:** The operator issued a PFLUSH command to discontinue processing the output from this job.

**System Action:** If a writer task was flushed, this message is printed on the console, and also on the printer if a list task was flushed.

**Programmer Action:** Check with the operator.

1Q40A ON cuu FORMS ffff NEEDED FOR jobname jobnumber

**Cause:** In a \* \$\$ LST (or PRT) statement or a \* \$\$ PUN statement the programmer has requested a special type of forms or cards for the output of the specified job.

**System Action:** This task waits for the operator to change the forms. Processing continues for other POWER/VS tasks.

**Operator Action:** Mount the forms that are indicated in the message. To check the forms alignment on the printer, issue a PSETUP command (you can adjust the alignment during the printing). Issue a PGO command to reactivate the job.

1Q41I WRONG PRINTER/PUNCH FOR jobname cuu

**Cause:** The printer or punch available to the list or punch task is not command compatible with the printer/punch assigned to the partition in which the job was executed.

**System Action:** The incompatible commands are ignored.

**Operator Action:** None, but if you did not want the commands ignored, enter a FLUSH,HOLD command to prepare for restarting on another printer.

1Q42I PAGE/CARD COUNT EXCEEDS END QUEUE ENTRY FOR cuu

**Cause:** A PRESTART command is entered with too great a forward page or card count.

**System Action:** The command is ignored.

**Operator Action:** Reenter the command with a smaller forward page or card count.

1Q43I END OF SPOOL TAPE {LST } ,cuu  
{PUN }

**Cause:** The logical end of a spool tape was reached for the indicated output writer task. (All data has been processed).

**System Action:** The tape is rewound and unloaded. The writer task is stopped.

**Operator Action:** If there are more tapes to be printed or punched, start a new writer task.

1Q44I BOOK library.bookname NOT FOUND

**Cause:** The programmer has requested a book that cannot be found in the specified (sub)library.

**System Action:** The job is flushed. Message OS01I is then issued.

**Operator Action:** None.

**Programmer Action:** Correct the bookname in the \* \$\$ SLI statement and resubmit the job, or ensure that the book is cataloged in the source statement library for the next time.

1Q45I SLI STATEMENT NOT SUPPORTED

**Cause:** An SLI statement was used in a job but POWER/VS was not generated with the SUBLIB parameter.

**System Action:** The job is flushed. Message OS01I is then issued.

**Operator Action:** None.

1Q46I DISP FORCED TO D FOR jobname,jobnumber

**Cause:** In the \* \$\$ LST or \* \$\$ PUN statement disposition N was specified but the SYSLST or SYSPCH device is not available.

**System Action:** The output is spooled.

**Operator Action:** None.

1Q47I {BG} jobname jobnumber [FROM remid] [user-information]  
{Fn}

**Cause:** An \* \$\$ JOB card was read and the JLOG option is in force.

**System Action:** None.

**Operator Action:** None.

1Q48I NO MATCHING SPOOL DEVICE {BG}  
{Fn}

**Cause:** The LST or PUN statement that is printed prior to this message addresses a partition for which no spool device has been set up by a PSTART command, or the device address in the LST or PUN statement does not agree with the spooling device assigned to the partition.

**System Action:** This LST or PUN statement is ignored.

**Operator Action:** Compare the statement with the device assignments. If PSTART has not been issued for the device, then flush the job, PSTOP the partition, enter the appropriate PSTART command, and resubmit the job. If the statement is an error, correct it and resubmit the job.

- 1Q49I INVALID DELIMITER  $\left\{ \begin{array}{l} \text{BG} \\ \text{Fn} \end{array} \right\}$
- Cause:** A parameter in the statement printed prior to this message is not followed by a blank or a comma.
- System Action:** The statement is ignored.
- Operator Action:** None, if the defaults for the statement are acceptable. Otherwise, correct the card and resubmit the job.
- 1Q50I UNKNOWN KEYWORD  $\left\{ \begin{array}{l} \text{BG} \\ \text{Fn} \end{array} \right\}$
- Cause:** POWER/VS does not recognize one of the keyword parameters in the JECL statement printed above this message.
- System Action:** The JECL statement is ignored.
- Operator Action:** Correct the statement and resubmit the job.
- 1Q51I INVALID pppppppp PARAMETER  $\left\{ \begin{array}{l} \text{BG} \\ \text{Fn} \end{array} \right\}$
- Cause:** The parameter pppppppp in the JECL statement printed above this message is invalid. pppppppp can be one of the following: DISP, CLASS, FNO, COPY, TADDR, LTAB, REMOTE, JSEP, RBM, RBS, UCS, FCB, LST, PUN, JNM, USER.
- System Action:** The JECL statement is ignored.
- Operator Action:** Correct the statement and resubmit the job.
- 1Q52I OUTPUT LIMIT EXCEEDED FOR jobname jobnumber  $\left\{ \begin{array}{l} \text{BG} \\ \text{Fn} \end{array} \right\}, \text{cuu}$
- Cause:** The number of list or punch records established by the RBM parameter in the LST or PUN statement (or otherwise by the STDLINE or STDCARD parameter of the POWER macro) is exceeded.
- System Action:** Processing of this job continues.
- Operator Action:** Issue a PSTOP or PFLUSH, if desired.
- 1Q53I OUTPUT SEGMENTED FOR jobname jobnumber  $\left\{ \begin{array}{l} \text{BG} \\ \text{Fn} \end{array} \right\}, \text{cuu}$
- Cause:** Output segmentation occurs because the number of records specified in the RBS parameter or in the \* \$\$ LST statement (or PRT) or in the \* \$\$ PUN statement is exceeded.
- System Action:** Processing of this job continues.
- Operator Action:** Start a writer task.
- 1Q54I  $\left\{ \begin{array}{l} \text{FCB} \\ \text{UCS} \end{array} \right\}$  ERROR FOR jobname jobnumber  $\left\{ \begin{array}{l} \text{BG} \\ \text{Fn} \end{array} \right\}$
- Cause:** The FCB or UCS parameter was specified in the \* \$\$ LST statement. The phase to be loaded either has an error or is not present in the core image library.
- System Action:** The parameter is ignored and LTAB is set to the default.
- Operator Action:** If the job is dependent on the FCB or UCS phase, correct the error and resubmit the job, or use the attention routine FCB or UCS commands when the job is directed to a printer.



**System Action:** The PACCOUNT command is not executed and the account file is still intact.

**Operator Action:** Correct the label information or the problem that caused the open failure and submit the PACCOUNT command again.

1Q61I UNRECOVERABLE I/O ERROR ON  $\left. \begin{array}{l} \{ \text{SYSTEM SSL} \} \\ \{ \text{PVT SSL cuu} \} \\ \{ \text{AFILE} \} \\ \{ \text{QFILE} \} \text{ n cuu} \\ \{ \text{DFILE} \} \\ \{ \text{PACCOUNT OUTPUT DEVICE} \} \\ \text{cuu} \end{array} \right\}$

**Cause:** An unrecoverable I/O error occurred on the file indicated in the message as the following:

SYSTEM SSL = system source statement library

PVT SSL cuu = private source statement library cuu

AFILE 0 cuu = account file SYS000 cuu

QFILE 1 cuu = queue file SYS001 cuu

DFILE 2(-6) cuu = data file SYS002 (-SYS006) cuu

PACCOUNT OUTPUT DEVICE = as specified in the PACCOUNT command

cuu = channel and unit number

**System Action:** Error bypass procedures will be performed for which another message will be issued. The task is terminated.

**Operator Action:** See explanation and recommended operator procedure for the accompanying messages.

1Q62I CANCEL FOR POWER/VS IGNORED

**Cause:** The CANCEL job control command was issued for the POWER/VS partition.

**System Action:** Processing continues.

**Operator Action:** To cancel POWER/VS, issue the PEND KILL command.

1Q63I UNRECOVERABLE I/O ERROR IN QUEUE MASTER REC cuu

**Cause:** Self-explanatory.

**System Action:** POWER/VS is terminated with message 1Q76I.

**Operator Action:** Perform a POWER/VS cold start with different extents for IJQFILE and IJDFILE. Resubmit all jobs not yet processed.

1Q64I JOB jobname  $\left. \begin{array}{l} \{ \text{RDR} \} \\ \{ \text{PUN} \} \\ \{ \text{LST} \} \end{array} \right\}$  SET DELETED

**Cause:** An unrecoverable I/O error was caused by a data record in a trackgroup or one of the queue records in the queue set.

**System Action:** The entire queue set for this job or job segment is deleted and the associated track groups from the POWER/VS data file will not be used again. The task is terminated with an additional message.

**Operator Action:** Resubmit the job indicated in the message (after performing back-up procedures, if necessary, on the user master file). If, after the last cold start, this message appears frequently, it may indicate that system performance is being affected by an increasing number of unusable track groups on the data files. Complete all jobs still in the queues, initialize the disk(s) on which IJQFILE and IJDFILE reside, assign alternate tracks as necessary, and perform a cold start. Different extents may be desirable for better performance if there are too many alternate tracks assigned for the original extents.

1Q65I UNKNOWN {RDR} SET DELETED  
                  {PUN}  
                  {LST}

**Cause:** An unrecoverable I/O error occurred on the queue file while POWER/VS was scanning the reader, punch, or list queues.

**System Action:** The queue set for this job or job segment is deleted and the associated track groups from the spool files will not be used again. The task is terminated.

**Operator Action:** Refer to action for message 1Q64I.

1Q66I ACCOUNT FILE KEPT

**Cause:** An unrecoverable I/O error occurred on the account file (IJAFILE) or on the PACCOUNT output device after a PACCOUNT command was issued. The IJAFILE file is still intact.

**System Action:** The PACCOUNT task is terminated with message 1Q72I.

**Operator Action:** If the I/O error occurred on the PACCOUNT output device, issue the PACCOUNT command again with a different output device.

1Q67I FREE SET NOT ACCESSIBLE

**Cause:** An unrecoverable I/O error occurred on the queue file (IJQFILE).

**System Action:** POWER/VS is terminated with message 1Q76I.

**Operator Action:** Perform a warm start and process all existing output queue entries (PUN and LST). Initialize the disk on which IJQFILE resides and assign alternate tracks as necessary. After a cold start, with different extents if necessary, resubmit all jobs in the input (RDR) queue.

1Q68I SEGMENTATION FORCED FOR jobname jobnumber {BG} ,cuu  
                                                          {Fn}

**Cause:** An unrecoverable I/O error occurred on the IJQFILE or IJDFILE.

**System Action:** Message 1Q69I is issued. The partition continues to run. If the I/O error occurred in an output queue set of this partition, it is deleted.

**Operator Action:** If message 1Q64I was also issued, cancel the job in the partition and resubmit it after performing backup procedures on the user master files.

1Q69I    DEFAULT OPTIONS TAKEN FOR jobname jobnumber  $\left\{ \begin{array}{l} \text{BG} \\ \text{Fn} \end{array} \right\}$  ,cuu

**Cause:** An unrecoverable I/O error occurred and after output segmentation all options in JECL commands are set to their defaults.

**System Action:** Processing continues on this basis.

**Operator Action:** If the defaults are not suitable, resubmit the job.

1Q70I    TASK FAILURE, STOPPED  $\left\{ \begin{array}{l} \text{BG} \\ \text{Fn} \end{array} \right\}$

**Cause:** An unrecoverable I/O error occurred on the queue file or the data file while reading input queues into the partition specified in the message.

**System Action:** A PSTOP is forced for the partition. If the message 1Q64I was not also issued, then the queue set is returned to the spool files in either the leave state (if it was in the keep state), or the hold state (if it was in the dispatchable state).

**Operator Action:** Issue an UNBATCH job control command followed by a POWER/VS PSTART command. The partition will continue with the next job in the queue. If the reader set was not deleted (as indicated by message 1Q64I), perform user file backup procedures as required and issue the PRELEASE command for this job.

1Q71I     $\left\{ \begin{array}{l} \text{RDR} \\ \text{LST} \\ \text{PUN} \end{array} \right\}$  ,cuu TERMINATED

**Cause:** An unrecoverable I/O error occurred on the queue file or the data file or the unit record device on cuu.

**System Action:** Processing continues with other tasks.

**Operator Action:** Take the appropriate corrective action for the device on cuu. Issue PSTART for the task and device (or another device of the same type) to continue processing. If a reader task was terminated, resubmit any jobs that were incompletely read.

1Q72I    PACCOUNT TERMINATED

**Cause:** An unrecoverable I/O error occurred during the PACCOUNT process.

**System Action:** The PACCOUNT task is terminated.

**Operator Action:** Try the PACCOUNT command again with a different output device if the I/O error occurred on the output device specified in the PACCOUNT command. Otherwise, if the error occurred on IJAFILE, see message 1Q74I.

1Q73I    STATUS DISPLAY TERMINATED

**Cause:** An unrecoverable I/O error occurred on the queue file or the printer.

**System Action:** The report is not continued. Processing continues with other tasks.

**Operator Action:** Reissue the PDISPLAY command.

1Q74I ACCOUNT SUPPORT FUNCTIONS TERMINATED

**Cause:** An unrecoverable I/O error occurred on the account file (IJAFILE).

**System Action:** POWER/VS accounting support is immediately terminated.

**Operator Action:** Issue a PEND command to finish processing of all currently running jobs (without accounting support). To regain accounting support and save the IJAFILE, do the following:

- 1) Perform a warm start (FORMAT QUEUES=NO).
- 2) After POWER/VS is initialized, issue a PACCOUNT (or J) command to save as much of the current account file as possible.
- 3) Perform a cold start on the IJAFILE with different extents and perform a warm start on IJQFILE and IJAFILE (FORMAT QUEUES=A).

1Q75I MULTIPLE TERMINATION OF TASK, POWER/VS TERMINATED

**Cause:** An unrecoverable I/O error occurred during processing of an earlier I/O error.

**System Action:** POWER/VS is terminated.

**Operator Action:** Check the previous message and determine if you want to warm start POWER/VS.

1Q76I POWER/VS CANNOT CONTINUE

**Cause:** An unrecoverable I/O error occurred on the queue file that cannot be bypassed.

**System Action:** Issue message 1Q30D.

**Operator Action:** Check the previous messages and determine if you want to warm start POWER/VS.

1Q77I INVALID SPOOL TAPE FOR {LST} ,cuu  
{PUN}

**Cause:** Either the spool tape is not a valid POWER/VS spool tape, or the tape does not correspond with the type of writer task (a list writer cannot be for a punch spool tape and a punch writer cannot be for a list spool tape).

**System Action:** The tape is rewound and unloaded. The writer task is stopped.

**Operator Action:** Mount the correct tape and start a writer task (using the PSTART command).

1Q78I NO REAL STORAGE AVAILABLE FOR ACT,cuu

**Cause:** The PACCOUNT command was issued but there is not enough real storage for the account file saving task to execute.

**System Action:** The account file saving task is terminated.

**Operator Action:** Retry the PACCOUNT command a few times. If it is still unsuccessful, do one of the following, whichever applies:

- 1) If you had entered the PACCOUNT command because the account file was 80% full (message 1Q31I), then enter the PSTOP command for one of the partitions under POWER/VS control. When this partition is stopped, re-issue the PACCOUNT

command.

2) If you had entered the PACCOUNT command because the account file was completely full (message 1Q32I), then enter the PSTOP command for one or more POWER/VS tasks. Reissue the PACCOUNT command.

1Q79I ACCOUNT FILE SAVED

**Cause:** The system has saved the account file as requested by the operator in a PACCOUNT command.

**System Action:** All DASD space allocated to the account file is made available for storing more account records. All tasks that are in the wait state, because they failed to write an account record, are automatically reactivated.

**Operator Action:** None.

1Q80I ACCOUNT FILE ERASED

**Cause:** The account file has been deleted in response to a PACCOUNT DEL command.

**System Action:** The account file extent is re-initialized. All tasks that are in the wait state, because they failed to write an account record, are automatically reactivated.

**Operator Action:** None.

1Q81I 'filename' EXTENT TOO SMALL, COMMAND NOT EXECUTED

**Cause:** A PACCOUNT command was issued to copy the account file to a disk extent (filename as indicated in the message), which is not large enough to contain all the account information.

**System Action:** If the account file was 100% full, the system remains in a wait state until the operator issues a valid PACCOUNT command. If the account file was not full, the system continues processing.

**Operator Action:** Issue the PACCOUNT command again, possibly specifying a different medium.

1Q82I I/O ERROR DURING  $\left\{ \begin{array}{l} \text{CP} \\ \text{IT} \\ \text{TT} \end{array} \right\}$ , POWER/VS TERMINATED

**Cause:** An unrecoverable I/O error occurred during execution of CP (command processor), IT (initialization), or TT (task termination).

**System Action:** The POWER/VS partition is canceled.

**Operator Action:** First, try to warm start POWER/VS. If this is unsuccessful, then cold start POWER/VS and resubmit all jobs that were not processed completely.

## 1Rxx = POWER/VS RJE AND POWER/VS COMMAND PROCESSOR MESSAGES

### Central Messages

#### 1R02I LINE cuu STOPPED

**Cause:** The central operator issued a PSTOP command to stop the line (cuu).

**System Action:** After this message is displayed at the central system, the system displays message 1R17I LINE HAS BEEN STOPPED for each remote-ID signed on.

**Operator Action:** None.

#### 1R03I INVALID DEVICE TYPE FOR cuu

**Cause:** The line address (cuu) specified in the PSTART command is known to the DOS/VS supervisor, but the device type is not 2701 or 2703.

**System Action:** The PSTART command is ignored.

**Operator Action:** Specify the correct line address in the PSTART command (if 2701 or 2703: otherwise, you will not succeed).

#### 1R04I NO PLINE MACRO GENERATED FOR cuu

**Cause:** The line address (cuu) specified in the PSTART command is known to the DOS/VS supervisor, but not to POWER/VS

**System Action:** The PSTART command is ignored.

**Operator Action:** Reassemble POWER/VS with a PLINE macro for this line.

### Remote Messages

#### 1R11I INVALID STOP COMMAND

**Cause:** Invalid parameters were specified with the STOP command.

**System Action:** The STOP command is ignored.

**Operator Action:** Correct the STOP command and re-issue it.

#### 1R12I INVALID CLASS SPECIFICATION

**Cause:** The class specified in a remote START command is invalid. Either (1) more than four characters were specified, (2) two of the same characters were specified, or (3) a character other than A-Z was specified.

**System Action:** The START command is ignored.

**Operator Action:** Reenter the correct START command.

1R13I INVALID TASK SPECIFICATION

**Cause:** The task specified in a remote START command is neither LST nor PUN nor MSG.

**System Action:** The START command is ignored.

**Operator Action:** Reenter the correct START command.

1R14I EOF ON THE READER

**Cause:** An end-of-file condition occurred at the terminal reader before EOJ.

**System Action:** The RJE reader routine is put in the wait state. If no operator response within n minutes (n is specified during POWER/VS generation in the TIMEOUT parameter), the terminal is signed off and the job is flushed.

**Operator Action:** Supply the reader with the rest of the job and ready the reader to continue.

1R15I REMOTE remind SIGNED ON

**Cause:** A remote user (identified by his remote-ID) entered a valid SIGNON command.

**System Action:** The terminal is logically attached to the central system.

**Operator Action:** None.

1R16I REMOTE remind SIGNED OFF

**Cause:** A remote user (identified by his remote-ID) entered a valid SIGNOFF command or he entered a SIGNON command at a processing terminal.

**System Action:** The terminal is logically detached from the central system.

**Operator Action:** None.

1R17I LINE HAS BEEN STOPPED

**Cause:** The central operator stopped the line.

**System Action:** The terminal is detached and all used storage space is released. The input queue entry in process is not made, or the output queue entry in process is preserved, depending on the processing mode.

**Operator Action:** Wait until the line is restarted centrally to sign on again, or check with the central operator.

1R18I REMOTE remind FORCED TO SIGN OFF

**Cause:** For n minutes no activity has taken place on the line (n is specified in the POWER/VS option TIMEOUT).

**System Action:** The terminal becomes inactive.

**Operator Action:** None.

1R19I FIRST CARD MUST BE SIGNON COMMAND

**Cause:** The first card in the input stream from a terminal that is currently not signed on is not a SIGNON command.

**System Action:** Flush the input stream.

**Operator Action:** Enter a SIGNON command.

1R20I MESSAGES DELETED

**Cause:** There is not enough room in the message queue for all the messages (probably due to the printer not being ready).

**System Action:** The messages for the remote-ID who is monopolizing the queue are deleted.

**Operator Action:** None.

1R21I SIGNON IGNORED. INVALID REMOTE-ID

**Cause:** The remote-ID specified in the SIGNON command is not known to POWER/VS or is not associated with this terminal.

**System Action:** Flush the input stream.

**Operator Action:** Enter a valid SIGNON command.

1R22I SIGNON IGNORED. INVALID PASSWORD

**Cause:** The security password specified in the SIGNON command does not match the password that the central operator defined for this line or the default password specified during POWER/VS generation.

**System Action:** Flush the input stream.

**Operator Action:** Enter a valid SIGNON command.

1R24I { START } COMMAND OUT OF SEQUENCE  
      { STOP }  
      { GO }

**Cause:** A command is entered at the wrong time. For example, a START command is issued on an active task, or another task management command is issued on an inactive task.

**System Action:** The command is ignored.

**Operator Action:** Correct the sequence of commands.

## Command Processor Messages

1R46I {status report  
{NOTHING TO DISPLAY}}

**Cause:** The status report is printed as a result of a PDISPLAY command.

**System Action:** none.

**Operator Action:** none.

1R47I {ttt, cuu message  
{NO MESSAGES PENDING}}

**Cause:** The PDISPLAY M command was issued.

**System Action:** The system lists all messages for which deferred operator reply is still required.

**Operator Action:** None. (Only after the entire list has been printed can you enter commands.)

1R48I {ttt, cuu class [jobname jobnumber class]  
{NO READER OR WRITER TASK CURRENTLY ACTIVE}}

**Cause:** The PDISPLAY A command was issued.

**System Action:** All active reader/writer tasks in the system are listed together with the classes on which they can operate. (For a physical list or punch task the jobname, jobnumber and class from the queue record being processed are also displayed.)

**Operator Action:** None.

1R49I FREE RECORDS QUEUE FILE nnnnn  
1R49I {ACCOUNT FILE nn% FULL }  
{NO ACCOUNTING SUPPORT}

**Cause:** The system responds to a PDISPLAY Q command.

**System Action:** None.

**Operator Action:** None.

1R50D pp READER=  
1R50D pp PRINTERS=  
1R50D pp PUNCHES=

**Cause:** Message 1R86I PLEASE SPECIFY DEVICES TO BE SPOOLED has just been issued.

**System Action:** The system waits for the operator's response.

**Operator Action:** Specify the physical devices to be spooled for the given partition. Specify the physical device address in the form cuu (channel and unit number). For READER, you can specify only one cuu or NO (NO defines a writer-only partition). If more than one printer or punch is to be spooled, separate the device addresses by a comma (up to eight device addresses may be specified). By specifying both PRINTERS and PUNCHES as NO you define a reader-only partition. NO can be specified once or twice, but must not be specified all three times.

1R51I commandcode OPERAND 1 DESIGNATES NON-EXISTING TASK

**Cause:** The task specified in a task management command does not exist.

**System Action:** The task management command is not executed.

**Operator Action:** None.

1R52I commandcode syntax error

**Cause:** A syntax error was made in an operator command. The syntax error may be one of the following:

NO SECOND OPERAND ALLOWED FOR PSTOP PARTITION

OPERAND 2 NEITHER DECIMAL NOR OMITTED

OPERAND n IS INVALID HEXADECIMAL TERM

OPERAND 2 NEITHER DECIMAL NOR OMITTED

OPERAND 2 NEITHER 'HOLD' NOR OMITTED

NO VALID KEYWORD SPECIFIED

OPERAND n NOT SPECIFIED AS KEYWORD

OPERAND 1 INVALID DESTINATION

OPERAND n IS INVALID KEYWORD

OPERAND 1 MISSING OR INVALID

OPERAND 2 NO DEVICE ADDRESS

OPERAND 2 NO VALID JOBNAME

OPERAND 1 NO VALID QUEUE

INVALID NUMBER OF COPIES

OPERAND 1 NOT 'STATUS'

SECOND OPERAND INVALID

OPERAND 3 NOT DECIMAL

INVALID DISPOSITION

INVALID REMOTE ID

OPERAND 1 INVALID

OPERAND 2 INVALID

INVALID PRIORITY

INVALID CLASS

**System Action:** The command is not executed.

**Operator Action:** Re-issue the corrected command.

1R53I PACCOUNT INVALID DENSITY

**Cause:** The tape density specified in a PACCOUNT (or J) command was invalid.

**System Action:** The command is not executed.

**Operator Action:** Re-issue the command with the correct tape density.

1R54I commandcode CLASS x INVALID

**Cause:** An invalid class was specified in a PSTART command.

Valid classes are as follows:

physical reader A - Z, 0 - 4

physical writer A - Z (1 to 4 alphabetic characters)

execution reader A - Z, 0 - 4 (1 to 4 alphanumeric characters). (0 for BG, 1 for F1, 2 for F2, 3 for F3, or 4 for F4)

**System Action:** The PSTART command is not executed.

**Operator Action:** Omit or correct the class in the PSTART command. (The default is A, except for the execution reader, whose default matches the partition (0 for BG, 1 for F1, etc.)

1R55I PACCOUNT INVALID FILENAME

**Cause:** The filename specified in the PACCOUNT (or J) command is invalid.

**System Action:** The command is not executed.

**Operator Action:** Re-issue the corrected PACCOUNT command.

1R56I xxx { DOES NOT OCCUR IN RJE LINE TABLE  
PROCESSING yyy  
NOT INITIATED  
INACTIVE }

**Cause:** The central operator issued a PINQUIRE command.

One message is issued for each RJE line address (xxx), or as specified. The meanings of each message text follow:

(a) DOES NOT OCCUR IN RJE LINE TABLE means that no PLINE macro was submitted for this RJE line address during POWER/VIS generation or that the line address is not present in the PUB table.

(b) PROCESSING yyy indicates that a user (remote-ID yyy) is currently signed on.

(c) NOT INITIATED means that the line was started, but no remote-id is signed on.

(d) INACTIVE means that the line was not started by the central operator (no PSTART was given).

**System Action:** Processing continues.

**Operator Action:** None.

1R57I commandcode COMMAND IGNORED, TASK IS AT JOB BOUNDARY

**Cause:** The PFLUSH or PRESTART command was issued for a task that has already reached end-of-job.

**System Action:** The command is not executed.

**Operator Action:** None.

1R58I commandcode DEVICE xxx { NOT KNOWN  
IN USE  
IS DOWN }

**Cause:** The device (xxx) specified in the command does not occur in the PUB table in the DOS/VIS supervisor, or is not available for use, or is down.

**System Action:** The command is not executed.

**Operator Action:** If the device is present and available, re-IPL and add the device to the PUB table. Otherwise, when the device is available, resubmit the command.

1R59I commandcode OPERAND 2 IGNORED

**Cause:** Two printer addresses were specified on a PEND command; only one is allowed.

**System Action:** If the first operand is correct, the command is executed. Otherwise, the command is not executed.

**Operator Action:** If the command was not executed, correct the command and resubmit it.

1R61I PFLUSH INVALID FOR WRITER-ONLY PARTITION

**Cause:** The PFLUSH command was issued for a writer-only partition (one that was started with READER=NO).

**System Action:** The PFLUSH command is ignored and processing continues.

**Operator Action:** None.

1R62I commandcode INVALID RJE PASSWORD

**Cause:** The password of a PSTART command for an RJE line was longer than 8 characters.

**System Action:** The command is not executed.

**Operator Action:** Re-enter the PSTART command with the correct password.

1R63I commandcode xx PRIORITY TOO HIGH

**Cause:** A PSTART command was issued for a partition whose priority is not lower than the priority of the POWER/VS partition.

**System Action:** The PSTART command is not executed.

**Operator Action:** Change the priority of the partitions with the PRTY attention command and then re-issue the PSTART command.

1R64I commandcode {SYSLST LUB NOT AVAILABLE}  
                  {NO FREE LUB AVAILABLE }

**Cause:** Self-explanatory.

**System Action:** The command is not executed.

**Operator Action:** Wait until a reader or writer task stops at which time a LUB will become available, or issue a PSTOP command for a physical reader or writer task. Then re-issue your original command.

1R65I commandcode RJE NOT SUPPORTED

**Cause:** An RJE command was issued in a non-RJE environment (no PLINE macros were specified during POWER/VS generation).

**System Action:** The command is not executed.

**Operator Action:** None.

1R66I commandcode {xxx LIST WRITER TASK DOES NOT EXIST}  
{NO WRITER TASK SPECIFIED}

**Cause:** A PSETUP or PRESTART command specified something other than a list writer task, or the writer task specification was omitted.

**System Action:** The command is not executed.

**Operator Action:** Correct the command and resubmit it.

1R67I commandcode {OPERAND 2 REDUCED TO 9999}  
{NUMBER OF PAGES REDUCED TO 99}

**Cause:** In the PRESTART command the number of pages or cards specified was greater than 9999, or in the PSETUP command the number of pages specified was greater than 99.

**System Action:** Self-explanatory.

**Operator Action:** None.

1R68I commandcode pp PARTITION NOT AVAILABLE

**Cause:** A PSTART command was issued for a partition that was already active.

**System Action:** The command is not executed.

**Operator Action:** Issue PDISPLAY A to check if the partition is already under POWER/VS control, then PSTOP it if necessary (to change the class specification in a new PSTART command). If the partition is not under POWER/VS control, wait until the current job finishes, issue STOP, and re-issue the PSTART command.

1R69I commandcode NO ACCOUNTING SUPPORT

**Cause:** The PACCOUNT command was entered, but ACCOUNT=YES was not specified during POWER/VS generation.

**System Action:** The PACCOUNT command is ignored.

**Operator Action:** None.

1R70I commandcode NO DEVICE ADDRESS SPECIFIED

**Cause:** The first operand of a PRESTART command was not a device address.

**System Action:** The PRESTART command is not executed.

**Operator Action:** Re-issue the corrected PRESTART command.

1R71I commandcode OPERAND x IS NO VALID READER/PRINTER/PUNCH

**Cause:** The operator response to message 1R50D is invalid. It must be NO or an applicable device address.

**System Action:** The system keeps prompting until a satisfactory reply is given.

**Operator Action:** Specify a correct device address or NO.

1R72I commandcode VIRTUAL xx SMALLER THAN 64K

**Cause:** The minimum size of a virtual partition to be started is 64K bytes.

**System Action:** The PSTART command is not executed.

**Operator Action:** Use the ALLOC command to change the partition size and then re-issue the PSTART command.

1R73I commandcode INVALID DEVICE TYPE FOR { RDR }  
{ LST }  
{ PUN }

**Cause:** Operands 1 and 2 of a PSTART command are inconsistent (you cannot read from a printer or write on a card reader).

**System Action:** The PSTART command is not executed.

**Operator Action:** Issue a valid PSTART command.

1R74I { PEND OPERAND n INVALID PRINTER SPECIFICATION }  
{ PACCOUNT INVALID TAPE SPECIFICATION }

**Cause:** Either operand n (1 or 2) in the PEND command was not the address of a printer, or in the PACCOUNT command an invalid tape address was specified.

**System Action:** The PEND or the PACCOUNT command is not executed.

**Operator Action:** Re-issue the corrected command.

1R75I pp AUTOSTARTED

**Cause:** The PSTART control card that was issued on SYSIPT has been processed for the pp partition.

**System Action:** Processing continues.

**Operator Action:** None.

1R76I commandcode NUMBER OF PAGES NOT DECIMAL

**Cause:** The number of pages specified in the PSETUP command was not specified in decimal notation.

**System Action:** The PSETUP command is not executed.

**Operator Action:** Re-issue the corrected command.

1R77I commandcode TASK NOT WAITING FOR OPERATOR

**Cause:** A PSETUP or PGO command was issued for a task that is not waiting for operator response.

**System Action:** The command is not executed.

**Operator Action:** None.

1R78I commandcode SHORT 'PEND' NO LONGER SUPPORTED

**Cause:** The E command was given.

**System Action:** The command is not executed.

**Operator Action:** Issue a PEND command.

1R79I commandcode ERRONEOUS AUTOSTART CARD(S) READ

**Cause:** During the AUTOSTART procedure an invalid PSTART command was detected, or an invalid spool device was specified for a partition to be started.

**System Action:** If the partition cannot be started, the command is not executed. If an invalid spool device was indicated, message 1R86D prompts the operator again to specify devices to be spooled.

**Operator Action:** None.

1R80I commandcode WARNING: CLASS SPECIFICATION IGNORED

**Cause:** The PSTART command with a class parameter was issued for a partition which is under POWER/VS control already, but which had been stopped with a STOP command (not PSTOP).

**System Action:** This PSTART command is equivalent to a DOS/VS START command, that is, the partition is activated, but the classes that applied to the previous PSTART command are still applicable.

**Operator Action:** None, unless the classes are not suitable, in which case issue PSTOP and then re-issue the desired PSTART command.

1R81I commandcode {MESSAGE DOES NOT START WITH QUOTE }  
{MESSAGE TOO LONG OR NO CLOSING QUOTE }

**Cause:** Self-explanatory.

**System Action:** The command is not executed.

**Operator Action:** Re-issue the PBRDCST command with the message enclosed in quotes.

1R82I commandcode 'PSETUP' OR 'PRESTART' IN PROGRESS

**Cause:** A PRESTART command was issued, but a PSETUP or PRESTART command is going on.

**System Action:** The PRESTART command is not executed.

**Operator Action:** Issue the PRESTART command again later.

1R83I commandcode OPERAND NEITHER 'ALL' NOR LINE ADDRESS

**Cause:** The PINQUIRE command was incorrect.

**System Action:** The command is not executed.

**Operator Action:** Re-issue the PINQUIRE command with the operand corrected.

1R84I commandcode DELETION NOT ALLOWED OR IMPOSSIBLE

**Cause:** The alluser message that was to be deleted does not exist or was not sent by the issuer of the PDELETE command.

**System Action:** The command is not executed.

**Operator Action:** None.

1R85I commandcode COMMAND INVALID FOR REMOTE OPERATOR

**Cause:** The GO, END, or ACCOUNT command was issued by the remote operator.

**System Action:** The command is not executed.

**Operator Action:** None.

1R86D PLEASE SPFCIFY DEVICES TO BE SPOOLED

**Cause:** A PSTART command has been entered for a partition.

**System Action:** Message 1R50D is issued immediately.

**Operator Action:** None.

1R87I commandcode TOO MANY CLASSES, ONLY FIRST n PROCESSED

**Cause:** In a PSTART command more than one class was specified for a reader task, or more than four classes were specified for a writer or execution reader task.

**System Action:** The extra classes are ignored.

**Operator Action:** If this is not acceptable, issue a PSTOP command, followed by a new PSTART with the correct classes.

1R88I { NOTHING TO RELEASE  
NOTHING TO DELETE  
CANNOT BE ALTERED  
JOB jobname jobnumber NOTHING TO ALTER  
CP READY }

**Cause:** The job specified in the PALTER command is being processed, or the job specified in the PALTER, PDELETE, or PRELEASE command is not present in the specified queue.

**System Action:** The command is not executed. For CP READY, the previous command has been processed.

**Operator Action:** None. If CP READY is the message text, you can issue another command.

1R89I commandcode POWER/VS INITIALIZATION NOT COMPLETED

**Cause:** The PEND command was given before POWER/VS initialization was complete.

**System Action:** The PEND command is not executed.

**Operator Action:** Wait until the initialization is finished before you try to issue PEND again.

1R90I commandcode INVALID TASK SPECIFICATION, operand

**Cause:** The first operand of a PSTART command was neither RDR, LST, PRT, PUN, RJE, BG, nor Fn. The invalid operand is shown in the message.

**System Action:** The command is not executed.

**Operator Action:** Correct the PSTART command and re-issue it.

1R91I commandcode TOO MANY OPERANDS, FIRST n PROCESSED

**Cause:** Self-explanatory.

**System Action:** The extra operands are ignored.

**Operator Action:** None.

1R92I ALLUSER MESSAGE QUEUE IS FULL

**Cause:** Self-explanatory.

**System Action:** The PBRDCST command is not executed.

**Operator Action:** Issue a PDISPLAY MSG to see what is in the queue, use the PDELETE MSG command to make space in the queue, and then re-issue the PBRDCST command.

1R93I REMOTE remid CURRENTLY NOT SIGNED ON

**Cause:** The PBRDCST command submitted refers to a remote-ID that is not signed on.

**System Action:** The message is not sent to the remote-ID. Processing continues.

**Operator Action:** To find out who is signed on, issue a PINQUIRE ALL.

1R97I commandcode COMMAND INVALID DURING SHUTDOWN PERIOD

**Cause:** Either a PSTART, PEND, PALTER, PRELEASE, or PDELETE command was issued during the shutdown period.

**System Action:** The command is not executed.

**Operator Action:** None.

1R98I commandcode INVALID POWER/VS COMMAND CODE

**Cause:** Some command was issued that is not known to POWER/VS.

**System Action:** The command is not executed.

**Operator Action:** Submit a correct command.

1R99I { POWER/VS IS IN THE SHUTDOWN PERIOD }  
{ POWER/VS HAS BEEN TERMINATED }

**Cause:** The PEND command was issued by the central operator, or PEND KILL was issued by the central operator.

**System Action:** Processing continues.

**Operator Action:** None.

## CHANGES TO OTHER DOS/VS MESSAGES

The following DOS/VS messages have been expanded for POWER/VS:

- . OS01I
- . 1A7nD/I
- . 1C5nI.

### OS01I JOB or SUB xxxxxxxx CANCELED DUE TO OPERATOR INTERVENTION

**Cause:** Operator typed CANCEL on the console keyboard to cancel a job, or the job has been flushed under POWER/VS.

**Note:** A separate subtask cannot be canceled by the operator. When the operator types CANCEL on the console-typewriter, the job is canceled due to operator intervention, and the subtasks are canceled due to maintask termination.

**System Action:** The job is canceled.

**Programmer Action:** None.

**Operator Action:** None.

### 1A7nD/I INVALID DEVICE STATUS

n indicates the field processed when the error was detected.

**Cause:** The previous ASSGN attempted to assign a physical unit that is in a 'down' or 'not operational' status resulting from a DVCDN command, or

The previous ASSGN attempted to assign SYSLNK or SYSCLB to a physical unit that is in a 'down' status resulting from a DVCDN command, or

The device specified in the DVCUP command was never previously placed in a down status by a DVCDN command, or

The previous MTC command specified a physical device assigned to a foreground program, or

The DVCDN command was issued for a device owned by POWER/VS.

**System Action:** If SYSLOG is assigned to a keyboard, the system waits for an operator response; otherwise, the invalid command or statement is ignored. If option ACANCEL is in force the job is canceled.

**Programmer Action:** None.

**Operator Action:** Issue a LISTIO command and use the output to determine which cause applies in this case. If you attempted to assign SYSPCH or SYSLST to a file protected tape, either mount a new tape or insert a file protect ring in the mounted tape, and reissue the ASSGN command, or

Type in a new assignment,

Type IGNORE or press END/ENTER to ignore the statement or command if you can do without it, and continue processing, or

Type CANCEL to cancel job.

If the problem recurs, have the

- . . LISTIO output
- . log sheet
- . job stream

available for problem determination.

## 1C5nI PROCESSING ROUTINE ACTIVE

n indicates the field processed when the error was detected.

**Cause:** External interrupt given, and external interrupt routine is currently active, or MSG command given, and foreground partition external interrupt routine is active, or a POWER/VS command was entered while the POWER/VS command processor was processing the previous command.

**System Action:** Processing continues.

**Programmer Action:** Check that an EXIT has been included for each STXIT (not applicable to POWER/VS).

**Operator Action:** Check that the MSG command was issued to the correct foreground partition or that the external interrupt key (see the operating procedures for your System/370 Model) was used to communicate with the background partition. For POWER/VS, wait for message 1R88I CP READY before entering the next command.

## INDEX

### A

account records, contents of 8.1  
account file  
  delete 6.15  
  description 2.8, 2.3  
  formatting of 4.6  
  jobs capacity 3.14  
  labels, example 3.8  
  save 6.15  
  size requirement 3.13  
accounting (see *job accounting*)  
addresses  
  spool devices, specify 4.6  
alarm  
  intervention required, terminal 7.2  
align  
  printer forms, RJE 7.12  
ALLOC  
  POWER/VS requirements 3.10  
  start-up 4.1, 4.5  
ALLOCR  
  POWER requirements 3.10  
  start-up procedure 4.1, 4.5  
ALLUSERS  
  broadcast message 6.15  
  broadcast message, RJE 7.12  
  delete message 6.14  
  delete message, RJE 7.10  
  display messages, RJE 7.8  
alter  
  attributes 6.11  
  attributes, RJE 7.9  
ALTER command 7.9  
ASCII  
  transmission, specify 3.5  
assign(ment)  
  file extents 4.1  
  inter-partition 2.10  
asterisk  
  entry being processed 6.9  
attribute(s)  
  alter 6.11  
  alter, RJE 7.9  
  class 2.9  
  description of 2.9  
  disposition 2.9  
  list output 5.7  
  punch output 5.11  
  specification of 5.3  
AUTOSTART procedure 4.3

### B

book statement  
  delete 5.15  
  insert 5.15  
  replace 5.15  
BRDCST command 7.12

broadcast message(s)  
  accept/ignore 7.4  
  central 6.15  
  remote 7.12

### C

calculations  
  real storage requirements 3.10  
cancel  
  codes 9.1  
  job entry 6.7  
  job, RJE 7.5  
carriage control (see *forms control*)  
card  
  images, specify limit 3.3  
  separator 5.13  
catalog JECL statements 5.1  
change (see *alter*)  
CICS/VS 4.7  
class  
  alter 6.12  
  alter, RJE 7.9  
  CTL statement 5.3  
  input, specify 5.4  
  job 1.3, 2.9  
  list output, specify 5.9  
  print 1.4  
  priority 2.9  
  punch 1.4  
  punch output, specify 5.13  
  specification 2.9, 5.3  
code(s)  
  cancel 9.1  
  operator commands 6.1  
  return, RDREXIT 14.1  
  SIGNOFF 10.1  
  transmission, specify 3.5  
coding conventions *i*  
command(s)  
  ALTER 7.9  
  BRDCST 7.12  
  DELETE 7.10  
  DISPLAY 7.5  
  FLUSH 7.5  
  format of 6.1  
  GO 7.5  
  list of 6.1  
  PACCOUNT 6.15  
  PALTER 6.11  
  parameters 6.2  
  PBRDCST 6.15  
  PCANCEL 6.6  
  PDELETE 6.13  
  PDISPLAY 6.9  
  PEND 6.6  
  PFLUSH 6.7  
  PINQUIRE 6.14  
  PRELEASE 6.14  
  PRESTART 6.8

command(s)—*continued*

- PSETUP 6.16
- PSTART 6.2
- PSTOP 6.5
- queue management 6.9
- RELEASE 7.11
- RESTART 7.5
- SETUP 7.12
- SIGNOFF 7.3
- SIGNON 7.3
- START 7.4
- STOP 7.4
- summary 2.10

comments

- JECL statements 5.3

compare POWER/VS with DOS/VS POWER 12.1

component selection 3.8

concepts

- POWER/VS 2.1

considerations, start-up 4.7

continuation

- JECL statement 5.2

control language

- summary 2.10

control statements (see *job entry control language*)

conventions, commands and statements *i*

copies

- alter number of 6.12
- alter number of, RJE 7.9

CTL statement 5.3

**D**

- D attribute 2.9
- DASD (see *disk*)
- data
  - block 3.2
  - block size, specify 2.8
  - block, storage requirement 3.11
  - flow, POWER/VS 2.1
  - transmission mode 1.5
- data file
  - description of 2.3
  - labels, example 3.8
  - size requirement 3.13
- DATA statement 5.15
- DBLK
  - size 2.8
  - specify 3.2
  - storage requirement 3.11
- delete
  - book statement 5.15
  - job entry 6.13
  - job entry, RJE 7.10
  - write entry 6.7
- DELETE command 7.10
- delimiter statements 5.15
- destination
  - output 2.10

device(s)

- spooling, specify 4.6
- support 2.11

differences, POWER/VS and DOS/VS POWER 12.1

direct

- print 5.8, 1.4
- punch 5.13, 1.4

disk

- DOS/VS libraries 3.13
- extents, POWER/VS files 2.5, 2.8
- POWER/VS files 2.3
- POWER/VS file requirements 3.13
- storage requirements 3.13

diskette (see *3540 diskette*)

dispatching priority 4.7

display

- cancel PDISPLAY 6.7
- command 6.9
- job number, specify 3.4
- messages 6.10
- status, RJE 7.7

DISPLAY command 7.6

disposition

- alter 6.12
- alter, RJE 7.9
- input, specify 5.4
- job entries 2.9
- list, specify 5.8
- output 2.9

direct access (see *disk*)

distributed POWER/VS program 3.8

DOS/VS

- JCL modification 2.1
- POWER comparison with POWER/VS 12.1
- supervisor generation 3.9, 4.7

dummy device

- assignment 2.11
- assignment in supervisor 4.7
- generation 2.10, 2.11
- LST statement 5.7

dump, POWER/VS termination 6.6

**E**

- early print start (see *segmentation*)
- EBCDIC 3.5
- end (see *stop, terminate or cancel*)
- end job 5.5
- entry
  - definition 2.5
  - delete, RJE 7.9
  - flush, RJE 7.5
  - LUB 3.9
  - queue 2.5
  - release, RJE 7.11
  - system, storage requirement 3.12
- EOJ statement
  - format 5.5
  - function 5.5

error  
 device 1.3  
 I/O, cancel code 9.1  
 example(s)  
 storage requirement calculations 3.13  
 execution  
 account record 8.3  
 list task 2.3  
 processor 2.3  
 punch task 2.3  
 read task 2.3  
 extents of POWER/VS files 3.8, 3.10, 2.5, 2.8

## F

### FCB

phasename, specify 5.10  
 support 2.12

features of POWER/VS 1.3

fields of JECL statements 5.2

### file(s)

account file (see *account file*)  
 data file (see *data file*)  
 extents, example 3.8  
 labels, example 3.8  
 organization 2.5  
 queue file (see *queue file*)  
 size requirement 3.13

fixable area 3.10

FLUSH command 7.5

### flush

task 6.7  
 task, RJE 7.5

### format

files, POWER/VS 4.1, 4.5  
 format statement 4.6  
 JECL statements 5.2  
 of commands 6.1  
 of RJE commands 7.1

### FORMAT statement

AUTOSTART procedure 4.5  
 format of 4.6

### forms

card, specify 5.13  
 printer, setup 6.16  
 printer, specify 5.9

### forms control

line table 5.10  
 LTAB parameter 3.2

### functions

POWER 2.1

## G

### generation

examples 17.1  
 file labels example 3.8  
 DOS/VS supervisor 3.8  
 POWER macro 3.1  
 POWER/VS 3.1  
 pre-assembled POWER/VS 3.8  
 macros 3.1

GO command 7.5

## H

H attribute 2.9

hardware device support 2.11

hints (see *considerations*)

### HOLD state

release 6.14

hot reader 1.5

hot writer 1.5, 7.2

## I

I attribute 2.9

idle RJE terminal 1.5, 7.2

### initialize

POWER/VS 4.3

### input

class, specify 5.4  
 disposition, specify 5.4  
 priority, specify 5.4

### insert

book statement 5.15  
 deck into source book 5.15  
 source book 5.15

interactive data communications 4.7

intermediate storage 2.1

inter-partition device assignment 2.10

## J

JAI (see *job accounting*)

JECL modification 2.1

JECL (see *job entry control language*)

### job(s)

account file capacity 3.14  
 attributes (see *attributes*)  
 entry (see *entry*)  
 name, specify 5.4  
 number 3.4  
 separator 1.4

### job accounting

adding user information 13.1  
 processing records 2.8  
 record, contents of 8.1  
 record types 2.8  
 save account file 6.15  
 support, specify 3.3  
 support, storage requirement 3.10  
 \$JOBACCT 13.1

### job entry control language

delimit jobs 5.1  
 function 5.1  
 list of statements 5.1  
 positional parameters 5.2  
 statement formats 5.2  
 treated as comments 5.1

### job number

display, specify 3.4

JOB statement 5.3

job stream(s)  
  delimit POWER/VS job 5.3  
  example 2.7, 15.1  
  processing 1.1

## K

K attribute 2.9  
KILL POWER/VS 6.6

## L

L attribute 2.9  
LFCB 2.5, 5.10  
LEAVE state, release 6.14  
limit  
  card images, specify 3.3  
  list records, specify 5.9  
  output records 2.10  
  print lines, specify 3.3  
  punch records, specify 5.14

### line(s)

  account record 8.5  
  RJE, define 3.5  
  RJE, number of 1.5  
  RJE, start-up 7.2  
  RJE, timeout 3.5

### line table

  LTAB, specify at generation 3.2

### list

  account record 8.2  
  entry 2.5  
  queue entry 2.5  
  queue set 2.5  
  tasks 2.3

### logical

  list 2.3  
  punch 2.3  
  read 2.3

### LST statement

  format 5.7  
  function 5.7

## M

### macro(s)

  PLINE 3.5  
  POWER 3.1  
  POWER/VS generation 3.1  
  PRMT 3.6

### main storage requirements 3.10

### message(s)

  broadcast 6.14, 7.4  
  display 6.10  
  format files 4.3  
  operator 18.1  
  POWER/VS 18.1  
  remote 7.1, 7.12

### modify book images 5.15

### multitasking support 2.1, 12.1

## multiple

  devices 1.3  
  list copies, specify 5.9  
  printers 1.3, 5.7  
  punch copies, specify 5.13  
  punches 1.3  
  multivolume tape files 5.9, 5.13

## N

N attribute 2.9  
name  
  logical unit, POWER/VS files 2.5, 2.8  
notation conventions *i*  
number-of-copies (see *copies*)

## O

object deck(s), transmit 3.5  
operator commands (see *commands*)  
operator messages 18.1  
output  
  destination 2.10  
  direct list 5.8  
  limit 2.10  
  list, attributes 5.7  
  remote list 5.9  
  remote punch 5.13  
  segmentation (see *segmentation*)  
  tape spooling 5.8

## P

PACCOUNT command 6.15  
page separator 5.9  
pageable area 3.10  
PALTER command 6.11  
partition(s)  
  layout, POWER/VS 3.10  
  POWER/VS controlled 2.1  
  size, POWER/VS 4.1, 4.5  
  start 6.2  
  stop 6.5  
password 3.5, 6.3, 7.3  
PBRDCST command 6.15  
PCANCEL command 6.7  
PDELETE 6.13  
PDISPLAY  
  cancel 6.7  
  format of command 6.9  
  function 6.9  
PEND command 6.6  
permanent area 3.10  
PFLUSH command 6.7  
physical  
  list 2.3  
  punch 2.3  
  read 2.3  
PINQUIRE command 6.14  
PLINE macro 3.5  
POCL (see *commands*)

- positional parameters, JECL statements 5.2
- POWER macro 3.1
- POWER/VS
  - commands 2.10
  - compare with DOS/V5 POWER 12.1
  - concepts 2.1
  - control language 2.10
  - data flow 2.1
  - definition 1.1
  - device support 2.11
  - disk files 2.3
  - facilities 2.1
  - features 1.3
  - file extents 2.5, 2.8
  - file labels, example 3.8
  - file requirements 3.13
  - functions 2.1
  - initialize 4.3
  - introduction 1.1
  - JECL statements 5.1
  - job delimiters 5.3
  - operator commands (see *commands*)
  - operator messages 18.1
  - pre-assembled program 3.8
  - processing with, without 1.1
  - queues 2.9
  - start-up 4.1
  - tasks 2.1
  - termination 6.6
- pre-assembled POWER/V5 3.8
- PRELEASE command 6.14
- print (see *list*)
- print lines
  - limit, specify 3.3
- printer(s)
  - dummy 2.10, 2.11
  - supported devices 2.11
  - UCS 2.12
- printer forms
  - FCB 2.12
  - specify 5.9, 1.4
  - setup 6.16
  - setup, RJE 7.12
- PRINTERS statement 4.6, 2.1
- printing
  - restart, RJE 7.6
- PRESTART command 6.8
- priority
  - alter 6.12
  - alter, RJE 7.9
  - attribute 2.10
  - class 2.9
  - default, specify 3.2
  - dispatching 2.1
  - input, specify 5.4
  - jobs 1.3
  - partitions 4.1, 4.7
  - specification 2.10
- PRMT macro 3.6
- procedure(s)
  - start-up with AUTOSTART 4.3
  - start-up without AUTOSTART 4.1

- program, pre-assembled POWER/V5 3.8
- PRT statement 5.7
- PSETUP command 6.16
- PSTART
  - AUTOSTART procedure 4.3
    - command 6.5
    - RJE lines 4.5
- PSTOP command 6.5
- PUN statement 5.11
- punch(es)
  - account record 8.3
  - dummy 2.10, 2.11
  - entry 2.5
  - queue entry 2.5
  - queue set 2.5
  - supported devices 2.11
- PUNCHES statement 4.6, 2.1
- punching
  - restart, RJE 7.6
- PUTACCT macro 13.1

## Q

- queue(s)
  - display 6.9
  - entries 2.5
  - file organization 2.5
  - management commands 6.9
  - POWER/V5 1.3, 2.9
  - record 2.5
  - set 2.5
- queue file
  - definition 2.3
  - extents 2.5
  - labels, example 3.8
  - logical unit name 2.5
  - organization 2.5
  - size requirement 3.13
- queue records
  - table, by device 3.14

## R

- RDR statement 5.5
- RDREXIT
  - coding of 14.1
  - storage requirement 3.10
  - support, specify 3.4
- read
  - account record 8.1
  - entry 2.5
  - queue entry 2.5
  - queue set 2.5
  - tasks 2.3
- READER statement 4.6, 2.1
- reader(s) (see also *read task*)
  - supported devices 2.11
- reader-only support 4.6
- real mode 2.1

- real storage
  - ALLOCR 3.10
  - ALLOCR, start-up 4.1, 4.5
- record(s)
  - account 8.1
  - account file, types 2.8
  - list, specify 5.9
  - output limit 1.4
  - punch, specify 5.14
  - queue 2.5
  - queue file, size 2.5
- re-display
  - messages 6.11
- release
  - job entry 6.14
  - job entry, RJE 7.11
- RELEASE command 7.11
- remote
  - output, list 5.9
  - output, punch 5.13
- remote-id
  - alter 6.12
  - alter, RJE 7.9
  - sign on terminal 7.3
- remote job entry (see *RJE*)
- replace book statement 5.15
- report
  - status, example 11.1
  - status, RJE line 6.14
- reprint 6.8
- restart
  - list/punch task 6.8
  - POWER/VS 1.3
  - printing 1.4
  - PSTOP parameter 6.5
  - punching 1.4
  - task at stopping point 6.5
  - task at stopping point, RJE 7.5
  - write task, RJE 7.6
- RESTART command 7.6
- return codes
  - RDREXIT 14.1
- RJE
  - commands, list 7.1
  - device support 2.12
  - facilities 1.5
  - hardware feature support 1.5
  - line, define 3.5
  - lines, number of 1.5
  - operational characteristics 7.1
  - PLINE macro 3.5
  - SIGNOFF codes 10.1
  - start task 6.2
  - status report 6.14
  - storage requirement 3.10
  - tasks 2.3
- ROCL (see *commands*)
- routine(s)
  - RDREXIT 14.1
  - \$JOBACCT 13.1

**S**

- save account file 6.15, 2.8
- segmentation
  - attribute 2.10
  - definition 2.5
  - feature 1.1
  - list output, specify 5.10
  - punch output, specify 5.14
  - separator pages/cards 3.4
  - specify at generation 3.4
- separator
  - cards, specify 5.13
  - cards, specify at generation 3.4
  - job 1.4
  - page, specify 5.9
  - pages, specify at generation 3.4
- sequence field, JECL statements 5.3
- setup
  - printer forms 6.16
  - printer forms, RJE 7.12
- SETUP command 7.12
- sign off terminal 7.3
- SIGNOFF
  - codes 10.1
  - command 7.3
  - sign on terminal 7.3
- SIGNON command 7.3
- six-bit transmission code 3.5
- SL update statement 5.15
- SLI support, specify 3.3
- SLI statement 5.14
- source
  - book, specify 5.14
  - library inclusion 5.14, 1.3
  - library update statements 5.15
  - library, SUBLIB 3.3
- split cylinders 2.5
- spool
  - device specification 4.6
  - tape, specify for list 5.8
  - tape, specify for punch 5.13
- spooling
  - definition 1.1
  - devices 2.11
- stacker select
  - device support 2.11
  - input 2.1
- start
  - partition 6.2
  - RJE task 6.2
  - task 6.2
  - write task, RJE 7.4
- START command 7.4
- start-up
  - considerations 4.7
  - POWER/VS 4.1
  - RJE lines 7.2, 4.5
  - with AUTOSTART 4.3
  - without AUTOSTART 4.1

- statement(s)
  - address specification 4.6
  - FORMAT 4.6
  - JECL (see *job entry control language*)
  - PRINTERS 4.6
  - PUNCHES 4.6
  - READER 4.6
- statistics
  - files, example 11.1
- status report
  - example 11.1
  - RJE 6.14, 7.6
- stop
  - command, DOS/VS 6.5
  - partition 6.5
  - POWER/VS 6.6
  - session 7.3
  - task 6.5
  - write task 7.4
- STOP command 7.4
- storage
  - intermediate 2.1
  - requirement, disk 3.13
  - requirement, main 3.10
  - requirement, POWER/VS 2.11
- sublibrary
  - SLI statement 5.14
  - SUBLIB parameter 3.3
- supervisor
  - DOS/VS generation requirements 3.9
  - standard defaults for POWER/VS 4.7

## T

- T attribute 2.9
- table(s)
  - deblock defaults 3.2
  - device support 2.11
  - display items, RJE 7.7
  - feature support, devices 2.11
  - JECL statements 5.1
  - job accounting 8.1
  - job entry display 6.9
  - line table 3.2, 5.10
  - operator commands 6.1
  - POWER/VS storage requirements 3.10
  - queue records by device 3.14
  - RDREXIT, return codes 8.1
  - RJE commands 7.1

## tape

- intermediate storage 2.1
- spool list output 5.9
- spool punch output 5.13
- spooling, specify for list 5.8

## task

- flush 7.5
- management commands 6.2, 6.1
- POWER/VS 2.1
- restart 6.8
- restart, RJE 7.6
- starting 6.2
- stop 6.5

- teleprocessing (see *RJE*)
- terminal
  - characteristics, specify 3.6
  - commands 7.3
  - identification 7.3
  - operation 7.2
  - supported devices 2.11
- terminate
  - POWER/VS 6.6
  - PDISPLAY 6.7
  - session 7.3
- timeout
  - RJE lines, specify 3.5
- track group(s)
  - definition 2.7
  - disk requirements 3.13
  - file labels example 3.8
  - TRACKGP parameter 3.2
- transparency support 3.5
- turn-around RJE line 7.2

## U

- UCS
  - phasename, specify 5.10
  - support 2.12
- UNBATCH command 6.5
- unit-record device(s)
  - assignments in supervisor 4.7
  - spooled, specify 4.6
- user information
  - account records 8.1
  - RJE, specify 7.3
  - specification 5.5
- user routine(s)
  - \$JOBACCT 13.1
  - RDREXIT 14.1

## V

- Virtual
  - mode 1.3, 2.1
  - storage ALLOC 3.10
  - storage, ALLOC at start-up 4.1

## W

- write task
  - start 6.2
  - start, RJE 7.4
  - stop, RJE 7.4
  - stop 6.5
- writer (see *write task*)
- writer-only support 4.6
- writer-only partition
  - definition 1.3
  - DATA statement 5.15
  - JECL 5.1

2311 disk support 12.1  
3540 diskette  
    input from 5.5  
    RDR statement 5.5  
    start read task 6.4  
    support 2.12  
3780 component selection 3.8  
\$JOBACCT  
    account record 8.4  
    coding of 13.1  
\* \$\$ /\* statement 5.15  
\* \$\$ /& statement 5.15

DOS/VS POWER/VS Installation and Operations

This sheet is for comments and suggestions about this manual. We would appreciate *your* views, favorable or unfavorable, in order to aid us in improving *this* publication. This form will be sent directly to the author's department. Please include your name and address if you wish a reply. Contact your IBM branch office for answers to technical questions about the system or when requesting additional publications. Thank you.

Name  
Address

How did you use this manual?

As a reference source

As a classroom text

As a self-study text

What is your occupation?

Your comments\* and suggestions:

---

\* We would especially appreciate your comments on any of the following topics:

Clarity of the text

Accuracy

Index

Illustrations

Appearance

Paper

Organization of the text

Cross-references

Tables

Examples

Printing

Binding

**YOUR COMMENTS, PLEASE . . .**

This manual is part of a library that serves as a reference source for systems analysts, programmers and operators of IBM systems. Your answers to the questions on the back of this form, together with your comments, will help us produce better publications for your use. Each reply will be carefully reviewed by the persons responsible for writing and publishing this material. All comments and suggestions become the property of IBM.

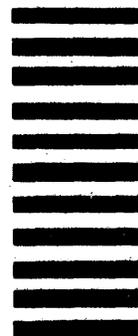
Please note: Requests for copies of publications and for assistance in utilizing your IBM system should be directed to your IBM representative or to the IBM sales office serving your locality.

Fold

Fold

FIRST CLASS  
PERMIT NO. 1359  
WHITE PLAINS, N. Y.

**BUSINESS REPLY MAIL**  
NO POSTAGE STAMP NECESSARY IF MAILED IN THE UNITED STATES



POSTAGE WILL BE PAID BY . . .

IBM Corporation  
1133 Westchester Avenue  
White Plains, N.Y. 10604

Attention: Department 813 U

Fold

Fold



**International Business Machines Corporation**  
**Data Processing Division**  
1133 Westchester Avenue, White Plains, New York 10604  
(U.S.A. only)

**IBM World Trade Corporation**  
821 United Nations Plaza, New York, New York 10017  
(International)

CUT ALONG THIS LINE

U.S./VS POWER/VS Installation and Operations, GC33-5403-1



**IBM**

**International Business Machines Corporation  
Data Processing Division  
1133 Westchester Avenue, White Plains, New York 10604  
(U.S.A. only)**

**IBM World Trade Corporation  
621 United Nations Plaza, New York, New York 10017  
(International)**